

Luca Longo  
Sebastian Lapuschkin  
Christin Seifert (Eds.)

Communications in Computer and Information Science

2155

# Explainable Artificial Intelligence


Second World Conference, xAI 2024  
Valletta, Malta, July 17–19, 2024  
Proceedings, Part III

Part 3

 Springer



Editorial Board Members

Joaquim Filipe , *Polytechnic Institute of Setúbal, Setúbal, Portugal*

Ashish Ghosh , *Indian Statistical Institute, Kolkata, India*

Lizhu Zhou, *Tsinghua University, Beijing, China*

## **Rationale**

The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are also considered for publication.

## **Topics**

The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

## **Information for Volume Editors and Authors**

Publication in CCIS is free of charge. No royalties are paid, however, we offer registered conference participants temporary free access to the online version of the conference proceedings on SpringerLink (<http://link.springer.com>) by means of an http referrer from the conference website and/or a number of complimentary printed copies, as specified in the official acceptance email of the event.

CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, and delivered in the form of printed books and/or electronically as USBs and/or e-content licenses for accessing proceedings at SpringerLink. Furthermore, CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library at <http://link.springer.com/bookseries/7899>. Conferences publishing in CCIS are allowed to use Online Conference Service (OCS) for managing the whole proceedings lifecycle (from submission and reviewing to preparing for publication) free of charge.

## **Publication process**

The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/files, authors have to strictly adhere to the Springer CCIS Authors' Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

## **Abstracting/Indexing**

CCIS is abstracted/indexed in DBLP, Google Scholar, EI-Compendex, Mathematical Reviews, SCImago, Scopus. CCIS volumes are also submitted for the inclusion in ISI Proceedings.

## **How to start**

To start the evaluation of your proposal for inclusion in the CCIS series, please send an e-mail to [ccis@springer.com](mailto:ccis@springer.com).

Luca Longo · Sebastian Lapuschkin ·  
Christin Seifert  
Editors


# Explainable Artificial Intelligence

Second World Conference, xAI 2024  
Valletta, Malta, July 17–19, 2024  
Proceedings, Part III

*Editors*

Luca Longo   
Technological University Dublin  
Dublin, Ireland

Christin Seifert   
University of Marburg  
Marburg, Germany

Sebastian Lapuschkin   
Fraunhofer Institute for Telecommunications,  
Heinrich-Hertz-Institut, HHI  
Berlin, Germany

ISSN 1865-0929                      ISSN 1865-0937 (electronic)  
Communications in Computer and Information Science  
ISBN 978-3-031-63799-5              ISBN 978-3-031-63800-8 (eBook)  
<https://doi.org/10.1007/978-3-031-63800-8>

© The Editor(s) (if applicable) and The Author(s), under exclusive license  
to Springer Nature Switzerland AG 2024

Chapter “CountARFactuals – Generating Plausible Model-Agnostic Counterfactual Explanations with Adversarial Random Forests” is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>). For further details see license information in the chapter.

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

# Preface

The field of eXplainable AI (XAI) has seen significant growth recently. As part of the larger field of Artificial Intelligence, it has evolved into a highly multi-disciplinary and inter-disciplinary active field of research. Artificial Intelligence and its methods have been used in many disciplines besides computer science, including medicine and neuroscience, chemistry, biology, education, psychology and philosophy. Thanks to its evolving subfield, machine learning, it has been applied in many real-world applications. This is due to its ability to learn and extract patterns automatically from sparse, complex, non-linear data. These include prediction and forecasting, classification and recommendation, to mention a few. However, in some critical applications, such as finance and health care, understanding machine-learned models and their underlying inferential mechanisms is paramount for creating trustworthy and responsible applications. In addition, the requirements imposed by the GDPR, the new AI act, and those regulations that will follow worldwide are leading scholars to develop methods for explaining AI systems and their outputs. This is evident in the hundreds of manuscripts submitted to the 2nd World Conference of Artificial Intelligence, their diverse methodology, techniques and approaches, and their application in various real-world contexts. Similarly to the first edition, the second edition has attracted considerable interest from scholars in academia and industry. The hundreds of authors, attendees and programme committee members from more than 40 countries make the conference a truly world event. With an acceptance rate of roughly ~40%, with 95 manuscripts being accepted from 204 submissions, it is our great privilege to present the proceedings of the second World Conference on eXplainable Artificial Intelligence (xAI 2024), held in Valletta, Malta, from the 17th to the 19th of July at the historic Mediterranean Conference Centre, a fascinating venue.

Split over four volumes, this book aggregates a collection of the best contributions received and presented at xAI 2024, describing recent approaches, methods and techniques for explainability. The accepted articles were selected through a rigorous, single-blind peer-review process. Each article received at least three reviews, with an average of four reviews per paper, from more than 250 scholars in academia and industry, with 99% of them holding a PhD in an area relevant to the topics of the conference. The programme committee chairs of the conference carefully selected the top contributions by ranking articles across several objective criteria and evaluating and triangulating the qualitative feedback left by the international reviewers. The peer-review process was exhaustive and intensive, ensuring that xAI-2024 adhered to the highest quality standards. All the accepted research contributions are included in these proceedings and were invited to give oral presentations.

Several special tracks and thematic sessions were organised, each proposed and chaired by various scholars, to aggregate highly innovative areas within the larger field of explainable Artificial Intelligence. A parallel track was organised for work in progress, specifically preliminary novel research studies relevant to xAI, which were presented as posters during the event. A demo track was held where scholars presented their software

prototypes on explainability or real-world applications of explainable AI-based systems. A doctoral consortium was organised, with lecturers to PhD scholars who submitted their doctoral proposals on future research related to eXplainable Artificial Intelligence. These scholars pitched their preliminary doctoral work and were matched with renowned scientists who provided guidance and constructive feedback. A separate programme committee was set up for the late-breaking work and demo and doctoral consortium tracks. Finally, a panel discussion was organised with renowned scholars in xAI, offering a multidisciplinary view while inspiring the attendees with tangible recommendations to tackle challenges toward designing responsible, trustworthy AI-based technologies through explainable AI.

A thank you to all the volunteers who helped in the organising committee for the 2nd World Conference on eXplainable Artificial Intelligence (xAI 2024). In particular, we would like to thank the local chair, Charlie Abela; the doctoral committee chair, Grégoire Montavon; the inclusion & accessibility chairs, Mario Brcic and Verena Klös; and the late-breaking work chair, Weiru Liu. Also, special thanks to the keynote speaker, Fosca Giannotti. A word of appreciation goes to the organisers of the special tracks and those who chaired them during the conference. Special thanks go to the researchers and practitioners who submitted their work, the various programme committee members who provided precious feedback during the peer-review process, and all who attended the event and turned it into a fantastic networking opportunity to share findings and learn from each other as a community.

The Mediterranean Conference Centre, a 16th-century marvel in historic Valletta, was initially built as a hospital by the Order of St. John. Known as the Sacra Infermeria, it was meant to receive Maltese and foreign patients and pilgrims travelling to the Holy Land. Analogously, explainable artificial intelligence (XAI) is intended to be a transit between current opaque AI-based technologies and the development of robust, transparent, fair and trustworthy AI for the benefit of humankind.

May 2024

Luca Longo  
Sebastian Lapuschkin  
Christin Seifert

# Organization

## Organizing Committee

### Program Committee Chairs

Luca Longo	Technological University Dublin, Ireland
Sebastian Lapuschkin	Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut, HHI, Germany
Christin Seifert	University of Marburg, Germany

### Doctoral Consortium Chair

Grégoire Montavon	Freie Universität Berlin, Germany
-------------------	-----------------------------------

### Late-Breaking Work/Demo Chair

Weiru Liu	University of Bristol, UK
-----------	---------------------------

### Inclusion and Accessibility Chairs

Mario Brcic	University of Zagreb, Croatia
Verena Klös	TU Dresden, Germany

### Local Chair

Charlie Abela	University of Malta, Malta
---------------	----------------------------

### General Chair

Luca Longo	Technological University Dublin, Ireland
------------	--



## Program Committee

Chirag Agarwal	University of Illinois at Chicago, USA
Arianna Agosto	University of Pavia, Italy
Jaumin Ajdari	South East European University at Tetovo, Republic of North Macedonia
Jesús Alcalá-Fdez	University of Granada, Spain
Elvio Gilberto Amparore	University of Turin, Italy
Christopher Anders	Technische Universität Berlin, Germany
Vincent Andrearczyk	HES-SO, Switzerland
Andrea Apicella	University of Naples Federico II, Italy
Annalisa Appice	University of Bari Aldo Moro, Italy
Gilles Audemard	CRIL - Université d'Artois, France
Hamed Ayoobi	Imperial College London, UK
Omran Ayoub	Scuola Universitaria Professionale della Svizzera Italiana, Switzerland
Werner Bailer	Joanneum Research Austria
Marco Baioletti	Università degli Studi di Perugia, Italy
Prasanna Balaprakash	Oak Ridge National Laboratory, USA
Antonio Jesús Banegas-Luna	Universidad Católica de Murcia, Spain
Marília Barandas	Fraunhofer Portugal AICOS, Portugal
Pietro Barbiero	University of Cambridge, UK
Sylvio Barbon Junior	University of Trieste, Italy
Francesco Barile	Maastricht University, The Netherlands
Nick Bassiliades	Aristotle University of Thessaloniki, Greece
Juri Belikov	Tallinn University of Technology, Estonia
Shai Ben-David	University of Waterloo, Canada
Malika Bendechache	University of Galway, Ireland
Jenny Benoist-Pineau	University of Bordeaux, France
Floris Bex	Utrecht University, The Netherlands
Marija Bezbradica	Dublin City University, Ireland
Przemek Biecek	University of Wrocław, Poland
Felix Biessmann	Berlin University of Applied Sciences, Germany
Stefano Bistarelli	Università di Perugia, Italy
Szymon Bobek	AGH University of Science and Technology, Poland
Diego Borro	University of Navarre, Spain
Sebastian Bosse	Fraunhofer Heinrich-Hertz-Institute, Germany
Henrik Boström	KTH Royal Institute of Technology, Sweden
Romain Bourqui	Université Bordeaux, France
Nicolas Boutry	EPITA Research Laboratory (LRE), Le Kremlin-Bicêtre, France

Dave Braines	IBM, UK
Mario Brcic	University of Zagreb, Croatia
Rob Brennan	University College Dublin, Ireland
Heike Buhl	Paderborn University, Germany
Adrian Byrne	CeADAR UCD/Idiro Analytics, Ireland
Federico Cabitza	Università degli Studi di Milano-Bicocca, Italy
Roberta Calegari	Alma Mater Studiorum Università di Bologna, Italy
Andrea Campagner	Università degli Studi di Milano-Bicocca, Italy
Roberto Capobianco	Sapienza University of Rome, Italy
Andrea Capotorti	Università di Perugia, Italy
F. Amílcar Cardoso	University of Coimbra, Portugal
Ramon Alberto Carrasco	Universidad Complutense de Madrid, Spain
Giuseppe Casalicchio	Ludwig Maximilian University of Munich, Germany
Gabriella Casalino	University of Bari Aldo Moro, Italy
Danilo Cavaliere	Università degli Studi di Salerno, Italy
Paola Cerchiello	University of Pavia, Italy
Debaditya Chakraborty	University of Texas at San Antonio, USA
Giovanni Ciatto	University of Bologna, Italy
Philipp Cimiano	Bielefeld University, Germany
Mario Giovanni C. A. Cimino	University of Pisa, Italy
Oscar Cordón	Universidad de Granada, Spain
Paulo Cortez	University of Minho, Portugal
Jane Courtney	Technological University Dublin, Ireland
Sabatina Criscuolo	University of Naples Federico II, Italy
Renato De Leone	Università di Camerino, Italy
Carmen De Maio	Università degli Studi di Salerno, Italy
Sarah Jane Delany	Technological University Dublin, Ireland
Tommaso Di Noia	Politecnico di Bari, Italy
Christos Dimitrakakis	University of Neuchâtel, Switzerland
Giovanna Dimitri	University of Siena, Italy
Ivan Donadello	Free University of Bozen, Italy
Alexandros Doumanoglou	Information Technologies Institute, Greece
Clemens Dubsloff	Eindhoven University of Technology, The Netherlands
Jonathan Dunne	IBM, Ireland
Ivana Dusparic	Trinity College Dublin, Ireland
Oliver Eberle	Technische Universität Berlin, Germany
Kris Ehinger	University of Melbourne, Australia
Charles Ellis	Center for Translational Research in Neuroimaging and Data Science, USA

Brígida Mónica Faria	ESS-P.PORTO, Portugal
Ad Feelders	Utrecht University, The Netherlands
Giuseppe Fenza	Università degli Studi di Salerno, Italy
Mexhid Ferati	Linnaeus University, Sweden
Alberto Fernández	University of Granada, Spain
Enrico Ferrari	Rulex Innovation Labs, Italy
Cesar Ferri	Universitat Politècnica de València, Spain
Gianna Figà-Talamanca	University of Perugia, Italy
Duarte Folgado Associação	Fraunhofer Portugal Research, Portugal
Kary Främling	Umeå University, Sweden
Valentina Franzoni	University of Perugia, Italy
Timo Freiesleben	Ludwig Maximilian University of Munich, Germany
Alberto Freitas	University of Porto, Portugal
Pascal Friederich	Karlsruhe Institute of Technology, Germany
Sebastian Fudickar	Lübeck University, Germany
Johannes Fürnkranz	Johannes Kepler University Linz, Austria
Domenico Furno	University of Salerno, Italy
Angelo Gaeta	Università di Salerno, Italy
Stéphane Galland	Université de Technologie de Belfort-Montbéliard, France
Mariacristina Gallo	University of Salerno, Italy
João Gama	University of Porto, Portugal
Edel Garcia-Reyes	CCG, Portugal
Pascal Germain	INRA, France
Melinda Gervasio	SRI International, USA
Gizem Gezici	Scuola Normale Superiore, Italy
Massimiliano Giacomini	University of Brescia, Italy
John Gilligan	Technological University Dublin, Ireland
Romain Giot	Université de Bordeaux, France
Paolo Giudici	University of Pavia, Italy
Martin Gjoreski	Università della Svizzera italiana, Switzerland
David Glass	University of Ulster, UK
Rocio Gonzalez-Diaz	University of Seville, Spain
Riccardo Guidotti	University of Pisa, Italy
Miguel A. Gutiérrez-Naranjo	University of Seville, Spain
Mark Hall	Airbus, UK
Barbara Hammer	Bielefeld University, Germany
Lars Kai Hansen	Technical University of Denmark, Denmark
Hanna Hauptmann	Utrecht University, The Netherlands
Yoichi Hayashi	Meiji University, Japan
Ciara Heavin	University College Cork, Ireland

Fredrik Heintz	Linköping University, Sweden
Marina Höhne	University of Potsdam, Germany
Andreas Holzinger	University of Natural Resources and Life Sciences, Austria
Georgiana Ifrim	University College Dublin, Ireland
Francesco Isgro	University of Naples Federico II, Italy
Richard Jiang	Lancaster University, UK
Ulf Johansson	Jönköping University, Sweden
José Manuel Juárez	Universidad de Murcia, Spain
Martin Jullum	Norwegian Computing Center, Norway
Ilir Jusufi	Blekinge Institute of Technology, Sweden
Severin Kacianka	Technical University of Munich, Germany
Nikos Karacapilidis	University of Patras, Greece
Sophia Karagiorgou	UBITECH Ltd., Greece
Gjergji Kasneci	Technical University of Munich, Germany
Zenun Kastrati	Linnaeus University, Sweden
Mark Keane	University College Dublin, Ireland
Mohammad Emtiyaz Khan	RIKEN, Japan
Hassan Khosravi	University of Queensland, Australia
Tomáš Kliegr	Prague University of Economics and Business, Czech Republic
Wolfgang Konen	Cologne University of Applied Sciences, Germany
Iordanis Koutsopoulos	Athens University of Economics and Business, Greece
Gitta Kutyniok	LMU Munich, Germany
Christophe Labreuche	Thales R&T, France
Andrew Lensen	Victoria University of Wellington, New Zealand
Francesco Leofante	Imperial College London, UK
Rosa Lillo	Universidad Carlos III de Madrid, Spain
Weiru Liu	University of Bristol, UK
Markus Loecher	Berlin School of Economics and Law, Germany
Tuwe Löfström	University of Jönköping, Sweden
Henrique Lopes	Cardoso University of Porto, Portugal
Jens Lundström	Halmstad University, Sweden
Lucie Charlotte Magister	University of Cambridge, UK
Klaus H. Maier-Hein	German Cancer Research Center, Germany
Avleen Malhi	University of Warwick, UK
Michail Mamalakis	University of Cambridge, UK
Marcelo Manzato	University of São Paulo, Brazil
Francesco Marcelloni	Università di Pisa, Italy
Stefano Mariani	Università di Modena e Reggio Emilia, Italy

Jose Paulo Marques dos Santos	University of Maia, Portugal
Chiara Masiero	Statwolf Data Science Srl, Italy
Manuel Mazzara	Innopolis University, Russia
Kevin McAreavey	University of Bristol, UK
Susan McKeever	Technological University Dublin, Ireland
Corrado Mencar	University of Bari Aldo Moro, Italy
Andrès Mendez	CNRS, France
Sandra Mitrovic	IDSIA, Switzerland
Jose M. Molina	Universidad Carlos III de Madrid, Spain
Sebastian Möller	TU Berlin, Germany
Maurizio Mongelli	CNR-IEIT, Italy
Anna Monreale	University of Pisa, Italy
Grégoire Montavon	Freie Universität Berlin, Germany
Antonio Moreno	URV, Spain
Vincenzo Moscato	University of Naples, Italy
Yazan Mualla	Université de Technologie de Belfort-Montbéliard, France
Jörg P. Müller	TU Clausthal, Germany
Emmanuel Müller	TU Dortmund, Germany
Pradeep Kumar Murukannaiah	Delft University of Technology, The Netherlands
Marco Muselli	Rulex Innovation Labs, Italy
Thomas Nagler	LMU Munich, Germany
Mohammad Naiseh	Bournemouth University, UK
Takafumi Nakanishi	Musashino University, Japan
Axel-Cyrille Ngonga Ngomo	Paderborn University, Germany
Slawomir Nowaczyk	Halmstad University, Sweden
Ruairi O'Reilly	Munster Technological University, Ireland
Declan O'Sullivan	Trinity College Dublin, Ireland
Eugénio Oliveira	Universidade do Porto, Portugal
Andrea Omicini	Alma Mater Studiorum Università di Bologna, Italy
Luca Oneto	University of Genoa, Italy
Chun Ouyang	Queensland University of Technology, Australia
Özgür Lütfü Özcep	University of Hamburg, Germany
Andres Paez	Universidad de los Andes, Colombia
Paolo Pagnottoni	University of Pavia, Italy
André Panisson	CENTAI Institute, Italy
Enea Parimbelli	University of Pavia, Italy
Sepideh Pashami	Halmstad University, Sweden
Miguel Angel Patricio	Universidad Carlos III de Madrid, Spain
Andrea Paziienza	NTT DATA Italia SpA & A3K Srl, Italy
Michael Pazzani	UCSD, USA

Felice Andrea Pellegrino	Università degli Studi di Trieste, Italy
Roberto Pellungri	University of Pisa, Italy
Alan Perotti	CENTAI Institute, Italy
Caroline Petitjean	Université de Rouen, France
Valentina Poggioni	Università di Perugia, Italy
Christopher Potts	Stanford University, USA
Roberto Prevede	University of Naples Federico II, Italy
Ricardo Prudencio	UFPE, Brazil
Georges Quénot	Laboratoire d'Informatique de Grenoble, CNRS, France
Santiago Quintana Amate	Airbus, UK
Astrid Rakow	German Aerospace Center (DLR), Germany
Wael Rashwan	Technological University Dublin, Ireland
Emanuele Ratti	University of Milan, Italy
Bujar Raufi	Technological University Dublin, Ireland
Oliver Ray	University of Bristol, UK
Daniele Regoli	Intesa Sanpaolo, Italy
Luis Paulo Reis	University of Porto/LIACC, Portugal
Alessandro Renda	Università degli Studi di Firenze, Italy
Rita P. Ribeiro	University of Porto, Portugal
Ita Richardson	Lero - Irish Software Engineering Research Centre, University of Limerick, Ireland
Tjitze Rienstra	Maastricht University, The Netherlands
Maria Riveiro	Jönköping University, Sweden
Lucas Rizzo	Technological University Dublin, Ireland
Katharina Rohlfing	University of Paderborn, Germany
Luís Rosado	Fraunhofer Portugal AICOS, Portugal
Matteo Rucco	Biocentis s.r.l., Italy
Amal Saadallah	TU Dortmund, Germany
Araceli Sanchis	Universidad Carlos III de Madrid, Spain
Ute Schmid	University of Bamberg, Germany
Christoph Schommer	University of Luxembourg, Luxembourg
Patrick Schramowski	TU Darmstadt, Germany
Carsten Schulte	University of Paderborn, Germany
Alexander Schulz	Bielefeld University, Germany
Pedro Sequeira	SRI International, USA
Telmo Silva Filho	University of Bristol, UK
Fabrizio Silvestri	University of Rome, Italy
Carlos Soares	University of Porto, Portugal
Tran Cao Son	New Mexico State University, USA
Francesco Sovrano	University of Zurich, Switzerland
Gerasimos Spanakis	Maastricht University, The Netherlands

Timo Speith	Universität Bayreuth, Germany
Gian Antonio Susto	Università degli Studi di Padova, Italy
Silke Szymczak	Universität zu Lübeck, Germany
Nikolay Tcholtchev	Fraunhofer Institute for Open Communication Systems (FOKUS), Germany
Jan Arne Telle	University of Bergen, Norway
Kirsten Thommes	Paderborn University, Germany
Alberto Tonda	INRA, France
Mariya Toneva	Max Planck Institute for Software Systems, Germany
Ruth Uerner	York University, Canada
Matias Valdenegro-Toro	University of Groningen, The Netherlands
Zita Vale	GECAD - ISEP/IPP, Portugal
Jasper van der Waa	TNO Human Factors - Perceptual and Cognitive Systems, The Netherlands
Niki van Stein	Leiden University, The Netherlands
Bruno Veloso	University of Porto & INESC TEC, Portugal
Giulia Vilone	Technological University Dublin, Ireland
Fabio Vitali	University of Bologna, Italy
Rosina Weber	Drexel University, USA
Robert Wille	Technical University of Munich & SCCH GmbH, Germany
Britta Wrede	Bielefeld University, Germany
Marvin Wright	Leibniz Institute for Prevention Research and Epidemiology – BIPS & University of Bremen, Germany
Bartosz Zielinski	Jagiellonian University, Poland

# Contents – Part III

## Counterfactual Explanations and Causality for eXplainable AI

Sub-SpaCE: Subsequence-Based Sparse Counterfactual Explanations for Time Series Classification Problems .....	3
<i>Mario Refoyo and David Luengo</i>	
Human-in-the-Loop Personalized Counterfactual Recourse .....	18
<i>Carlo Abrate, Federico Siciliano, Francesco Bonchi, and Fabrizio Silvestri</i>	
COIN: Counterfactual Inpainting for Weakly Supervised Semantic Segmentation for Medical Images .....	39
<i>Dmytro Shvetsov, Joonas Ariva, Marharyta Domnich, Raul Vicente, and Dmytro Fishman</i>	
Enhancing Counterfactual Explanation Search with Diffusion Distance and Directional Coherence .....	60
<i>Marharyta Domnich and Raul Vicente</i>	
CountARFactuals – Generating Plausible Model-Agnostic Counterfactual Explanations with Adversarial Random Forests .....	85
<i>Susanne Dandl, Kristin Blesch, Timo Freiesleben, Gunnar König, Jan Kapar, Bernd Bischl, and Marvin N. Wright</i>	
Causality-Aware Local Interpretable Model-Agnostic Explanations .....	108
<i>Martina Cinquini and Riccardo Guidotti</i>	
Evaluating the Faithfulness of Causality in Saliency-Based Explanations of Deep Learning Models for Temporal Colour Constancy .....	125
<i>Matteo Rizzo, Cristina Conati, Daesik Jang, and Hui Hu</i>	
CAGE: Causality-Aware Shapley Value for Global Explanations .....	143
<i>Nils Ole Breuer, Andreas Sauter, Majid Mohammadi, and Erman Acar</i>	
<b>Fairness, Trust, Privacy, Security, Accountability and Actionability in eXplainable AI</b>	
Exploring the Reliability of SHAP Values in Reinforcement Learning .....	165
<i>Raphael C. Engelhardt, Moritz Lange, Laurenz Wiskott, and Wolfgang Konen</i>	



Categorical Foundation of Explainable AI: A Unifying Theory .....	185
<i>Francesco Giannini, Stefano Fioravanti, Pietro Barbiero, Alberto Tonda, Pietro Liò, and Elena Di Lavore</i>	
Investigating Calibrated Classification Scores Through the Lens of Interpretability .....	207
<i>Alireza Torabian and Ruth Urner</i>	
XentricAI: A Gesture Sensing Calibration Approach Through Explainable and User-Centric AI .....	232
<i>Sarah Seifi, Tobias Sukianto, Maximilian Strobel, Cecilia Carbonelli, Lorenzo Servadei, and Robert Wille</i>	
Toward Understanding the Disagreement Problem in Neural Network Feature Attribution .....	247
<i>Niklas Koenen and Marvin N. Wright</i>	
ConformaSight: Conformal Prediction-Based Global and Model-Agnostic Explainability Framework .....	270
<i>Fatima Rabia Yapicioglu, Alessandra Stramiglio, and Fabio Vitali</i>	
Differential Privacy for Anomaly Detection: Analyzing the Trade-Off Between Privacy and Explainability .....	294
<i>Fatima Ezzeddine, Mirna Saad, Omran Ayoub, Davide Andreoletti, Martin Gjoreski, Ihab Sbeity, Marc Langheinrich, and Silvia Giordano</i>	
Blockchain for Ethical and Transparent Generative AI Utilization by Banking and Finance Lawyers .....	319
<i>Swati Sachan, Vinicius Dezem, and Dale Fickett</i>	
Multi-modal Machine Learning Model for Interpretable Malware Classification .....	334
<i>Fahmida Tasnim Lisa, Sheikh Rabiul Islam, and Neha Mohan Kumar</i>	
Explainable Fraud Detection with Deep Symbolic Classification .....	350
<i>Samantha Visbeek, Erman Acar, and Floris den Hengst</i>	
Better Luck Next Time: About Robust Recourse in Binary Allocation Problems .....	374
<i>Meirav Segal, Anne-Marie George, Ingrid Chieh Yu, and Christos Dimitrakakis</i>	
Towards Non-adversarial Algorithmic Recourse .....	395
<i>Tobias Leemann, Martin Pawelczyk, Bardh Prenkaj, and Gjergji Kasneci</i>	

Communicating Uncertainty in Machine Learning Explanations:  
A Visualization Analytics Approach for Predictive Process Monitoring ..... 420  
*Nijat Mehdiyev, Maxim Majlatow, and Peter Fettke*

XAI for Time Series Classification: Evaluating the Benefits of Model  
Inspection for End-Users ..... 439  
*Brigt Håvardstun, Cèsar Ferri, Kristian Flikka, and Jan Arne Telle*

**Author Index** ..... 455

# **Counterfactual Explanations and Causality for eXplainable AI**



# Sub-SpaCE: Subsequence-Based Sparse Counterfactual Explanations for Time Series Classification Problems

Mario Refoyo<sup>(✉)</sup>  and David Luengo 

Universidad Politécnica de Madrid (UPM), C/Nikola Tesla s/n, 28031 Madrid, Spain  
m.refoyo@upm.es

**Abstract.** The interpretation of existing machine learning models has become a critical task to facilitate the widespread adoption of AI across different domains, leading to the emergent field of eXplainable AI (XAI). However, dedicated approaches for time series data have received limited attention compared to XAI methods for images or tabular data. Moreover, current approaches often overlook the unique challenges present in time series classification problems. In this paper, we introduce Subsequence-based Sparse Counterfactual Explanations (Sub-SpaCE), a novel method tailored for time series classification problems. Sub-SpaCE employs genetic algorithms, with customized mutation and initialization processes, promoting changes in a small number of subsequences to generate highly sparse and plausible counterfactual explanations. Our empirical evaluations on various datasets demonstrate Sub-SpaCE's excellent performance, achieving a good balance between sparsity and plausibility in counterfactual explanations for time series data.

**Keywords:** eXplainable Artificial Intelligence (XAI) · Counterfactual Explanations · Genetic Algorithm Optimization · Time Series Classification

## 1 Introduction

In recent years, the field of Machine Learning, and especially Deep Learning, has witnessed remarkable advancements, significantly impacting real-world applications [21]. The increased capacity and complexity of these models, however, introduce a notable challenge: their interpretability. As a result, the field of eXplainable Artificial Intelligence (XAI) has emerged to enhance human understanding of these models [17]. The utilization of XAI methods is always desirable, and can even become imperative in critical domains such as healthcare, financial and military applications.

Despite the increasing attention to XAI, there is a noted gap in the development of methods tailored for time series data compared to tabular or image data [19, 20]. Time series data, commonly used in specialized domains, requires expert knowledge, hindering the development of practical explanation methods

[22]. Regardless of their particular nature, most of the methods designed to work with time series are adaptations of the popular methods used in computer vision [19]. That approximation has already been noted as not appropriate for the time-ordered structure characteristic of time series inputs [14].

This trend extends to counterfactual explanations, a particular type XAI approach that tries to find the minimum changes that should be applied to an original input in order to modify the classification outcome of a black-box model [26]. Counterfactuals stand out by their alignment with human cognitive processes, offering alternative scenarios to explain model outputs and aiding in the detection of cause-effect relations [4]. Despite the interest and success of counterfactuals, their adaptation to time series problems remains an active area of research, with only a handful of recently proposed methods. These approaches often focus on modifying inputs with contiguous sequences of changes, driven by the observation that most time series classification problems can be solved through the identification of class-specific patterns [9].

However, relying on a single sequence of changes, as most methods do, or not limiting the number of independent subsequences of changes, can lead to a decrease in the plausibility and/or understandability of the counterfactual. Our novel and key observation is that the reduction of both, the number of points changed and the number of subsequences used, must be performed jointly, something that none of the current methods does. On the other hand, current approaches leverage generative models [15, 18] or prototype losses [10] to guide the search away from outlier examples. However, this often drives counterfactuals to the center of the data manifold, neglecting the fact that the original input might be far away from that center (or even that it could be out-of-sample), thus increasing the total amount of changes needed to reach a plausible counterfactual.

In response to these challenges, we introduce Subsequence-based Sparse Counterfactual Explanations (Sub-SpaCE), a new counterfactual explanation method tailored for time series classification problems that focuses on providing highly sparse counterfactuals in the form of contiguous subsequences of changes to the original input, while also mitigating the negative impact of plausibility on the number of changes. Our contributions include designing a new loss function emphasizing minimum changes and contiguity, developing a plausibility term less conflicting with sparsity, and implementing a genetic algorithm with novel initialization and mutation processes to achieve better convergence. Through empirical evaluation of several datasets from the UCR repository (see Sect. 4), we demonstrate Sub-SpaCE’s ability to achieve an excellent balance between sparsity, contiguity, and plausibility in counterfactual explanations.

## 2 Related Work

The most common and basic definition of counterfactual explanation is tailored to classification problems with binary labels. Given an input instance,  $x \in X$ , and predictive black-box model,  $b: X \rightarrow [0, 1]$ , that produces the output  $y = b(x)$ ,

a counterfactual explanation is the smallest variation of the original instance  $x'$  that is able to change the predicted outcome  $y' = b(x')$  in such a way that  $y' \neq y$  [12]. Initially, the proposed algorithms treated counterfactual generation as optimization problems, primarily emphasizing the proximity between  $x$  and  $x'$ . However, researchers soon started to focus on designing methods that generate explanations that fulfill additional properties, such as plausibility, sparsity, or diversity (see [12, 25] for a detailed review).

The vast majority of these methods are designed for their application in tabular data sets, characterized by lower dimensionality. While extending these methods to high-dimensional problems is feasible, and has been the typical approach for time series classification problems, the resulting counterfactuals are usually challenging to interpret [7] due to numerous individual time steps being altered without forming coherent sequences. To avoid this issue, the contiguity of changes is considered an additional counterfactual desideratum in time series problems, as it can enhance the understandability of explanations [7, 10, 24].

An early work offering contiguous subsequences of changes as an explanation is Native Guide [7], which finds the nearest unlike neighbor (NUN) of the input sample  $x$  -the closest existing instance of the desired class- and substitutes its shortest and most relevant subsequence into the original sample to create a counterfactual. Boubrahimi *et al.* [10] improve on Native Guide by optimizing the values of the most relevant subsequence of the input sample  $x$ , instead of substituting the subsequence of the NUN. This method also incorporates an additional term in the objective function to guide the search towards plausible solutions. Additionally, Ates *et al.* [1] extend Native Guide to multivariate time series. They present an optimization problem to decide which -entire- variables from the original sample should be substituted by values from the NUN. However, none of these works consider the use of multiple subsequences of changes, which might be too limiting for complex datasets.

Recent research has focused specifically on modifying several subsequences. Some works leverage the concept of the shapelets [2, 3] to this end. They find the most relevant shapelets of a class, spatially locate them in the input to explain, and substitute them with the values of the NUN to alter the classification outcome. Another notable approach is LASTS [23], which works in the latent space of a Variational Autoencoder (VAE) to generate a set of exemplars and counterfactuals in the neighborhood of the instance to explain. Then, they derive classification rules based on the apparition of specific shapelets. Although these methods are capable of generating explanations based on multiple subsequences, they neither optimize for sparsity nor limit the number of different sub-sequences used. This can easily result in counterfactuals with unnecessary changes with respect to the original input, thus hindering the assimilation of the explanation. This limitation also extends to AB-CF [16], which, instead of using shapelets, creates a set of fixed-length subsequences using a sliding window over the instance to explain. Subsequently, padding is added, the subsequences are passed to the black-box model to discover the most relevant ones to the current class, and they are replaced by the values of the NUN. Finally, TSEvo [13] is the

only method optimizing for sparsity and plausibility while using sub-sequences of changes obtained from the training dataset. However, it does not minimize the number of subsequences employed, and the mutation strategies that it uses result in extremely high execution times, even for a single explanation.

In this paper, we present Subsequence-based Sparse Counterfactual Explanations (Sub-SpaCE). Our approach generates counterfactuals by using multiple sub-sequences of changes, jointly minimizing the amount of sub-sequences and the global sparsity of the explanation. It also accounts for plausibility of the generated counterfactuals by incorporating a custom term in the objective function. Sub-SpaCE employs a genetic algorithm with initialization and mutation techniques specifically tailored to this problem.

### 3 Proposed Method

#### 3.1 Problem Formulation

Let  $x = \{x_{1:L}\} \in \mathbb{R}^L$  be the time series to explain, where  $L$  is its length,  $m = \{m_{1:L}\} \in \{0, 1\}^L$  be the binary mask that specifies the points that will be changed to generate the counterfactual  $x'$  by using a generic function  $f: \mathbb{R}^L \rightarrow \mathbb{R}^L$  such that  $x' = f(x|m)$ . In this work, we adopt the concept of the Nearest Unlike Neighbor (NUN) proposed by Delaney *et al.* [7] to define the form of  $f$ . Letting  $x^n = \{x_{1:L}^n\} \in \mathbb{R}^L$  be the Nearest Unlike Neighbor (NUN), found as the Nearest Neighbor of a different class of the original sample in the Euclidean space, we can define ( $\forall i \in \{1, \dots, L\}$ ):

$$x'_i = f(x_i|m_i, x_i^n) = \begin{cases} x_i^n, & \text{if } m_i = 1; \\ x_i, & \text{else.} \end{cases}$$

The main objective is to find the lowest number of changes in the original signal that modify the classification outcome of the black-box classifier,  $b: X \rightarrow Y$ , to the true label assigned to the NUN:  $b(x') = b(x^n) = y^n$ . We also assume that we can access each class classification probability,  $p_b(x, y)$ . As highlighted in the previous section, other desirable properties are also pursued. In particular, we focus on acquiring sparse, plausible and structurally meaningful counterfactuals for time series data in the form of contiguous subsequences of changes. To achieve this, the following optimization problem is solved:

$$\begin{aligned} \min_m \quad & -\alpha L_{adv} + \beta L_{spa} + \eta L_{sub} + \lambda L_{os} \\ \text{s.t.} \quad & b(x') = y^n \\ & m_i \in \{0, 1\} \quad \forall i \in \{1, \dots, L\} \end{aligned} \tag{1}$$

where  $\alpha$ ,  $\beta$ ,  $\eta$ , and  $\lambda$  are parameters that control the weight given to each term in the objective function, and where:

- $L_{adv} = p_b(x', y^n)$  guides the counterfactual search to the desired  $y^n$  class by maximizing its output probability, which is by definition bound to  $[0, 1]$ . This

term is widely used in optimization-based counterfactual generation methods as a relaxation of the class restriction [12]. We also use it to guide the convergence when the solution does not lie within the feasible region.

- $L_{spa} = \frac{\|m\|_0}{L}$ , where  $\|m\|_0$  indicates the  $\ell_0$  pseudo-norm of  $m$  (i.e., the number of non-zero elements in  $m$ ), leads the solution to sparse counterfactuals, where the mask  $m$  is activated only in a few time steps. The term is scaled by the length of the time series, so the range is  $[0, 1]$ .
- $L_{sub} = \left(\frac{\sum_{i=2}^L \mathbb{1}_i}{L/2}\right)^\gamma$ , where  $\mathbb{1}_i$  is an indicator function that takes the value 1 whenever a new sub-sequence begins (i.e.,  $m_{i-1} = 0$  and  $m_i = 1$ ). The contiguity of the changes is also desired for a time series problem, as the counterfactual would be easier to comprehend for a human. Therefore, this term seeks to minimize the number of individual sub-sequences that appear in the solution. Again, the loss range is scaled to  $[0, 1]$  by dividing by the maximum number of possible sub-sequences,  $L/2$ . Additionally, the  $\gamma$  hyperparameter sets the way in which the loss evolves with the number of existing sub-sequences. In the case of  $\gamma = 1$ , the value of the loss increases linearly with respect to the normalized number of sub-sequences, while values  $\gamma < 1$  or  $\gamma > 1$  change the relation to be convex or concave, respectively.
- $L_{os}$  is used to ensure that the counterfactuals lie in the same data manifold as the original examples. Multiple approaches have been proposed in the literature to achieve this. In this work, we follow the approach of [24], by leveraging an additional autoencoder,  $f_{AE} : \mathbb{R}^L \rightarrow \mathbb{R}^L$ , trained to reconstruct the same training set used to train the black-box classifier. The  $\ell_2$  reconstruction error,  $\|x' - f_{AE}(x')\|_2$ , can then be interpreted as an outlier score metric which, when used as a loss function, will penalize out-of-distribution counterfactuals. However, instead of using the reconstruction error directly, we calculate the increment in the reconstruction error of  $x$  with respect to the reconstruction error of the original input:  $L_{os} = \frac{e(x, x')}{e_{max}}$ , where the term  $e(x, x')$  is the increment in the outlier score,  $e(x, x') = \min(0, \|x' - f_{AE}(x')\|_2 - \|x - f_{AE}(x)\|_2)$ , and  $e_{max}$  is the maximum value of the AE’s error on the training set, used to scale the values of the loss term to  $[0, 1]$ . Using this approach, we normalize the metric with respect to the examples, acknowledging that some of them can actually be far away from the data manifold learned by the autoencoder (i.e., they can be outliers).

All the loss terms are scaled to be within  $[0, 1]$ , to ease their weight balancing through the parameters  $\alpha$ ,  $\beta$ ,  $\eta$ , and  $\lambda$ , whose sum must be equal to one.

### 3.2 Sub-SpaCE: Subsequence-Based Sparse Counterfactual Explanations

We propose to use a genetic algorithm to solve the optimization problem outlined in (1), as their efficiency and adaptability make them a compelling choice to tackle complex, high-dimensional problems in which the solution should be produced in a reasonable amount of time.



When working with genetic algorithms, the convention is to define a fitness function, which is typically maximized. Also, the constraints are not natively supported, so they must be relaxed and added to the fitness function. Consequently, we have derived the following fitness function from (1):

$$\mathcal{F}(m) = \alpha L_{adv} - \beta L_{sparsity} - \eta L_{subsequences} - \lambda L_{os} - \nu \cdot \mathbb{1}_{class}(x, m, y^n) \quad (2)$$

where  $\mathbb{1}_{class}(x, m, y^n)$  is an indicator function that takes a value equal to 1 when the black-box classifier does not classify the counterfactual as the desired class. Note that the original restriction for obtaining the desired class is now a penalization term, where  $\nu$  is a scalar set to a large integer value. Note also how the signs are inverted because a fitness function is typically maximized.

In its simplest form, a genetic algorithm starts from a random population of candidate solutions, which are iteratively mixed, simulating reproduction (*crossover*), and mutated to achieve higher values of the fitness score [11]. In this work, we used a simple implementation that also incorporates *elitism* to preserve the best individuals of each iteration [8]. The rest of the population undergoes the standard steps of a genetic algorithm, including parent selection, crossover, and mutation. The iterative process continues until the specified number of generations (iterations) is achieved. For parent selection and crossover, we employ simple methods: roulette wheel and single point crossover respectively [11]. The rest of the steps include adaptations tailored to this problem, which are detailed in the following.

**Initialization.** Let  $P \in \mathbb{R}^{N \times L}$  be the initial population, where each row can be interpreted as a candidate solution to the mask  $m$ . In our binary setting for  $m$ , a basic random initialization would assign 1’s to random positions in  $P$ . The initialization step is a way of introducing prior knowledge about the solution, and a way of reducing the number of iterations required to reach a local maximum of the fitness function. A natural way to incorporate prior knowledge about the correct solution is to leverage feature-attribution methods, which indicate the relevance of different time steps to the black-box classifier [17], something already exploited by Native Guide [7] and TimeX [10].

Let  $f_a : \mathbb{R}^L \rightarrow \mathbb{R}^L$  be a generic function that represents the feature importance generation of an input  $x$  (obtained using GradCAM++ in our case, see Sect. 4). The idea is to increase the probability of setting to 1 the values of  $m$  where the arithmetic mean of the importance of the original input  $x$  and the NUN  $x^n$  is higher:  $A = \frac{f_a(x) + f_a(x^n)}{2A_{\max}} \in \mathbb{R}^L$ , where  $A_{\max}$  is the maximum value of  $f_a$  over the whole dataset. To initialize the population, we take that importance as the initial value of each individual, and add Gaussian noise,  $\epsilon \sim \mathcal{N}(0, 1)$ , to introduce diversity. Subsequently, we set to 1 the  $h\%$  most activated values and set to 0 the rest, resulting in the initial population  $P \in \mathbb{R}^{N \times L}$ .

The choice of  $h$  influences the initialization: a low value would lead to starting the counterfactual search from an already sparse solution, which would result in highly sparse explanations. However, starting from an already sparse solution can also make it difficult to find a valid counterfactual in cases where the true regions

of interest to generate the counterfactuals are distributed across multiple sections of the input. To automatically adapt the initialization to the complexities of the datasets and the feature importance of individual samples, we dynamically adapt the value of  $h$  during the optimization: we begin with a low value ( $h = 20$ ) and, if a valid solution has not been found within 50 iterations, reinitialize the algorithm with a value 20 points higher than the previous initialization.

**Subsequence-Based Mutation.** The most naive mutation approach consists of randomly changing the values of the candidate mask with a low probability, usually controlled by a hyperparameter. For this particular problem, a random mutation is likely to add new independent sequences to  $m$ , which would result in abrupt changes in the  $L_{sub}$  term that could slow convergence as demonstrated in the ablation study shown in Appendix 4.3. Therefore, we propose a modification to restrict the mutation to the points where the sequences of ones can be extended or shortened. That limits the exploration of the mutation step toward solutions with a similar amount of subsequences, increasing the stability and convergence properties of the algorithm.

## 4 Numerical Experiments

### 4.1 Set Up

**Datasets.** We selected several publicly available data sets from the University of California at Riverside (UCR) Archive [6], which is widely used in the evaluation of XAI approaches for time series, to test the proposed method: CBF, Chinatown, Coffee, ECG200 and Gunpoint. The selected univariate time series data sets are characterized by different lengths, classes, and types of patterns, and are the same ones used by Delaney *et al.* [7].

**Baseline Methods.** The proposed method will be compared with three open-source approaches proposed in the literature: (i) **W-CF**, proposed in [26], is one of the first methods of counterfactual explanations. It complies with the basic definition of counterfactuals and only takes into account the proximity property during the search. We use the implementation available in the Native Guide repository; (ii) **Native Guide**, proposed by [7], is the first method that directly searches for contiguity in the explanations. We use the original implementation, released by the authors; (iii) **AB-CF**, proposed in [16], a recent work that generates the counterfactuals using multiple subsequences of changes. Again, we use the author’s implementation with the original parameters.

**Evaluation Metrics.** Multiple metrics have been proposed in the literature to assess the quality of the generated counterfactuals:

- **Validity:** measured as the percentage of counterfactuals that change the original output class:  $\frac{1}{N} \sum_N \mathbb{1}_{class}$ , where  $\mathbb{1}_{class}$  is equal to 1 when the output class of the original instance and the counterfactual’s class differ.

- **Proximity**: quantified with the  $\ell_2$  distance between the original sample  $x$  and the counterfactual  $x'$ .
- **Sparsity**: evaluated as the number of changes of the counterfactual, normalized by the total length of the time series. A lower score is preferred. This aligns with one of the loss terms of equation (1):  $\frac{\|m\|_0}{L}$ .
- **Increment in Outlier Score (IOS)** to quantify plausibility as depicted in equation (1):  $\min(0, \|x' - f_{AE}(x')\|_2 - \|x - f_{AE}(x)\|_2) / e_{\max}$ . Ideally, the increment should be close to 0.
- **Number of Subsequences (NoS)** of changes in the counterfactual:  $\sum_{i=2}^L \mathbb{1}_i$ , where  $\mathbb{1}_i$  equals 1 when a new subsequence of changes is present in the counterfactual  $x'$ . The lower the number of subsequences, the easier it is to understand the explanation.

**Implementation Details.** For each of the data sets, a black-box classifier,  $b$ , and an Autoencoder,  $f_{AE}$ , are trained using the TensorFlow framework<sup>1</sup>. The black-box classifier is the same one used by Delaney *et al.* [7] in their experiments: a fully convolutional neural network. Regarding the feature importance computation ( $f_a$ ), the well-known feature attribution method GradCAM++ is used [5]. The choice resides in the ease of implementation and the fast generation of explanations that characterizes this method. Nonetheless, any other feature attribution method could be used as well. On the other hand, the values given to the hyperparameters of the genetic algorithm are shared between instances and data sets and are set empirically. The population size is set to 100 individuals and the maximum number of iterations is also set to 100. The probability of mutation is set to 5% and the weights of the loss function are set to  $\alpha = 0.2, \beta = 0.24, \eta = 0.36$ , and  $\lambda = 0.2$ . Also, since every term in the fitness function is scaled to the range  $[0, 1]$ , we have noticed that  $\nu = 100$  is large enough for the wrong class penalization term to behave like the original restriction. Finally, to quickly penalize the increment in the number of sub-sequences in the solution, we set  $\gamma = 0.25$ .

## 4.2 Results

This section shows the comparison of the proposed method with the baselines across the five data sets used in our experiments. Average metrics for counterfactuals on the complete test sets are summarized in Table 1 (with the performance of the best method highlighted in boldface and the one of the second best underlined). The table is partitioned into five sub-tables, with each one corresponding to a specific metric.

The proposed method is capable of delivering valid counterfactuals that outperform the baselines in terms of sparsity, consistently ranking as the top performer across data sets and exhibiting substantial improvements with respect to W-CF, NG and AB-CF. It also achieves an excellent performance in terms

<sup>1</sup> The code of Sub-SpaCE, the experiments, and the results can be found on <https://github.com/MarioRefoyo/Sub-SpaCE>.

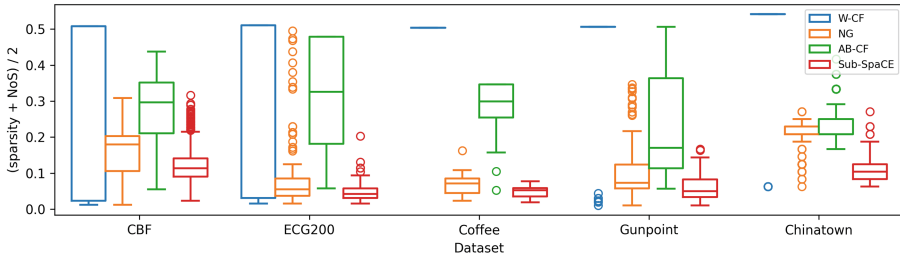
**Table 1.** Results of method evaluation on the complete test set.

Method	CBF	Chinatown	Coffee	ECG200	Gunpoint
Validity					
W-CF	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
NG	<u>0.96 ± 0.19</u>	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
AB-CF	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	<u>0.94 ± 0.24</u>	1.00 ± 0.00
Sub-SpaCE	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Sparsity					
W-CF	0.68 ± 0.46	0.97 ± 0.18	1.00 ± 0.00	0.61 ± 0.48	0.94 ± 0.23
NG	<u>0.31 ± 0.13</u>	<u>0.33 ± 0.08</u>	<u>0.13 ± 0.06</u>	<u>0.17 ± 0.22</u>	<u>0.19 ± 0.15</u>
AB-CF	0.52 ± 0.18	0.38 ± 0.07	0.56 ± 0.16	0.59 ± 0.31	0.43 ± 0.27
Sub-SpaCE	<b>0.20 ± 0.09</b>	<b>0.12 ± 0.04</b>	<b>0.08 ± 0.03</b>	<b>0.07 ± 0.04</b>	<b>0.09 ± 0.06</b>
Proximity ( $\ell_2$ )					
W-CF	6.10 ± 13.22	662.70 ± 211.41	<u>1.10 ± 0.25</u>	<b>2.22 ± 1.71</b>	<b>1.00 ± 1.04</b>
NG	<b>5.60 ± 1.44</b>	<b>393.09 ± 124.49</b>	1.26 ± 0.24	2.62 ± 1.48	2.34 ± 1.74
AB-CF	7.46 ± 1.97	430.97 ± 123.65	1.38 ± 0.36	3.77 ± 1.85	2.92 ± 1.86
Sub-SpaCE	<u>5.88 ± 1.40</u>	<u>399.87 ± 120.52</u>	<b>1.00 ± 0.25</b>	<u>2.41 ± 1.30</u>	<u>1.64 ± 1.12</u>
IOS					
W-CF	0.24 ± 0.56	0.19 ± 0.14	0.06 ± 0.04	0.12 ± 0.14	<u>0.09 ± 0.09</u>
NG	<u>0.10 ± 0.09</u>	<u>0.02 ± 0.04</u>	0.04 ± 0.04	<u>0.07 ± 0.11</u>	0.14 ± 0.16
AB-CF	0.13 ± 0.11	<u>0.02 ± 0.04</u>	<b>0.01 ± 0.02</b>	0.09 ± 0.11	0.11 ± 0.11
Sub-SpaCE	<b>0.07 ± 0.06</b>	<b>0.02 ± 0.03</b>	<u>0.02 ± 0.03</u>	<b>0.05 ± 0.06</b>	<b>0.05 ± 0.05</b>
NoS					
W-CF	<u>1.14 ± 0.44</u>	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>	<u>1.21 ± 0.48</u>	<u>1.08 ± 0.38</u>
NG	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>	<b>1.00 ± 0.00</b>
AB-CF	2.46 ± 0.75	1.10 ± 0.31	1.54 ± 0.74	1.61 ± 0.78	1.98 ± 0.52
Sub-SpaCE	2.39 ± 0.93	1.21 ± 0.41	3.07 ± 0.86	1.31 ± 0.61	2.19 ± 0.81

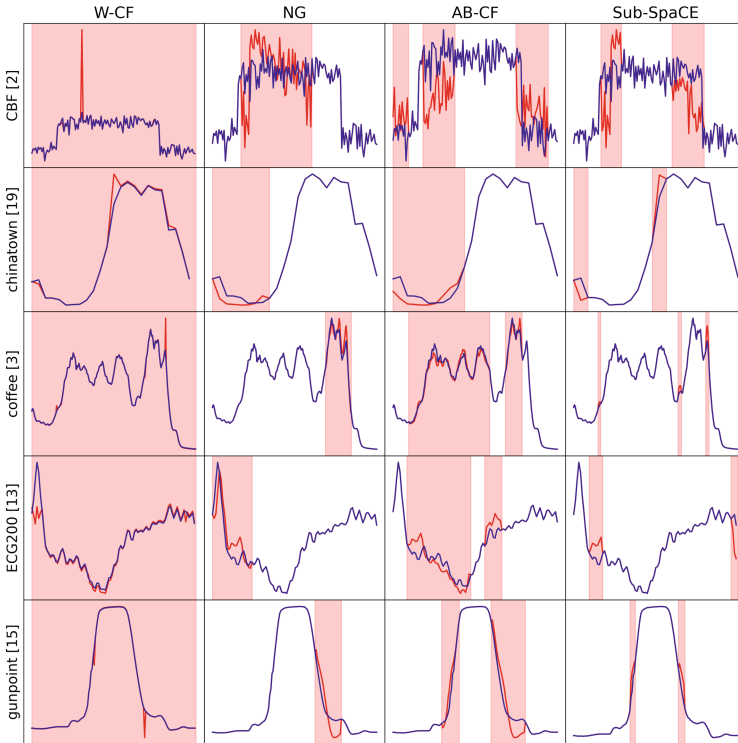
of plausibility (as measured by the IOS), although the improvements are less significant, especially for the Coffee (in which it is the second best) and Chinatown data sets. Regarding proximity metric, W-CF or NG are usually the best performers, while Sub-SpaCE takes the second position in general, and occupies the first position in the Coffee dataset. This is primarily due to Sub-SpaCE not directly optimizing the proximity during the counterfactual generation.

Finally, the number of sub-sequences modified is crucial to the understandability of the generated explanations. Therefore, sparsity and the number of subsequences must be comprehended jointly: an increase in the number of subsequences should only be performed if this leads to a reduction in the total number of points being changed, while a large amount of subsequences should also be discouraged. To support this intuition, we can compute a simple metric that summarizes the interaction between sparsity and the number of subsequences: the arithmetic mean between their max-scaled values. Figure 1 shows that comparison using a box plot. Our proposed method is consistently able to obtain the smallest value, indicating a good balance between sparsity and the number

of subsequences. Figure 2 shows examples where this behavior is evident when comparing W-CF, NG, AB-CF, and Sub-SpaCE.



**Fig. 1.** Arithmetic mean between sparsity and scaled number of subsequences.



**Fig. 2.** Examples of counterfactual explanations. The red and blue lines represent the original input and the generated counterfactual respectively. The background in red indicates the timestamps in which the counterfactual value differs from the original one. (Color figure online)

### 4.3 Ablation Study

In this section, we show the results of an ablation study that demonstrates the effectiveness of the proposed mutation and initialization processes. We study the results of Sub-SpaCE under different configurations:

- **Sub-SpaCE (Basic)**. The proposed method is in its simplest form, without including the proposed mutation or the initialization processes. That is, both the initialization and the mutation are completely random.
- **Sub-SpaCE (Mut.)**. The proposed method, including the mutation process and excluding the initialization based on feature attribution methods. That is, the populations are initialized completely at random.
- **Sub-SpaCE (Init.)**. The proposed method without the adaptation in the mutation process, but including the initialization based on feature attribution methods. The mutation is now completely at random.
- **Sub-SpaCE**. The full version of the proposed method, the one used for the experiments in Sect. 4.2.

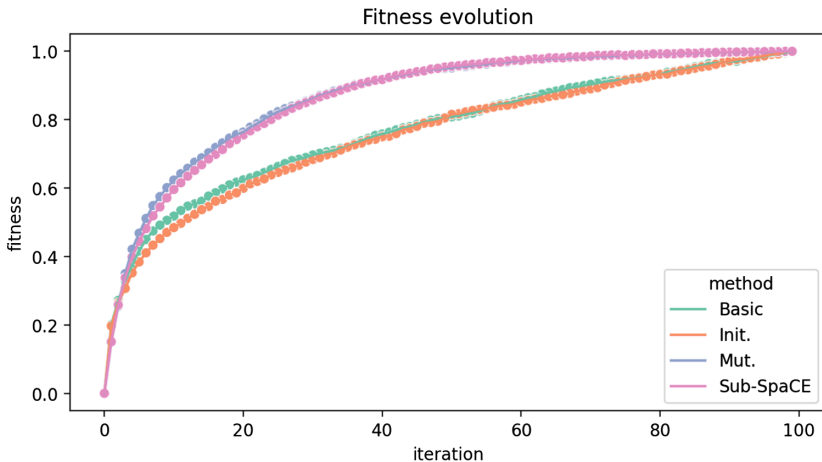
We employed the same metrics as those utilized in the experiments outlined in Sects. 4.1 and 4.2. The outcomes for each configuration are presented in Table 2. The results highlight how the proposed mutation significantly improves the performance with respect to the Basic configuration, even surpassing the full version of Sub-SpaCE in most of the metrics. However, this improvement comes at the expense of not achieving a perfect validity score in some data sets, something critical for counterfactual explanations. On the other hand, the proposed initialization always results in an improvement across metrics and data

**Table 2.** Ablation study results.

Dataset	Method	Sparsity	Proximity ( $L_2$ )	Validity	IOS	NoS
CBF	Basic	$0.33 \pm 0.09$	$6.23 \pm 1.28$	$0.96 \pm 0.21$	<b><math>0.05 \pm 0.06</math></b>	$21.27 \pm 2.79$
	Init	$0.30 \pm 0.10$	$6.21 \pm 1.35$	<b><math>1.00 \pm 0.00</math></b>	$0.06 \pm 0.06$	$19.74 \pm 3.44$
	Mut	<b><math>0.15 \pm 0.05</math></b>	<b><math>5.05 \pm 1.07</math></b>	$0.61 \pm 0.49$	$0.06 \pm 0.05$	<u><math>2.54 \pm 1.07</math></u>
	Sub-SpaCE	<u><math>0.20 \pm 0.09</math></u>	<u><math>5.83 \pm 1.45</math></u>	<b><math>1.00 \pm 0.00</math></b>	<b><math>0.05 \pm 0.06</math></b>	<b><math>2.42 \pm 0.96</math></b>
Chinatown	Basic	$0.13 \pm 0.05$	$411.48 \pm 131.24$	<b><math>1.00 \pm 0.00</math></b>	$0.02 \pm 0.03$	$1.40 \pm 0.51$
	Init	$0.13 \pm 0.04$	$401.1 \pm 123.88$	<b><math>1.00 \pm 0.00</math></b>	$0.02 \pm 0.03$	$1.32 \pm 0.48$
	Mut	<b><math>0.12 \pm 0.04</math></b>	<b><math>396.66 \pm 102.47</math></b>	$0.99 \pm 0.08$	$0.02 \pm 0.03$	<b><math>1.21 \pm 0.40</math></b>
	Sub-SpaCE	<b><math>0.12 \pm 0.04</math></b>	<u><math>399.57 \pm 120.63</math></u>	<b><math>1.00 \pm 0.00</math></b>	$0.02 \pm 0.03$	<u><math>1.21 \pm 0.41</math></u>
Coffee	Basic	$0.30 \pm 0.07$	$1.22 \pm 0.23$	<b><math>1.00 \pm 0.00</math></b>	<b><math>0.02 \pm 0.03</math></b>	$51.64 \pm 6.98$
	Init	$0.25 \pm 0.06$	$1.16 \pm 0.22$	<b><math>1.00 \pm 0.00</math></b>	$0.03 \pm 0.03$	$47.07 \pm 7.19$
	Mut	<b><math>0.07 \pm 0.02</math></b>	<b><math>0.99 \pm 0.29</math></b>	$0.71 \pm 0.46$	<b><math>0.02 \pm 0.03</math></b>	<b><math>2.80 \pm 0.95</math></b>
	Sub-SpaCE	<u><math>0.08 \pm 0.03</math></u>	<u><math>1.02 \pm 0.25</math></u>	<b><math>1.00 \pm 0.00</math></b>	$0.03 \pm 0.04$	<u><math>2.86 \pm 1.01</math></u>
ECG200	Basic	$0.19 \pm 0.06$	$2.66 \pm 1.23$	<b><math>1.00 \pm 0.00</math></b>	$0.10 \pm 0.09$	$12.32 \pm 2.36$
	Init	$0.17 \pm 0.05$	$2.71 \pm 1.29$	<b><math>1.00 \pm 0.00</math></b>	$0.08 \pm 0.07$	$9.81 \pm 2.19$
	Mut	<b><math>0.06 \pm 0.04</math></b>	<b><math>2.29 \pm 1.18</math></b>	$0.99 \pm 0.1$	<b><math>0.06 \pm 0.07</math></b>	<u><math>1.38 \pm 0.67</math></u>
	Sub-SpaCE	<b><math>0.06 \pm 0.04</math></b>	<u><math>2.34 \pm 1.16</math></u>	<b><math>1.00 \pm 0.00</math></b>	<u><math>0.07 \pm 0.08</math></u>	<b><math>1.37 \pm 0.65</math></b>
Gunpoint	Basic	$0.26 \pm 0.10$	$1.94 \pm 1.28$	<b><math>1.00 \pm 0.00</math></b>	$0.13 \pm 0.12$	$23.15 \pm 4.24$
	Init	$0.24 \pm 0.10$	$2.00 \pm 1.19$	<b><math>1.00 \pm 0.00</math></b>	$0.13 \pm 0.12$	$21.13 \pm 4.93$
	Mut	<b><math>0.07 \pm 0.05</math></b>	<b><math>1.23 \pm 0.83</math></b>	$0.82 \pm 0.39$	<b><math>0.05 \pm 0.05</math></b>	<u><math>2.11 \pm 0.86</math></u>
	Sub-SpaCE	<u><math>0.09 \pm 0.05</math></u>	<u><math>1.57 \pm 1.07</math></u>	<b><math>1.00 \pm 0.00</math></b>	<b><math>0.05 \pm 0.05</math></b>	<b><math>2.09 \pm 0.79</math></b>

sets, although its impact is marginal with respect to the mutation adaptation. Notably, the validity of counterfactuals is consistently maintained at the highest possible score. The combination of both configurations (Sub-SpaCE) strikes the desired balance, achieving a perfect validity score while benefiting from the improvement in other metrics facilitated by the mutation adaptation.

Another interesting way of comparing the settings is by assessing the rate of convergence to the solutions. To accomplish this, we exclusively consider valid counterfactuals, storing the highest fitness score for each sample in every iteration, and normalizing the curve by the last fitness value corresponding to the given solution. This normalization eases the comparison between different settings and data sets. Note that the focus is not comparing the quality of the solution, as in Table 2, but rather inspecting the algorithm’s convergence towards that solution. Figure 3 illustrates the average normalized fitness evolution across data sets. The mutation adaptation is observed to significantly increase the convergence velocity of the algorithm, while initialization does not provide any improvement in this sense. Additionally, we show the initial fitness values across data sets in Fig. 4. As expected, the initialization results in better starting fitness scores. By jointly analyzing both figures, we can observe how Sub-SpaCE maintains the rate of convergence of the mutation configuration, while also benefiting from the better starting points achieved by the proposed initialization, thus demonstrating the effectiveness of this combination that leads to the performance depicted in Table 1.



**Fig. 3.** Average normalized fitness score evolution with the number of iterations.

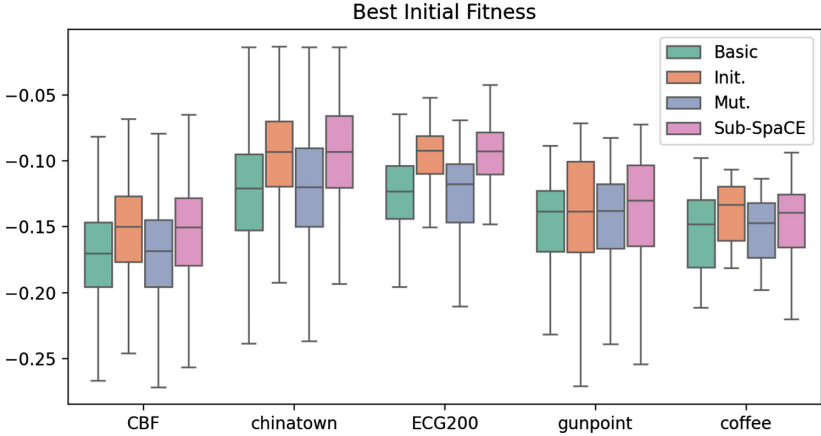


Fig. 4. Box plot of the highest fitness score at the first iteration for every configuration.

## 5 Conclusions and Future Work

Our work addresses a gap in XAI applied to time series problems, where the form of the explanations plays a key role in their understandability. The proposed method, Sub-SpaCE, employs a genetic algorithm, with custom mutation and initialization, to maximize an innovative fitness function, tailored for the generation of highly sparse counterfactuals, while also imposing the explanation to be formed by a small number of contiguous subsequences of changes. The performed experiments demonstrate that Sub-SpaCE attains an excellent compromise between sparsity and plausibility in counterfactual explanations, while also respecting the sequential nature of time series data.

As part of future work, we intend to conduct more comprehensive experiments using additional and diverse datasets, as well as explore more thoroughly the impact of the initialization and mutation strategies developed. We also plan to extend our approach to address multivariate problems. Another interesting avenue would be the inclusion of generative models to generate the counterfactual values in those positions specified by the mask, instead of relying on a Nearest Unlike Neighbor.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.




## References

1. Ates, E., Aksar, B., Leung, V.J., Coskun, A.K.: Counterfactual explanations for multivariate time series. In: International Conference on Applied Artificial Intelligence (ICAPAI), pp. 1–8 (2021)
2. Bahri, O., Boubrahimi, S.F., Hamdi, S.M.: Shapelet-based counterfactual explanations for multivariate time series. arXiv preprint [arXiv:2208.10462](https://arxiv.org/abs/2208.10462) (2022)
3. Bahri, O., Li, P., Boubrahimi, S.F., Hamdi, S.M.: Temporal rule-based counterfactual explanations for multivariate time series. In: 21st IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1244–1249 (2022)
4. Byrne, R.M.J.: Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI), pp. 6276–6282 (2019)
5. Chattopadhyay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Gradcam++: generalized gradient-based visual explanations for deep convolutional networks. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 839–847 (2018)
6. Chen, Y., et al.: The UCR time series classification archive. arXiv preprint [arXiv:1810.07758](https://arxiv.org/abs/1810.07758) (2015)
7. Delaney, E., Greene, D., Keane, M.T.: Instance-based counterfactual explanations for time series classification. In: Case-Based Reasoning Research and Development: 29th International Conference (ICCBR), pp. 32–47 (2021)
8. Dumitrescu, D., Lazzarini, B., Jain, L., Dumitrescu, A.: Evolutionary Computation. CRC Press (2000)
9. Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. *Data Min. Knowl. Disc.* **33**, 917–963 (2018)
10. Filali Boubrahimi, S., Hamdi, S.M.: On the mining of time series data counterfactual explanations using barycenters. In: 31st ACM International Conference on Information and Knowledge Management (CIKM), pp. 3943–3947 (2022)
11. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st edn (1989)
12. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining Knowl. Discov.* (2022)
13. Höllig, J., Kulbach, C., Thoma, S.: Tsevo: evolutionary counterfactual explanations for time series classification. In: 21st IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 29–36 (2022)
14. Ismail, A.A., Gunady, M., Bravo, H.C., Feizi, S.: Benchmarking deep learning interpretability in time series predictions. In: 34th International Conference on Neural Information Processing Systems (NeurIPS) (2020)
15. Lang, J., Giese, M.A., Ilg, W., Otte, S.: Generating sparse counterfactual explanations for multivariate time series. In: International Conference on Artificial Neural Networks and Machine Learning (ICANN), pp. 180–193 (2023)
16. Li, P., Bahri, O., Boubrahimi, S.F., Hamdi, S.M.: Attention-based counterfactual explanation for multivariate time series. In: Wrembel, R., Gamper, J., Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) *Big Data Analytics and Knowledge Discovery*, pp. 287–293. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-39831-5\\_26](https://doi.org/10.1007/978-3-031-39831-5_26)
17. Molnar, C.: *Interpretable Machine Learning*, 2nd edn. (2022). <https://christophm.github.io/interpretable-ml-book>
18. Nagesh, S., Mishra, N., Naamad, Y., Rehg, J.M., Shah, M.A., Wagner, A.: Explaining a machine learning decision to physicians via counterfactuals. In: Conference

- on Health, Inference, and Learning. Proceedings of Machine Learning Research, vol. 209, pp. 556–577 (2023)
19. Rojat, T., Puget, R., Filliat, D., Ser, J.D., Gelin, R., Rodríguez, N.D.: Explainable artificial intelligence (XAI) on timeseries data: a survey. arXiv preprint [arXiv:2104.00950](https://arxiv.org/abs/2104.00950) (2021)
  20. Saeed, W., Omlin, C.: Explainable AI (XAI): a systematic meta-survey of current challenges and future opportunities. *Knowl.-Based Syst.* **263**, 11027 (2023)
  21. Sarker, I.: AI-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems. *SN Comput. Sci.* **3** (2022)
  22. Siddiqui, S.A., Mercier, D., Munir, M., Dengel, A., Ahmed, S.: TSViz: demystification of deep learning models for time-series analysis. *IEEE Access* **7**, 67027–67040 (2019)
  23. Spinnato, F., Guidotti, R., Monreale, A., Nanni, M., Pedreschi, D., Giannotti, F.: Understanding any time series classifier with a subsequence-based explainer. *ACM Trans. Knowl. Discov. Data* **18**(2), 1–34 (2023)
  24. Van Looveren, A., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. arXiv preprint [arXiv:1907.02584](https://arxiv.org/abs/1907.02584) (2019)
  25. Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review. arXiv preprint [arXiv:2010.10596](https://arxiv.org/abs/2010.10596) (2020)
  26. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. arXiv preprint [arXiv:1711.00399](https://arxiv.org/abs/1711.00399) (2017)



# Human-in-the-Loop Personalized Counterfactual Recourse

Carlo Abrate<sup>1,2</sup> , Federico Siciliano<sup>1</sup> , Francesco Bonchi<sup>2</sup> ,  
and Fabrizio Silvestri<sup>1</sup> 

<sup>1</sup> CENTAI, Torino, Italy

{carlo.abrate, francesco.bonchi}@centai.eu

<sup>2</sup> Sapienza University of Rome, Rome, Italy

{siciliano, fsilvestri}@diag.uniroma1.it

**Abstract.** We introduce a new framework for generating counterfactual recourse in machine learning that embraces a “human-in-the-loop” approach by incorporating user preferences. Traditional counterfactual tools neglect individual user preferences when adjusting features. To address this, we tackle recourse generation as a multi-objective optimization problem, integrating conventional constraints with user preferences. Our framework, termed HIP-CORE, is specifically crafted to estimate these preferences during the counterfactual generation phase. We also introduce the “Personal Validity” as a measure of the effectiveness of recourse for individual users. Through extensive theoretical and empirical analysis, we validate the benefits of our proposal. Overall, this work enhances counterfactual reasoning and paves the way for more personalized algorithmic recourse. Code is available at <https://github.com/federicosiciliano/hip-core.git>.

**Keywords:** Personalized Counterfactual · Algorithmic Recourse · Explainability

## 1 Introduction

Algorithmic decision-making systems have become ubiquitous, influencing myriad aspects of our lives, from personalized content recommendations to high stakes decisions in finance, healthcare, and justice. While these algorithms offer efficiency and scalability, their opaqueness often leads to concerns regarding fairness, accountability, and transparency. As a response to these concerns, *eXplainable Artificial Intelligence* (XAI) aims to clarify the complex workings of machine learning models, making their decisions transparent, understandable, and interpretable for end-users, including human-in-the-loop processes [23].

*Human-in-the-loop* refers to a collaborative approach that integrates human judgment, feedback, and decision-making into automated processes, acknowledging that there are instances where human intervention and expertise are crucial for ensuring

---

C. Abrate and F. Siciliano—Contributed equally to the paper.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 18–38, 2024.

[https://doi.org/10.1007/978-3-031-63800-8\\_2](https://doi.org/10.1007/978-3-031-63800-8_2)

quality, fairness, and ethical considerations of AI systems. Incorporating *human-in-the-loop* processes can enhance the accountability and the transparency of AI systems, making them more reliable and aligned with human values and predilections.

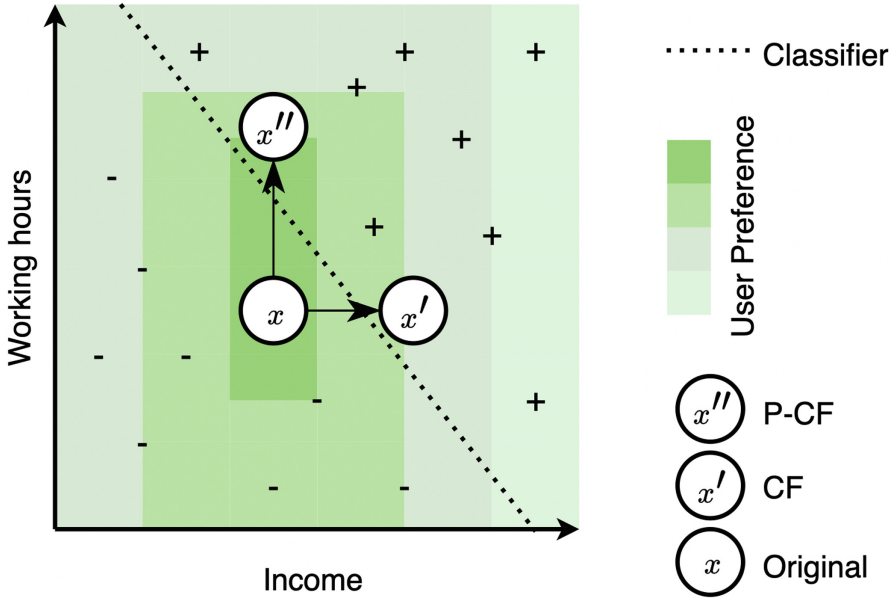
As a result of the emergence of international regulation (i.e., GDPR), increasing attention has been devoted to the *right to recourse* [19]: i.e., in the event that an individual receives an unfavorable decision from a model, he/she is also entitled to receive an actionable explanation that can make him/her proactively adapt his/her features to get a positive outcome from the model in the future. Central to this XAI endeavor is the idea of *counterfactual explanations*, a form of example-based explainability that provides insights by presenting alternative scenarios in which a given decision would change [20]. When applied for algorithmic recourse [8], rather than merely explaining why a decision was made, counterfactuals empower users with actionable insights by suggesting how they might alter inputs to achieve a desired outcome [9, 13].

Considering a real case in which a user applies for a loan and a credit-scoring model gives as output “denied”, but the user is presented with a counterfactual recourse. The counterfactual recourse must allow the user to change the output to “accepted” (i.e. *validity*), while not being too different from the user’s initial status, e.g. suggesting to change only few features, *sparsity*, and asking to change the features values minimally, *proximity*. Since the solution may not be unique depending on the definition of the problem and method used for the solution, to avoid eclipsing some potential explanations and relevant alternatives to the user (*personalization*) [22], multiple counterfactuals can be presented to the user (*diversity*) [12].

In this context, an individual has always been considered as a *rational* agent, so the objective assumptions of *proximity*, *sparsity*, and other constraints related to the underlying model or generic assumptions (a.k.a. user-agnostic), do not consider the irrationality and the subjectivity of human judgment of an algorithm output. Therefore, in this paper we consider the problem of creating counterfactual recourse that are tailored to the subjective and irrational tastes of the user: a personalized counterfactual recourse that includes the user in the generation process to estimate and integrate personal preferences in the solution.

**Example.** In Fig. 1, we show a dummy example of the advantage of generating a **personalized** counterfactual recourse. The plane represents a 2D projection of the feature space hyper-plane for two features: *Working hours* and *Income*. The user  $x$  is classified in the plane as negative – by the simplified line classifier (dashed line). The counterfactual *CF* data point  $x'$  is the optimal solution of a user-agnostic counterfactual recourse problem, where the  $x'$  counterfactual recourse solution recommends that the user increases the *Income* feature to get a positive + output. Given the feasibility of interacting with the user, we consider the user preference a central factor, represented on the plane as a heat map: the user prefers darker areas. A counterfactual recourse method based on user sentiment would output  $x''$ , that is the optimal solution considering both generating a valid counterfactual (i.e. positive output) and maximizing user intent to change the feature (increasing *Working hours* instead of *Income*), thus leading to a solution more easily achievable for the user.

The goal of this paper is to propose such a system, which we dub HIP-CORE (Human-In-the-Loop Preference COUNTERfactual REcourse).



**Fig. 1.** Dummy example of personalized counterfactual recourse: given the original instance  $x$ , classified in the negative class  $-$ , a counterfactual recourse algorithm produces  $x'$ , that asks the user to increase the `income`. The heat-map on the 2D plane represents the user preference over the counterfactual feature space. Our HIP-CORE framework, which takes the user bias into account, produces  $x''$ , that is an optimal solution considering the trade-off between counterfactual validity and user-preference.

**Summary of Contribution.** The contributions of this work are summarized as follows:

1. We formalize the problem of Personalized Counterfactual Recourse (Sect. 3), as a multi-objective optimization problem aggregating optimization functions for both user-agnostic metrics (i.e., validity, sparsity, proximity) and user-level metrics (i.e., preference).
2. We present our algorithmic framework, HIP-CORE to generate preference-driven counterfactual while estimating the preferences of the user (Sect. 4).
3. Key to the development of HIP-CORE is a mathematical framework to represent and estimate user preferences over the complex space of counterfactual feature change (Sect. 4.3).
4. We introduce a new metric called *Personal Validity*, a natural extension of Validity to incorporate users' preferences in the evaluation.
5. We assess our framework empirically on widely used benchmarks, comparing with a user-agnostic baseline, confirming the importance of including user preferences in the counterfactual recourse generation process (Sect. 5).

## 2 Related Work

**Counterfactual Recourse.** Defining and searching for the target counterfactual, it is not trivial in the complex feature space of the instances and the black-box classifier [1]. Traditional assumptions to search a good counterfactual instance include the classification into the opposite class (termed as *Validity*) and its similarity to the original instance [20]. The latter constraint is frequently expressed in terms of *sparsity*, trying to minimize the number of features changed, and *proximity*, which tries to minimize the magnitude of feature change. Other definitions of counterfactual instance are related to the underlying features model, such as a Structural Causal Model [5, 9], where the solution has to be coherent with respect to the causal constraints between features [25]. If the distribution of the feature space is known (not data-agnostic), more feasible counterfactuals can be created with the a priori knowledge, such as through *data manifold closeness* [18].

**Actionable and Diverse Counterfactual Recourse.** Actionability of counterfactual recourse [16] refers to taking into account only those feature changes that an agent can feasibly implement. The personalization of counterfactual recourse is closely related to the local actionability for a user, that in other works is pursued with an ex-ante [16] or post-hoc [12] filter on the generated counterfactuals. We integrate user preferences directly into counterfactual recourse generation, addressing the issue of actionability through explicit human judgment. Another strategy is to create a set of different counterfactuals, therefore, a set of acceptable solutions that maximize a *diversity* function [10, 12] is proposed to the user that choose the most appropriate one. We include the diversity in the main multi-objective problem to accelerate avoid eclipsing relevant counterfactual solution from the user preference evaluation.

**Personalized Counterfactual Recourse.** Few recent attempts have tried to incorporate user tastes in the generation process of counterfactual recourse. The cost of adoption of the counterfactual change is the pivotal point to discriminate among user-centered approach, where the preference is incorporated in the cost function, and user-agnostic method, where the cost function does not take into account the user preferences. To incorporate the preference, some approaches request users to specify it over a set of solutions [12, 24], or attempt to quantify the cost of potential changes in advance [17, 21]. Such methodologies do not incorporate the user within the counterfactual generation loop, consequently neglecting the exploration of the counterfactual preference space. Our method, instead, encompasses the estimation of user preferences within the counterfactual recourse generation process. Our framework is not comparable to methods that do not encompass user preference, e.g. [11, 15, 20]. In terms of personalized counterfactual, our approach can only outperform methods that do not optimize for it.

**Human-in-the-Loop Algorithmic Recourse.** Research on the human-in-the-loop approach, combining preference elicitation to create solutions that align with user preferences, is notably limited. In contrast from previous work, we offer a broader approach that does not estimate preference through the use of experts [17] but directly through the interaction with the user. Furthermore, we do not rely on Structural Causal Models [5], or impose constraints on preference modeling [25], to provide a more general framework. These approaches are not compatible with our modeling because they rely on

defining a cost of recourse. In contrast, we define the user’s preference on recourse based on probability. This definition is more comprehensive, as it does not simply presuppose a (linear) weight, varying for each feature, that changes with the distance between the action required to obtain recourse and the current values of the user’s features. Consequently, we can describe a broader class of preferences.

In summary, we define preference at a broader and more comprehensive level compared to the limited scope of current works. Moreover, none of the existing approaches encompass, within a unified multi-objective problem, properties intrinsic to counterfactual recourse with user preference: user-centered (i.e., preference), general aspect (i.e., validity), and data-specific elements (i.e. proximity, sparsity).

### 3 Problem Statement

A user  $u \in \mathcal{U}$  is described as a point  $x_u \in \mathcal{X}$  in a feature space  $\mathcal{X} \subseteq \mathbb{R}^n$ , with  $n \in \mathbb{N}^+$ . Users are subjected to evaluation by a black-box classifier<sup>1</sup>  $f : \mathcal{X} \rightarrow [0, 1]$ . The *algorithmic recourse problem* is defined when a user  $u$  gets a negative outcome  $f(x_u) < \tau$  and needs to receive a recourse. The *counterfactual* formulation of the recourse problem provides the recourse as a new counterfactual configuration of the user point  $x'_u \in \mathcal{X}$  that allows the user to get a positive outcome  $f(x'_u) \geq \tau$ , with  $\tau \in [0, 1]$  (generally  $\tau = 0.5$ ). Differentiating  $x$  according to the user is important because two users  $u, v \in \mathcal{U}$  represented by identical vectors  $x_u = x_v$ , might have different preferences (see following sections). Nevertheless, in the absence of ambiguity, the subscript will be omitted.

Several requirements are typically incorporated into the standard counterfactual recourse problem: the counterfactual  $x'$  must be close to the original point  $x$  (*Proximity*),  $x'$  must change the minimum number of features of  $x$  (*Sparsity*), when producing multiple counterfactuals for the same  $x$ , they must be diverse in nature (*Diversity*). Besides these standard requirements, we introduce user preference as a key property to generate user-centered counterfactual recourse.

**Definition 1.** *The preference of a user  $u \in \mathcal{U}$  for a counterfactual  $x'_u \in \mathcal{X}$  is a probability  $\Pi_u(x'_u) = P(x'_u | x_u, u)$  that the user accepts the counterfactual instance  $x'_u \in \mathcal{X}$  as a recourse, with  $\Pi_u : \mathcal{X} \rightarrow [0, 1]$ .*

We do not define an absolute preference of a user within the space  $\mathcal{X}$ , but rather how willing the user  $u$  is to alter their current state  $x$  and in what manner.

We are now ready to formalize the main problem:

*Problem 1 (Personalized Counterfactual Recourse Problem).* Given a user  $u \in \mathcal{U}$ , with  $x_u \in \mathcal{X}$ , that received a negative outcome  $f(x_u) < \tau$ , from a black-box classifier

---

<sup>1</sup> Our formulation is also applicable when  $f$  is a multi-class classifier by employing the one-vs-all technique. In the opposite classification fashion,  $1 - f(x_u)$  can simply be used as the classifier.

$f : \mathcal{X} \rightarrow [0, 1]$ , find a set of  $k \in \mathcal{N}^+$  counterfactual data point  $C = \{x^{(1)}, \dots, x^{(k)}\}$  such that

$$\begin{aligned}
 & \max_{x^{(i)}} \mathcal{Y}(x^{(i)}, x_u) && \text{proximity} \\
 & \quad \forall i \in \{1, \dots, k\} \\
 & \min_{x^{(i)}} \Gamma(x^{(i)}, x_u) && \text{sparsity} \\
 & \quad \forall i \in \{1, \dots, k\} \\
 & \max_{x^{(i)}, x^{(j)}} \Delta(x^{(i)}, x^{(j)}) && \text{diversity} \\
 & \quad \forall i, j \in \{1, \dots, k\}, i \neq j \\
 & \max_{x^{(i)}} \Pi_u(x^{(i)}) && \text{preference} \\
 & \quad \forall i \in \{1, \dots, k\} \\
 & \text{s.t. } f(x^{(i)}) \geq \tau && \text{validity} \\
 & \quad \forall i \in \{1, \dots, k\}
 \end{aligned} \tag{1}$$

where  $\mathcal{Y}, \Gamma, \Delta, \Pi_u : \mathcal{X}^2 \rightarrow [0, 1]$ .

Since in the recourse setting, the user’s preference is initially unknown, Problem 1 cannot be solved as is. Instead, the user’s preference needs to be estimated by querying user predilections (Sect. 4).

**Counterfactual Metrics** - Despite the generality of Problem 1, in this paper we adopt metrics widely used in the counterfactual literature:

- **Proximity** can be defined as the Euclidean norm between a counterfactual  $x'$  and the original value  $x$ :

$$\mathcal{Y}(x', x) = \frac{1}{\|x' - x\|_2 + 1}$$

This is inverted to map it to 0, 1 and to align with maximization.

- **Sparsity**, representing the number of modified features, can be expressed using the zero norm:

$$\Gamma(x', x) = \|x' - x\|_0$$

Since this norm is non-differentiable, it is preferable to use the absolute-value norm  $\|\cdot\|_1 = |\cdot|$ .

- **Diversity**, as in [12] we use a distance between the generated counterfactual, that is the cosine distance  $\forall x^{(i)}, x^{(j)} \in C, i \neq j$ :

$$\Delta(x^{(i)}, x^{(j)}) = 1 - \frac{x^{(i)} \cdot x^{(j)}}{\|x^{(i)}\|_2 \|x^{(j)}\|_2}$$

We introduce an additional metric that replaces the traditional concept of validity, which is defined as  $f(x^{(i)}) \geq \tau$ . With the introduction of user preferences, it becomes imperative to redefine the notion of a *valid* counterfactual: if the user does not *accept* the counterfactual, can it truly be considered valid? Building upon this premise, we introduce the following measure:

**Definition 2 (Personal Validity).** *Given a user  $u \in \mathcal{U}$ , with  $x_u \in \mathcal{X}$  and the user preference probability  $\Pi_u(x'_u) = P(x'_u | x_u, u)$ , the Personal Validity for the classifier  $f : \mathcal{X} \rightarrow [0, 1]$  on counterfactual recourse  $x'_u \in \mathcal{X}$  is defined as:*

$$PV(x'_u) = \Pi_u(x'_u) \cdot f(x'_u)$$



This novel metric preserves the properties of both Validity and Preference. The non-binary nature of the recourse probability captures the nuances between full acceptance and rejection of a counterfactual, thus providing a more detailed measure of validity compared to the traditional binary definition.

## 4 Framework

In this section, we present our iterative algorithm (Sect. 4.4), dubbed HIP-CORE (Human-In-the-Loop COUNTERfactual REcourse), designed to estimate user preference  $\Pi_u$  (Sect. 4.3) while generating candidate counterfactuals  $C$  (Sect. 4.1). Our approach is agnostic to both model and data, making it applicable for generating personalized counterfactual recourse with any black-box model.

### 4.1 Personalized Counterfactual Generation

Given the complexity of solving a multi-objective problem such as Problem 1, that in some setting has been show to be NP-hard [9], we transform Problem 1 into a single-objective problem, by considering a linear combination of the metrics, with the signs appropriately inverted for those metrics that are to be minimized (i.e., sparsity). Additionally, the constraint of the class flipping (i.e. score) is directly incorporated, removing the dependence on  $\tau$ . Consequently, we provide the following single-objective problem.

*Problem 2 (Relaxed Personalized Counterfactual Recourse Problem).* Given the same setting as in Problem 1, the problem can be relaxed as follows:

$$\begin{aligned} \max_C \quad & \frac{1}{k} \sum_{i=1}^k [\lambda_Y \Upsilon(x^{(i)}, x) + \lambda_\Gamma (1 - \Gamma(x^{(i)}, x)) + \\ & + \lambda_\Pi \Pi(x^{(i)}) + \lambda_f f(x^{(i)}) + \\ & + \lambda_\Delta \frac{1}{k-i-1} \sum_{j=i+1}^k \Delta(x^{(i)}, x^{(j)})] \end{aligned} \quad (2)$$

where  $\Upsilon, \Gamma, \Delta, \Pi : \mathcal{X}^2 \rightarrow [0, 1]$ ,  $\lambda_Y, \lambda_\Gamma, \lambda_\Delta, \lambda_\Pi, \lambda_f \in [0, 1]$ , such that  $\lambda_Y + \lambda_\Gamma + \lambda_\Delta + \lambda_\Pi + \lambda_f = 1$

A counterfactual can be generated by solving the presented maximization problem. To effectively address this, several optimization algorithms can be employed. In our experiments, we solved this problem using the Powell Method [14].

The coefficients  $\lambda$  in Eq. 2 allow for adjusting the importance assigned to individual metrics. They generate a challenging trade-off, between the user-agnostic counterfactual properties (i.e. score, proximity, sparsity, diversity) and the user preference. In the experimental evaluation, we discuss the implication of the trade-off.

## 4.2 Preference Modeling

In this section, we introduce a set of assumptions and the resulting propositions, to facilitate the modeling of preference and elaborate more on the framework. However, our proposed framework is intentionally designed to be agnostic to preference modeling, emphasizing its versatility across various preference representation schemes, as elaborated in Limitations 4.6.

**Assumption 1.** *The preference  $\Pi_u$  of a user  $u \in \mathcal{U}$  remains stable in the explanation process.*

Introducing a temporal component to the problem is not a straightforward task because it would require considering users  $u$  who change both their instances  $x_u$  and their preferences  $\Pi_u$  over time [6]. Consequently, the same counterfactuals generated may no longer be valid at different times. For this reason, in the current work, we will not account for the temporal component.

**Assumption 2.** *For all users  $u \in \mathcal{U}$ , there exists a counterfactual explanations  $x' \in \mathcal{X}$ , such that the user's preference  $\Pi_u(x')$  is equal to 1.*

Enforcing the preference to have a value of 1 allows us to evaluate preference as if it were a normalized metric, thus facilitating a better assessment of the quality of a counterfactual and determining if the preference optimum (1) has been reached.

**Assumption 3.** *The preference  $\Pi_u(x')$  is maximal when  $x' = x$ .*

This assumption is based on the idea that users tend to maintain their current state, making the maximum preference corresponding to minimal state change. However, since they aim to flip their classification, they are willing to yield, take actions that move them away from their current state, thereby reducing their initial preference.

**Proposition 1.** *Let  $\pi_u : \mathcal{X} \rightarrow [0, 1]$  a probability distribution. Then,  $\Pi_u(x') = \frac{\pi_u(x')}{\max_{x'} \pi_u(x')}$  represents a model for a preference of a user  $u \in \mathcal{U}$ .*

*Proof.* In order for  $\Pi_u(x')$  to be a preference, we need to check that it respects the above two assumptions. To check Assumption 1 it suffices to observe that  $\pi_u(x') \in [0, \max_{x' \in \mathcal{X}} \pi_u(x')]$  therefore  $\Pi_u(x') \in [0, 1]$ . To check Assumption 2 note that  $\Pi_u(x^*) = 1$  for the counterfactual  $x^*$  such that  $\pi_u(x^*) = \max_{x'} \pi_u(x')$ , implying that  $\Pi_u(x^*) = 1$ .

The reason for introducing a probability distribution in the above proposition is to imbue the preference with some desired properties. The probability distribution  $\pi_u$  allows us to sample counterfactuals where the probability is directly proportional to the user's preference value.

However, we cannot directly model the preference using a density function since this would render values incomparable among different users. Let us consider two users,  $i$  and  $j$ , where one user absolutely avoids taking actions to obtain the recourse, while the other is willing to take any action to acquire it. User  $i$  would have a preference of

1 at  $x_i$ , and 0 elsewhere. On the other hand, the latter user would exhibit a preference of  $1/N$  for any value  $x'$  (but only if a finite number  $N$  of actions exists, as if the set were countable and infinite, even a uniform distribution would not exist). Not only would these two preferences be incomparable, but we would not even know if we had maximized the preference of user  $j$ , as the preference would have a different maximum depending on the user.

Similarly, we cannot directly model the preference using a Cumulative Distribution Function (CDF). This limitation arises from the specific behavior of a CDF, as defined: it increases monotonically with the input, without any decrease. Consequently, we cannot represent, for instance, a user's preference that tends to favor staying around value  $x$ , diminishing as one moves away—essentially a bell-shaped function.

It is reasonable to consider that the joint probability distribution  $\pi_u$  might be complex, and the assumption of feature independence rarely holds in reality. However, from a joint distribution, it is still possible to derive a marginal probability density function for each feature. These marginal probability density functions can often be related to or approximated by known distributions. In the remainder of this section, we will introduce a series of examples illustrating how one can model the preference of different types of features using well-known distributions.

First, we can introduce two extreme scenarios, represented by features for which the user has no desire or ability to change (e.g., place of birth) or holds no specific preference.

**Proposition 2.** *If a user  $u$  has no intention to or can not change a feature  $i$ , their preference  $\Pi_u(x'_i)$  can be modeled using a degenerate probability distribution  $\pi_u$  over  $x_i$ , such that:  $\Pi_u(x'_i) = \begin{cases} 1 & \text{if } x'_i = x_i \\ 0 & \text{otherwise} \end{cases}$*

*Proof.* If a user  $u$  is unwilling to change a feature  $i$ , it is natural to assume that  $\Pi_u(x'_i) = 0$  for all  $x'_i \neq x_i$ . Referring back to Definition 1, we can rewrite  $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\max_{x'_i} \pi_u(x'_i)}$ . This leads to  $\pi_u(x'_i) = 0$  for all  $x'_i \neq x_i$ . Since  $\pi_u$  is a probability distribution and must satisfy the constraint  $\sum_{x'_i \in \mathcal{X}} \pi_u(x'_i) = 1$ , it follows that  $\pi_u(x_i) = 1$ , and thus  $\Pi_u(x_i) = 1$ . This also agrees with Assumption 3, as  $\Pi_u(x'_i)$  indeed attains its maximum value at  $\Pi_u(x_i)$ .

**Proposition 3.** *If a user  $u$  has no preference for changing feature  $i$ , their preference  $\Pi_u(x'_i)$  can be modeled using a uniform probability distribution  $\pi_u$ :  $\Pi_u(x'_i) = 1 \quad \forall x'_i \in \mathcal{X}_i$ .*

*Proof.* If  $\pi_u$  is a continuous distribution,  $\pi_u(x'_i) = \frac{1}{|\mathcal{X}_i|} \quad \forall x'_i \in \mathcal{X}_i$  (we are not considering the case when  $\mathcal{X}_i$  is an infinite set). Since that same value is also the maximum of  $\pi_u$ , we would get  $\Pi_u(x'_i) = 1$ .

In these extreme cases, preference is essentially estimated by definition, as there are no unknown parameters to estimate. Expanding to non-extreme cases we can assume that preference  $\Pi_u$  is, in fact, dependent on an unknown set of parameters  $\theta$ . For instance, in the following section, we can define preferences for continuous features.

**Proposition 4.** *If a feature  $i$  is continuous, the preference  $\Pi_u(x'_i)$  can be modeled using a normal distribution  $\pi_u$  with mean  $\theta_1 = x_i$  and variance  $\theta_2 > 0$ , such that:  $\Pi_u(x'_i) = e^{-\frac{1}{2}\left(\frac{x'_i - x_i}{\theta_2}\right)^2}$*

*Proof.* Referring back to Definition 1, we can write  $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\max_{x'_i} \pi_u(x'_i)}$ . If  $\pi_u$  follows a normal distribution, it has a mean  $\theta_1 = x_i$  according to Assumption 3. Thus, it can be expressed as  $\pi_u(x'_i) = \frac{1}{\theta_2\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x'_i - x_i}{\theta_2}\right)^2}$ . Since the maximum is reached at  $x'_i = x_i$ , i.e.,  $\max_{x'_i} \pi_u(x'_i) = \pi_u(x_i) = \frac{1}{\theta_2\sqrt{2\pi}}$ , we obtain  $\Pi_u(x'_i) = e^{-\frac{1}{2}\left(\frac{x'_i - x_i}{\theta_2}\right)^2}$ .

Proposition 4 proves that it is not critical to estimate the position of preference, as it is always centered around the current value  $x_i$ . What matters, instead, is the variance  $\theta_2$ , which directly models the user’s willingness to deviate from the current value  $x_i$ .

**Proposition 5.** *If a continuous feature  $i$  can only increase<sup>2</sup>, the preference  $\Pi_u(x'_i)$  can be modeled using an exponential distribution with rate  $\theta > 0$ , such that:  $\Pi_u(x'_i) = \begin{cases} e^{-\theta(x'_i - x_i)} & \text{if } x'_i \geq x_i \\ 0 & \text{otherwise} \end{cases}$*

*Proof.* If  $\pi_u$  follows an exponential distribution, we can write:

$$\pi_u(x'_i) = \begin{cases} \theta e^{-\theta(x'_i - x_i)} & \text{if } x'_i \geq x_i \\ 0 & \text{otherwise} \end{cases}$$

Noting that the maximum value  $\pi_u(x'_i) = \pi_u(x_i) = \theta$  does not violate Assumption 3, we can write  $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\theta}$ , following from Definition 1.

**Proposition 6.** *If a feature  $i$  is categorical with  $K$  categories, the preference  $\Pi_u(x'_i)$  can be modeled using a categorical distribution  $\pi_u$  with parameters  $\theta_1, \dots, \theta_K$ , such that:  $\Pi_u(x'_i) = \frac{\theta_k}{\theta_{x_i}} \quad \forall k \in \{1, \dots, K\}$ .*

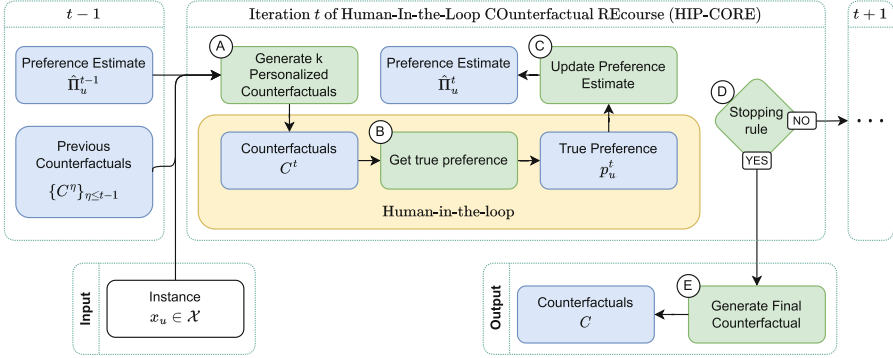
*Proof.* Given that  $\pi_u(x'_i) = \theta_i \forall k \in \{1, \dots, K\}$ , based on Assumption 3 we derive that  $\max_{x'_i} \pi_u(x'_i) = \theta_{x_i}$ . Consequently, we obtain  $\Pi_u(x'_i) = \frac{\pi_u(x'_i)}{\theta_{x_i}}$ , which attains value of 1 if  $x'_i = x_i$ , satisfying Assumption 2.

### 4.3 Preference Estimation

In the recourse setting, there is no access to  $\Pi_u$ , and we cannot invoke it at will. Furthermore, there could be a maximum number of feasible interaction to ask the user’s preferences. Therefore, it is fundamental to be able to estimate  $\Pi_u$ .

The preference can be estimated by solving the following system of equations:

<sup>2</sup> The extension to non-increasing features is trivial.



**Fig. 2.** Iteration  $t$  of HIP-CORE. During each iteration, a distinct set of personalized counterfactuals is generated and presented to the user to elicit preferences. This iterative process refines the preference estimate incrementally. Upon termination, the algorithm leverages the cumulative preference estimate obtained throughout the iterations to generate a final set of counterfactuals.

**Definition 3.** Given a user  $u \in \mathcal{X}$ , a set of preference values  $p_u$  and a set of counterfactuals  $C$ , preference can be estimated by solving the following system of equations in  $\theta$ :

$$\hat{\Pi}_u(x^{(i)}|\theta) = p_u^{(i)} \quad \forall i \in \{1, \dots, |C|\} \quad (3)$$

Solving this problem depends on both the quantity of generated counterfactuals  $|C|$  for which true preferences  $p_u$  are available and the number of parameters  $\theta$  that comprise  $\Pi_u$ . These parameters are contingent on how the preference is defined, as exemplified in Propositions 2, 3, 4, 5 and 6. In our experiments, we solved this problem by minimizing the mean squared difference between  $\Pi_u(x^{(i)}|\theta)$  and  $p_u^{(i)}$  for all  $i \in 1, \dots, |C|$  using the Powell Method [14] at each iteration  $t$ . We initialized the parameters at each iteration using the estimates from the previous iteration  $\theta^{t-1}$ .

#### 4.4 HIP-CORE Framework

Figure 2 provides a graphical schematization of the functioning of HIP-CORE, while its pseudocode is provided in Algorithm 1. At an high-level, at each iteration  $t$ , HIP-CORE refines the estimate of user preference  $\hat{\Pi}_u^t$  with the human-in-the-loop true preference  $p_u^t$ , while generating more personalized counterfactual recourse  $C^t$ .

**Initialization** - The initialization of  $\hat{\Pi}$  is defined as uniform over the entire set  $\mathcal{X}$ . However, if some data are available, it could be initialized as uniform over all data points in the dataset or only for those where the counterfactual is valid.

**Iteration  $t$**  - At each iteration  $t$ , the algorithm receives as input the original instance  $x \in \mathcal{X}$ , the set of counterfactuals generated in the previous iterations  $\{C^\eta\}_{\forall \eta < t}$ , and the estimated user preference  $\hat{\Pi}^{t-1}$ .

At each iteration  $t$  HIP-CORE performs the following steps:

Ⓐ A set of  $k \in \mathcal{N}^+$  personalized counterfactual recourse  $C^t$  are produced with `get_c` (check Sect. 4.1).

**Algorithm 1.** HIP-CORE

---

**Require:** a user identifier  $u \in \mathcal{U}$ , a user feature point  $x \in \mathcal{X}$ ; a classifier  $f$ ; a number of counterfactual generated at each iteration  $k \in \mathbb{N}^+$ ; a maximum number of iterations  $T \in \mathbb{N}^+$ .

**Ensure:** A personalized counterfactual recourse  $\{x'\}$  and an estimation of user preference  $\hat{\Pi}_u$

- 1:  $\hat{\Pi}_u^0 \leftarrow g : \mathcal{X} \rightarrow 1$  s.t.  $g(x) = \frac{1}{|\mathcal{X}|}$  if  $x \in \mathcal{X}$  else 0 Initialize the estimate of user preferences;
  - 2:  $C^0 \leftarrow \{\}$  Initialize counterfactuals' set
  - 3:  $t \leftarrow 1$ ;
  - 4: **while** ( $t \leq T$ ) **do**
  - 5:    $C^t \leftarrow \text{get\_c}(x, \hat{\Pi}_u^{t-1}, k, \{C^\eta\}_{\forall \eta < t})$  Generate counterfactuals;
  - 6:    $p_u^t \leftarrow \{I_u(x')\}_{\forall x' \in C^t}$  Ask user preference;
  - 7:    $\hat{\Pi}_u^t \leftarrow \text{update\_pref}(\hat{\Pi}_u^{t-1}, \{p_u^\eta\}_{\forall \eta \leq t}, \{C^\eta\}_{\forall \eta \leq t})$  Update preference estimation;
  - 8:    $t \leftarrow t + 1$
  - 9:  $C \leftarrow \text{get\_c}(x, \hat{\Pi}_u^T, 1)$  Generate final counterfactual;
  - 10: **return**  $C, \hat{\Pi}_u^T$
- 

(B)  $C^t$  is provided to the user for a human-in-the-loop interaction and he expresses the preferences over the counterfactuals. At this stage, the true preferences  $p_u^t = \{I_u(x')\}_{\forall x' \in C^t}$  of the user are stored.

(C) The algorithm updates the estimate of the user preference  $\hat{\Pi}_u^t$  given the new true preferences  $p_u^t$  with *update\_pref* (check Sect. 4.3).

(D) The stopping rule is a straightforward maximum number of iterations  $T$ . Alternatively, it may depend on other factors, such as whether the generated counterfactuals match those from the previous iteration.

(E) Once the stopping criteria are met, the final personalized counterfactual  $C = \{x'\}$  is generated.

#### 4.5 Complexity of User Feedback

The mental effort required to offer a feedback, such as assigning a numerical value to a hypothetical scenario on a scale from 0 to 10, can be substantial. Hence, expecting users to provide a density function when interrogated is realistically infeasible. There are various approaches to model this user difficulty in providing the precise value of their preference for a specific counterfactual. We have chosen to model it as the number of decimal places to which the true preference is rounded when the user is queried. Given that the preference resides within the interval  $[0, 1]$ , if the decimal places are set to 0, the user is essentially indicating whether they would accept the counterfactual (1) or not (0). Using 1 decimal place would **correspond** to asking for a score within the range  $[0, 10]$ , while using 2 decimals would **equate** to requesting a score within the range  $[0, 100]$ , and so forth.

#### 4.6 Limitations

The assumptions regarding the preference made in Sect. 4 could be perceived as mere limitations of our framework. However, in our view, these assumptions are essential in

imparting convenient properties to the preference, subsequently benefiting both the optimization problem and the Personal Validity. Nevertheless, HIP-CORE is more general and applies beyond these assumptions. More specifically, assumption 3 can be easily circumvented by also modeling the value  $x^*$  where the maximum preference is located. For instance, in the case of a continuous variable expressed by a Gaussian distribution 4, estimating the mean  $\theta_1$  would suffice. Assumption 2, as previously mentioned, imparts characteristics desired for the problem. However, if a user’s preference does not reach a value of 1, HIP would still be applicable. Lastly, assumption 1 is an area where substantial enhancement of the framework could occur. It is important to note that even if preferences were to change over time, the new inquiries made to users would still reflect their true preferences. Therefore, the framework could still converge effectively.

## 5 Experiments

**Table 1.** Comparison of HIP-CORE and baseline model performance for each one of the classifiers using maximum decimal precision. The direction of arrows indicates what is considered the best performance:  $\uparrow/\downarrow$  denotes that higher/lower values are better. Best-performing values in each category are highlighted in **bold**.

Dataset	Model	Validity( $\uparrow$ )	Preference( $\uparrow$ )	Sparsity( $\downarrow$ )	Proximity( $\uparrow$ )	Personal Validity( $\uparrow$ )
XGBoost						
Adult Income	HIP-CORE	0.904	<b>0.386 <math>\pm</math> 0.107</b>	<b>0.695 <math>\pm</math> 0.136</b>	0.945 $\pm$ 0.064	<b>0.317 <math>\pm</math> 0.150</b>
	Baseline	<b>0.943</b>	0.346 $\pm$ 0.108	0.757 $\pm$ 0.135	<b>0.975 <math>\pm</math> 0.036</b>	0.302 $\pm$ 0.099
GiveMeSomeCredit	HIP-CORE	<b>0.585</b>	<b>0.053 <math>\pm</math> 0.006</b>	<b>0.759 <math>\pm</math> 0.143</b>	<b>0.970 <math>\pm</math> 0.152</b>	<b>0.030 <math>\pm</math> 0.027</b>
	Baseline	0.002	0.0 $\pm$ 0.001	1.000 $\pm$ 0.007	<b>0.970 <math>\pm</math> 0.152</b>	0.0 $\pm$ 0.0
HELOC	HIP-CORE	<b>0.515</b>	<b>0.070 <math>\pm</math> 0.043</b>	<b>0.880 <math>\pm</math> 0.086</b>	0.702 $\pm$ 0.254	<b>0.037 <math>\pm</math> 0.050</b>
	Baseline	0.342	0.031 $\pm$ 0.047	0.925 $\pm$ 0.117	<b>0.761 <math>\pm</math> 0.261</b>	0.005 $\pm$ 0.008
Logistic Regression						
Adult Income	HIP-CORE	0.277	0.203 $\pm$ 0.123	0.832 $\pm$ 0.14	0.907 $\pm$ 0.122	<b>0.051 <math>\pm</math> 0.105</b>
	Baseline	<b>0.351</b>	<b>0.268 <math>\pm</math> 0.152</b>	<b>0.77 <math>\pm</math> 0.168</b>	<b>0.957 <math>\pm</math> 0.055</b>	0.046 $\pm$ 0.074
GiveMeSomeCredit	HIP-CORE	0.452	<b>0.058 <math>\pm</math> 0.01</b>	<b>0.644 <math>\pm</math> 0.107</b>	<b>0.96 <math>\pm</math> 0.165</b>	<b>0.027 <math>\pm</math> 0.031</b>
	Baseline	<b>0.512</b>	0.014 $\pm$ 0.014	0.955 $\pm$ 0.051	0.902 $\pm$ 0.205	0.001 $\pm$ 0.002
HELOC	HIP-CORE	<b>0.04</b>	<b>0.058 <math>\pm</math> 0.041</b>	<b>0.931 <math>\pm</math> 0.089</b>	0.681 $\pm$ 0.259	<b>0.003 <math>\pm</math> 0.023</b>
	Baseline	0.006	0.014 $\pm$ 0.008	0.96 $\pm$ 0.041	<b>0.738 <math>\pm</math> 0.264</b>	0.0 $\pm$ 0.0
MultiLayer Perceptron						
Adult Income	HIP-CORE	<b>0.098</b>	<b>0.186 <math>\pm</math> 0.104</b>	<b>0.855 <math>\pm</math> 0.12</b>	0.873 $\pm$ 0.168	<b>0.024 <math>\pm</math> 0.081</b>
	Baseline	0.056	0.109 $\pm$ 0.071	0.946 $\pm$ 0.074	<b>0.927 <math>\pm</math> 0.072</b>	0.005 $\pm$ 0.011
GiveMeSomeCredit	HIP-CORE	<b>0.004</b>	<b>0.052 <math>\pm</math> 0.006</b>	<b>0.799 <math>\pm</math> 0.125</b>	<b>0.971 <math>\pm</math> 0.152</b>	<b>0.0 <math>\pm</math> 0.004</b>
	Baseline	<b>0.004</b>	0.005 $\pm$ 0.012	0.983 $\pm$ 0.039	<b>0.971 <math>\pm</math> 0.152</b>	0.0 $\pm$ 0.0
HELOC	HIP-CORE	<b>0.592</b>	<b>0.069 <math>\pm</math> 0.046</b>	<b>0.91 <math>\pm</math> 0.06</b>	0.673 $\pm$ 0.251	<b>0.042 <math>\pm</math> 0.05</b>
	Baseline	0.338	0.025 $\pm$ 0.049	0.938 $\pm$ 0.121	<b>0.718 <math>\pm</math> 0.274</b>	0.006 $\pm$ 0.017

## 5.1 Experimental Setting

To evaluate HIP-CORE, we define an experimental setting as follows. Each user  $u \in \mathcal{U}$  is described by a user feature data  $x_u \in \mathcal{D}$  and the true user-preference distribution  $\Pi_u$  is simulated as described in Sect. 4.2. Furthermore, we have made the assumption of feature independence. Despite being an oversimplification, explicitly expressing a joint distribution with dependencies can be exceedingly complex, particularly for high-dimensional feature sets. Frequently, it necessitates specific modeling choices that rely on the nature and interrelationships among the features under consideration.

More details about the experimental setup can be found in the appendix. To get the feature data  $\mathcal{D}$ , we used existing real-world datasets: Adult [2], GiveMeSomeCredit [4] and HELOC [7]. Given the model-agnostic nature of HIP-CORE, we employed various classifiers in our experimentation: XGBoost [3], Logistic Regression, a MultiLayer Perceptron (MLP). Each classifier is trained on an appropriate subset of the complete data.

To solve the personalized counterfactual recourse step of HIP-CORE, as defined in Problem 2, as well as the preference estimation step, as defined in Problem 3, we have chosen to employ the Powell’s method [14]. Furthermore, we have run a randomized search for the  $\lambda$  parametrization in Eq. 2, to explore the trade-off between the different properties. We ran HIP-CORE for a maximum of 100 iterations. Furthermore, we tested the framework in scenarios where there were no limits on the decimal places for preference, as well as decimal places in the set  $\{0, 1, 2\}$ .

The experiment are performed for the tested dataset with two distinct setting: one user-agnostic (the baseline), i.e.  $\lambda_{\Pi} = 0$ , and one including the preference, i.e.  $\lambda_{\Pi} > 0$ , to highlight the importance of using preference in generating personalized recourse. The results are shown for the combination of  $\lambda$  parameters that achieves the maximum Personalized Validity.

The comprehensive code required to replicate the experiments is accessible in our GitHub repository<sup>3</sup>.

## 5.2 Overall Performance

In Table 1, we report the main results of our experiments for the tested dataset for HIP-CORE with preference and a user-agnostic version. Metrics are generally improved by the HIP-CORE across all datasets, with the exception of the proximity, and validity in some cases.

Sparsity value is decreased, meaning that on average less features are modified by HIP-CORE: we generated more concise counterfactual recourse using features that the user prefers. Proximity is slightly decreased compared to the baseline. However, given that proximity is a data-driven measure that do not consider the subjective and potentially irrational user preference, we encourage the community to increase the relevance of the user preference with respect to the proximity.

Finally, the preference is substantially enhanced by HIP-CORE compared with the baseline. This underscores the importance of including the preference in counterfactual recourse generation process.

<sup>3</sup> <https://anonymous.4open.science/r/hip-core-FS20>.



Only on the Adult Income dataset and solely using Logistic Regression, HIP-CORE yields worse results than the baseline across all metrics, except for Personal Validity. Considering that Personal Validity is, in fact, the primary metric to optimize as it represents the genuine validity of the counterfactual for the user, we can observe that HIP-CORE consistently outperforms the baseline.

### 5.3 Model-Agnostic Validation

HIP-CORE’s performance doesn’t seem to be significantly affected by the different types of classifiers used (XGBoost, Logistic Regression, or MultiLayer Perceptron). We can observe that it struggles more to outperform the baseline in Validity when the classifier is Logistic Regression. Additionally, the metric values, in general, appear to be better when XGBoost is the chosen classifier, possibly because XGBoost is notably more effective on these tabular datasets.

### 5.4 Study on the Number of Iterations

In Fig. 3, the percentage of optimizations reaching convergence at each iteration is illustrated. This represents the relative number of users for whom no new counterfactuals are being generated, indicating that their preference can no longer be updated.

Regarding the differences between models, it is noticeable that there is almost no difference in the convergence of the three, further demonstrating how our optimization algorithm operates independently of the classifier.

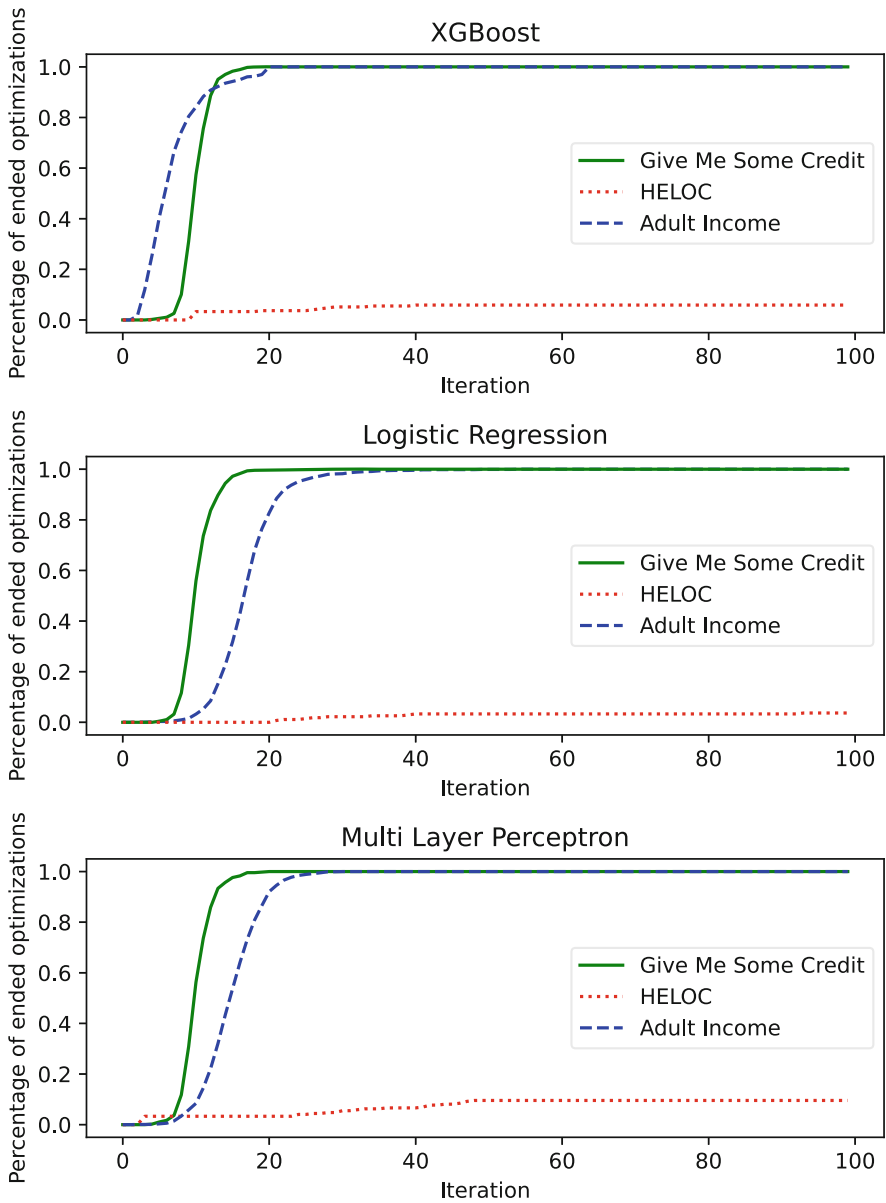
When discussing datasets, we observe that the Give Me Some Credit and Adult Income datasets converge around the 15th and 20th iterations, respectively. However, the HELOC dataset appears more complex, with only about 10% of the samples completing optimization. This does not imply invalidity in the generated counterfactuals, as demonstrated in Table 1. It merely suggests that extended iterations might yield even more refined outcomes.

A noteworthy observation is that HIP-CORE converges more quickly with the combination of XGBoost and the Adult Income dataset.

### 5.5 Study on the Number of Decimal Places

Table 2 illustrates that the overall performance across all metrics remains largely unaffected by the reduced decimal precision.

Table 2 illustrates the performance of HIP-CORE applied to logistic regression classifiers using different decimal precision of the preferences provided by the user during interrogation. Comparing the results with those presented in Table 1, where a precision of 16 decimal places is used, we find that the overall performance across all metrics remains largely unaffected by the reduced decimal precision. Surprisingly, some results even show an improvement. Furthermore, when we examine the metric variations with increasing decimal precision, we see a consistent upward trend in improvement. This is consistent with the concept that the more precise the user’s preference input, the more accurately the model can estimate it.



**Fig. 3.** Percentage of optimizations reaching convergence at each iteration, for each model type and for the three datasets. The y-axis shows the percentage of users for whom no further counterfactuals are produced, indicating that their preference can no longer be updated, while the x-axis represents the number of iterations. Almost no difference is noticeable between the three models.

**Table 2.** HIP-CORE framework performance for the Logistic Regression classifier using different decimal precision. *DC* indicates decimal precision for the preference given by the user.

DC	Dataset	Validity( $\uparrow$ )	Preference( $\uparrow$ )	Sparsity( $\downarrow$ )	Proximity( $\uparrow$ )	Personal Validity( $\uparrow$ )
0	Adult Income	0.341 $\pm$ 0.439	0.196 $\pm$ 0.134	0.839 $\pm$ 0.15	0.905 $\pm$ 0.114	0.058 $\pm$ 0.107
	GiveMeSomeCredit	0.368 $\pm$ 0.482	0.019 $\pm$ 0.011	0.972 $\pm$ 0.049	0.949 $\pm$ 0.14	0.006 $\pm$ 0.009
	HELOC	0.022 $\pm$ 0.134	0.058 $\pm$ 0.037	0.93 $\pm$ 0.055	0.704 $\pm$ 0.259	0.002 $\pm$ 0.019
1	Adult Income	0.256 $\pm$ 0.406	0.215 $\pm$ 0.129	0.826 $\pm$ 0.148	0.911 $\pm$ 0.12	0.049 $\pm$ 0.104
	GiveMeSomeCredit	0.165 $\pm$ 0.371	0.05 $\pm$ 0.006	0.847 $\pm$ 0.095	0.962 $\pm$ 0.14	0.009 $\pm$ 0.021
	HELOC	0.033 $\pm$ 0.179	0.06 $\pm$ 0.047	0.937 $\pm$ 0.057	0.68 $\pm$ 0.258	0.003 $\pm$ 0.019
2	Adult Income	0.305 $\pm$ 0.46	0.204 $\pm$ 0.121	0.833 $\pm$ 0.137	0.906 $\pm$ 0.121	0.054 $\pm$ 0.098
	GiveMeSomeCredit	0.769 $\pm$ 0.5	0.06 $\pm$ 0.006	0.612 $\pm$ 0.126	0.957 $\pm$ 0.134	0.047 $\pm$ 0.03
	HELOC	0.037 $\pm$ 0.188	0.076 $\pm$ 0.05	0.912 $\pm$ 0.059	0.684 $\pm$ 0.256	0.004 $\pm$ 0.025

## 5.6 Discussion and Ethical Implications

In the new preference-based framework, traditional metrics like Sparsity and Proximity have undergone a significant transformation. Previously, they served as automatic methods to gauge a rational user’s preference. However, when applied in this new setting, they risk providing solutions that may not align with the user’s actual preferences. So, with the introduction of a more realistic modeling of user preference and Personal Validity, these metrics become outdated.

When considering the ethical implications of our work, several key aspects deserve attention.

- Privacy and Data Handling: Users have the option to keep their preferences confidential, but expressing preferences accurately is important for optimal recourse. Failing to provide preferences can affect preference estimation and recourse quality. The algorithm should prioritize data security, not retaining user data beyond creating recourse, to ensure user privacy.
- The presence of bias or unfairness in the treatment of features within the model hinges on its design. To enhance fairness, a null preference for specific features can be integrated, addressing potential bias or unfairness in the approach.

The broader issue of ethics in counterfactuals is multifaceted. However, we maintain that it falls beyond the scope of our current work. Our primary focus is on the development of a methodology rather than the creation of an operational product. The assurance of ethical practices ultimately hinges on the specifics of implementation.

## 6 Conclusions

In this study, we introduced HIP-CORE (Human-In-the-Loop Preference COunterfactual REcourse) to incorporate user preference in the generation of counterfactual recourse through a human-in-the-loop process. We have formalized the modeling of

preference, positioning it as a fundamental property in the creation of personalized counterfactuals. Acknowledging that user preference is not known a priori, we have mathematically formalized the estimation of user preferences, establishing a foundation for new opportunities in this area.

In future works, we plan to further investigate the mathematical implication of the modeling and the estimation of the user preference in the counterfactual recourse setting. For instance, we want to provide a more comprehensive analysis of the preference estimate, considering more specific types of features, and exploring scenarios where the problem might have solutions, and of which type (unique or multiple solutions might exist).

We also intend to investigate possible solutions to the multi-objective problem, which may lead to the identification of a variety of trade-off solutions across the objectives. Furthermore, we will evaluate and integrate other counterfactual metrics in the multi-objective problem with an in-depth analysis of the trade-off between them.

Similarly, we intend to expand the experiments to scenarios where feature independence is not assumed, exploring potential feature interactions, as well as real-world applications. We will also explore modeling preference and, consequently, counterfactual recourse while considering the element of time.

In conclusion, we earnestly believe that this study underscores the paramount importance of considering users and their preferences when generating recourse. We hope this could serve as an encouragement for the counterfactual recourse community to adopt our proposed modeling approach and incorporate user preferences into the counterfactual recourse framework.

**Acknowledgments.** This work was partially supported by projects FAIR (PE0000013) and SERICS (PE0000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU. Supported also by the ERC Advanced Grant 788893 AMDROMA, EC H2020RIA project “SoBigData++” (871042), PNRR MUR project IR0000013-SoBigData.it. This work has been supported by the project NEREO (Neural Reasoning over Open Data) project funded by the Italian Ministry of Education and Research (PRIN) Grant no. 2022AEFHA

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## A Appendix

Here, we outline how we modeled and simulated the preferences for the variables in the datasets.

- Gaussian Preferences:  $\theta_2 \in (0, 10]$
- Exponential Preferences:  $\theta \in (0, 10]$
- Categorical Preferences:  $\theta_i \in (0, 1) \quad \forall i \in \{1, \dots, K\} \quad s.t. \sum_{i=1}^K \theta_i = 1$ , where  $K$  is the number of categories the feature has

**Adult Income**

- age: exponential
- workclass: categorical
- education: categorical
- marital\_status: categorical
- occupation: categorical
- race: degenerate
- gender: degenerate
- hours\_per\_week: gaussian
- income: target

**GiveMeSomeCredit**

- RevolvingUtilizationOfUnsecuredLines: gaussian
- age: exponential
- NumberOfTime30-59DaysPastDueNotWorse: exponential
- DebtRatio: gaussian
- MonthlyIncome: gaussian
- NumberOfOpenCreditLinesAndLoans: gaussian
- NumberOfTimes90DaysLate: exponential
- NumberRealEstateLoansOrLines: gaussian
- NumberOfTime60-89DaysPastDueNotWorse: exponential
- NumberOfDependents: exponential
- SeriousDlqin2yrs: target

**HELOC**

- ExternalRiskEstimate: gaussian
- MSinceOldestTradeOpen: gaussian
- MSinceMostRecentTradeOpen: gaussian
- AverageMInFile: gaussian
- NumSatisfactoryTrades: exponential
- NumTrades60Ever2DerogPubRec: gaussian
- NumTrades90Ever2DerogPubRec: gaussian
- PercentTradesNeverDelq: gaussian
- MSinceMostRecentDelq: gaussian
- MaxDelq2PublicRecLast12M: gaussian
- MaxDelqEver: exponential
- NumTotalTrades: exponential
- NumTradesOpeninLast12M: gaussian
- PercentInstallTrades: gaussian
- MSinceMostRecentInqexcl7days: gaussian
- NumInqLast6M: gaussian
- NumInqLast6Mexcl7days: gaussian
- NetFractionRevolvingBurden: gaussian

- NetFractionInstallBurden: gaussian
- NumRevolvingTradesWBalance: gaussian
- NumInstallTradesWBalance: gaussian
- NumBank2NatlTradesWHighUtilization: gaussian
- PercentTradesWBalance: gaussian
- RiskPerformance: target

Regarding the classifiers employed, we adopted the default parameter settings from the Python scikit-learn library implementations.

## References

1. Artelt, A., Hammer, B.: On the computation of counterfactual explanations - a survey. arXiv preprint [arXiv:1911.07749](https://arxiv.org/abs/1911.07749) (2019)
2. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996). <https://doi.org/10.24432/C5XW20>
3. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), pp. 785–794. ACM, New York (2016). <https://doi.org/10.1145/2939672.2939785>
4. Credit Fusion, W.C.: Give me some credit (2011). <https://kaggle.com/competitions/GiveMeSomeCredit>
5. De Toni, G., Viappiani, P., Lepri, B., Passerini, A.: Generating personalized counterfactual interventions for algorithmic recourse by eliciting user preferences. arXiv preprint [arXiv:2205.13743](https://arxiv.org/abs/2205.13743) (2022)
6. Fonseca, J.A., Bell, A., Abrate, C., Bonchi, F., Stoyanovich, J.: Setting the right expectations: algorithmic recourse over time. In: Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO 2023). Association for Computing Machinery, New York (2023). <https://doi.org/10.1145/3617694.3623251>
7. Holter, S., Gomez, O., Bertini, E.: FICO Explainable Machine Learning Challenge. FICO CCommunity (2018). <https://community.fico.com/s/explainable-machine-learning-challenge>
8. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. arXiv preprint [arXiv:2010.04050](https://arxiv.org/abs/2010.04050) (2020)
9. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 353–362 (2021)
10. Laugel, T., Jeyasothy, A., Lesot, M.J., Marsala, C., Detyniecki, M.: Achieving diversity in counterfactual explanations: a review and discussion. In: Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, pp. 1859–1869 (2023)
11. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: Inverse classification for comparison-based interpretability in machine learning. arXiv preprint [arXiv:1712.08443](https://arxiv.org/abs/1712.08443) (2017)
12. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. arXiv preprint [arXiv:1905.07697](https://arxiv.org/abs/1905.07697) (2019)
13. Pawelczyk, M., Leemann, T., Biega, A., Kasneci, G.: On the trade-off between actionable explanations and the right to be forgotten. arXiv preprint [arXiv:2208.14137](https://arxiv.org/abs/2208.14137) (2022)
14. Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Comput. J.* **7**(2), 155–162 (1964). <https://doi.org/10.1093/comjnl/7.2.155>

15. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 344–350 (2020)
16. Rasouli, P., Yu, I.C.: CARE: coherent actionable recourse based on sound counterfactual explanations. arXiv preprint [arXiv:2108.08197](https://arxiv.org/abs/2108.08197) (2021)
17. Rawal, K., Lakkaraju, H.: Beyond individualized recourse: interpretable and interactive summaries of actionable recourses. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 12187–12198. Curran Associates, Inc. (2020). [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/8ee7730e97c67473a424ccfeff49ab20-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/8ee7730e97c67473a424ccfeff49ab20-Paper.pdf)
18. Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review. arXiv preprint [arXiv:2010.10596](https://arxiv.org/abs/2010.10596) (2020)
19. Voigt, P., von dem Bussche, A.: Introduction and ‘Checklist’. In: Voigt, P., von dem Bussche, A. (eds.) The EU General Data Protection Regulation (GDPR), pp. 1–7. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-57959-7\\_1](https://doi.org/10.1007/978-3-319-57959-7_1)
20. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017)
21. Wang, Z.J., Vaughan, J.W., Caruana, R., Chau, D.H.: GAM coach: towards interactive and user-centered algorithmic recourse. In: Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems. ACM (2023). <https://doi.org/10.1145/3544548.3580816>
22. Watson, D.S., Floridi, L.: The Explanation Game: A Formal Framework for Interpretable Machine Learning. In: Ethics, Governance, and Policies in Artificial Intelligence, pp. 185–219. Springer International Publishing, Cham (2021). [https://doi.org/10.1007/978-3-030-81907-1\\_11](https://doi.org/10.1007/978-3-030-81907-1_11)
23. Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., He, L.: A survey of human-in-the-loop for machine learning. arXiv preprint [arXiv:2108.00941](https://arxiv.org/abs/2108.00941) (2021)
24. Yadav, P., Hase, P., Bansal, M.: Low-cost algorithmic recourse for users with uncertain cost functions. arXiv preprint [arXiv:2111.01235](https://arxiv.org/abs/2111.01235) (2021)
25. Yetukuri, J., Hardy, I., Liu, Y.: Towards user guided actionable recourse. In: Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society, pp. 742–751 (2023)



# COIN: Counterfactual Inpainting for Weakly Supervised Semantic Segmentation for Medical Images

Dmytro Shvetsov<sup>1</sup> , Joonas Ariva<sup>1,2,3</sup> , Marharyta Domnich<sup>1</sup> ,  
Raul Vicente<sup>1</sup> , and Dmytro Fishman<sup>1,2,3</sup> 

<sup>1</sup> Institute of Computer Science, University of Tartu, Tartu, Estonia  
shvetsovdi2@gmail.com, marharyta.domnich@ut.ee

<sup>2</sup> Better Medicine, Tartu, Estonia

<sup>3</sup> STACC, Tartu, Estonia

**Abstract.** Deep learning is dramatically transforming the field of medical imaging and radiology, enabling the identification of pathologies in medical images, including computed tomography (CT) and X-ray scans. However, the performance of deep learning models, particularly in segmentation tasks, is often limited by the need for extensive annotated datasets. To address this challenge, the capabilities of weakly supervised semantic segmentation are explored through the lens of Explainable AI and the generation of counterfactual explanations. The scope of this research is development of a novel counterfactual inpainting approach (COIN) that flips the predicted classification label from abnormal to normal by using a generative model. For instance, if the classifier deems an input medical image  $X$  as abnormal, indicating the presence of a pathology, the generative model aims to inpaint the abnormal region, thus reversing the classifier's original prediction label. The approach enables us to produce precise segmentations for pathologies without depending on pre-existing segmentation masks. Crucially, image-level labels are utilized, which are substantially easier to acquire than creating detailed segmentation masks. The effectiveness of the method is demonstrated by segmenting synthetic targets and actual kidney tumors from CT images acquired from Tartu University Hospital in Estonia. The findings indicate that COIN greatly surpasses established attribution methods, such as RISE, ScoreCAM, and LayerCAM, as well as an alternative counterfactual explanation method introduced by Singla et al. This evidence suggests that COIN is a promising approach for semantic segmentation of tumors in CT images, and presents a step forward in making deep learning applications more accessible and effective in healthcare, where annotated data is scarce.

**Keywords:** Explainable AI · Counterfactual explanations · GANs · Semantic Segmentation · Medical Imaging · CT scans · Kidney Tumour



## 1 Introduction

Deep learning is revolutionizing the field of medical imaging [5, 26] and radiology [12], offering the potential to aid radiologists by triaging incoming patients and detecting various pathologies from medical images like computed tomography (CT) or X-ray scans. However, the accuracy of deep learning models for medical images hinges on access to substantial annotated datasets [10]. Collecting such datasets is hard for two reasons: 1) labeling medical images accurately requires the knowledge of trained medical professionals such as radiologists, 2) accurate manual image labeling is very labor intensive. These problems are further amplified by the limited access to medical image datasets due to data protection laws [7].

Considering these challenges, there is a pressing need for methods that can automate or simplify the manual data labelling process. Dense pixel-level annotations, though highly informative, are particularly time-consuming to create [34]. In contrast, image-level annotations, which indicate the presence or absence of certain organs or pathologies, are more feasible to obtain and can be derived from accompanying radiology reports. This scenario raises an intriguing question: Can one generate detailed pixel labels based solely on image-level labels?

This task falls under weakly supervised semantic segmentation (WSSS) and it is often tackled with methods known from Explainable AI (XAI) [1, 8, 9, 32]. In computer vision, the XAI’s task is usually to explain a “black box” classifier by highlighting the most important regions of the images for classifiers decisions. The principle of generating saliency maps from a classifier is partly transferable over to the task of WSSS as the saliency maps can be seen as segmentation masks. However, as noted in [11], saliency maps, while visually intuitive, often blend useful and non-useful information, making it hard to identify the specific image features that are important for model’s decisions. This makes it difficult to generate precise segmentations from the saliency maps.

Counterfactual explanations recently emerged that seeks for minimal change in input to flip the decision output of classifier [24, 36]. This approach aims not only to flip the decision of a model in adversarial attack manner, but to ensure that modifications are meaningful and interpretable in a real-world context as highlighted in [17, 39]. First trained counterfactual explanation models with usage of conditional Generative Adversarial Network (cGAN) [33] showed great promise in providing insights into decision-making processes of classifiers and uncovering potential biases or failure modes. Building upon this work, this research explores the potential of adapting the counterfactual explanation framework for the domain of WSSS. It is argued that the difference between original input and its respective counterfactual image can serve as implicit segmentation masks, while also revealing critical features for classification decisions.

In this context, this study extends the application of the generative counterfactual explanation method to facilitate the generation of segmentation labels. The original methodology and architecture are refined to better suit the needs of WSSS, thereby eliminating the dependence on pre-existing segmentation masks for training. This approach allows for the development of a more efficient, weakly

supervised learning framework, enhancing the precision of segmentation outcomes. The method is tested by segmenting kidney tumors from CT images. The adapted counterfactual pipeline efficiently produces accurate segmentation labels from straightforward classification models. This innovation is substantiated through comprehensive testing and validation on a synthetic dataset. Furthermore, the novel counterfactual pipeline is compared against established attribution methods—RISE [29], Score-CAM [37], and LayerCAM [18]—to affirm its enhanced capability and practical applicability in generating segmentation labels under weak supervision.

## 2 Related Works

### 2.1 Weakly Supervised Semantic Segmentation

WSSS has gathered a lot of attention as the idea of segmentation masks from image-level labels makes it very cost-effective. WSSS methods often utilize class activation maps (CAMs). In CAM methods the saliency maps are generated from classifier model and input image to indicate the most important image regions for the classifiers decision for each class. Frequently used CAM methods include GradCAM, ScoreCAM and LayerCAM [18,31,37]. CAM based WSSS methods consist of the following steps: 1) Training of the classifier 2) Extracting saliency maps with a CAM method 3) Refining the saliency maps by postprocessing. Additionally, sometimes these refined maps are used to train a segmentation model and then the model is used to acquire the final segmentations.

However, CAM methods are not ideal for WSSS and are not without issues. One of the major problem with CAMs is that they highlight only the most discriminative regions of the image and not the full object that represents the class. This can lead to poor segmentation performance. Secondly, the resolution of saliency maps from CAMs is tied to the resolution of the activation maps from the classifier model. Using the activations from deepest layers of the model usually yields semantically more representative saliency maps. At the same time these activations have low resolution which creates low resolution saliency maps. Moreover, saliency maps can remain unchanged even when the underlying model predictions are significantly affected by adversarial attacks as pointed in [11], raising questions about their reliability as explanations.

The potential way to overcome these issues in WSSS is to use counterfactual explanations instead of CAMs.

### 2.2 Counterfactual Explanations

Counterfactual explanation is model-agnostic instance-based method that answers the question “What is the minimum input change that leads to the flip of the prediction outcome?”. Originating from the fields of cognitive science [6], psychology [20], and causality research [28], counterfactual explanations have been explored as a way to understand AI’s decision-making processes. Wachter et

al. [36], introduced a formal counterfactual optimization function for generating explanations in continuous data, marking a significant milestone in XAI. Furthermore, Tim Miller’s insights from social sciences [23] highlighted the criticality of contrastive explanations for human reasoning and decision-making processes and the necessity of counterfactual explanations for XAI. Extensive surveys on counterfactual explanations by Guidotti et al. [13] and Karimi et al. [19] showed big variety of counterfactual explanation frameworks primarily focused on discrete data. These frameworks either solve optimization problems or employ heuristics search strategies to find counterfactual explanations. In image domain a notable advancement introduced Akula et al. [2], who highlighted the importance of Theory of Mind for Explainability and developed a pipeline for generating counterfactuals in images by identifying and modifying minimal semantic-level features, such as altering the stripes on a zebra.

The application of Generative Adversarial Networks (GANs) for creating counterfactual explanation has gained increasing attention, demonstrating the versatility across a variety of domains. For instance, Kenny et al. [21] PIECE method demonstrated this on MNIST data by altering exceptional image features to generate plausible counterfactuals for black-box CNN classifiers. In the domain of autonomous driving, a field where explainability is crucial due to the safety-critical nature of its applications, Zemni et al. [39] proposed an object-centric framework for counterfactual generation. Their method was specifically designed for images with many objects, such as urban scenes common in autonomous driving. By encoding the query image into a structured latent space, this approach facilitates object-level manipulations, making it highly suitable for complex scenes. The method was tested on counterfactual explanation benchmarks for driving scenes, demonstrating its capability to adapt beyond classification to explain semantic segmentation models. Jeanneret et al. [17] focused on transforming adversarial attacks into semantically meaningful perturbations for facial expression data. Their work hypothesized that Denoising Diffusion Probabilistic Models could regularize adversarial attacks to generate actionable and understandable image modifications, such as making sad people happy. Bischof et al. [4] proposed a unified framework leveraging image-to-image translation GANs to address interpretability and robustness in neural image classifiers. This framework was designed and assessed on two specific applications: a semantic segmentation task for concrete cracks and a fruit defects detection problem. Through this, they produced saliency maps for interpretability and demonstrated improved model robustness against adversarial attacks.

In the medical field, counterfactual explanations have shown great promise for diagnostic purposes, particularly in analyzing chest X-ray images. The work by Atad et al., [3] employed a StyleGAN-based approach, StyleEx, to manipulate specific latent directions in chest X-ray images. They demonstrated the use of counterfactual explanations in analyzing chest X-ray images helping to identify the patterns that models rely on for diagnoses, which was clinically evaluated with radiologists. Similarly Singla et al. [33] used counterfactual explanation generation for chest X-rays images to explain the decision-making processes of

image classifiers. They trained a generative model capable of producing images that would lead to a different classification by the original model by preserving original context from an instance. Their generative model was trained and validated on chest X-ray images, with a human-grounded study confirming the usefulness of the generated explanations in a medical context.

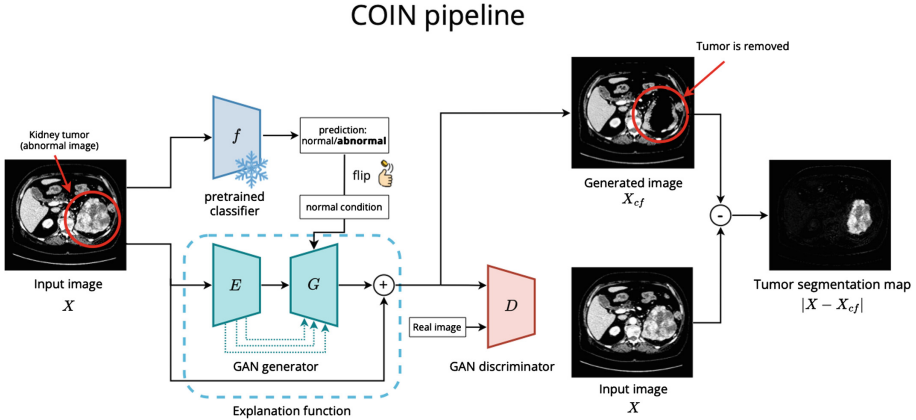
While these works have illustrated the prominent capabilities of counterfactual explanations across various tasks, their application from the perspective of WSSS remains unexplored. Most methods have relied on segmentation masks within GAN training, which may not be available in this context. COIN aims to bridge this gap by adapting the counterfactual approach for segmentation purposes, particularly focusing on the work of Singla et al.’s [33] methodology and adapting it to generate segmentation labels without using pre-existing masks.

### 3 Counterfactual Approach for WSSS

The counterfactual inpainting approach is introduced for producing semantic segmentation masks. Counterfactual approach originates from Singla et al. [33] with significantly enhanced architecture, adding perturbation-based generator, skip connections, another loss measure and discarding the usage of segmentation masks to make it more viable for WSSS.

#### 3.1 Method Formulation

Our method is defined in the case of a binary classification. Let’s denote a black-box classifier as  $f$  with assumption that it is differentiable with access to its output value and gradient. The pre-trained binary classifier  $f$  accepts an input image  $X$  and outputs whether an image  $X$  is **normal** ( $y = 0$ ) if  $f(X) < t$  or **abnormal** ( $y = 1$ ) if  $f(X) \geq t$ , where  $t$  is a threshold used to binarize the prediction output of  $f$ . In Singla et al. [33], the generative model (cGAN) accepts an input image  $X$  and a tweakable parameter  $\delta$ , such that the counterfactual image  $X_{cf} = \mathcal{E}(X, \delta)$  flips the prediction class  $y$  and  $\mathcal{E}(\cdot)$  is used as an explanation function. Specifically, if the classifier predicts **abnormal** for an image  $X$ , the generator aims to inpaint or remove the abnormal region, effectively **flipping** the classifier’s prediction. Similarly, if the classifier predicts **normal**, the generated counterfactual image should add the **abnormal** region, **flipping** the classifier’s prediction. In contrast, this research argues that only the inpainting case should be considered for WSSS purposes. Our generative model predicts a **normal** counterfactual image  $X_{cf}$  only when the classifier deems the input image  $X$  as **abnormal**. Subsequently, the absolute difference between the counterfactual image  $X_{cf}$  and the original image  $X$  serves as a weak segmentation label. The overview of the COIN method is depicted in Fig. 1.



**Fig. 1.** Overview of the proposed counterfactual inpainting (COIN) pipeline. Given the input image  $X$  and black-box classifier  $f$  that produces a classification label, the image-to-image model (GAN) generates a counterfactual image  $X_{cf}$  with  $y = 0$ . If  $X$  is **abnormal**, it is expected that  $X_{cf}$  no longer contains the abnormal part of the input image. Computing the absolute difference of the original image  $X$  and counterfactual image  $X_{cf}$  results in a weak tumor segmentation map. While training the pipeline, only GAN weights are updated. Classifier predictions are used for classifier consistency loss calculation.

### 3.2 Image Generation Architecture

The SNGAN [25] architecture is adapted for the generator and discriminator networks. The original generative model (cGAN) consists of an encoder  $E(X) = z$  and a decoder  $G(z, \delta) = X_{cf}$ . Consequently, the explanation function can be expressed as  $\mathcal{E}(X, \delta) = G(E(X), \delta)$ . The encoder transforms an input image  $X$  into a latent representation  $z$ , which is then passed into the decoder along with a condition label  $\delta$  to produce a counterfactual image  $X_{cf}$ . Unlike the approach by Singla et al. [33], where the parameter  $\delta$  is utilized to provide unique offsets by discretizing the  $[0; 1]$  range into  $N = 10$  equal bins—each corresponding to a unique condition vector  $c$ —COIN simplifies the architecture to handle only one condition ( $c_{inp}$  - to inpaint or remove the abnormal region). The model is trained as a simple GAN that produces counterfactual image  $X_{cf}$  where  $f(X_{cf}) < t$ . In this case, the flipping happens only in one direction when the  $X$  is **abnormal** and  $X_{cf}$  becomes **normal**, removing the area that affects the classifier’s prediction. This approach addresses the challenge of needing a condition-balanced training set for the cGAN by reducing the complexity and data requirements. The benefits of decreasing number of condition is showcased in Appendix A Table 3.

Moreover, COIN distinguishes itself by implementing a perturbation-based image generation within the GAN framework. The generator architecture is unique with slight modification of having a residual connection of the model’s input to the outputs. Instead of regenerating the complete image in the decoder from the latent variable, the generator is trained to produce only the perturba-

tion map which is fused into the input image. Therefore, the explanation function is modified such that  $\mathcal{E}(X, \delta) = G(E(X), \delta) + X$ . This technique contrasts with conventional counterfactual generation models [4, 17, 39], which typically reconstruct the entire image. While full reconstruction can be effective in some contexts, it often introduces artifacts - unintended alterations that can skew the classification model’s interpretation and analysis. The impact of perturbation-based architecture is illustrated in Appendix A Fig. 4 and Table 3.

Additionally, in contrast to reported Singla et al. architecture, the skip-connections are integrated into the encoder-decoder model [30] together with perturbation-based approach. This enhancement facilitates the generation of more accurate perturbations, thereby improving reconstruction quality and preserving the original image details. The impact of skip-connections is showcased in Appendix A Fig. 5 and Table 3. This approach is beneficial for WSSS as it ensures the generation of precise, artifact-free counterfactual explanations.

### 3.3 Loss Function for Training GAN

The same loss functions are inherited as described by Singla et al. [33], and introduce an additional Total-Variation loss [16] to enforce smoothness in the generated images. The complete objective function for the counterfactual inpainting pipeline is given as follows:

$$\min_{E,G} \max_D (\lambda_{GAN} L_{GAN} + \lambda_f \mathcal{L}_f + \lambda_{idt} \mathcal{L}_{idt} + \lambda_{tv} \mathcal{L}_{tv}), \quad (1)$$

where  $E$ ,  $G$  and  $D$  is the Encoder, Decoder and Discriminator of GAN;  $f$  is a pre-trained classifier;  $L_{GAN}$  is a data consistency loss term;  $\mathcal{L}_f$  is a classification model consistency loss term;  $\mathcal{L}_{idt}$  is a domain-aware self-consistency loss term, and  $\mathcal{L}_{tv}$  is a Total-variation loss term;  $\lambda_{GAN}$ ,  $\lambda_f$ ,  $\lambda_{idt}$ ,  $\lambda_{tv}$  are respective hyper-parameters to configure contribution of each term.

**Data Consistency Loss Term.** Generated images should look similar to the images in the training dataset. For GAN, the two networks are trained: generator, which consists of encoder  $E(\cdot)$  and decoder  $G(\cdot)$ , and discriminator  $D(\cdot)$  and compute binary cross-entropy loss on the real/fake labels.

$$\mathcal{L}_{GAN} = \mathbb{E}_{X \sim P(X)} [\log(D(X))] + \mathbb{E}_{X \sim P(X_{cf})} [\log(1 - D(E(G(X))))], \quad (2)$$

where  $P(X)$  and  $P(X_{cf})$  denote distributions of real and generated images respectively.

**Classification Model Consistency Loss Term.** A GAN should generate the counterfactual images that influence the classifier predictions in the desired manner. The general formulation of the objective function computes the KL divergence between the predicted and expected distributions of probabilities. In the proposed reformulation for the inpainting pipeline, the condition-aware loss simplifies to:

$$\mathcal{L}_f = D_{\text{KL}}(f(\mathcal{E}(X)) \parallel 0), \quad (3)$$

where  $\mathcal{E}(X)$  is a counterfactual explanation derived from an input image  $X$ .

**Domain-Aware Self-consistency Loss Term.** Similarly to Singla et al. [33], the main idea behind the objective function is to let the model learn cyclically consistent counterfactual images when applying a series of counterfactual generations. One cycle of generations is computed to produce  $\mathcal{E}(X)$  and  $\mathcal{E}(\mathcal{E}(X))$ , given that  $\mathcal{E}(\cdot)$  - is an explanation function. Both counterfactual images should retain as much details as possible from the input image perturbing it only if it is abnormal. Additionally, Singla et al. [33] uses segmentation masks to enforce local consistency over the foreground pixels of different segmentation labels present in the image. However, this research employs a simpler supervision, not requiring segmentation masks, to compute the reconstruction loss over the whole image instead of local regions. The domain-aware self-consistency loss for the proposed counterfactual inpainting pipeline is given by:

$$\mathcal{L}_{\text{idt}} = \mathcal{L}_1(X, \mathcal{E}(X)) + \mathcal{L}_1(X, \mathcal{E}(\mathcal{E}(X))), \quad (4)$$

where

$$\mathcal{L}_1(X, X') = \frac{\|X - X'\|_1}{HW}, \quad (5)$$

where  $H$  and  $W$  represent height, width of the images  $X$  and  $X'$ .

**Total-Variation Loss Term.** To further improve the smoothness of the segmentation masks, a Total-Variation (TV) loss [16] is adopted and computed directly from the difference maps. TV loss enforces smoothness for the generated counterfactuals suppressing the noise and preserving the edges at the same time. The formula for the objective function is as follows:

$$L_{tv} = \frac{1}{HW} \left( \sum_{i=1}^{H-1} \sum_{j=1}^W (x_{i+1,j} - x_{i,j})^2 + \sum_{i=1}^H \sum_{j=1}^{W-1} (x_{i,j+1} - x_{i,j})^2 \right), \quad (6)$$

where  $x_{i,j}$  is the intensity of a pixel at position  $(i, j)$  in the input image  $X$ .

TV loss serves as a regularization term and improves consistency in the raw difference maps enforcing the model to perturb only densely located regions.

## 4 Experiments

### 4.1 Datasets

**TotalSegmentator.** The TotalSegmentator [38] dataset is an extensive collection of CT imaging data, particularly designed to train and evaluate algorithms for the task of image segmentation. Originating from a wide array of sources,

it encompasses a diverse set of medical scans, including those of the chest, abdomen, and pelvis regions, among others. In addition to manually labelled ground truth masks, a great portion of the dataset is annotated with pre-trained segmentation models, which introduces some level of noise to the annotations.

Synthetic anomalies are generated inside the kidneys of these scans for development of the model before moving to segmentation of real tumors. Generation of synthetic anomalies is described in Sect. A.2. The dataset’s scans are split randomly into training and validation sets with a ratio of 80%/20%.

**TUH.** The Tartu University Hospital kidney tumor dataset contains contrast enhanced CT scans of 291 kidney tumor cases and 300 control cases with pixel-level annotations for classes *kidney*, *malignant lesion* and *benign lesion*. Dataset was annotated by five radiologist and each scan was annotated by at least two of them. Final labels were produced by combining the two versions. If any major disagreements presented themselves they were resolved in direct discussion between radiologists. From pixel-level labels, the image-level labels are extracted of whether the slice contains a malignant lesion (kidney tumor) or not and used these labels for training of the classifier. Pixel-level labels were only used for evaluating the final segmentations quality. The dataset’s scans are split randomly into training and validation sets with a ratio of 80%/20% stratified by the total tumor area in voxels present in each scan.

## 4.2 Evaluation

**Realism of Generated Images.** The Fréchet inception distance (FID) score [15] is a widely used for measuring the similarity between generated images and a real sets of images. It involves the use of a pre-trained deep learning model to extract feature vectors from both sets of images. These features encapsulate various aspects of the images, such as textures, edges, and patterns. The similarity is defined as the distance between the activation distributions of the real image  $x$  and the synthetic explanations  $x_c$  as,

$$FID(x, x_\delta) = \|\mu_{\mathbf{x}} - \mu_{\mathbf{x}_\delta}\|_2^2 + \text{Tr}(\Sigma_{\mathbf{x}} + \Sigma_{\mathbf{x}_\delta} - 2(\Sigma_{\mathbf{x}}\Sigma_{\mathbf{x}_\delta})^{\frac{1}{2}}), \quad (7)$$

where  $\mu$ ’s and  $\Sigma$ ’s are mean and covariance of the activation vectors derived from the penultimate layer of a pre-trained Inception v3 network.

**Classifier Consistency.** Counterfactual Validity (CV) score is a metric defines the fraction of counterfactual explanations that successfully flipped the classification decision, e.g. if the input image is positive, the explanation should be predicted as negative. Prediction flip is considered successful when the difference of predictions  $|f(X) - f(X_{cf})|$  is over a certain threshold  $\tau$ . Similarly to the original study [33],  $\tau = 0.8$  is used.



**Segmentation Metric.** Intersection Over Union (IoU) score is a widely adopted metric for evaluating segmentation accuracy. It is computed as follows:

$$IoU(S, S_c) = \frac{|S \cap S_c|}{|S \cup S_c|} = \frac{TP}{TP + FP + FN}, \quad (8)$$

where  $S$  and  $S_c$  are ground truth and predicted segmentation masks respectively and  $TP$ ,  $FP$ ,  $FN$  stand for true positive, false positive and false negative predictions.

### 4.3 Implementation Details

All neural networks were implemented in PyTorch [27] and were trained on the High Performance Computing (HPC) cluster of the University of Tartu on a single Nvidia A100-SXM-40 GB GPU. Only kidney slices with kidney mask area of at least 32 pixels per image are used from each dataset, which are resized to  $256 \times 256$  with bilinear interpolation and sampled into batches of 16 images. The training pipeline consists of two independent stages, namely classifier and explanation model training. The parameters of the classifier remain frozen throughout training process of the GAN. As for the classifier architecture, the models like ResNet18 [14] for the synthetic anomalies and EfficientNet-V2 [35] for real tumors are used respectively. The Adam [22] optimizer is used for optimizing the objective function with parameters  $\alpha = 0.0002$ ,  $\beta_1 = 0$ ,  $\beta_2 = 0.9$ . The segmentation masks from counterfactual generation are extracted as taking the absolute difference between the counterfactual and input images, and thresholding it with a fixed value.

### 4.4 Comparison with Modified Singla et al.\* Method

To establish a baseline comparison, the main purpose is to contrast the proposed method with that of Singla et al. However, the direct comparison is challenging due to the absence of the model’s code, necessitating a unique implementation. To establish fair comparison, the reliance on segmentation masks should additionally be removed in the loss function, as the goal is to detect the masks for WSSS without using them explicitly. Thus, the loss terms were adapted accordingly. The reconstruction loss is computed as a plain  $L_1$  function averaging over all the pixels in the generated and input images. This modification, however, resulted in poor segmentation performance. Suspecting unreported skip connections in their cGAN model, the baseline was enhanced by incorporating skip connections. In the given results, this adjusted model is referred as the modified Singla et al\*.

## 5 Results

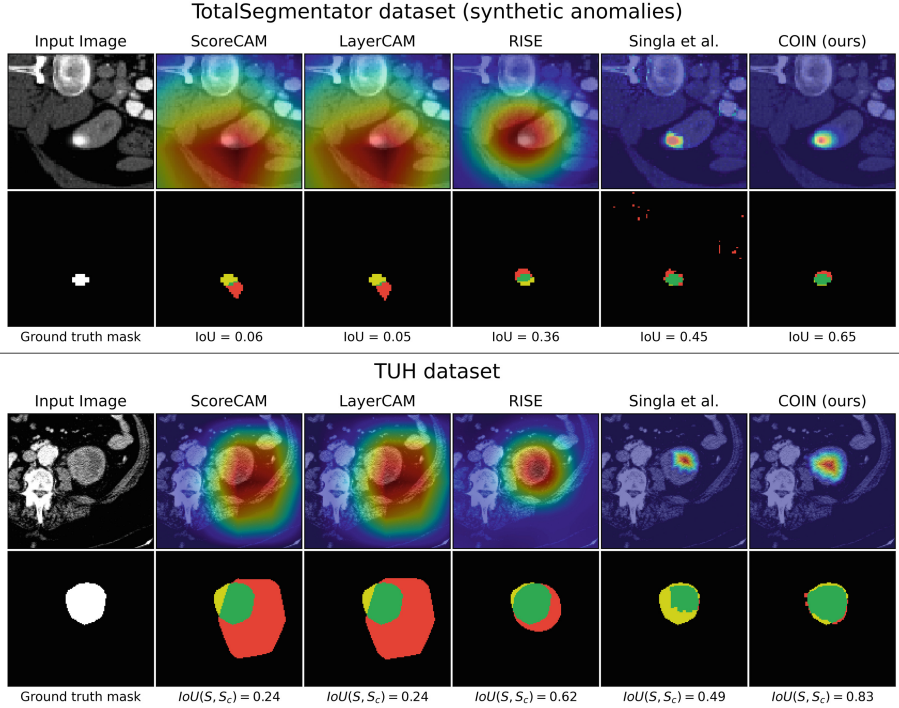
In this chapter, evaluation results are given for the proposed COIN method compared to attribution methods [18, 29, 37]. To convert the attention maps produced by all the methods, the outputs are normalized to the  $[0; 1]$  range and threshold with a fixed value. The 0–1 range sweep is performed to find the best binarization threshold that maximizes IoU for each method. For the counterfactual methods, a morphological postprocessing of closing and opening is applied to suppress noise and retain only one largest connected component in the masks. The results are presented in the Table 1 and the Fig. 2 for synthetic anomalies and real tumors respectively on a test set. In all the comparisons, COIN outperforms the alternative methods by a large margin in terms of IoU of up to 60% on synthetic anomalies, and up to 14% on real tumors. Moreover, while attribution methods like CAM and RISE are computationally inexpensive, as they do not require training an auxiliary model, they generally yield less accurate results and do not have static inference times, requiring multiple forward passes to compute saliency maps, which makes them less efficient for large datasets. In contrast, COIN, despite requiring approximately 20 h of training on a single GPU, necessitates only a single forward pass and significantly enhances performance, achieving superior results that justify the additional computation time and making it more efficient for extensive data applications.

**Table 1.** Metric results for the attribution methods and the proposed counterfactual inpainting pipeline on TUH dataset. Since CAMs and RISE do not create counterfactual images, FID and CV metrics cannot be computed for these methods.

Datasets	Methods	FID↓	CV↑	IoU↑
TotalSegmentator	ScoreCAM	–	–	0.030
	LayerCAM	–	–	0.026
	RISE	–	–	0.397
	Singla et al.*	0.047	0.998	0.445
	<b>COIN</b>	<b>0.003</b>	<b>0.997</b>	<b>0.646</b>
Tartu University Hospital	ScoreCAM	–	–	0.293
	LayerCAM	–	–	0.296
	RISE	–	–	0.294
	Singla et al.*	0.203	0.992	0.352
	<b>COIN</b>	<b>0.036</b>	<b>0.980</b>	<b>0.432</b>

### 5.1 Ablation Experiments

In the ablation study of this research, the importance of each loss term’s contribution to the final image realism is assessed, classifier predictions flip rate and segmentation accuracy for the proposed counterfactual inpainting pipeline. This



**Fig. 2.** Visualization of the attribution and the proposed counterfactual inpainting pipeline methods’ predictions on TotalSegmentator and TUH datasets. For each dataset, the bottom row depicts thresholded masks obtained from saliency maps from each method. For each masks, colors represent outcomes in terms of true positive (green), false positive (red) and false negative (yellow) predictions. White masks denote ground truth labels. Images are zoomed in for better clarity. (Color figure online)

**Table 2.** Ablation study results on each loss term contribution based on TotalSegmentator dataset with synthetic anomalies.

Experiment	FID↓	CV↑	IoU↑
$\lambda_{idt} = 0$	0.0024	0.997	0.500
$\lambda_f = 0$	0.0032	0.642	0.424
$\lambda_{tv} = 0$	0.0178	0.997	0.427
COIN baseline	<b>0.0029</b>	<b>0.997</b>	<b>0.646</b>

was achieved by zeroing out the weights of each loss term independently and compare the metric results. The Table 2 presents the results of the experiment.

Classifier consistency loss plays a crucial role for learning good counterfactual images, since excluding it from the loss practically removes the influence of the classifier on the generator outputs, resulting in CV score degradation by 35%.

Self-Consistency and TV losses are important for the model to avoid random perturbations and force the model to focus on perturbing only densely located regions, which enforces minimum change and increases IoU by up to 22%.

## 6 Discussion

In this study, a novel counterfactual inpainting approach is introduced for weakly supervised semantic segmentation, which demonstrated to outperform existing attribution methods and the baseline counterfactual method in the segmentation of synthetic anomalies and real tumors on Tartu University Hospital dataset. Aiming for a fair comparison, the best performing threshold is meticulously selected for all attribution methods. Still, CAM methods substantially underperformed compared to the proposed approach on both synthetic anomalies and real tumors. They primarily rely on high-level classification-relevant image areas and often fail to localize the central parts of anomalies accurately. RISE proved to be more effective than ScoreCAM and LayerCAM methods as it produces the saliency map based on the direct perturbations of the input image with pre-generated masks. The intentions for comparison with original Singla et al. approach were challenged by the unavailability of the code. This is addressed through a custom implementation, enhancing their method with added skip connections. It is assumed that the original methodology might have underreported architectural details, given its poor initial performance. The modified Singla et al.\* approach achieved high segmentation results after postprocessing but produced lower fidelity images resulting in high FID score. Overall, COIN generates consistent difference maps, enhancing the accuracy and confirming suitability for the WSSS task. Figure 2 offers a qualitative evaluation of the generated counterfactuals for the discussed methods.

**Limitations and Future Works.** A key limitation of the effectiveness of the developed counterfactual inpainting pipeline is its dependency on the performance of the underlying black-box classifier. Although simpler than training a segmentation model, training a classifier to a satisfactory level of accuracy and robustness may become a non-trivial task that requires substantial amounts of labeled data, computational resources, and careful tuning of model parameters. Any imperfections in the classifier, such as biases in the training data, overfitting, or underfitting, can adversely affect the quality of the generated counterfactuals. This, in turn, can lead to poor segmentation labels, which may not accurately reflect the underlying data distribution or the specific features that are of interest in the analysis.

Another notable limitation of the current pipeline is its restriction to 2D analysis, despite the inherently 3D nature of the CT scans dataset. This simplification can lead to a loss of spatial context and information that is crucial for accurate segmentation and analysis of medical images. This limitation highlights the need for extending the pipeline to accommodate 3D data directly which is planned for the future works. Developing methods that can efficiently process

and generate counterfactuals in 3D would significantly enhance the applicability and effectiveness of the pipeline in medical imaging contexts. Recognizing this, future work will focus on extending COIN to accommodate 3D data, thereby enhancing the precision of segmentation masks as 3D input will provide a more comprehensive understanding of the input. Additionally, the weakly supervised segmentation pipeline with counterfactual inpainting should not be confined to tumor data and medical domain in general. Future studies will assess the generality of COIN method, testing its effectiveness across diverse applications and datasets. This exploration is anticipated to be particularly beneficial in scenarios where acquiring segmentation masks is more challenging than obtaining classification labels.

## 7 Conclusion

In this study, a novel strategy is proposed that utilizes explainability for weakly supervised semantic segmentation task for medical domain. By adopting a counterfactual method as a foundation, the method is enhanced with a perturbation-based generator, simplified conditioning for inpainting or removing abnormality, elimination of the need for segmentation masks, and a new loss term for enforcing smoothness in the counterfactuals. All these additions contributed to precise generation of segmentation masks, demonstrating superiority over attribution methods and original counterfactual approach. This innovative approach enhances the capability to generate more nuanced and detailed counterfactual examples resulting in a significant contribution to the field of weakly supervised learning. By addressing the inherent limitations of sparse annotations and leveraging the power of counterfactual reasoning, the inpainting pipeline offers a robust solution for improving semantic segmentation models without the need for extensive manually labeled datasets. The code is released in the public repository:

<https://github.com/Dmytro-Shvetsov/counterfactual-search>

**Acknowledgments.** The data has been collected as part of the Clinical Investigation that has been approved by The University of Tartu Research Ethics Committee (no 332/T-9, dated 21.12.2020), we thank Better Medicine for providing the dataset. This research was supported by the Estonian Research Council Grants PRG1604, the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 952060 (Trust AI), by the Estonian Centre of Excellence in Artificial Intelligence (EXAI), funded by the Estonian Ministry of Education and Research, by Better Medicine and by STACC.

**Disclosure of Interests.** Dmytro Fishman owns stock in Better Medicine. All other authors have no competing interests to declare that are relevant to the content of this article.

## Appendix A

In this chapter, a detailed experimentation is given for the iterative improvements to the Singla et al.\* method to obtain the COIN pipeline. Table 3 summarizes all the performed experiments and obtained metrics based on the TotalSegmentator and synthetic anomalies.

**Table 3.** Metric results for the iterative improvements of the original Singla et al. and the COIN methods. Experiment H refers to the modified Singla et al.\* method.

ID & iterative changes	uses masks	perturbations	skip connections	conditions	FID↓	CV↑	IoU↑
A	✓		0	2	1.6190	0.992	0.020
B (+ perturbations)	✓	✓	0	2	<b>0.5500</b>	<b>0.934</b>	<b>0.086</b>
C (+ skip connection)	✓	✓	1	2	0.3254	0.968	0.284
D (+ skip connection)	✓	✓	2	2	0.1751	0.933	0.480
E (+ skip connection)	✓	✓	3	2	0.0849	0.961	0.463
F (+ skip connection)	✓	✓	4	2	<b>0.0253</b>	0.9542	<b>0.509</b>
G (- masks)		✓	4	2	0.0493	0.996	0.528
H (- perturbations)			4	2	0.0466	0.998	0.445
COIN		✓	4	1	<b>0.0029</b>	<b>0.997</b>	<b>0.646</b>

### A.1 Loss Function for Dual-Conditioning in Singla et al.\*

In this chapter, the main difference between COIN and the original Singla et al.\* is described in terms of loss functions for training the image generation model. Firstly, the model of Singla et al.\* with two conditions accepts an input image  $X$  and a condition parameter  $\delta$ , so that the counterfactual example  $X_{cf} = \mathcal{E}(X, 1 - f(X))$  yields a **normal** image if  $X$  is **abnormal** or an **abnormal** image if  $X$  is **normal**. To achieve this, the **classification model consistency loss** is introduced as follows:

$$\mathcal{L}_f = D_{\text{KL}}(f(X_{cf}) || 1 - f(X)), \quad (9)$$

In terms of **domain-aware self-consistency loss**, to achieve corresponding counterfactual images when applying a series generations, the objective functions is given as follows:

$$\mathcal{L}_{\text{idt}} = \mathcal{L}_{\text{rec}}(X, \mathcal{E}(X, f(X))) + \mathcal{L}_{\text{rec}}(X, \mathcal{E}(X, 1 - f(X)), f(X)), \quad (10)$$

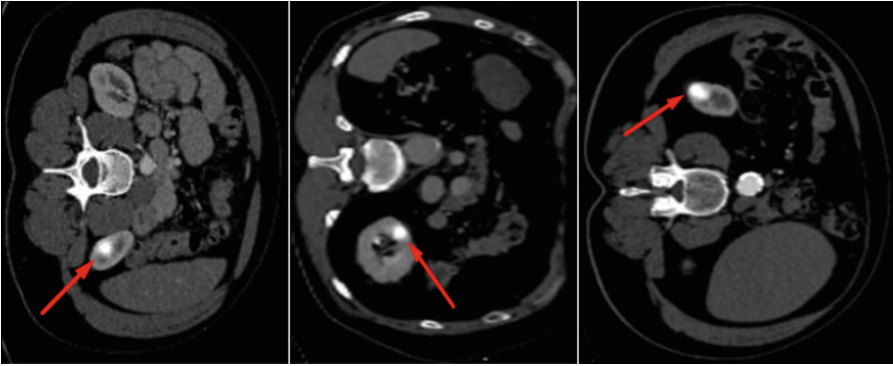
where both components enforce the model to predict the identity image. However, the first component should generate one if conditioned on the  $f(X)$ , whereas the second one should generate the identity when doing counterfactual generation two times in a row. For the experiments where it is referred

that segmentation masks are used, the  $\mathcal{L}_{\text{rec}}$  function is defined as the average L1 distance computed between foreground pixels of segmentation mask  $S$  for label  $j$ .

$$\mathcal{L}_{\text{rec}}(X, X') = \sum_j \frac{S_j(X) \cdot \|X - X'\|_1}{\sum_j S_j(X)}, \quad (11)$$

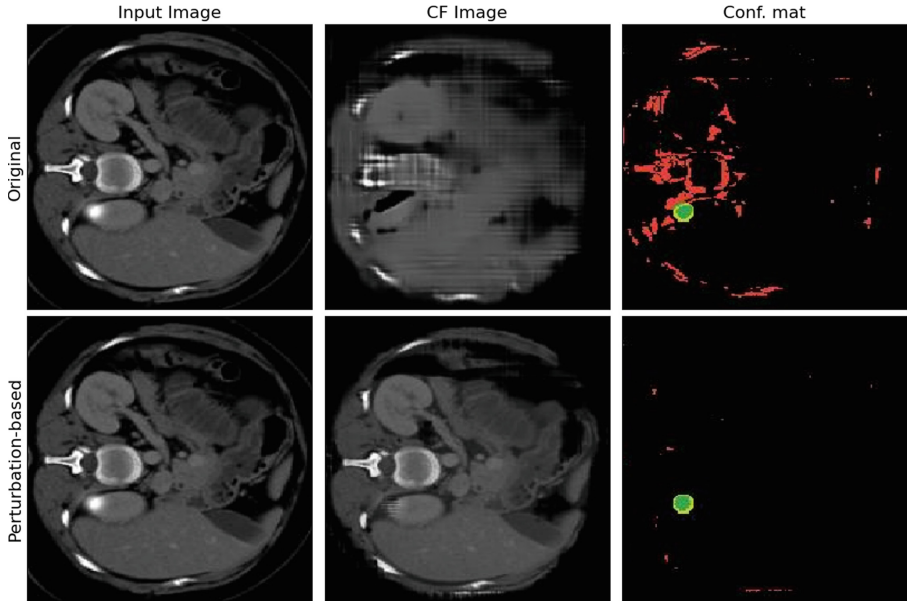
In the case of the experiment with no masks used,  $\mathcal{L}_{\text{rec}}$  is the  $L_1$  function similar to COIN.

## A.2 Synthetic Anomaly Generation



**Fig. 3.** Examples of the synthetic anomalies injected randomly inside kidneys for the TotalSegmentator dataset.

In order to establish a robust benchmark for assessing the performance of the modeling decisions, this research introduces a synthetic anomaly, meticulously designed and integrated into the CT scans datasets. The synthetic anomaly is conceptualized as a Gaussian blob, characterized by a fixed sigma and radius. This design choice is deliberate, aiming to mimic typical radiological findings that present as circular or ellipsoid structures in medical imaging. The synthetic anomaly is sampled at random positions within one of the kidneys in the abdominal slices of the scans in the TotalSegmentator dataset. This approach ensures a diverse and unpredictable distribution of anomalies, closely simulating the randomness and variability. To further enhance the complexity and variability of the synthetic anomaly, a series of transformation and augmentation techniques are employed, including random grid distortions, scaling and rotations. The examples of resulting gaussian blobs are visualized in Fig. 3.



**Fig. 4.** Examples of images generated with original and perturbation-based Singla et al.\* pipelines.

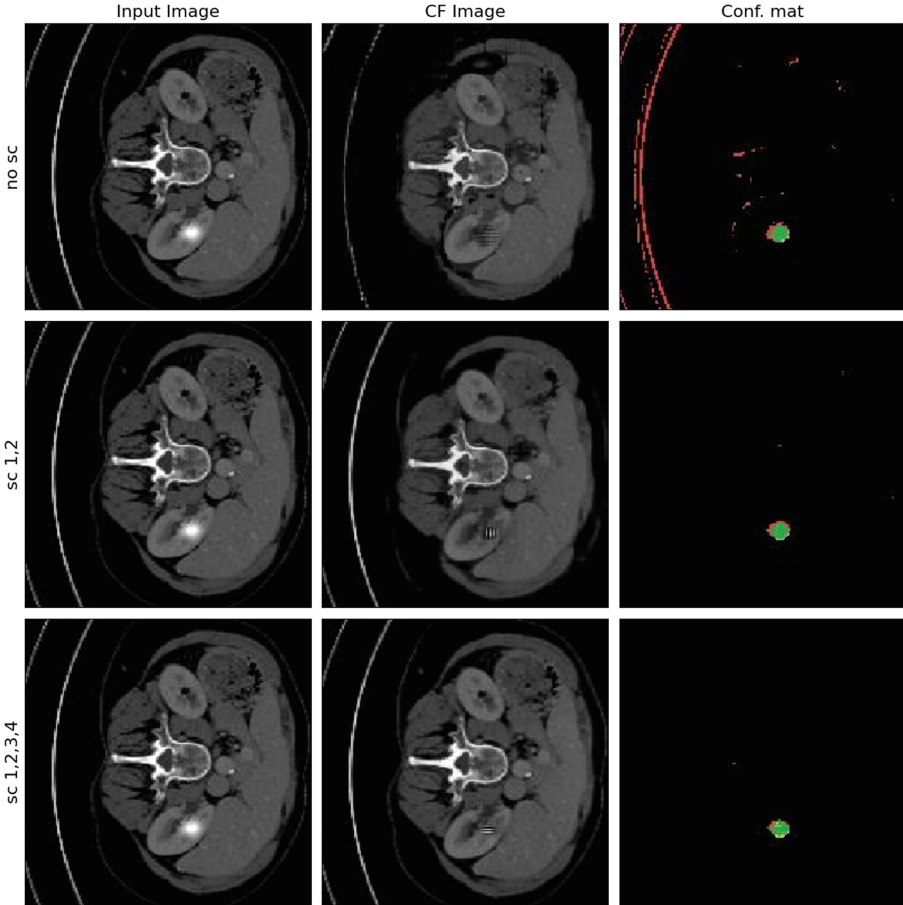
### A.3 Original vs Perturbation-Based Generator

Within this experiment, the counterfactual explainer of Singla et al. is taken to validate the significant improvement employing the perturbation-based generation in terms of FID and IoU scores. The perturbation-based image generation generates much higher fidelity images. Instead of reconstructing the whole input image from scratch, the decoder learns to output only the changes needed to flip classifier decision. Figure 4 gives qualitative evaluation of the generated images following the two approaches.

### A.4 Influence of Skip Connections on the Generated Images Quality

In this experiment, the baseline of Singla et al. employing perturbation-based counterfactual generation from the previous section is taken to showcase the importance of skip connections in the generator network to mitigate drastic distortions of the input images. During the down-sampling process of the encoder, the information loss is inevitable, so reconstructing the counterfactual images with minimum perturbations becomes a challenge. Therefore, the skip connections between different down-sampling and up-sampling layers are gradually injected to show the improvements in terms of FID and IoU scores. The perturbation-based image generation leveraging skip connections results in less





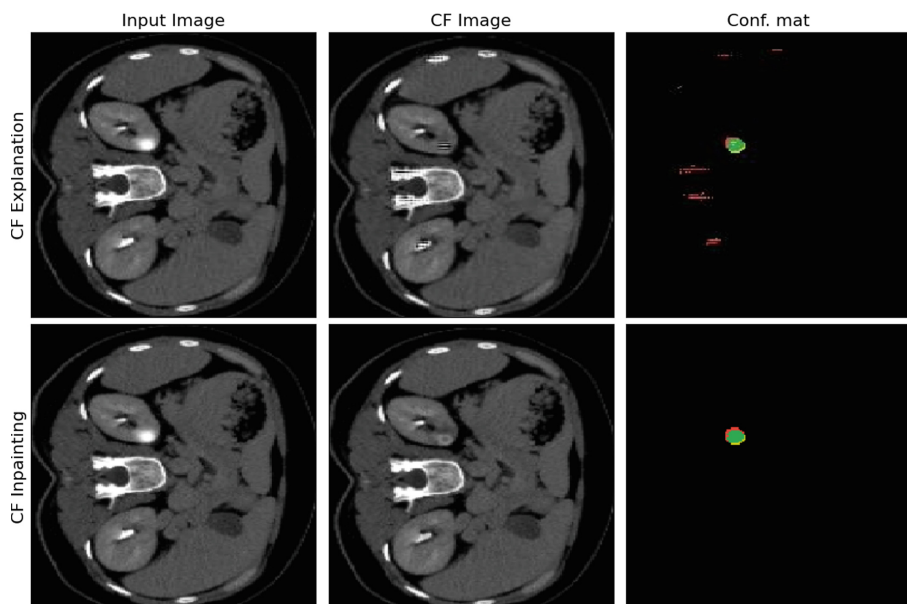
**Fig. 5.** Examples of images generated with and without skip-connections between encoder-decoder layers of the perturbation-based Singla et al.\* pipeline.

distorted images, hence, in lower FID score. Figure 5 gives qualitative evaluation of the generated images with and without adoption of skip-connections.

### A.5 Counterfactual Explanation vs Counterfactual Inpainting Segmentation Accuracy

This experiment proves that the proposed counterfactual inpainting pipeline outperforms the base counterfactual explanation approach. Both methods are trained and evaluated in terms of segmentation accuracy for the extracted weak segmentation labels from the counterfactual images.

The benefits of using the counterfactual inpainting are two-fold. First, it does not require segmentation masks for enforcing local consistency. Second, the IoU score is much higher due to the fact that the model is simplified to only



**Fig. 6.** Examples of images generated with perturbation-based Singla et al.\* method equipped with skip connections and with the proposed counterfactual inpainting approach.

either inpaint the anomaly or not to produce the segmentation mask. Figure 6 gives qualitative evaluation of the generated counterfactuals following the two approaches.

## References

1. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4981–4990 (2018)
2. Akula, A.R., et al.: CX-ToM: counterfactual explanations with theory-of-mind for enhancing human trust in image recognition models. *IScience* **25**(1), 103581 (2022)
3. Atad, M., et al.: CheXplaining in style: counterfactual explanations for chest X-rays using StyleGAN. arXiv preprint [arXiv:2207.07553](https://arxiv.org/abs/2207.07553) (2022)
4. Bischof, R., Scheidegger, F., Kraus, M.A., Malossi, A.C.I.: Counterfactual image generation for adversarially robust and interpretable classifiers (2023). [http://arxiv.org/abs/2310.00761](https://arxiv.org/abs/2310.00761)
5. Burton, R.J., Albur, M., Eberl, M., Cuff, S.M.: Using artificial intelligence to reduce diagnostic workload without compromising detection of urinary tract infections. *BMC Med. Inform. Decis. Mak.* **19**, 1–11 (2019)
6. Byrne, R.M.: Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: *IJCAI*, pp. 6276–6282 (2019)
7. Chaddad, A., Peng, J., Xu, J., Bouridane, A.: Survey of explainable AI techniques in healthcare. *Sensors* **23**(2), 634 (2023)

8. Chen, L., Wu, W., Fu, C., Han, X., Zhang, Y.: Weakly supervised semantic segmentation with boundary exploration. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020, Proceedings, Part XXVI 16*, pp. 347–362. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58574-7\\_21](https://doi.org/10.1007/978-3-030-58574-7_21)
9. Chen, Z., Tian, Z., Zhu, J., Li, C., Du, S.: C-CAM: causal CAM for weakly supervised semantic segmentation on medical image. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11676–11685 (2022)
10. Cui, H., Wei, D., Ma, K., Gu, S., Zheng, Y.: A unified framework for generalized low-shot medical image segmentation with scarce data. *IEEE Trans. Med. Imaging* **40**(10), 2656–2671 (2020)
11. Ghassemi, M., Oakden-Rayner, L., Beam, A.L.: The false hope of current approaches to explainable artificial intelligence in health care. *Lancet Digit. Health* **3**(11), e745–e750 (2021)
12. Gidde, P.S., et al.: Validation of expert system enhanced deep learning algorithm for automated screening for COVID-Pneumonia on chest X-rays. *Sci. Rep.* **11**(1), 23210 (2021)
13. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. *Data Min. Knowl. Disc.* 1–55 (2022)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2016). <http://arxiv.org/abs/1512.03385>
15. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium (2019). <http://arxiv.org/abs/1706.08500>
16. Javanmardi, M., Sajjadi, M., Liu, T., Tasdizen, T.: Unsupervised total variation loss for semi-supervised deep learning of semantic segmentation (2016). <http://arxiv.org/abs/1605.01368>
17. Jeanneret, G., Simon, L., Jurie, F.: Adversarial counterfactual visual explanations. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16425–16435. IEEE (2023). <https://doi.org/10.1109/CVPR52729.2023.01576>, <https://ieeexplore.ieee.org/document/10205255/>
18. Jiang, P.T., Zhang, C.B., Hou, Q., Cheng, M.M., Wei, Y.: LayerCAM: exploring hierarchical class activation maps for localization. *IEEE Trans. Image Process.* **30**, 5875–588 (2021). <https://doi.org/10.1109/TIP.2021.3089943>, <https://ieeexplore.ieee.org/document/9462463/>
19. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Comput. Surv.* **55**(5), 1–29 (2022)
20. Keil, F.C.: Explanation and understanding. *Annu. Rev. Psychol.* **57**, 227–254 (2006)
21. Kenny, E.M., Keane, M.T.: On generating plausible counterfactual and semi-factual explanations for deep learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11575–11585 (2021)
22. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014). <http://arxiv.org/abs/1412.6980>
23. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019)
24. Miller, T.: Contrastive explanation: a structural-model approach. *Knowl. Eng. Rev.* **36**, e14 (2021)
25. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks (2018). <http://arxiv.org/abs/1802.05957>

26. Musen, M.A., Middleton, B., Greenes, R.A.: Clinical decision-support systems. In: Shortliffe, E.H., Cimino, J.J. (eds.) *Biomedical Informatics: Computer Applications in Health Care and Biomedicine*, pp. 795–840. Springer, Cham (2021). [https://doi.org/10.1007/0-387-36278-9\\_20](https://doi.org/10.1007/0-387-36278-9_20)
27. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library (2019). <http://arxiv.org/abs/1912.01703>
28. Pearl, J.: The seven tools of causal inference, with reflections on machine learning. *Commun. ACM* **62**(3), 54–60 (2019)
29. Petsiuk, V., Das, A., Saenko, K.: RISE: randomized input sampling for explanation of black-box models (2018). <http://arxiv.org/abs/1806.07421>
30. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation (2015). <http://arxiv.org/abs/1505.04597>
31. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626 (2017)
32. Shen, W., et al.: A survey on label-efficient deep image segmentation: bridging the gap between weak supervision and dense prediction. *IEEE Trans. Pattern Anal. Mach. Intell.* (2023)
33. Singla, S., Eslami, M., Pollack, B., Wallace, S., Batmanghelich, K.: Explaining the black-box smoothly—a counterfactual approach (2021). <http://arxiv.org/abs/2101.04230>
34. Tajbakhsh, N., Jeyaseelan, L., Li, Q., Chiang, J.N., Wu, Z., Ding, X.: Embracing imperfect datasets: a review of deep learning solutions for medical image segmentation. *Med. Image Anal.* **63**, 101693 (2020)
35. Tan, M., Le, Q.V.: EfficientNetV2: smaller models and faster training (2021). <http://arxiv.org/abs/2104.00298>
36. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL & Tech.* **31**, 841 (2017)
37. Wang, H., et al.: Score-CAM: score-weighted visual explanations for convolutional neural networks (2019). <http://arxiv.org/abs/1910.01279>
38. Wasserthal, J., et al.: TotalSegmentator: robust segmentation of 104 anatomical structures in CT images. *Radiol. Artif. Intell.* **5**(5), e23002 (2023). <https://doi.org/10.1148/ryai.230024>, <http://arxiv.org/abs/2208.05868>
39. Zemni, M., Chen, M., Zablocki, E., Ben-Younes, H., Perez, P., Cord, M.: OCTET: object-aware counterfactual explanations. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15062–15071. IEEE (2023). <https://doi.org/10.1109/CVPR52729.2023.01446>, <https://ieeexplore.ieee.org/document/10205035/>



# Enhancing Counterfactual Explanation Search with Diffusion Distance and Directional Coherence

Marharyta Domnich<sup>(✉)</sup>  and Raul Vicente 

Institute of Computer Science, University of Tartu, Tartu, Estonia  
marharyta.domnich@ut.ee

**Abstract.** A pressing issue in the adoption of AI models is the increasing demand for more human-centric explanations of their predictions. To advance towards more human-centric explanations, understanding how humans produce and select explanations has been beneficial. In this work, inspired by insights of human cognition we propose and test the incorporation of two novel biases to enhance the search for effective counterfactual explanations. Central to our methodology is the application of diffusion distance, which emphasizes data connectivity and actionability in the search for feasible counterfactual explanations. In particular, diffusion distance effectively weights more those points that are more interconnected by numerous short-length paths. This approach brings closely connected points nearer to each other, identifying a feasible path between them. We also introduce a directional coherence term that allows the expression of a preference for the alignment between the joint and marginal directional changes in feature space to reach a counterfactual. This term enables the generation of counterfactual explanations that align with a set of marginal predictions based on expectations of how the outcome of the model varies by changing one feature at a time. We evaluate our method, named Coherent Directional Counterfactual Explainer (CoDiCE), and the impact of the two novel biases against existing methods such as DiCE, FACE, Prototypes, and Growing Spheres. Through a series of ablation experiments on both synthetic and real datasets with continuous and mixed-type features, we demonstrate the effectiveness of our method.

**Keywords:** Explainable AI · Counterfactual explanations · Diffusion distance · Feasibility · Directional coherence · Post-hoc explanations · Model agnostic explanations · Tabular data · Interpretable machine learning

## 1 Introduction

The demand for explainability of AI models has reached a new level of urgency with the rapid adoption of AI in different domains. Fueled by deep learning algorithms, applications in critical areas such as medical imaging [23, 33], law

[4], or finance [11] are seeking to automate and assist in decision-making [39]. However, the ability to explain why a certain prediction or decision was inferred is a fundamental prerequisite for responsible deployment in high-stake fields where accountability, transparency, and trust are valued [7, 32].

Developing explanations of AI models for humans confront us with the complexity of human explanatory processes [12, 19]. Moreover, humans excel at recognizing patterns from limited examples, even in uncertain situations, and can generalize concepts to address new problems [14]. While producing and evaluating explanations is natural to us, the underlying processes depend on complex mental models of the world which assist in inferring meaning from incomplete information based on previous experience [18]. Ideally, accessing a human’s world model would allow for producing explanations that fill the specific gaps in understanding, by contrasting input with prior knowledge of one’s mental model. Without access to an accurate “theory of mind” model [2] as part of one’s explainable model, another pathway to more human-centric explanations is to incorporate the preferences or biases that humans have reported in their judgment of explanations [34]. Following this perspective, our work incorporates two cognitive biases in a novel way: feasibility with diffusion distance and directional coherence as consistency of proposed changes with prediction direction.

Counterfactual explanations have emerged as a powerful tool in Explainable AI (XAI). Research in social sciences [22] highlights that human explanations are usually contrastive, presented in a format “X because of Y rather than Z”, where the “rather than Z” part is not always explicitly mentioned. A significant subtype of contrastive explanations are counterfactual explanations which answer questions such as “What would have happened if X had not occurred?”. In the context of machine learning model explainability, counterfactual explanations are used as local explanations [10, 16]. Given a particular instance and a trained model, they answer the question “What should the input have been in order to change the decision outcome?”. The power of counterfactual explanations lies in their ability to present alternative realities, aligning closely with our natural hypothetical reasoning [42].

Despite significant advancements in the development of counterfactual methods such as DiCE [25], FACE [27], DACE [15], CLUE [1], Guided Prototypes [36], CARE [30] and others, a complete integration of coherence within these approaches continues to be challenging. There is a consensus within the community that coherence is a crucial aspect distinguishing a basic explanation from an effective one. However, the multifaceted nature of coherence complicates its definition and application within AI explanations. Zemla et al. [42] highlight this complexity by distinguishing between internal and external coherence, where the internal coherence relates to the consistency among the components of an explanation, while external coherence concerns the explanation’s alignment with the user’s pre-existing knowledge. While existing approaches have attempted to model external coherence through user-defined constraints or by tracing the directional correlations within the feature space, the influence of the marginal direction of features concerning model predictions has not been explored. On

top of that, the use of diffusion distance ensures that counterfactual points are obtained through feasible paths of connected points respecting underlying geometry of the data manifold.

Our work contributes to this field by introducing CoDiCE, a framework designed to generate Directionally Coherent Counterfactual Explanations. This approach diverges from traditional methods by:

1. Replacing standard  $L_p$  distance measures used for proximity with diffusion distance, prioritizing connectivity and feasibility of transitions towards counterfactual scenarios.
2. Incorporating directional coherence as a constraint. This guides the selection of counterfactual explanations which joint changes (changing multiple features simultaneously) aligned with desired marginal changes (model of how outcome should vary if one changes one feature at a time).

## 2 Related Work

Our work builds upon the literature about counterfactual explanations, by focusing on their interpretation and integration of two cognitive biases: feasibility and coherence.

To account for **feasibility**, different notions of distances between the original point and counterfactual point were proposed. An early approach by Wachter et al. [38] posits Manhattan distance, adjusted by the inverse median absolute deviation, as a measure. Alternatives to this include the Euclidean (L2) or Gower distances, with some methods employing a blend of L1 and L2 distances weighted variably as an elastic net penalty. Despite accounting for scale variability, these metrics might neglect data density variations, potentially placing the counterfactual outside the data manifold. To mitigate this, CEM [6] propose training an auto-encoder on the desired class data, introducing a novel objective function term that penalizes the deviation of a counterfactual from its auto-encoded representation. Similarly, [36] expands upon this concept by identifying a prototype or class-representative instance through the autoencoder’s latent space. Furthermore, the DACE method [15] utilizes Mahalanobis distance, which accounts for data correlations and Local Outlier Factor that penalize points that are out of distribution.

Nonetheless, these methodologies do not sufficiently consider the transition path from the original instance to its counterfactual. In contrast, FACE [27] leverages k-NN to construct a connectivity graph, subsequently employing Dijkstra’s algorithm to identify the most feasible counterfactual pathway. This strategy not only affirms the feasibility of the counterfactual point but also of its pathway. A notable limitation of the FACE algorithm is its reliance on endogenous data points, which can complicate feasibility in sparse datasets or higher-dimensional spaces. Inspired by FACE, our approach utilizes diffusion distance, initially training a diffusion map to generate a transition graph within the diffusion space. This strategy accounts for data flow, enabling the projection of new

points onto diffusion coordinates without the constraints imposed by reliance on endogenous data.

An explanation is deemed **coherent** if it aligns with the recipient’s existing beliefs and knowledge, essentially reflecting the user’s mental model of the application domain. Echoing Zemla et al. [42] understanding that there are two types of coherence, Forster et al. [8] suggest that an explanation gains coherence from consistency with the user’s knowledge. Additionally, when the counterfactual scenario depicted is both realistic and typical of the alternative class distribution, they incorporate a loss term based on density estimate and external knowledge, adding pre-defined constraints into the search. Alternatively, CARE framework [30] interprets coherence as the consistency between the altered and unaltered features from the original to the counterfactual point by training model of correlations which guide the search towards more correlated features. Various methods interpret the coherence measure as ensuring the point remains within the same class distribution, modifying proximity with the Mahalanobis distance [5], or incorporating auto-encoders [6,36] to maintain the counterfactual within the data manifold.

Beyond data distribution conformity, coherence involves the transition from the original point to the plausible counterfactual point. Some methods, such as that proposed by [21], advocate for incorporating partial causal knowledge into the search process, suggesting learning feasibility constraints from user feedback. This approach, while emphasizing feasibility, indirectly fosters coherence by ensuring transitions adhere to plausible causal relationships. Additionally, Raman et al. [29] utilize a Bayesian approach to model the relationships between variables using conditional distributions. This allows for sampling counterfactuals from the posterior density while preserving domain-specific constraints. Primary methods underscore the significance of coherence by ensuring counterfactual explanations adhere to partial domain constraints. However, most approaches overlook the coherence of transitions from factual to counterfactual points with respect to the model’s output.

### 3 Incorporating Novel Biases in Counterfactual Search

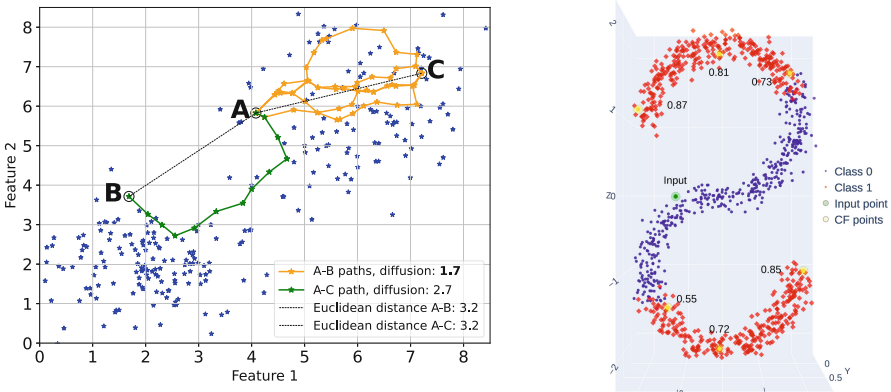
This section introduces our approach to refining the search for counterfactual explanations by incorporating two terms that account for a feasibility and external coherence biases in a novel way. The methodology aims to generate more intuitive and human-centric explanations. Finally, we detail the integration of these biases into the counterfactual objective function and optimization strategy.

#### 3.1 Using Diffusion Distance to Search for More Feasible Transitions

A strategy for generating meaningful counterfactual explanations is to develop methodology that emphasizes the feasibility, coherence, and actionability of possible explanations. We propose the utilization of diffusion distance as a metric to



assess the connectivity and potential actionability of counterfactual transitions. Unlike traditional distance metrics such as Euclidean (L2), Manhattan (L1), or shortest-path distance on the data manifold (as used in FACE or Isomap [35]), diffusion distance offers a nuanced understanding of the data manifold by prioritizing transitions between data points that are interconnected through numerous, short paths. This approach brings points that are highly connected by numerous short paths into closer proximity, and hence highlighting points for which numerous short routes exist to transition from one point to the other while being on the data manifold. The concept of diffusion distance and its role in detecting counterfactual points that are more “accessible” from the original instance (in the sense of the existence of numerous short distance routes between the points) is illustrated in Fig. 1.



**Fig. 1.** Illustration of the concept of diffusion distance and its use for counterfactual search. Left panel: Points connected by numerous short distance paths (A-C) exhibit a shorter diffusion distance than pairs of points which connections pass through a bottleneck or low density region (A-B). Note that evaluated by Euclidean distance the pairwise distance A-C and A-B would be exactly the same. Right panel: 3D S-shaped synthetic dataset with two classes. The input point belongs to class 0, the diffusion distances between such point and 6 counterfactual candidates are displayed.

The formal definition of diffusion distance between two points  $x$  and  $y$  at time  $t$  is given by:

$$D_{\text{diff}}(x, y, t)^2 = \sum_z \frac{(p_t(x|z) - p_t(y|z))^2}{\phi_0(z)}, \tag{1}$$

where  $p_t(x|z)$  represents the probability of transitioning from point  $z$  to  $x$  in  $t$  steps following a diffusion process (random walk on the graph), and  $\phi_0(z)$  is the stationary distribution of the diffusion process at point  $z$ . This formula highlights the diffusion distance’s capacity to account for the data’s intrinsic geometry through probabilistic transitions.

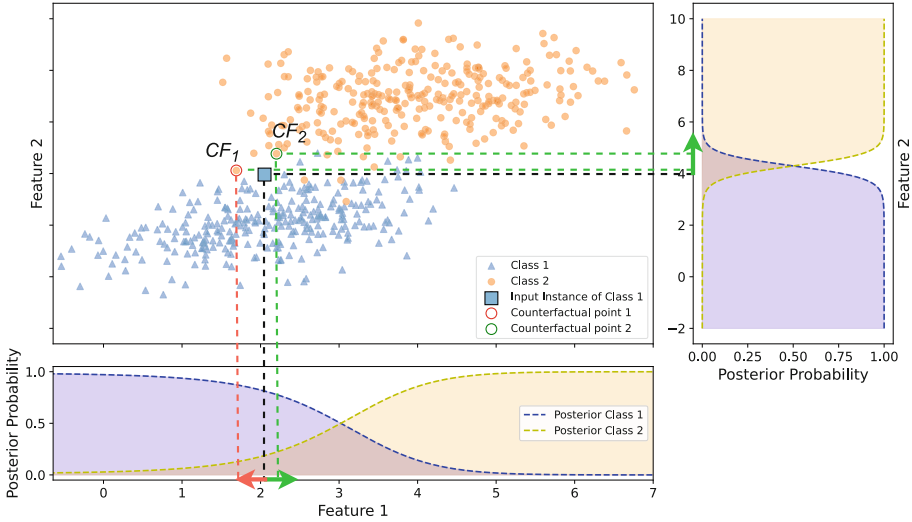
**Employing Diffusion Distance with Self-Tuning Kernel for Local Scaling.** Our implementation incorporates a self-tuning kernel within the diffusion distance framework [41], adjusting dynamically to the variance in the data. This adjustment ensures the robustness across different data domains and reduce the number of parameters needed for fitting, as the only parameter we ask is number of nearest neighbors, which is intuitive to set up.

Diffusion distance is rooted in the concept of diffusion processes on graphs, encapsulating the connectivity and density of data points within a dataset. This metric quantifies the ease of traversing the data landscape from one point to another, factoring in the multitude of potential paths and their associated probabilities. The key advantage of diffusion distance over the shortest-path distance on the manifold as used by FACE or Isomap is its robustness to noise, which is particularly valuable in high-dimensional settings where data sparsity and noise are prevalent challenges. By facilitating the exploration of multiple paths, diffusion distance looks for counterfactuals through a sequence of realistic transitions which favour to in-distribution feasible point.

### 3.2 Directional Coherence

Directional Coherence formulates a bias designed to maintain consistency between the marginal (one feature at a time) and joint (multiple features simultaneously) directions in feature space needed to flip the outcome of the model’s prediction. This coherence facilitates the generation of counterfactual explanations that not only adhere to the model’s predictions for individual feature alterations but also align with the overall direction of change necessary to shift to a desired counterfactual state. Such term can be use to tune the importance of aligning counterfactual paths with intuitive human reasoning about a set of causal expectations when changes are produced in marginal directions (changing one feature at a time).

To illustrate this concept, consider the scenario of applying for a home loan, where it is intuitively expected that an increase in income for either the applicant or co-applicant would improve the chances of loan approval. We would be shocked to learn that a bank advises increasing the applicant’s income, but decrease a co-applicant income. This counterintuitive recommendation could arise from the specific nature of the data distribution, reflecting scenarios where other people with these factual scenario in the past got loan approval. We argue that although observing such point is possible, it would represent an undesirable direction for counterfactual explanation. Figure 2 illustrates this conceptual situation. An input point highlighted with rectangular shape and belonging to Class 1, has two counterfactual candidates  $CF_1$  and  $CF_2$ , which are associated with the desired Class 2. The data spread indicates that increases in Feature 1 and Feature 2 are correlated with a higher likelihood of predicting Class 2. Consequently,  $CF_2$  point is directionally coherent, as the joint increase in these features aligns with the marginal direction of probability of Class 2. On the other hand,  $CF_1$  is directionally incoherent, since the change in Feature 1 leads to decrease in the posterior probability of predicting Class 2.



**Fig. 2.** Illustration of Directional Coherence. The input point belongs to Class 1. Given counterfactual candidates  $CF_1$  and  $CF_2$  at equal distance from the original input point, we deem  $CF_1$  as incoherent with respect to the expected effect of changing Feature 1. Intuitively,  $CF_1$  suggests to decrease Feature 1, while the effect of increasing either Feature 1 or Feature 2 is to increase the posterior probability of predicting Class 2. For the other counterfactual ( $CF_2$ ), there is an agreement between the direction of marginal changes (changing one feature at a time) and the joint direction of changes resulting in a more coherent counterfactual point.

Thus, directional coherence is predicated on the intuition that certain feature alterations should consistently lead to predictable changes in the model’s output. This is especially crucial in complex domains where interpretability and actionability of counterfactual explanations are required. By assessing the directional impact of each feature independently, we can ascertain the collective influence exerted by all features on the transition towards the desired counterfactual state.

Mathematically, we formulate directional coherence as a term that quantifies the preference for alignment between joint and marginal directional changes in the feature space necessary to achieve a counterfactual outcome. For clarity, we introduce here the case of a classifier. The corresponding formulation for a regression model is an straightforward extension.

Let us denote  $f : \mathcal{X} \rightarrow \mathcal{Y}$  the classification model. Given an original instance as a vector  $x = (x_1, x_2, \dots, x_n) \in \mathcal{X} \subseteq \mathbb{R}^n$  and a counterfactual instance  $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in \mathcal{X} \subseteq \mathbb{R}^n$  that brings the desired outcome. The goal is to evaluate the coherence of the transition from  $x$  to  $x^*$  in achieving a specified outcome label  $y$  with a set of expected marginal transitions  $\{x_i \rightarrow x'_i \mid f(y|x_1, x_2, \dots, x'_i, \dots, x_n) \geq f(y|x_1, x_2, \dots, x_i, \dots, x_n), 1 \leq i \leq n\}$ . Notably, while marginal transitions are typically derived from the model, user-

specified marginal transitions, when provided, take precedence over those suggested by the model.

Then, the Directional Coherence score counts the excess of features which have aligned marginal (') and joint (\*) directions to increase the model's prediction probability towards the desired outcome  $y$ :

$$dcoherence = \frac{1}{n} \sum_{i=1}^n \text{sgn}((x_i^* - x_i)(x'_i - x_i)) . \quad (2)$$

The information about incoherent features can be leveraged to introduce new constraints or refine the model. Additionally, repeated patterns of incoherence could indicate areas where the model's sensitivity to changes in feature values needs further investigation or adjustment. The implementation of calculating directional coherence is illustrated in Appendix A Algorithm 2.

### 3.3 Bringing Feasibility and Directional Coherence into Counterfactual Objective Function

Next we formalize the incorporation of the feasibility and coherence biases within the objective function for the counterfactual search. We denote by  $f$  a trained predictor function that maps the input space  $\mathcal{X}$  to the output space  $\mathcal{Y}$ , i.e.,  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . Given a factual point or the original input point  $x = (x_1, x_2, \dots, x_n) \in \mathcal{X} \subseteq \mathbb{R}^n$ , our objective is to identify a counterfactual point  $c^* = (c_1^*, c_2^*, \dots, c_n^*) \in \mathcal{X} \subseteq \mathbb{R}^n$  that yields the desired label  $y$  while minimising a weighted sum of diffusion distance, sparsity, and directed coherence penalties. The optimization problem is defined as follows:

$$c = \arg \min_{x^*} \left( \text{loss}(f(x^*), y) + \lambda_1 \text{diffusion\_dist}(x^*, x) + \lambda_2 \text{sparsity}(x^*, x) + \lambda_3 (1 - \text{dcoherence}(x^*, x)) \right), \quad (3)$$

where:

- $\text{loss}(f(x^*), y)$  is the loss term that checks if the counterfactual outcome is equal to the desired outcome, we utilize commonly used loss measures hinge-loss [9] for classification and mean squared error [28] for regression.
- $\text{diffusion\_dist}(x^*, x)$  quantifies the diffusion distance between the original point  $x$  and the counterfactual point  $x^*$  (see formula (1)).
- $\text{sparsity}(x^*, x)$  computes the  $l_0$  distance to count the number of features that have been modified.
- $\text{dcoherence}(x^*, x)$  assesses the directional coherence by aligning the joint direction of the counterfactual point with its marginals. Since we are interested in minimizing objective function, we take the penalty measure  $(1 - \text{dcoherence})$ .

The terms are weighted by hyperparameters  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ , which can be adjusted or set to 0 if a particular constraint is not applicable.

There is a natural division between continuous and categorical features which have different properties. For **categorical features**, the notion of diffusion distance does not apply, and measuring distance is challenging, necessitating a context-specific approach. We employ the  $L_0$  norm to detect category changes, which could be further refined based on the number of categories and the difficulty of changing each feature.:

$$dist_{cat}(x, x^*) = \frac{1}{m} \sum_{j=1}^m I(x_j^* \neq x_j) \quad (4)$$

It is notable that the distance for categorical features overlap with the sparsity term, we intend to keep it that way, as the user might want to tune down sparsity weight term without accounting for its effect on the search of counterfactual explanation. Therefore, the objective function for counterfactual search for mix-type data is the following:

$$c = \arg \min_{x^*} \left( \text{loss}(f(x^*), y) + \lambda_1 \text{diffusion\_dist}_{cont}(x^*, x) + \lambda_1 dist_{cat}(x^*, x) + \lambda_2 \text{sparsity}(x^*, x) + \lambda_3 (1 - \text{dcoherence}(x^*, x)) \right) \quad (5)$$

We optimize this expression with genetic algorithm, similarly to DiCE 'genetic' optimization [25] with the details of implementation described in Appendix A Algorithm 3.

### 3.4 Evaluation Metrics

To assess the quality and compare the performance of the generated counterfactual explanations, we utilize commonly accepted metrics (Validity, Weighted L1 Continuous, Categorical L0) as well as our novel metrics inspired by our integrated biases: Diffusion Distance and Directional Coherence. For their detailed description see Appendix A.

## 4 Experiments

We conducted experiments on two synthetically generated datasets to visualize the effect of diffusion distance. After that, we applied our framework for commonly used in counterfactual explanation literature classification datasets, such as Diabetes, Breast Cancer [40] and mixed-type features Adult [3] and German [13]. Additionally, we used energy consumption prediction dataset [31] for testing regression settings as for this case we also had access to domain experts for feedback.

## 4.1 Synthetic Datasets

To illustrate the effect of diffusion distance we generated two synthetic datasets: an S surface and a Swiss roll, utilizing the `sklearn.datasets` module for their creation. We thresholded the parameter of the generated shapes, dividing each dataset into two distinct classes. The dataset were partitioned into training and test subsets with a 80/20 split. We trained a Support Vector Machine (SVM) classifier with radial basis kernel on these datasets resulting in 97% accuracy on the test set for a S surface and 99% accuracy for a Swiss roll.

## 4.2 Datasets with Continuous Features

**The Diabetes dataset**, sourced from the UCI Machine Learning Repository, consists of diagnostic measurements for predicting the onset of diabetes within a Pima Indian population. It features 768 instances, each with 8 numeric predictor variables such as the number of pregnancies, plasma glucose concentration, blood pressure, and body mass index, among others. The outcome variable is binary, indicating the presence or absence of diabetes.

**The Breast Cancer Wisconsin (Diagnostic) dataset**, also from the UCI Machine Learning Repository, comprises features computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. It includes 569 instances with 30 continuous features, describing characteristics of the cell nuclei present in the image. The prediction task is binary, distinguishing between malignant and benign tumors.

**Energy consumption dataset** is a time series data of residential building energy consumption contains of 473 instances with indoor temperature, outdoor temperature, active electricity historical observations. The prediction feature is future electricity consumption. The counterfactual question is what should have been the input to decrease energy consumption. Details of the dataset are available here [31].

## 4.3 Classification Datasets with Mix-Type Features

**The Adult Income dataset**, often referred to as the “Census Income” dataset, contains demographic information from the 1994 Census database. It consists of 48842 instances and 14 features (6 numerical and 8 categorical), including age, work class, education, marital status, occupation, and hours per week, among others. The dataset’s target variable is binary, predicting whether an individual’s income exceeds \$50K/year.

**The German Credit** dataset comprises financial and demographic data for 1,000 loan applicants. Each instance is described by 20 attributes, a mix of continuous and categorical variables, such as credit history, savings account balance, employment duration, and purpose of the loan. The objective is to classify individuals into good or bad credit risks.

**Compas** dataset encompasses information related to defendants involved in the criminal justice system. It consists of 7214 instances with 11 features (4

continuous and 7 categorical), such as age, race, gender, criminal history, charge degree, etc. The target variable is the risk of recidivism (high or low). The dataset has been a point of analysis in discussions on the fairness, bias, and transparency of predictive algorithms in legal settings.

#### 4.4 Benchmarking with Other Frameworks

We compared our results with various counterfactual frameworks, focusing on validity, diffusion distance, weighted L1 distance, directional coherence, and L0 categorical for mixed-type data. The compared methods are:

**Diverse Counterfactual Explanations (DiCE)** [25], the most popular method in literature, generates diverse counterfactual instances using weighted L1 for proximity on continuous features and L0 for categorical. Acknowledging the trade-off between diversity and proximity, we generated a single, optimal counterfactual explanation using DiCE’s original implementation.

**Feasible and Actionable Counterfactual Explanations (FACE)** [27] considers feasible paths using the Dijkstra algorithm and produces counterfactual points from existing training data. We ran the FACE algorithm using the CARLA benchmark library [26] for algorithm comparison.

**Guided Prototypes (Prototypes)** [36] integrates the notion of coherence by training an auto-encoder to select a prototype instance, ensuring the typicality of the point. Prototypes use a combination of weighted L1 and L2 as an elastic net regularizer for proximity.

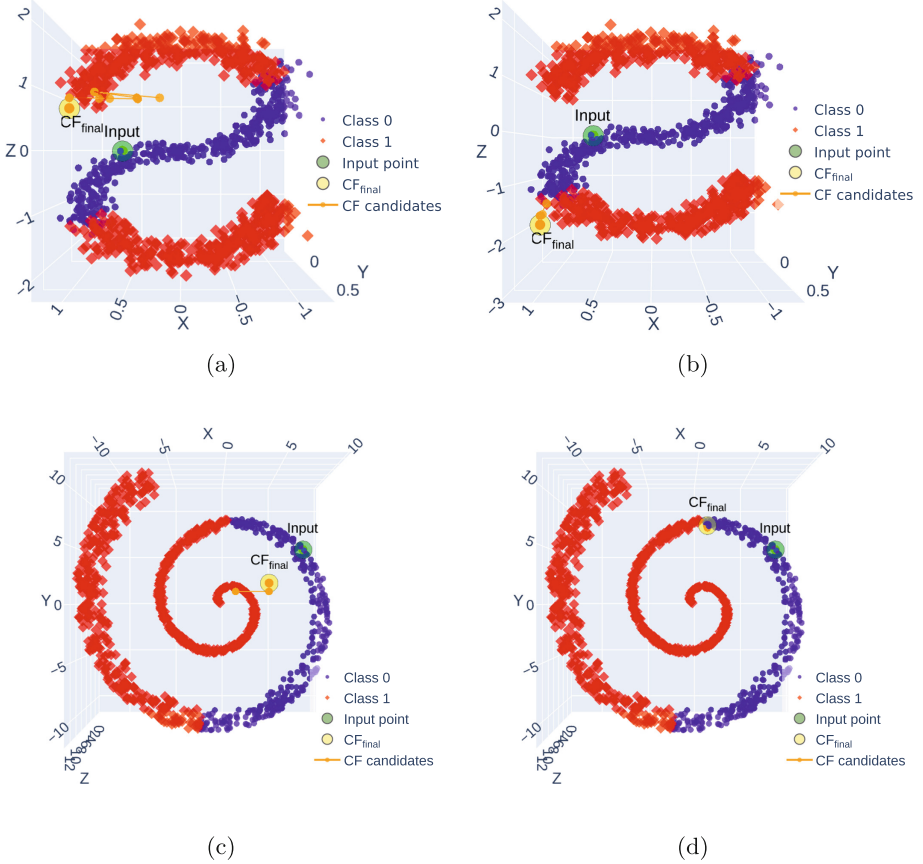
**Growing Spheres (GS)** [20] employs weighted  $L_2$  and  $L_0$  norms between counterfactuals and candidates. It is often reported to generate the closest counterfactuals in terms of proximity. Importantly, GS does not support mixed-feature datasets.

We ran Prototypes and GS methods from [24], a library created for benchmarking counterfactual methods.

## 5 Results

### 5.1 Diffusion Distance and Directional Coherence on Synthetic and Diabetes Datasets

We applied the CoDiCE framework on synthetic datasets to illustrate the effect of diffusion distance on geometrically structured data. Figure 3 shows the “S” surface and Swiss roll shape partition in two classes. We took original Input point from class 0 and searched for counterfactual point minimising a  $L_1$  distance (Fig. 3 (a), (c)), and diffusion distance (Fig. 3 (b), (d)). As indicated in the figure, the counterfactual obtained using  $L_1$  distance crosses a low-density data region and ignores geometrical structure of the data, while counterfactuals found using diffusion distance respects the connectivity of the data manifold.

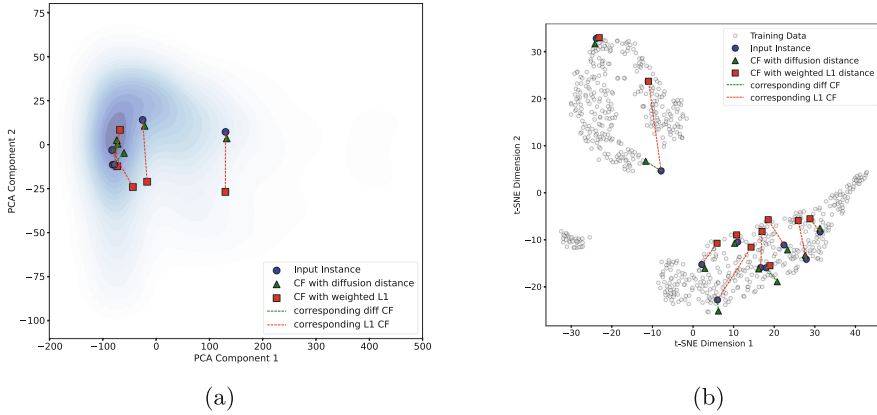


**Fig. 3.** Counterfactual search on synthetic datasets. The counterfactuals obtained for the S surface and Swiss roll illustrate the role of diffusion distance (panels b and d) to take into account the connectivity of the data manifold as opposed to  $L_1$  (panels a and b) or more generally other  $L_p$  distances.

To illustrate the effect of diffusion distance on the counterfactual search for high-dimensional data, we also applied dimensionality reduction techniques such as PCA and t-SNE. For the trained Logistic Regression model and diabetes test set we searched for counterfactual explanations with weighted  $L_1$  ( $L_1$  norm with the inverse of the median absolute deviation that is commonly used in counterfactual methods as it originates from Wachter et al. [38]) as well as diffusion distance. We evaluated the search using 5 random samples as original inputs and visualize them together with their counterfactuals on PCA and 10 random samples visualized on t-SNE coordinates (Fig. 4). The counterfactuals obtained using diffusion distance point are significantly closer in both PCA and t-SNE coordinates than the corresponding obtained with  $L_1$  distance. This type



of visualizations further suggest that diffusion distances are useful to capture the connectivity and clustering structure of the dataset.



**Fig. 4.** Counterfactual search on the Diabetes dataset projected onto PCA coordinates (a) and t-SNE (b). The plot has triplets of points connected by dotted lines Input instance (blue circle) their respective Counterfactual point obtained with diffusion distance (green triangle) and Counterfactual point obtained with weighted  $L_1$  distance (red square). (Color figure online)

## 5.2 Comparison of CoDiCE with Other Counterfactual Methods on Various Datasets

Beyond the visual illustrations, we set to compare the effect of diffusion distance and directional coherence against various popular counterfactual frameworks. In particular, we evaluated the following statistics: validity, diffusion distance, weighted  $L_1$  distance for continuous features,  $L_0$  distance for categorical features and directional coherence.

To make a fair comparison we adjusted the preprocessing of the data and our choice of model to be compatible with other frameworks. For all datasets we apply OneHotEncoding for categorical features and standardization for continuous features. After that we split the data into 80/20 train test split to train Logistic Regression model. In our experiments model is treated as a black-box and can be replaced with any other model. Finally, we generated counterfactual explanations for the first 100 instances from the same test set across all methods for every dataset.

To measure the effect of directional coherence isolated from that of the diffusion distance, we implemented two versions of CoDiCE framework:  $\text{CoDiCE}_{\text{diff}}$  that uses diffusion distance for proximity and directional coherence  $\text{CoDiCE}_{L_1}$  that uses a weighted  $L_1$  proximity measure similarly to DiCE and directional

**Table 1.** Evaluation metrics comparison across different frameworks for datasets with continuous features. For distance and coherence metrics we report average and standard deviation over 100 samples. Validity is expressed in %.

Dataset	Metric	Validity $\uparrow$	Diffusion $\downarrow$	L1 continuous $\downarrow$	DCoherence $\uparrow$
Diabetes	CoDiCE <sub>diff</sub>	<b>100%</b>	<b>0.38 <math>\pm</math> 0.22</b>	1.11 $\pm$ 0.53	0.64 $\pm$ 0.15
	CoDiCE <sub>L1</sub>	<b>100%</b>	0.72 $\pm$ 0.45	<b>0.29 <math>\pm</math> 0.16</b>	0.76 $\pm$ 0.16
	DiCE	54%	1.62 $\pm$ 0.73	1.10 $\pm$ 0.34	0.68 $\pm$ 0.13
	FACE	70%	1.64 $\pm$ 0.67	1.08 $\pm$ 0.36	0.72 $\pm$ 0.14
	Prototypes	26%	2.18 $\pm$ 0.88	2.12 $\pm$ 0.61	<b>0.84 <math>\pm</math> 0.07</b>
	GS	<b>100%</b>	0.67 $\pm$ 0.31	0.39 $\pm$ 0.19	0.57 $\pm$ 0.12
Breast Cancer	CoDiCE <sub>diff</sub>	60%	2.87 $\pm$ 1.21	2.18 $\pm$ 0.39	<b>0.78 <math>\pm</math> 0.10</b>
	CoDiCE <sub>L1</sub>	60%	2.62 $\pm$ 1.07	1.22 $\pm$ 0.37	0.67 $\pm$ 0.09
	DiCE	46%	<b>2.13 <math>\pm</math> 0.87</b>	<b>0.97 <math>\pm</math> 0.28</b>	0.72 $\pm$ 0.08
	FACE	63%	2.91 $\pm$ 1.08	0.98 $\pm$ 0.38	0.74 $\pm$ 0.10
	Prototypes	31%	4.72 $\pm$ 0.45	2.22 $\pm$ 0.37	<b>0.79 <math>\pm</math> 0.01</b>
	GS	<b>100%</b>	2.25 $\pm$ 1.31	0.47 $\pm$ 0.28	0.58 $\pm$ 0.08

coherence. Table 1 shows the comparison of methods for Diabetes and Breast Cancer datasets, both of which represent data with continuous features.

We report that for the Diabetes dataset only CoDiCE<sub>diff</sub>, CoDiCE<sub>L1</sub>, and GS obtained 100% Validity, and hence all of the proposed counterfactual points for these methods did actually flip the outcome. It is important to note that the metrics are computed for valid points (points for which the desired outcome change is realized). As expected CoDiCE<sub>diff</sub> has the lowest diffusion distance ( $0.38 \pm 0.22$ ) and both CoDiCE<sub>L1</sub> has the lowest weighted L1 distance. FACE showed to be more efficient method than Prototypes having higher Validity, which resulted in the highest directional coherence score, given that it was computed on a very few valid counterfactuals ( $0.84 \pm 0.07$ ). Surprisingly, GS is comparable with CoDiCE<sub>L1</sub> in terms of proximity scores, however directional coherence is the lowest.

For Breast Cancer dataset we found that it is quite difficult to find both directionally coherent and feasible points. The validity of all methods was low except for GS, which produces the least coherent counterfactuals. The highest performing methods in terms of directional coherence were Prototypes ( $0.78 \pm 0.10$ ) and CoDiCE<sub>diff</sub> ( $0.79 \pm 0.01$ ). However, Prototypes' validity only reached 31%, and hence it discovered less than half the number of valid counterfactuals than CoDiCE methods found (which reached 60% validity). Overall, as we will explicitly show in ablation experiments directional coherence is in a trade-off with diffusion distance (see Fig. 5) which resulted in the algorithm not having lowest diffusion distance for this setting (specific weights for each term in the cost function).

**Table 2.** Evaluation metrics comparison across different frameworks for dataset with mixed-type of features (continuous + categorical). For distance and coherence metrics we report average and standard deviation over 100 samples. Validity is expressed in %.

Dataset	Metric	Validity $\uparrow$	Diffusion $\downarrow$	$L_1$ cont $\downarrow$	$L_0$ cat $\downarrow$	DCoherence $\uparrow$
Adult	CoDiCE <sub>diff</sub>	<b>98%</b>	<b>0.001 <math>\pm</math> 0.004</b>	3.9 $\pm$ 1.4	0.2 $\pm$ 0.1	0.84 $\pm$ 0.09
	CoDiCE <sub>L<sub>1</sub></sub>	92%	0.005 $\pm$ 0.013	1.1 $\pm$ 0.6	0.4 $\pm$ 0.1	0.82 $\pm$ 0.08
	DiCE	78%	0.005 $\pm$ 0.011	1.2 $\pm$ 0.6	<b>0.1 <math>\pm</math> 0.1</b>	<b>0.98 <math>\pm</math> 0.02</b>
	FACE	82%	0.007 $\pm$ 0.013	<b>0.7 <math>\pm</math> 0.3</b>	0.5 $\pm$ 0.2	0.81 $\pm$ 0.11
	Prototypes	19%	0.013 $\pm$ 0.012	1.6 $\pm$ 0.4	0.7 $\pm$ 0.1	0.85 $\pm$ 0.05
German	CoDiCE <sub>diff</sub>	<b>100%</b>	<b>4.3 <math>\pm</math> 3.4</b>	1.2 $\pm$ 0.5	<b>0.1 <math>\pm</math> 0.1</b>	0.93 $\pm$ 0.04
	CoDiCE <sub>L<sub>1</sub></sub>	<b>100%</b>	6.4 $\pm$ 3.8	<b>0.7 <math>\pm</math> 0.4</b>	<b>0.1 <math>\pm</math> 0.1</b>	<b>0.94 <math>\pm</math> 0.04</b>
	DiCE	49%	8.2 $\pm$ 3.2	1.2 $\pm$ 0.4	0.5 $\pm$ 0.1	0.78 $\pm$ 0.07
	FACE	63%	7.6 $\pm$ 2.4	1.0 $\pm$ 0.4	0.5 $\pm$ 0.1	0.79 $\pm$ 0.07
	Prototypes	34%	10.1 $\pm$ 3.4	1.1 $\pm$ 0.6	0.7 $\pm$ 0.1	0.73 $\pm$ 0.05
Compas	CoDiCE <sub>diff</sub>	<b>100%</b>	0.03 $\pm$ 0.05	5.2 $\pm$ 1.9	<b>0</b>	0.92 $\pm$ 0.08
	CoDiCE <sub>L<sub>1</sub></sub>	<b>100%</b>	0.03 $\pm$ 0.05	<b>0.8 <math>\pm</math> 0.4</b>	<b>0</b>	<b>0.93 <math>\pm</math> 0.07</b>
	DiCE	49%	0.03 $\pm$ 0.04	1.0 $\pm$ 0.6	0.4 $\pm$ 0.2	0.83 $\pm$ 0.11
	FACE	18%	0.04 $\pm$ 0.06	1.2 $\pm$ 0.6	0.5 $\pm$ 0.2	0.79 $\pm$ 0.11
	Prototypes	18%	<b>0.01 <math>\pm</math> 0.01</b>	1.3 $\pm$ 0.7	0.6 $\pm$ 0.1	0.76 $\pm$ 0.09

Table 2 shows the comparison of counterfactual methods on mixed-type datasets, such as Adult, German, Compas. Similarly as for the continuous case, counterfactual instances were generate for the first 100 instances of the fixed test set for the same Logistic Regression model.

Notably, for all datasets CoDiCE<sub>diff</sub> and CoDiCE<sub>L<sub>1</sub></sub> results in the highest validity. For both Adult and German dataset CoDiCE<sub>diff</sub> has the lowest diffusion distance. However, we note that for Adult and Compas datasets the standard deviation of the diffusion distance is usually larger than its average. This indicates that diffusion distance distribution is significantly skewed for these datasets and the standard deviation might not fully capture the nature of its variability. Also for Compas the low validity for DiCE, FACE, and Prototypes resulted in the metrics for such methods to be evaluated from relatively few samples.

Furthermore, mixed-type data brings the question of the weighting between the proximity metrics for continuous and categorical features. Depending on such weighting the method can results in cases in which a higher preference for changing categorical features results in a smaller difference in continuous features needed to flip the prediction. Hence, for mixed-type of data the pressure to minimize diffusion distance (which is exclusively computed for continuous features) is also a function of the weighting between continuous and categorical features proximity biases.

As an additional metric, the running time for searching counterfactual explanations using these methods was measured. We report the running time for the most challenging dataset in terms of diffusion distance: Breast Cancer, which has 30 continuous features. The average speed of finding one counterfactual point on such a challenging dataset averaged 27 s for CoDiCE<sub>diff</sub> and 26 s for CoDiCE<sub>L<sub>1</sub></sub>. The running times are comparable since the diffusion distance is calculated for the entire dataset once. It is important to note that no code optimization efforts have been made thus far. For other methods, FACE requires, on average, 13 s to identify a counterfactual point, GS took 0.44 s, and the highly optimized DiCE took 0.11 s per counterfactual point. Although not the focus of our current investigation, we believe that code optimization efforts could significantly enhance running times, an aspect we aim to explore in forthcoming studies.

**Table 3.** Evaluation metrics comparison across different frameworks for the Energy consumption dataset (regression problem). For distance and coherence metrics we report average and standard deviation over 100 samples. Validity is expressed in %.

Dataset	Metric	Validity $\uparrow$	Diffusion $\downarrow$	$L_1$ continuous $\downarrow$	DCoherence $\uparrow$
Energy	CoDiCE <sub>diff</sub>	100%	0.005 $\pm$ 0.03	0.52 $\pm$ 0.33	<b>0.67 <math>\pm</math> 0.04</b>
	CoDiCE <sub>L<sub>1</sub></sub>	100%	<b>0.003 <math>\pm</math> 0.02</b>	<b>0.41 <math>\pm</math> 0.26</b>	0.63 $\pm$ 0.11
	DiCE	100%	1.64 $\pm$ 2.21	0.51 $\pm$ 0.47	0.62 $\pm$ 0.11

The energy the consumption dataset was fitted with a genetic programming model (symbolic tree) trained with GP-GOMEA library [37]. It is a regression problem, which similarly to the logistic regression models for previous datasets, we treat as a black-box for counterfactual search. The target of interest was decreasing energy consumption by 5%. We ran counterfactual search for 20 test instances, where for every instance we targeted an energy consumption decrease in the range [10%, 5%] of that predicted at the original input. Among methods used in our comparison only DiCE supports a regression problem. The comparison with DiCE for this model is shown in Table 3.

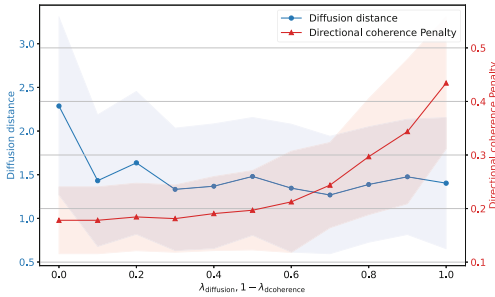
### 5.3 Ablation Experiments

To systematically explore the influence of each term within the CoDiCE objective function (5), we conducted ablation studies on the Diabetes dataset reported in Table 4. The ablations consist of setting the weights of all terms except the one under investigation to zero. Specifically,  $\lambda_1$  corresponds to the weight assigned to the diffusion distance,  $\lambda_2$  to the weight assigned to sparsity, and  $\lambda_3$  to the weight assigned to directional coherence.

**Table 4.** Evaluation metrics under various ablations of diffusion, sparsity, and directional coherence terms for the Diabetes dataset.

Dataset	Inactive terms	Validity	Diffusion	$L_1$ continuous	Sparsity	DCoherence
Diabetes	$\lambda_2, \lambda_3$	100%	<b><math>1.49 \pm 0.55</math></b>	<b><math>1.01 \pm 0.48</math></b>	1	$0.56 \pm 0.13$
	$\lambda_1, \lambda_3$	100%	$2.38 \pm 0.86$	$1.92 \pm 0.43$	<b>0.85</b>	$0.56 \pm 0.13$
	$\lambda_1, \lambda_2$	100%	$2.19 \pm 0.98$	$1.73 \pm 0.44$	1	<b><math>0.82 \pm 0.06</math></b>

Figure 5 also illustrates the outcomes of trade-off experiments by evaluating the impact of varying the weights assigned to diffusion distance and directional coherence. Given a fixed sparsity weight  $\lambda_2 = 0.5$ , we varied the weight of diffusion distance  $\lambda_1$  while setting the directional coherence weight to be  $\lambda_3 = 1 - \lambda_1$ . As the weight on diffusion distance increases from 0 to 1, effectively reducing the emphasis on directional coherence, we observe a notable trade-off.

**Fig. 5.** Trade-off between diffusion distance and directional coherence penalty is explored as the diffusion weight is increased.

## 6 Discussion

This study proposes novel incorporations into an objective function of counterfactual search through the integration of diffusion distance and directional coherence. These are aimed to enhance the feasibility and alignment with intuitive expectations of a candidate counterfactual point. The introduction of diffusion distance as a component of the objective function ensures that the search for counterfactual points takes into account the underlying geometry of the data manifold. Additionally, diffusion distance is robust to noise which is particularly valuable in high-dimensional settings. The directional coherence term promotes that the counterfactual suggestion aligns with internal constraints or intuitions

about how individual feature changes the model’s outcome. After systematic experiments with various datasets (with both continuous and mixed-type data) we found that these terms work as intended and overall promote counterfactuals with shorter diffusion separation (being well connected or same cluster as the original input) and higher coherence (respecting marginal constraints). However, it is important to note that the inclusion of several terms simultaneously can result in a trade-off between both aims. The ablation experiments demonstrated notable trade-off between these two components, emphasizing the importance of a balanced approach to counterfactual explanation generation. The search for explanation can be adjustable by the user identifying and weighing different biases, given that there are different usages for generating explanations or modus of construal [18], such as debugging the model behaviour or suggesting the end user to act on explanation.

**Future Work** could involve a deeper investigation into the role of diffusion distance in enhancing the robustness of counterfactual explanations. Given the importance of ensuring that generated counterfactuals closely mirror the underlying data distribution, diffusion distance may also contribute to making explanations more resistant to variations in data or model perturbations. It would be interesting to assess the impact of diffusion distance on the stability of counterfactual explanations under varying conditions of data noise and model uncertainty. While this study has highlighted the importance of balancing diffusion distance and directional coherence, the generation of diverse counterfactuals that equally satisfy these criteria presents a challenge. Addressing this challenge could involve the adoption of multi-objective optimization strategies, where diffusion distance and directional coherence are simultaneously optimized. That would enrich the set of available counterfactual explanations and allow to adapt towards user preferences, enhancing applicability.

## 7 Conclusion

We explored the effect of integrating diffusion distance and directional coherence into the counterfactual explanation generation process. The proposed approach produces explanations that are more aligned with human intuition about transition from factual to counterfactual point and take into account the intrinsic geometry of the data. The findings highlight a crucial trade-off between these two factors, underlining the necessity for a balanced integration to optimize explanation quality. By formalising insights from human cognitive processes into AI explanation frameworks, the field is advancing towards AI systems that offer both more human-centric and intuitive explanations.

**Acknowledgments.** This research was supported by the Estonian Research Council Grants PRG1604, the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No. 952060 (Trust AI), by the Estonian Centre of Excellence in Artificial Intelligence (EXAI), by the Estonian Ministry of Education and Research.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## A Appendix

To ensure reproducible research, the code for CoDiCE with all experiments is released in the public repository:

<https://github.com/anitera/CoDiCE>

### Diffusion distance with Self-tuning kernel

Our implementation incorporates a self-tuning kernel within the diffusion distance framework [41], adjusting dynamically to the variance in the data. The main difference of self-tuning kernel approach is the kernel used for local scaling:

$$K(x, y) = \exp\left(-\frac{\|x - y\|^2}{\sigma_x \sigma_y}\right). \quad (6)$$

Here,  $K(x, y)$  denotes the kernel similarity between points  $x$  and  $y$ , while  $\sigma_x$  and  $\sigma_y$  are the local scaling parameters for  $x$  and  $y$ , respectively. These parameters are typically determined based on the distances to their  $k$ -th nearest neighbors, allowing the kernel to adaptively modulate its influence across different data densities.

The transition matrix  $P$ , pivotal for the diffusion process, is derived from the kernel matrix  $K$  through normalization:

$$P = D^{-1}K \quad (7)$$

where  $D$  is the diagonal degree matrix of kernel similarity  $D_{ii} = \sum_j K_{ij}$ . This step converts the kernel similarities into transition probabilities, facilitating the computation of diffusion distances.

For the exact implementation details of the method see Algorithm 1.

**Algorithm 1.** Self-Tuning Diffusion Maps (STDM)

---

**Require:** Data matrix  $X \in \mathbb{R}^{n \times d}$ , number of neighbors  $k$ , diffusion time scale  $\alpha$   
**Ensure:** Diffusion map embeddings  $DMap$ , eigenvalues  $\Lambda$ , eigenvectors  $V$

- 1: Initialize *STDiffusionMap* object with  $k, \alpha$
- 2:  $K \leftarrow \text{construct\_affinity\_matrix}(X, k)$
- 3:  $L \leftarrow \text{construct\_transition\_matrix}(K, \alpha)$
- 4:  $DMap, V, \Lambda \leftarrow \text{make\_diffusion\_coords}(L)$
- 5: **function** CONSTRUCT\_AFFINITY\_MATRIX( $X, k$ )
- 6:   Compute  $k$ -nearest neighbors for  $X$
- 7:   Calculate local scales for each data point
- 8:   Form affinity matrix  $K$  using self-tuning kernel
- 9:   **return**  $K$
- 10: **end function**
- 11: **function** CONSTRUCT\_TRANSITION\_MATRIX( $K, \alpha$ )
- 12:   Calculate right normalization vector  $q$
- 13:   Normalize  $K$  to form transition matrix  $P$
- 14:   Derive Laplacian  $L$  from  $P$
- 15:   **return**  $L$
- 16: **end function**
- 17: **function** MAKE\_DIFFUSION\_COORDS( $L$ )
- 18:   Compute eigenvalues and eigenvectors of  $L$
- 19:   Select top eigenvalues and corresponding eigenvectors
- 20:   Calculate diffusion map embeddings  $DMap$
- 21:   **return**  $DMap, V, \Lambda$
- 22: **end function**

---

**Evaluation metrics for counterfactual explanations**

**Validity** measures the proportion of generated counterfactuals that successfully achieve the desired outcome when applied to the model and defined as:

$$\text{Validity} = \frac{\text{Number of successful counterfactuals}}{\text{Total number of counterfactuals generated}} * 100\% \quad (8)$$

In evaluating counterfactual **proximity** to the original instance, it is standard practice to use the Weighted L1 distance for continuous features, as suggested by Wachter [38], and the  $L_0$  norm for categorical features [10, 17]. While less common, some approaches incorporate the Mahalanobis distance [15] to assess the typicality of the counterfactual within the desired class distribution. For assessing continuous features, we use Weighted L1 Distance to maintain consistency with prior research, but we also add Diffusion distance. As compared to Mahalanobis distance it is offering additional insights by considering the connectivity along the entire path from the factual point.

**The Weighted L1** is defined by adjusting the L1 norm with the inverse of the median absolute deviation (MAD) for each feature. The formula is given by:

$$L1_{\text{Wachter}}(x, x') = \sum_{i=1}^M \left( \frac{|x_i - x'_i|}{\text{MAD}_i} \right) \quad (9)$$



---

**Algorithm 2.** Compute Directional Coherence Score

---

**Require:** Input instance  $x$ , Counterfactual instance  $x'$ , Required label  $y'$ , Model  $M$ , List of features  $F$

**Ensure:** Directional coherence penalty  $dir\_coherence$ , Uncoherent suggestions  $S$

- 1: Initialize  $marginal\_signs \leftarrow \{\}$  ▷ Dictionary with the marginal direction of prediction change for each feature
- 2:  $N \leftarrow \text{Length}(F)$
- 3: **for each**  $f$  in  $F$  **do** ▷ Iterate over all features
- 4:      $marginal\_signs[f] \leftarrow \text{MARGINAL\_PRED\_SIGNS}(x, x', f, y', M)$
- 5: **end for**
- 6:  $dir\_coherence \leftarrow \frac{1}{N} \sum_{f \in F} [marginal\_signs[f] \neq -1]$  ▷ Calculate directional coherence score as ratio of coherent features
- 7:  $S \leftarrow \{f \mid f \in F, marginal\_signs[f] = -1\}$  ▷ Features not changing as expected
- 8: **return**  $dir\_coherence, S$
- 9: **function**  $\text{MARGINAL\_PRED\_SIGNS}(x, x', f, y', M)$
- 10:      $control \leftarrow \text{copy}(x)$
- 11:      $control[f].value \leftarrow x'[f].value$  ▷ Change only the current feature value
- 12:     **if**  $M.type == \text{"classification"}$  **then**
- 13:          $original\_pred \leftarrow M.predict\_proba\_instance(x)$  ▷ If model gives access to probability prediction instead of thresholded value, it gives more precise result
- 14:          $control\_pred \leftarrow M.predict\_proba\_instance(control)$
- 15:          $prob\_sign \leftarrow \text{SIGN}(control\_pred[y'] - original\_pred[y'])$
- 16:     **else**
- 17:          $original\_pred \leftarrow M.predict\_instance(x)$
- 18:          $control\_pred \leftarrow M.predict\_instance(control)$
- 19:          $prob\_sign \leftarrow \text{SIGN}(control\_pred - original\_pred)$
- 20:     **end if**
- 21:     **return**  $prob\_sign$  ▷ Sign of prediction change due to the feature's alteration
- 22: **end function**

---



---

**Algorithm 3.** Genetic Algorithm for Counterfactual Generation

---

**Require:**  $model, original\_instance, desired\_output, population\_size, max\_iterations$

**Ensure:**  $counterfactuals$

- 1: Initialize population with  $population\_size$  members for  $original\_instance$
- 2: Evaluate fitness of each member in population using  $model$  and  $desired\_output$
- 3: **for**  $iteration = 1$  to  $max\_iterations$  **do**
- 4:     Select top half of population based on fitness scores
- 5:     Generate offspring through crossover and mutation operations
- 6:     Evaluate fitness of new members
- 7:     Select  $population\_size$  members for the next generation
- 8:     **if** convergence criteria met **or**  $iteration = max\_iterations$  **then**
- 9:         Extract counterfactuals meeting  $desired\_output$
- 10:     **break**
- 11:     **end if**
- 12: **end for** **return**  $counterfactuals$

---

where  $M$  is total number of points,  $MAD_i$  is the median absolute deviation of the  $i$ -th feature across the dataset, and  $x_i$  and  $x'_i$  are the values of the  $i$ -th feature in the original and counterfactual instances, respectively.

**L0 Categorical** counts the number altered features and defined as:

$$L0(x, x') = \|\{i \mid x_i \neq x'_i\}\|_0 \quad (10)$$

indicating the count of non-zero differences between corresponding features of  $x$  and  $x'$ .

**The Diffusion Distance** captures the proximity of the counterfactual to the original instance, taking into account the intrinsic geometry of the data manifold. It is calculated as:

$$D_{\text{diff}}(x, x') = \sqrt{\sum_{i=1}^M \frac{(p_t(x, i) - p_t(x', i))^2}{\phi_0(i)}} \quad (11)$$

where  $M$  is number of instances,  $p_t(x, i)$  denotes the transition probability from point  $x$  to  $i$  in  $t$  steps, and  $\phi_0(i)$  represents the stationary distribution.

**Directional Coherence** assesses the consistency between the prediction changes of counterfactual features (jointly) and their per feature marginal directions with respect to the model prediction. Given an original instance as a vector  $x = (x_1, x_2, \dots, x_n) \in \mathcal{X} \subseteq \mathbb{R}^n$  and a counterfactual instance  $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in \mathcal{X} \subseteq \mathbb{R}^n$  that brings the desired outcome. The goal is to evaluate the coherence of the transition from  $x$  to  $x^*$  in achieving a specified outcome label  $y$  with a set of expected marginal transitions

$$\{x_i \rightarrow x'_i \mid f(y|x_1, x_2, \dots, x'_i, \dots, x_n) \geq f(y|x_1, x_2, \dots, x_i, \dots, x_n), 1 \leq i \leq n\}.$$

Then, the Directional Coherence score counts the excess of features which have aligned marginal ( $'$ ) and joint ( $*$ ) directions to increase the model's prediction probability towards the desired outcome  $y$ :

$$dcoherence = \frac{1}{n} \sum_{i=1}^n \text{sgn}((x_i^* - x_i)(x'_i - x_i)). \quad (12)$$

## References

1. Antorán, J., Bhatt, U., Adel, T., Weller, A., Hernández-Lobato, J.M.: Getting a CLUE: a method for explaining uncertainty estimates (2020). <https://doi.org/10.48550/ARXIV.2006.06848>
2. Aru, J., Labash, A., Corcoll, O., Vicente, R.: Mind the gap: challenges of deep learning approaches to theory of mind. *Artif. Intell. Rev.* **56**(9), 9141–9156 (2023)
3. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996). <https://doi.org/10.24432/C5XW20>
4. Chalkidis, I., Kampas, D.: Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artif. Intell. Law* **27**(2), 171–198 (2019)

5. Cheng, F., Ming, Y., Qu, H.: DECE: Decision explorer with counterfactual explanations for machine learning models. *IEEE Trans. Visual. Comput. Graph.* **27**(2), 1438–1447 (2021). <https://doi.org/10.1109/TVCG.2020.3030342>
6. Dhurandhar, A., et al.: Explanations based on the missing: towards contrastive explanations with pertinent negatives. *Adv. Neural Inf. Process. Syst.* **31** (2018)
7. Dignum, V.: Introduction. In: Dignum, V. (ed.) *Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way*, pp. 1–7. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-30371-6\\_1](https://doi.org/10.1007/978-3-030-30371-6_1)
8. Förster, M., Hühn, P., Klier, M., Kluge, K.: User-centric explainable AI: design and evaluation of an approach to generate coherent counterfactual explanations for structured data. *J. Decis. Syst.* **32**(4), 700–731 (2023)
9. Gentile, C., Warmuth, M.K.: Linear hinge loss and average margin. *Adv. Neural Inf. Process. Syst.* **11** (1998)
10. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking . <https://doi.org/10.1007/s10618-022-00831-6>
11. Heaton, J.B., Polson, N.G., Witte, J.H.: Deep learning for finance: deep portfolios. *Appl. Stoch. Model. Bus. Ind.* **33**(1), 3–12 (2017)
12. Hilton, D.J.: Mental models and causal explanation: judgements of probable cause and explanatory relevance. *Think. Reason.* **2**(4), 273–308 (1996)
13. Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994). <https://doi.org/10.24432/C5NC77>
14. Holzinger, A., Saranti, A., Angerschmid, A., Finzel, B., Schmid, U., Mueller, H.: Toward human-level concept learning: pattern benchmarking for AI algorithms. *Patterns* (2023)
15. Kanamori, K., Takagi, T., Kobayashi, K., Arimura, H.: DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In: *IJCAI*, pp. 2855–2862 (2020)
16. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Comput. Surv.* **55**(5), 1–29 (2022)
17. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 353–362 (2021)
18. Keil, F.C.: Explanation and understanding. **57**, 227–254 (2006). <https://doi.org/10.1146/annurev.psych.57.102904.190100>
19. Kulesza, T., Stumpf, S., Burnett, M., Yang, S., Kwan, I., Wong, W.K.: Too much, too little, or just right? ways explanations impact end users’ mental models. In: *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pp. 3–10. IEEE (2013)
20. Laugel, T., Lesot, M.-J., Marsala, C., Renard, X., Detyniecki, M.: Comparison-based inverse classification for interpretability in machine learning. In: Medina, J., et al. (eds.) *Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2018*, pp. 100–111. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-91473-2\\_9](https://doi.org/10.1007/978-3-319-91473-2_9)
21. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277* (2019)
22. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019). <https://doi.org/10.1016/j.artint.2018.07.007>

23. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J.T.: Deep learning for health-care: review, opportunities and challenges. *Brief. Bioinform.* **19**(6), 1236–1246 (2018)
24. Moreira, C., Chou, Y.L., Hsieh, C., Ouyang, C., Jorge, J., Pereira, J.M.: Benchmarking counterfactual algorithms for XAI: from white box to black box. arXiv preprint [arXiv:2203.02399](https://arxiv.org/abs/2203.02399) (2022)
25. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT\* 2020)*, pp. 607–617. Association for Computing Machinery (2020). <https://doi.org/10.1145/3351095.3372850>, event-place: Barcelona, Spain
26. Pawelczyk, M., Bielawski, S., Heuvel, J.v.d., Richter, T., Kasneci, G.: CARLA: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms (2021). <https://doi.org/10.48550/ARXIV.2108.00783>
27. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., Bie, T.D., Flach, P.: FACE. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM (2020). <https://doi.org/10.1145/3375627.3375850>
28. Prasad, N.N., Rao, J.N.: The estimation of the mean squared error of small-area estimators. *J. Am. Stat. Assoc.* **85**(409), 163–171 (1990)
29. Raman, N., Magazzeni, D., Shah, S.: Bayesian hierarchical models for counterfactual estimation. In: *International Conference on Artificial Intelligence and Statistics*, pp. 1115–1128. PMLR (2023)
30. Rasouli, P., Chieh Yu, I.: Care: coherent actionable recourse based on sound counterfactual explanations. *Int. J. Data Sci. Analyt.* **17**(1), 13–38 (2024)
31. Sakkas, N., et al.: Explainable approaches for forecasting building electricity consumption. *Energies* **16**(20), 7210 (2023)
32. Schmidt, P., Biessmann, F., Teubner, T.: Transparency and trust in artificial intelligence systems. *J. Decis. Syst.* **29**(4), 260–278 (2020)
33. Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. *Annu. Rev. Biomed. Eng.* **19**, 221–248 (2017)
34. Sokol, K., Flach, P.: Explainability fact sheets: a framework for systematic assessment of explainable approaches. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 56–67 (2020)
35. Tenenbaum, J.B., Silva, V.D., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
36. Van Looveren, A., Klaise, J.: Interpretable counterfactual explanations guided by prototypes. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (eds.) *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021*, pp. 650–665. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86520-7\\_40](https://doi.org/10.1007/978-3-030-86520-7_40)
37. Virgolin, M., Alderliesten, T., Witteveen, C., Bosman, P.A.N.: Improving model-based genetic programming for symbolic regression of small expressions. *Evol. Comput.* **29**(2), 211–237 (2021)
38. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017)
39. Whittaker, M., et al.: *AI Now Report 2018*. AI Now Institute at New York University New York (2018)

40. Wolberg, W., Mangasarian, O., Street, N., Street, W.: Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository (1995). <https://doi.org/10.24432/C5DW2B>
41. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. *Adv. Neural Inf. Process. Syst.* **17** (2004)
42. Zemla, J.C., Sloman, S., Bechlivanidis, C., Lagnado, D.A.: Evaluating everyday explanations. *Psychon. Bull. Rev.* **24**, 1488–1500 (2017)



# CountARFactuals – Generating Plausible Model-Agnostic Counterfactual Explanations with Adversarial Random Forests

Susanne Dandl<sup>1,2</sup> , Kristin Blesch<sup>3,4</sup> , Timo Freiesleben<sup>6</sup> ,  
Gunnar König<sup>7</sup> , Jan Kapar<sup>3,4</sup> , Bernd Bischl<sup>1,2</sup> ,  
and Marvin N. Wright<sup>3,4,5</sup> 

<sup>1</sup> Department of Statistics, LMU Munich, Munich, Germany

<sup>2</sup> Munich Center for Machine Learning (MCML), Munich, Germany

<sup>3</sup> Leibniz Institute for Prevention Research and Epidemiology - BIPS,  
Bremen, Germany  
[wright@leibniz-bips.de](mailto:wright@leibniz-bips.de)

<sup>4</sup> Faculty of Mathematics and Computer Science, University of Bremen,  
Bremen, Germany

<sup>5</sup> Department of Public Health, University of Copenhagen, Copenhagen, Denmark

<sup>6</sup> Cluster: Machine Learning for Science, University of Tübingen, Tübingen, Germany

<sup>7</sup> Department of Computer Science and Tübingen AI Center, University of Tübingen,  
Tübingen, Germany

**Abstract.** Counterfactual explanations elucidate algorithmic decisions by pointing to scenarios that would have led to an alternative, desired outcome. Giving insight into the model’s behavior, they hint users towards possible actions and give grounds for contesting decisions. As a crucial factor in achieving these goals, counterfactuals must be plausible, i.e., describing realistic alternative scenarios within the data manifold. This paper leverages a recently developed generative modeling technique – adversarial random forests (ARFs) – to efficiently generate plausible counterfactuals in a model-agnostic way. ARFs can serve as a plausibility measure or directly generate counterfactual explanations. Our ARF-based approach surpasses the limitations of existing methods that aim to generate plausible counterfactual explanations: It is easy to train and computationally highly efficient, handles continuous and categorical data naturally, and allows integrating additional desiderata such as sparsity in a straightforward manner.

**Keywords:** Counterfactual explanations · Explainable artificial intelligence · Interpretable machine learning · Adversarial random forest · Tabular data · Plausibility · Model-agnostic

---

S. Dandl, K. Blesch, T. Freiesleben and G. König—Equal contribution as first authors.

© The Author(s) 2024

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 85–107, 2024.

[https://doi.org/10.1007/978-3-031-63800-8\\_5](https://doi.org/10.1007/978-3-031-63800-8_5)

## 1 Introduction

Machine learning (ML) algorithms are increasingly used in high-stakes scenarios. For example, they help to decide whether you receive a loan, if you are suitable for a job, or even which disease you are diagnosed with. While ML-based systems are powerful at detecting complex patterns in data, the reasoning behind their predictions is often not easy to discern for humans. Many ML models are black boxes with a complex mathematical structure that do not follow transparent logical rules [8].

The emerging field of *interpretable machine learning* (IML) (also known as explainable artificial intelligence or XAI for short) promises to open up these black boxes and aims to make the decisions of ML models transparent to humans (see [2, 37] for overviews). A particularly simple approach is to explain algorithmic decisions to end-users via so-called *counterfactual explanations* [48].

**Example:** Imagine you apply for a loan. You enter characteristics such as your age, salary, loan amount, etc. in the online application form and after a few seconds you receive the decision – your loan application has been denied. A counterfactual explanation could be: If your salary had been €5,000 higher, your loan would have been approved.

More generally, a counterfactual explanation points to a close alternative scenario (the so-called *counterfactual*) that, in contrast to the actual scenario, would have resulted in the desired outcome. Counterfactual explanations may be employed for various purposes, such as helping to guide a person’s actions [27, 32], enabling them to contest adverse decisions [34], and providing insights into the decision behavior of the model [36]. For all these goals, counterfactuals must be *plausible*, which means the alternative scenarios they depict are realistic. For instance, in the example above, suggesting a negative loan amount or a real estate loan with an amount of €500 would not be very plausible counterfactuals.

When adding plausibility as another objective for generating counterfactuals, its trade-off with other objectives like proximity, i.e., that the counterfactual is close to the point of interest, should be taken into account. Dandl et al. [11] were one of the first to address this trade-off by framing the counterfactual search as a multi-objective optimization problem. Their approach – multi-objective counterfactual explanations (MOC) – returns not just a single counterfactual, but a Pareto set of counterfactuals (see [50] for a definition of Pareto-optimality), which is advisable to account for the Rashomon effect, i.e., that multiple, diverse, equally good counterfactuals may exist [6].

An intuitive approach to plausibility is searching for only those counterfactuals that are close to actual instances in the dataset [22]. To operationalize this goal, one objective in MOC minimizes the distance between counterfactuals and the actual instances. However, as presented in Sect. 3.1, this approach has its limitations if, for example, there are low-density gaps close to the point of interest between high-density regions. Other approaches model plausibility via the joint probability density. They rely on computationally intensive neural network

architectures such as variational autoencoders (VAEs) [7, 25, 35, 39] or generative adversarial networks (GANs) [38, 45]. While these architectures have merits for high-dimensional tensor data (e.g., images or text), they are less suited for tabular data (see our discussion in Sect. 3.2).

*Contributions.* We leverage a tree-based technique from generative modeling called *adversarial random forests* (ARF) [49] to generate plausible counterfactuals in a mixed (i.e., categorical and continuous) tabular data setting. We call these countARFactuals and propose two model-agnostic algorithms to generate them:

1. We integrate ARF into the multi-objective counterfactual explanation (MOC) framework [11] to speed up the counterfactual search and find more plausible counterfactuals (see Sect. 4.1).
2. We tailor ARF to directly generate plausible counterfactuals without an optimization algorithm (see Sect. 4.2).

A simulation study shows the advantages in plausibility and efficiency of our ARF-based approaches compared to competing methods (Sect. 5). Moreover, we apply our method on a real-world dataset, namely to explain coffee quality predictions (Sect. 6).

## 2 Related Work

There is widespread agreement in the counterfactual community that plausibility is an important concern [16, 22, 27, 29, 44, 47]. Various suggestions have been made to incorporate plausibility into the counterfactual search, for example using causal knowledge [32, 35], case-based reasoning [30], outlier detectors [26], restricting the search space [3], imputing feature combinations from real instances [20], respecting paths between datapoints [40], or, as described above, staying close to the training data [11].

Many define plausibility theoretically through the joint probability density [47]. Some works rely on VAEs or standard autoencoders: they directly generate counterfactuals [35, 39], use VAEs in the optimization [25] or just for measuring plausibility [7]. Other works rely on GANs to generate counterfactuals [38, 45]. However, these approaches differ substantially from our work, as they are tailored for neural network models [35], focus only on plausibility thereby ignoring other objectives like sparsity [35, 39] (see Sect. 3.1), or work only for continuous data [25, 38]. The closest works to ours are Brughmans et al. [7] and Dandl et al. [11]. Both are designed to generate plausible and sparse counterfactuals in mixed tabular data settings. Brughmans et al. [7] use the autoencoder reconstruction loss as a plausibility measure and Dandl et al. [11] use the distance to the  $k$ -nearest neighbors to evaluate plausibility. We show in our experiments in Sect. 5 that utilizing ARF to generate counterfactuals improves plausibility compared to those approaches while being computationally fast.



### 3 Background

Before we present our approaches, we provide background on the two methods we build upon: multi-objective counterfactual explanations (MOC) [11] and adversarial random forests (ARF) [49].

We consider a supervised learning setup with a binary classification or regression problem.<sup>1</sup>  $\mathcal{X}$  denotes a  $p$ -dimensional feature space. The respective vector  $\mathbf{X} := (X_1, \dots, X_p)^T$  of random variables may contain both continuous and categorical features. With  $Y \in \mathbb{R}$ , we denote a random variable reflecting the outcome. In case of a binary classification model, we restrict  $Y$  to  $\{0, 1\}$ .

To predict  $Y$  from  $\mathbf{X}$ , we trained an ML model  $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$  on a dataset  $D_{\text{train}} := \{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(n_{\text{train}})}, y^{(n_{\text{train}})})\}$  with  $n_{\text{train}}$  observations. For binary classification, the model output is restricted to  $\hat{f}(\mathbf{x}) \in [0, 1]$ , reflecting the probability for  $Y = 1$ . Most counterfactual explanation methods require access to a dataset for generating counterfactuals. To reflect that this dataset *can* differ to  $D_{\text{train}}$ , we denote it as  $D$  in the following and assume it to consist of  $n$  observations.

#### 3.1 Multi-objective Counterfactual Explanations

Suppose we want to explain why a certain datapoint of interest  $\mathbf{x}^*$  was predicted as  $\hat{f}(\mathbf{x}^*)$  instead of a desired prediction within  $Y_{des} \subset \mathbb{R}$ . Wachter et al. [48] define counterfactuals as the closest possible input vector  $\mathbf{x}^{cf}$  to  $\mathbf{x}^*$  according to some distance on  $\mathcal{X}$  such that  $\hat{f}(\mathbf{x}^{cf}) \in Y_{des}$ . This definition does not explicitly demand sparse or plausible changes. When integrating all these desiderata into an objective to generate counterfactuals, trade-offs between the different objectives must be taken into account since the objectives conflict each other. Figure 1a illustrates this for the properties plausibility and proximity to the original instance  $\mathbf{x}^*$ . If all high-density regions are far away from the decision boundary, enforcing proximity leads to unrealistic counterfactuals.

To consider these trade-offs, Dandl et al. [11] turned the search for counterfactuals into a multi-objective optimization problem:

$$\mathbf{x}^{cf} \in \arg \min_{\mathbf{x} \in \mathcal{X}} \left( o_{\text{valid}}(\hat{f}(\mathbf{x}), Y_{des}), o_{\text{prox}}(\mathbf{x}, \mathbf{x}^*), o_{\text{plaus}}(\mathbf{x}, D), o_{\text{sparse}}(\mathbf{x}, \mathbf{x}^*) \right). \quad (1)$$

The different objectives denote:

1. **Validity:** Counterfactuals should have a predicted outcome in  $Y_{des}$

$$o_{\text{valid}}(\hat{f}(\mathbf{x}), Y_{des}) := \inf_{y \in Y_{des}} |\hat{f}(\mathbf{x}) - y|. \quad (2)$$

2. **Proximity:** Counterfactuals should be close to  $\mathbf{x}^*$  according to the Gower distance  $d_{\text{Gower}}$  [19]

$$o_{\text{prox}}(\mathbf{x}, \mathbf{x}^*) := d_{\text{Gower}}(\mathbf{x}, \mathbf{x}^*). \quad (3)$$

---

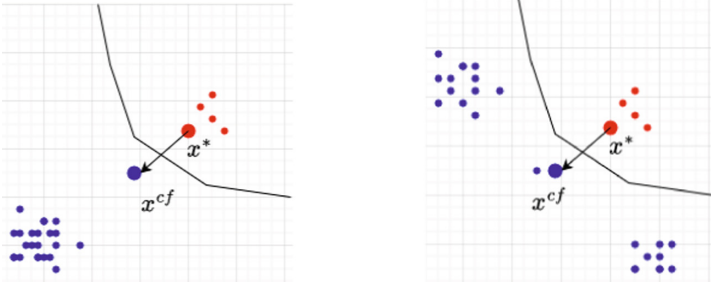
<sup>1</sup> Our framework also generalizes to multi-class problems; we restrict ourselves here only for the sake of simplicity and notation.

3. **Plausibility:** Counterfactuals should describe a realistic data instance, with  $\mathbf{x}^{[1]}, \dots, \mathbf{x}^{[k]}$  indicating the  $k$ -nearest neighbors to  $\mathbf{x}$  within data  $D$  and  $w_i$  denoting weights with  $\sum_{i=1}^k w_i = 1$

$$o_{\text{plaus}}(\mathbf{x}, D) := \sum_{i=1}^k w_i d_{\text{Gower}}(\mathbf{x}, \mathbf{x}^{[i]}). \quad (4)$$

4. **Sparsity:** Counterfactuals should vary from  $\mathbf{x}^*$  in only a few features

$$o_{\text{sparse}}(\mathbf{x}, \mathbf{x}^*) := \|\mathbf{x} - \mathbf{x}^*\|_0 = \sum_{j=1}^p \mathbb{1}_{x_j \neq x_j^*}. \quad (5)$$



(a) Plausibility-proximity trade-off      (b) Limitation of MOC's plausibility

**Fig. 1.** (a) Proximity and plausibility can be conflicting objectives [11]; enforcing proximity may lead to unrealistic counterfactuals  $\mathbf{x}^{cf}$ . (b) To have high proximity (i.e., low  $o_{\text{prox}}$  in Eq. (3)) and high plausibility (i.e., low  $o_{\text{plaus}}$  in Eq. (4), with  $k = 1$ ), the counterfactual may be in a low-density region.

Dandl et al. [11] adapted the nondominated sorting genetic algorithm or short NSGA-II of Deb et al. [12] to solve the multi-objective optimization problem. This algorithm follows three steps:

1. It generates a set of candidate instances close to the point of interest  $\mathbf{x}^*$ . Among these, it recombines and mutates the candidates that perform best according to the above criteria. Per default, the mutator does not take feature dependencies into account. To enhance plausibility, mutation can be optionally performed by sampling from conditional distributions learned on  $D$  by conditional trees [24] – we refer to this MOC version as MOCCTREE.
2. Both new and old candidates are ranked using nondominated and crowding distance sorting. Nondominated sorting ranks according to optimality with respect to the above objectives (with the option to penalize invalid counterfactuals) and crowding distance ranks according to diversity.

- Based on these rankings, optimal and diverse candidates are selected for the next iteration. The search for counterfactuals ends after either a fixed number of predefined iterations or when the generated counterfactuals are not significantly better according to the hypervolume of the objectives above. As a final step, the algorithm outputs the Pareto optimal set of counterfactuals over the generations.

The conceptualization of plausibility as Eq. (4) has its limitations as, e.g., illustrated in Fig. 1b: With  $k = 1$  (the default in MOC), counterfactuals with low values in Eq. (4) might still end up in low-density regions.

### 3.2 Generative Modeling and Adversarial Random Forests

Generative modeling is concerned with models that generate synthetic data  $\tilde{D}$  that mimic the appearance of real data  $D$ . A well-known approach are VAEs [31], which encode original data instances into a set of low-dimensional distribution parameters and then reconstruct these instances with a decoder neural network from samples of these distributions. Another common technique are GANs [18], where two different neural network models play a zero-sum game – the generator network aims to generate realistic instances, and the discriminator network aims to discriminate these instances from real data. Other generative models based on neural networks include normalizing flows [41], diffusion probabilistic models [23] and transformer-based models [46] (see [4, 15] for overviews). While there exist adaptations of neural network models to tabular data, tree-based approaches may be better suited [5, 21, 43].

ARFs are a tree-based procedure for generative modeling [49]. Similarly to GANs, the ARF approach relies on an adversarial training procedure. However, instead of neural networks, ARF relies on random forests, using the parameterization learned by the discrimination also for the generator (see [49], for details). An ARF is trained in three steps: (1) Fitting an unsupervised random forest [42], which generates a naive synthetic dataset  $\tilde{D}_1$  and subsequently trains a random forest  $\hat{g}_1$  to distinguish between  $D$  and  $\tilde{D}_1$ . (2) Sampling feature values marginally from the instances in the leaves of  $\hat{g}_1$  to obtain a more realistic synthetic dataset  $\tilde{D}_2$ . Another random forest  $\hat{g}_2$  is trained to distinguish between  $D$  and  $\tilde{D}_2$ . (3) This process is repeated until the random forest classifier can no longer distinguish synthetic from real data. We denote the final ARF model as  $\hat{g}^*$ . As opposed to GANs, ARFs allow for both density estimation and generative modeling. The two algorithms are called *forests for density estimation* (FORDE) and *forests for generative modeling* (FORGE), respectively.

*Density estimation with FORDE* leverages the mutual independence across features in the leaves after algorithm convergence, which allows modeling the joint density  $p(\mathbf{x})$  as a mixture of univariate feature densities:

$$\text{FORDE}(\mathbf{x}) := \hat{p}(\mathbf{x}) = \sum_{l:\mathbf{x} \in X_l} \pi_l \prod_{j=1}^p \hat{p}_{l,j}(x_j), \quad (6)$$

where  $X_l$  is the hyperrectangle defined by the  $l$ -th leaf, the corresponding mixture weights  $\pi_l$  are calculated as the share of real datapoints that fall into leaf  $l$  normalized over all trees, and  $\hat{p}_{l,j}$  are (locally) estimated univariate density/mass functions for the  $j$ -th feature in leaf  $l$ . The convergence of FORDE to the real data distribution of  $\mathbf{X}$  for infinite data is proven under some mild conditions in Watson et al. [49]. A conditional density under a set of conditions  $\mathcal{C}$ , e.g., fixed values or intervals for certain features  $C \subseteq \{1, \dots, p\}$ , can be derived from Eq. (6) in the following way:

$$\text{FORDE}(\mathbf{x} \mid \mathcal{C}) := \hat{p}(\mathbf{x} \mid \mathcal{C}) = \sum_{l:\mathbf{x} \in X_l} \pi'_l \prod_{j=1}^p \hat{p}_{l,j}(x_j \mid \mathcal{C}_j), \quad (7)$$

where  $\mathcal{C}_j \subseteq \mathcal{C}$  denotes the subset of conditions concerning feature  $j \in C$ , and the mixture weights  $\pi'_l$  are updated to reflect how likely their corresponding leaves fulfill the condition. More formally, the mixture weights are updated and normalized using the univariate marginals by

$$\pi'_l := \frac{\pi_l \prod_{j=1}^p \hat{p}_{l,j}(\mathcal{C}_j)}{\sum_{m:\mathbf{x} \in X_m} \pi_m \prod_{j=1}^p \hat{p}_{m,j}(\mathcal{C}_j)} \quad (8)$$

if the denominator does not equal 0 and by  $\pi'_l := 0$  otherwise. Note that in the case of conditioning on a fixed value or interval for a continuous feature  $j$ , the univariate densities  $\hat{p}_{l,j}$  collapse to the indicator function  $\mathbb{1}_{\mathcal{C}_j}$  or the unconditional densities truncated on the conditioning interval, respectively.

*Generative modeling with FORGE* is based on drawing a leaf  $l$  from the forest according to the mixture weights in FORDE and sampling feature values from the estimated univariate (conditional) densities  $\hat{p}_{l,j}$ . Thereby, FORGE allows drawing samples that adhere to FORDE as an approximation to the real distribution of  $\mathbf{X}$  or  $\mathbf{X} \mid \mathcal{C}$ .

## 4 Methods

Our proposal is to leverage ARF for the efficient generation of counterfactual explanations, i.e., countARFactuals, in mixed tabular data settings. More specifically, we use and modify ARF to account for the desiderata that we discussed in Sect. 3.1:

1. **Validity:** We train ARF on  $D$  but replace the target  $Y$  with the predictions  $\hat{Y}$ . Here,  $\hat{Y}$  is treated just as any other feature in the data. Since FORGE allows for conditional sampling, we can sample from  $\mathbf{X}$  conditioned on our desired outcomes  $\hat{Y} \in Y_{des}$ . Note, however, that ARF may not learn a perfect representation of the prediction function  $\hat{Y} := \hat{f}(\mathbf{X})$ . It therefore is not guaranteed that ARF-samples are valid, it only becomes more likely. In our algorithms, we only return those candidates with predictions in  $Y_{des}$ .

2. **Proximity:** We restrict the output of our two methods to those counterfactuals in the Pareto set, defined over the four objectives of Sect. 3.1, including proximity (Eq. (3)). In the first algorithm described below, we additionally use ARF combined with MOC, which accounts for proximity, as described in Sect. 3.1.
3. **Plausibility:** ARF allows us to both evaluate the plausibility of datapoints using FORDE (which is also used to determine the returned Pareto set) and efficiently generate plausible data with FORGE.
4. **Sparsity:** FORGE allows sampling feature values  $X_S$  based on the observation  $X_C = x_C$ . By fixing certain features  $C$  to the value of  $\mathbf{x}_C^*$ , we only change feature values in the sparse set  $S := \{1, \dots, p\} \setminus C$ .

With the desiderata in place, several decisions need to be made: Should we integrate plausibility via density estimation (FORDE) or generative modeling (FORGE)? What is an optimal trade-off between proximity and other objectives, such as plausibility and sparsity? How should we search for the conditioning set  $C$  for features that should not be changed? In the following, we provide two algorithms that decide on these questions in different ways. The first integrates ARF into MOC (Sect. 4.1). The second uses ARF as a standalone counterfactual generator (Sect. 4.2).

#### 4.1 Algorithm 1: Integrating ARF into MOC

In MOC’s optimization problem (Eq. (1)), we substitute the plausibility measure (Eq. (4)) by the density estimator of FORDE (Eq. (6)). Since the individual objectives in MOC must map to a zero-one interval (with low values denoting desired properties), we transform  $\hat{p}(\mathbf{x})$ , as estimated by FORDE, with the negative exponential function<sup>2</sup>

$$o_{\text{plaus}}^*(\mathbf{x}) := e^{-\hat{p}(\mathbf{x})}. \quad (9)$$

We use FORGE as described above to sample plausible candidates in MOC in the mutation step of the NSGA-II. This is a strategy to efficiently limit the search space of MOC to plausible counterfactuals. Concerning sparsity, we find the conditioning set  $C$  through iterated mutation and recombination, just like in MOC, and we select candidates using NSGA-II according to optimality and diversity. At last, the output comprises only the valid Pareto-set of counterfactuals over the generations, i.e., counterfactuals that have a prediction in  $Y_{des}$  and are not dominated by other candidates that were generated (w.r.t.  $o_{\text{prox}}$ ,  $o_{\text{sparse}}$  and  $o_{\text{plaus}}^*$ ). For details, we refer to the pseudocode in Appendix A.

#### 4.2 Algorithm 2: ARF Is All You Need

For this algorithm, we leverage the ability of our modified ARF sampler to directly and efficiently generate many relevant counterfactuals. As described

<sup>2</sup> The negative exponential has the advantage that small changes at low values of  $\hat{p}(\mathbf{x})$  are more distinctive than at high values. Other transformation methods are also possible but a comparison is beyond the scope of this paper.

above, the modified FORGE method allows generating plausible datapoints. To enforce sparsity, we sample  $m$  features with probabilities according to their local feature importance, calculated as the standard deviation of the individual conditional expectation (ICE) curve [11,17]. The  $m$  selected features describe the features  $S$  we aim to change because they, according to the local feature importance, impact the prediction the most. The remaining features then form the conditioning set  $C = \{1, \dots, p\} \setminus S$ .

As for Algorithm 1, we output only the valid and Pareto-optimal set of counterfactuals (w.r.t.  $o_{\text{prox}}$ ,  $o_{\text{sparse}}$  and  $o_{\text{plaus}}^*$ ). The pseudocode for this method is given in Appendix B.

## 5 Experiments

We evaluate the quality of our proposed methods with respect to the following research questions:

- RQ (1) Do our proposed ARF-based methods generate more plausible counterfactuals compared to competing methods without major sacrifices in sparsity ( $o_{\text{sparse}}$ ), proximity ( $o_{\text{prox}}$ ) and the runtime?
- RQ (2) Does  $o_{\text{plaus}}^*$  (Eq. (9)) better reflect the true plausibility compared to  $o_{\text{plaus}}$  (Eq. (4))?

To objectively evaluate the plausibility of the generated counterfactuals, we require access to the ground-truth likelihood. Because ground-truth likelihoods are usually unavailable for real-world data, we evaluate our methods on synthetic data. An illustrative real-world application follows in Sect. 6.

### 5.1 Data-Generating Process

For the experiments, we constructed three illustrative two-dimensional datasets, namely `cassini` (inspired by [14]), `two sines` (inspired by the two moons dataset), and `three blobs` (inspired by [39]). Moreover, we generated four datasets from randomly sampled Bayesian networks of dimensionality 5, 10, and 20, namely `bn_5`, `bn_10`, and `bn_20`, which all include both continuous and categorical features as well as nonlinear relationships. An XGBoost model was fitted on sampled datasets  $D_{\text{train}}$  of size 5 000 [9]. For each data-generating process (DGP), ten additional points were sampled as instances of interest  $\mathbf{x}^*$ . The counterfactual generation methods received access to newly sampled datasets  $D$  of size 5 000. Details on the dataset generation and model fit can be found in Appendix C and in the repository accompanying this paper.<sup>3</sup>

### 5.2 Competing Methods

We compare our proposed MOC version based on ARF of Sect. 4.1 (referred to as MOCARF) and the standalone ARF generator of Sect. 4.2 (referred to as ARF)

<sup>3</sup> <https://github.com/bips-hb/countARFactuals>.

to the following competitors: MOC and MOCCTREE (MOC with a conditional sampler, see Sect. 3.1) [11] and NICE [7] with a plausibility reward function (see Eq. (4) in [7]). NICE generates counterfactuals by iteratively replacing one feature after the other in  $\mathbf{x}^*$  by the values of  $\mathbf{x}^{\text{mn}}$ , which denotes a nearest neighbor of  $\mathbf{x}^*$  in  $D$  with  $\hat{f}(\mathbf{x}^{\text{mn}}) \in Y_{des}$ . In each iteration, the algorithm keeps the feature change with the highest plausibility reward.

To allow for a fair comparison, all methods generate a *set* of counterfactual candidates. For NICE, we apply the extension of Dandl et al. [10]; instead of stopping the search once the point with the highest reward has a prediction in  $Y_{des}$ , the search continues until  $\mathbf{x}^{\text{mn}}$  is recovered and all intermediate instances with predictions in  $Y_{des}$  are returned. If possible, we selected the hyperparameters for the methods such that each method generated an equal number of candidates – namely, 1000.<sup>4</sup> ARF requires a maximum set size for  $S$ , reflecting how many features are maximally allowed to be changed. We set it according to the number of features  $p$  as  $m_{max} := \min(\lceil \sqrt{p} + 3 \rceil, p)$ . Since also for all MOC-based methods the maximum number can be specified, we used the same  $m_{max}$  for MOC, MOCARF and MOCCTREE. For the evaluation, we focused only on the unique counterfactuals that have predictions in  $Y_{des}$ . We further reduced this set to the Pareto set, i.e., the set of counterfactuals that are non-dominated according to proximity ( $o_{\text{prox}}$ ), sparsity ( $o_{\text{sparse}}$ ) and plausibility. The definition of the plausibility objective differed between the methods, with  $o_{\text{plaus}}^*$  for ARF and MOCARF,  $o_{\text{plaus}}^{\text{plaus}}$  for MOC and MOCCTREE, and the autoencoder reconstruction error for NICE (as proposed by [7]).

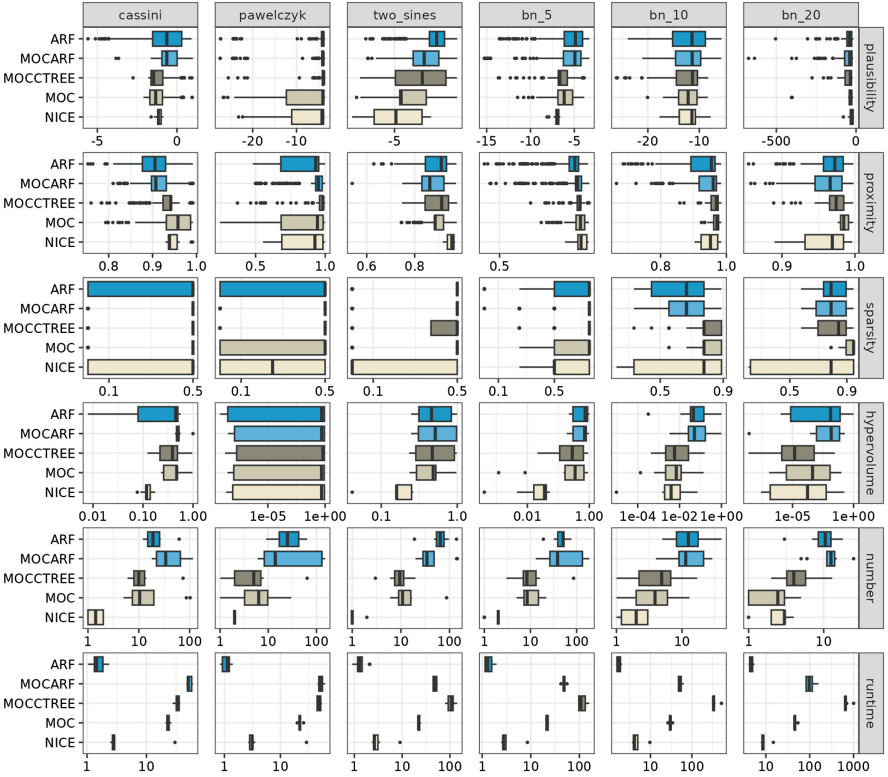
### 5.3 Evaluation Criteria

To answer RQ (1), we evaluated the generated counterfactuals with respect to the ground-truth likelihood (denoted as plausibility, in the following), proximity  $o_{\text{prox}}$  and sparsity  $o_{\text{sparse}}$  (see Sect. 3.1). We aggregated the results per method, dataset and instance of interest  $\mathbf{x}^*$  by computing (scaled) dominated hypervolumes [50]. We also measured the number of nondominated counterfactuals and the runtime. To investigate the trade-off between plausibility and proximity, and between plausibility and sparsity, we also computed median attainment surfaces according to López-Ibáñez et al. [33] for each method and dataset. It reveals how two objectives are distributed on average over the different  $\mathbf{x}^*$ . To answer RQ (2), all generated counterfactuals were evaluated with respect to  $o_{\text{plaus}}^*$  and  $o_{\text{plaus}}$ . Per method, dataset and  $\mathbf{x}^*$ , we computed Spearman-rank correlations between the true plausibility and  $o_{\text{plaus}}^*$  and between the true plausibility and  $o_{\text{plaus}}$ . With a Wilcoxon signed rank test, we tested whether  $o_{\text{plaus}}^*$  has higher correlations to the true plausibility than  $o_{\text{plaus}}$ .

### 5.4 Results

Figure 2 presents the results for RQ (1) and shows the objective values per counterfactuals as well as the hypervolume, number of nondominated counterfactuals

<sup>4</sup> Specifying the exact number was possible for all methods besides NICE [7].



**Fig. 2.** Boxplots of the logarithmic plausibility, proximity ( $1 - o_{\text{prox}}$ ), sparsity ( $1 - o_{\text{sparse}}$ ), hypervolume, number of nondominated counterfactuals and runtime for each method and dataset. Higher values are better, except for runtime.

and runtime. On average, ARF and MOCARF generated more plausible counterfactuals compared to the other MOC-based approaches and NICE. In alignment with previous literature [11, 13], our results suggest that higher plausibility might be associated with lower proximity and sparsity. For further investigations on the trade-offs, Fig. 5 and Fig. 6 in Appendix D detail the median attainment surfaces per dataset and method. The plots illustrate Pareto dominance across individual objectives within the multi-objective optimization problem, revealing that ARF and MOCARF on average dominate the other methods in proximity, sparsity and plausibility, with the differences being greatest in plausibility. The hypervolume was on average similar for the different methods for low-dimensional datasets (ARF had lower hypervolumes in *cassini* due to its inferiority in proximity and sparsity), for higher-dimensional datasets (*bn\_10* and *bn\_20*), ARF and MOCARF performed better than the competing methods. Concerning runtime, ARF generated counterfactuals the fastest on average, followed by NICE and MOC. MOCARF was faster than MOCCTREE for datasets with more than two features. The runtime differences increased with higher dimensional data. On average, ARF and MOCARF generated the largest set of nondominated



counterfactuals compared to the other methods (see Fig. 2). Note that for all methods, the results on the objectives have high variance. The reason for this variance stems from individual instances  $\mathbf{x}^*$  in the set of explained instances that have particularly high variances, as can be inferred from the results of each  $\mathbf{x}^*$  separated per method and dataset. ARF and MOCARF have particularly high variance in the results, as they provide larger sets of nondominated counterfactuals.

Considering RQ (2), the Wilcoxon rank sum test had a p-value close to 0 ( $7.16e - 06$ ), i.e., the correlation of our proposed plausibility measure  $o_{\text{plaus}}^*$  to the true plausibility was significantly higher than that of  $o_{\text{plaus}}$ . The median correlation to the true plausibility over all methods and datasets was 0.84 for  $o_{\text{plaus}}^*$  and 0.69 for  $o_{\text{plaus}}$ . Figure 4 in the Appendix shows a more detailed comparison of the distributions of the correlations.

Overall, our study shows that on average our proposed methods – ARF and MOCARF – generate a more plausible set of counterfactuals compared to our competitors without major sacrifices in sparsity and proximity. Notably, ARF achieves this with superiority in runtimes.

## 6 Real Data Example

We illustrate our approach on the publicly available coffee quality dataset.<sup>5</sup> The data details the characteristics of several Arabica coffee beans, such as the country of origin and altitude at which the beans were cultivated. Further, the dataset includes information on a quality review score (*cup points*) specified by an expert jury within the Coffee Quality Institute [1].

In this example, we use a random forest to predict coffee quality from selected, actionable characteristics of the coffee beans. For simplicity, we binarize the target score *cup points*. Aiming for balanced classes of *good* and *bad* quality, we use the dataset’s median value of *cup points* as a cut-off point, i.e.,

$$\text{quality} = \begin{cases} \textit{good} & \text{if } \textit{cup points} \geq \text{median}(\textit{cup points}) \\ \textit{bad} & \text{otherwise.} \end{cases} \quad (10)$$

For illustration, we generate counterfactual explanations for an instance of bad coffee quality, answering the question: Which characteristics would need to be changed to rate as good quality coffee?

This example illustrates the importance of taking into account the multiple objectives of counterfactual explanations, such as sparsity and plausibility. For example, a company that aims to improve the quality of their coffee may want to make as sparse changes to the coffee characteristics as possible for economic reasons. Similarly, some changes might not be plausible, think of changing the country of origin independently of the altitude of the coffee plantations or the variety of beans cultivated in the respective country (since the variety must suit the natural conditions in the respective country).

<sup>5</sup> <https://github.com/jldbc/coffee-quality-database>.

	country of origin	harvest year	variety	processing method	moisture	altitude mean meters	quality
point of interest $x^*$	Taiwan	2014	Typica	Washed / Wet	0.10	800.00	bad
countARFactual #1	Mexico ↔	2013 ↓	Typica	Washed / Wet	0.11 ↑	1498.68 ↑	good ↔
countARFactual #2	Taiwan	2014	Typica	Washed / Wet	0.07 ↓	1172.96 ↑	good ↔
countARFactual #3	Colombia ↔	2013 ↓	Caturra ↔	Washed / Wet	0.01 ↓	1735.77 ↑	good ↔
countARFactual #4	Taiwan	2013 ↓	Typica	Washed / Wet	0.11 ↑	1268.73 ↑	good ↔
countARFactual #5	Taiwan	2014	Typica	Washed / Wet	0.11 ↑	1013.48 ↑	good ↔
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
countARFactual #11	Taiwan	2014	Typica	Washed / Wet	0.10	952.60 ↑	good ↔
countARFactual #12	Taiwan	2014	Typica	Washed / Wet	0.10	965.36 ↑	good ↔

**Fig. 3.** Exemplary countARFactuals for an instance of *bad* coffee quality. Arrows indicate changes in comparison to  $x^*$ , i.e., a feature’s value increase ↑, decrease ↓ or change in category ↔.

The generation of counterfactual explanations in this example is performed using Algorithm 2 detailed in Sect. 4.2. In Fig. 3, we present a set of the generated countARFactual explanations for an instance of coffee beans belonging to the *bad* class that originate from Taiwan. Figure 3 illustrates that countARFactuals yield plausible counterfactual explanations. For instance, for countARFactual #3, both the country of origin is changed from Taiwan to Colombia and the variety from Typica to Caturra. This seems reasonable because Typica was grown in only few Colombian instances in the training dataset, and instead, Caturra was the most frequently grown variety in Colombia. Further, the altitude at which the beans are grown is elevated only a little within Taiwanese countARFactuals (# 4 - 12), but more drastically for countries that – given the data – grow coffee on higher altitudes on average, such as Mexico (# 1) and Colombia (# 3).

## 7 Discussion

In this paper we show that adversarial random forests (ARF) can be modified to generate plausible counterfactuals, both as a subroutine to multi-objective counterfactual explanations (MOC) and as a standalone approach. Our experiments in Sect. 5 demonstrate that ARF can improve the plausibility of counterfactuals and the efficiency in their generation without substantially sacrificing other desiderata such as proximity and sparsity. In contrast to other generative modeling approaches for plausible counterfactuals, ARF handles mixed tabular data directly without, e.g., one-hot-encoding categorical features, thereby improving data-efficiency. Moreover, ARF-based counterfactual generation allows for sparsity via conditional sampling and is an off-the-shelf methodology that requires minimal efforts in tuning and computational resources.

Our work faces some limitations. For example, we define the plausibility of counterfactuals via the joint density. However, as highlighted by Keane et al. [29], there are different conceptualizations of plausibility, for example, based on the feasibility of actions or user perceived plausibility [27, 32, 40]. One might even question if staying in the manifold is always desirable, e.g., if changing the class requires extrapolation so should our counterfactuals. It should be noted that

plausible counterfactuals, in general, cannot be interpreted as action recommendations. Although they provide hints about which alternative feature values would yield acceptance by the predictor, they do not guide the user on which interventions yield the desired change in the real world. To guide action, causal knowledge is required [28]. Furthermore, in the context of recourse, *improvement* of the underlying target is more desirable than *acceptance* by a specific predictor, which counterfactual explanations do not target [32].

Proximity and plausibility are conflicting objectives [7, 11]. Oftentimes, there is only little data close to the decision boundary, and jumping just over the boundary can lead to implausible counterfactuals [16]. A trade-off between the two objectives is desirable, which we implicitly address by generating a Pareto-optimal set of diverse counterfactuals. Considering the low-density problem in Fig. 1a, the estimation of the joint density with FORDE and consideration of the empirical coverage when generating data with FORGE can circumvent low density regions, however, proximity is not directly taken into account. In future work, one could already incorporate such trade-offs in the counterfactual generation, e.g., by a parameter that directly controls for the proximity-plausibility trade-off. One option would be to set a threshold for plausibility instead of a trade-off parameter, as suggested by Brughmans et al. [7].

Like all works on counterfactual explanations, we face the Rashomon effect: There exist many plausible counterfactuals that explain the same datapoint. This raises the question of which one we should show to the user [29, 47]. As a bottom line, we return only the Pareto-optimal set of counterfactuals, which at least guarantees that no strictly dominated option is shown. In future work, integrating user preferences or considering additional objectives may improve the final selection.

Our framework is tailored to mixed tabular data settings. For other data modalities like image or text data, we advise for using neural network based approaches for density estimation and generative modeling such as VAEs and GANs. Finally, our framework is designed for binary classification and regression but can be extended to multi-class classification.

In future work, we plan to investigate the role of the ML model in the ARF approach to counterfactuals. We could also generate counterfactuals with ARF without the model by directly training ARF on  $Y$  rather than the predictions  $\hat{Y}$ . We would then get plausible counterfactuals that hint towards improvement instead of acceptance [32]. While such counterfactuals appear different from those discussed in the XAI literature so far, in fact, they essentially just turn the generative model that conditions on  $\mathbf{X} = \mathbf{x}$  into a prediction algorithm.

**Acknowledgements.** MNW and KB were supported by the German Research Foundation (DFG), Grant Number 437611051. MNW was supported by the German Research Foundation (DFG), Grant Number 459360854. KB was supported by a PhD grant of the Minds, Media, Machines Integrated Graduate School Bremen. MNW and JK were supported by the U Bremen Research Alliance/AI Center for Health Care, financially supported by the Federal State of Bremen. GK and TF were supported by the German Research Foundation through the Cluster of Excellence “Machine Learning - New Perspectives for Science” (EXC 2064/1 number 390727645). TF was supported by the Carl Zeiss Stiftung.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## A Algorithm 1: Integrating ARF into MOC

The following pseudocode is based on Algorithm 1 in [10]. Blue lines highlight the steps that differ from the original MOC algorithm proposed by [11].

---

### Algorithm 1. MOC with ARF-based Sampler and Evaluation

---

**Inputs:**

Datapoint to explain prediction for  $\mathbf{x}^* \in \mathcal{X}$

Desired outcome (range)  $Y_{des}$

Prediction function  $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$

Observed data  $D$

ARF  $\hat{g}^*$  trained on  $(\mathbf{x}_i, \hat{f}(\mathbf{x}_i))_{i=1}^n$  with  $\mathbf{x}_i \in D$

MOC hyperparameters: Number of generations  $n_{generations}$ , size of population  $\mu$ , recombination and mutation methods including probabilities, selection method for features in the conditioning set and initialization method, stopping criterion

(Additional user inputs, e.g., range of numerical features, immutable features, distance function, see [11])

- 1: Initialize population  $P_0$  with  $|P_0| = \mu$  (ICE-curve-based, see [11])
  - 2: Evaluate candidates according to four objectives:
    - Validity ( $L_1$ )
    - Sparsity ( $L_0$ )
    - Proximity (Gower distance)
    - Plausibility (ARF-based likelihood transformed with  $e^{-x}$ )
  - 3: Set  $t = 0$
  - 4: **for**  $r \in \{1, \dots, n_{iterations}\}$
  - 5:  $C_t = \text{create\_offspring}(P_t)$ ,  $|C_t| = \mu$  with given probabilities
    1. Select best candidates (acc. to validity objective)
    2. Recombine these pairwise
    3. Mutate values jointly using  $\hat{g}^*$ : generate new datapoints with FORGE
  - 6: Combine parents and offspring  $R_t = C_t \cup P_t$
  - 7: Assign candidates to a front according to their objective values:  
 $(F_1, F_2, \dots, F_m) = \text{nondominated\_sorting}(R_t)$
  - 8: **for**  $i = 1, \dots, m$
  - 9: Sort candidates acc. to diversity (objective and feature space):  
 $\tilde{F}_i = \text{crowding\_distance\_sort}(F_i)$
  - 10: **end for**
  - 11: Set  $P_{t+1} = \emptyset$  and  $i = 1$
  - 12: **while**  $|P_{t+1}| + |\tilde{F}_i| \leq \mu$
  - 13:  $P_{t+1} = P_{t+1} \cup \tilde{F}_i$
  - 14:  $i = i + 1$
  - 15: **end while**
  - 16: Choose first  $\mu - |P_{t+1}|$  elements of  $\tilde{F}_i$ :  $P_{t+1} = P_{t+1} \cup \tilde{F}_i[1 : (\mu - |P_{t+1}|)]$
  - 17:  $t = t + 1$
  - 18: **end for**
  - 19: Return unique, nondominated (w.r.t.  $o_{prox}$ ,  $o_{sparse}$  and  $o_{plaus}^*$ ) candidates of  
 $\bigcup_{k=0}^t P_k \setminus \mathbf{x}^*$  with  $\hat{f}(\mathbf{x}_{CF}) \in Y_{des}$
-

## B Algorithm 2: ARF Is All You Need

---

### Algorithm 2. ARF-based Counterfactual Generator

---

**Inputs:**

 Datapoint to explain prediction for  $\mathbf{x}^* \in \mathcal{X}$ 

 Desired outcome (range)  $Y_{des}$ 

 Prediction function  $\hat{f} : \mathcal{X} \rightarrow \mathbb{R}$ 

 Observed data  $D$ 

 ARF  $\hat{g}^*$  trained on data  $(\mathbf{x}_i, \hat{f}(\mathbf{x}_i))_{i=1}^n$  with  $\mathbf{x}_i \in D$ 

 Maximum number of feature changes  $m_{max}$ 

 Number of iterations  $n_{iterations}$ 

 Number of samples generated in each iteration  $n_{synth}$ 

(Additional user inputs, e.g., immutable features)

- 1: Derive local importances  $(f_j)_{j=1}^p$  for each feature  $j \in \{1, \dots, p\}$  (ICE-curve-based, see [11])
  - 2: **for**  $r \in \{1, \dots, n_{iterations}\}$
  - 3:      $m \leftarrow \text{sample}(1, \dots, m_{max})$
  - 4:     Select set  $C \subset \{1, \dots, p\}$  by randomly sampling  $m$  features with probability proportional to how *unimportant* feature is
  - 5:      $CF \leftarrow$  sample  $n_{synth}$  observations with FORGE derived from  $\hat{g}^*$  under condition that  $\forall j \in C : X_j = x_j^* \ \& \ \hat{Y} \in Y_{des}$
  - 6:  $\mathbf{X}_{CF} \leftarrow (\mathbf{X}_{CF}, CF)$
  - 7: **end for**
  - 8: Return unique, nondominated (w.r.t.  $o_{prox}$ ,  $o_{sparse}$  and  $o_{plaus}^*$ ) candidates  $\mathbf{x}_{CF} \in \mathbf{X}_{CF}$  with  $\hat{f}(\mathbf{x}_{CF}) \in Y_{des}$
- 

## C Synthetic Data

As follows, we describe the three illustrative datasets as well as the sampling of the randomly generated data-generating processes. An general overview on the datasets is given in Table 1. The code that was used to generate the datasets and pair plots visualizing their distribution can be found in the repository accompanying the paper (<https://github.com/bips-hb/countARFactuals>).<sup>6</sup>

---

<sup>6</sup> For an explanation of how to run the code, we refer to `python/README.md`. The visualization can be found in the folder `python/visualizations/`.

**Table 1.** Overview of synthetic datasets.

Name	No. continuous	No. binary
cassini	2	0
two_sines	2	0
pawelzyk	2	0
bn_5	3	1
bn_10	7	2
bn_20	15	4

### C.1 Illustrative Datasets

*Cassini.* The DGP, inspired by [14], is defined as follows:

$$\begin{aligned}
 Y &\sim Y_1 + Y_1 Y_2 \quad \text{with} \quad Y_1 \sim \text{Bern}(2/3), Y_2 \sim \text{Bern}(0.5) \\
 X_1 | Y_1 &\sim \begin{cases} N(0, 0.2), & \text{if } Y_1 = 0 \\ N(0, 0.5), & \text{otherwise} \end{cases} \\
 X_2 | X_1, Y_1, Y_2 &\sim \begin{cases} N(0, 0.2) & \text{if } Y_1 = 0 \\ N((-1)^{Y_2} \cos(X_1), 0.2) & \text{otherwise} \end{cases}
 \end{aligned}$$

*Two Sines.* The DGP, inspired by the two moons dataset, is specified as:

$$\begin{aligned}
 Y &\sim \text{Bern}(0.5), \\
 X_1 | Y &\sim N(Y, 3.0), \\
 X_2 | Y, X_1 &\sim N(\sin(X_1) - 2Y + 1, 0.3)
 \end{aligned}$$

*Pawelczyk.* The DGP, taken from [39], is defined as:

$$\begin{aligned}
 L &\sim \text{Cat}\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \\
 X | \mu &\sim N(\mu, I_2), \quad \mu | L = \begin{cases} (-10, 5)^T & \text{if } L = 0 \\ (0, 5)^T & \text{if } L = 1 \\ (0, 0)^T & \text{otherwise} \end{cases} \\
 Y(X) &:= X_2 > 6
 \end{aligned}$$

Here  $I_2$  refers to the  $2 \times 2$  identity matrix and  $\text{Cat}(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$  to the uniform categorical distribution with values 0, 1, 2.

### C.2 Randomly Generated DGPs

For the generation of **bn\_5**, **bn\_10**, and **bn\_20**, we randomly sample Bayesian networks with categorical and continuous distributions as well as linear and nonlinear relationships.

1. First, we randomly sample a Directed Acyclic Graph (DAG) using the `networkx` package. We select  $Y$  as the root node. To make sure that  $Y$  is related to many of the features, for each node that is not directly neighboring  $Y$ , a directed edge is added with probability 0.5 (directed such that the graph remains acyclic).
2. From all nodes, 20% are randomly selected to be categorical nodes;  $Y$  is always selected to be a categorical node.
3. For every node  $j$ , an aggregation function  $g$  is sampled that maps the parent values  $x_{pa(j)}$  to an aggregate, which then parameterizes the distribution of the respective node.

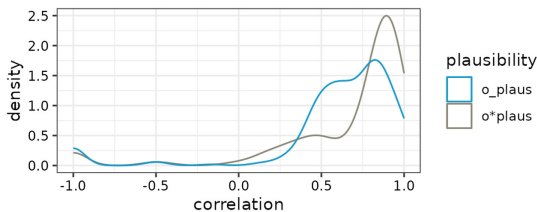
$$g(x) = \beta + \beta_1 h(x) + \beta_2 h(x)^2 \quad \text{with} \quad h(x) = \sin \left( \sum_{i \in pa(j)} w_i x_i \right) \quad (11)$$

The weights  $w$  are sampled from  $Unif(-1, 1) + 3Bern(3/d)$ . To make it more likely that  $Y$  can be predicted well from its covariates, weights concerning  $Y$  are increased by  $d \sim Unif(3, 4)$  with probability 0.1. The weight vector  $w$  is normalized. The coefficients  $\beta$  are sampled from  $Unif(-1, 1)$ .

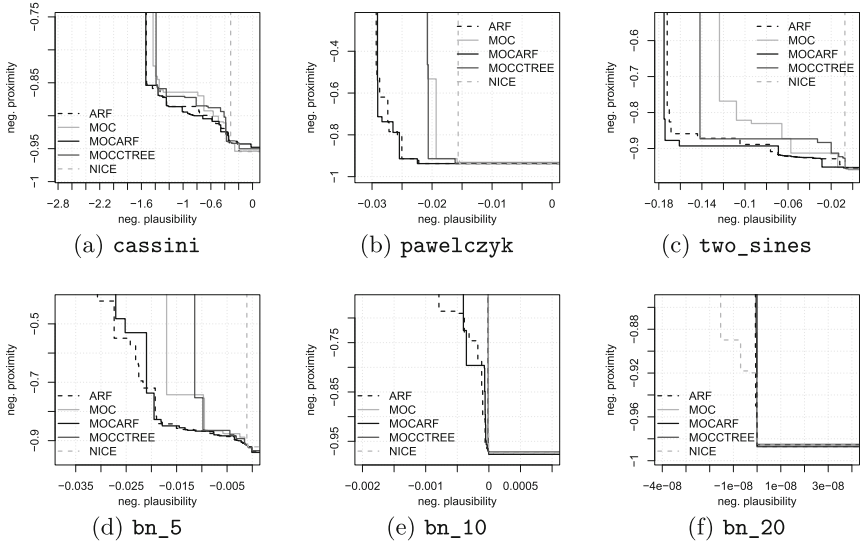
4. If the feature is categorical, the respective Bernoulli is parameterized with the sigmoid of the aggregate of the parents  $Bern(\zeta(g(x)))$ . Continuous features follow  $N(\mu, \sigma)$  with  $\mu \sim N(g(x), 1)$  and  $\sigma \sim N(0, 2)$ .

To ensure that a prediction model fitted on the data can discriminate between the classes and that changing the prediction to the desirable class is feasible, we randomly generated datasets until we found one with balanced labels ( $0.4 < E[Y] < 0.6$ ), and where a `xgboost` model demonstrated good accuracy ( $> 0.95$ ) and balanced predictions ( $0.3 < E[\hat{Y}] < 0.7$ ).

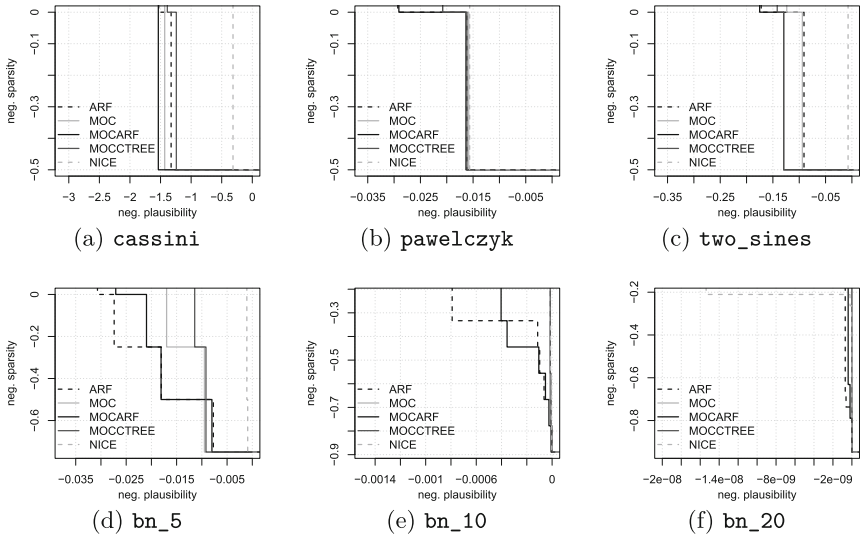
## D Additional Empirical Results



**Fig. 4.** Density plots of correlations of true plausibility and  $o_{\text{plaus}}$  or  $o^*_{\text{plaus}}$ .



**Fig. 5.** Median empirical attainment function [33] for the negative plausibility and negative proximity. Lower values are better.



**Fig. 6.** Median empirical attainment function [33] for the negative plausibility and negative sparsity. Lower values are better.



## References

1. Coffee Quality Institute. <https://www.coffeeinstitute.org/>. Accessed 12 Mar 2024
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018)
3. Artelt, A., Hammer, B.: Convex density constraints for computing plausible counterfactual explanations. In: Farkaš, I., Masulli, P., Wermter, S. (eds.) *Artificial Neural Networks and Machine Learning–ICANN 2020: 29th International Conference on Artificial Neural Networks*, Bratislava, Slovakia, 15–18 September 2020, Proceedings, Part I 29, pp. 353–365. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-61609-0\\_28](https://doi.org/10.1007/978-3-030-61609-0_28)
4. Bond-Taylor, S., Leach, A., Long, Y., Willcocks, C.G.: Deep generative modelling: a comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(11), 7327–7347 (2021)
5. Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G.: Deep neural networks and tabular data: a survey. *IEEE Trans. Neural Netw. Learn. Syst.* (2022)
6. Breiman, L.: Statistical modeling: the two cultures (with comments and a rejoinder by the author). *Stat. Sci.* **16**(3), 199–231 (2001). <https://doi.org/10.1214/ss/1009213726>
7. Brughmans, D., Leyman, P., Martens, D.: NICE: an algorithm for nearest instance counterfactual explanations. *Data Min. Knowl. Disc.* 1–39 (2023)
8. Burrell, J.: How the machine ‘thinks’: understanding opacity in machine learning algorithms. *Big Data Soc.* **3**(1), 2053951715622512 (2016)
9. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 785–794. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2939672.2939785>
10. Dandl, S., Hofheinz, A., Binder, M., Bischl, B., Casalicchio, G.: Counterfactuals: an R package for counterfactual explanation methods. *arXiv preprint arXiv:2304.06569* (2023)
11. Dandl, S., Molnar, C., Binder, M., Bischl, B.: Multi-objective counterfactual explanations. In: Bäck, T., et al. (eds.) *PPSN 2020. LNCS*, vol. 12269, pp. 448–469. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58112-1\\_31](https://doi.org/10.1007/978-3-030-58112-1_31)
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
13. Del Ser, J., Barredo-Arrieta, A., Díaz-Rodríguez, N., Herrera, F., Saranti, A., Holzinger, A.: On generating trustworthy counterfactual explanations. *Inf. Sci.* **655**, 119898 (2024). <https://doi.org/10.1016/j.ins.2023.119898>
14. Dimitriadou, F.L.E.: *MLBench: machine learning benchmark problems* (2021). <https://CRAN.R-project.org/package=mlbench>. R package version 2.1-3.1
15. Foster, D.: *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*, 2nd edn. O’Reilly Media, Inc., Upper Saddle River (2022)
16. Freiesleben, T.: The intriguing relation between counterfactual explanations and adversarial examples. *Mind. Mach.* **32**(1), 77–109 (2022)
17. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* **24**(1), 44–65 (2015). <https://doi.org/10.1080/10618600.2014.907095>

18. Goodfellow, I., et al.: Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, vol. 27 (2014)
19. Gower, J.C.: A general coefficient of similarity and some of its properties. *Biometrics* **27**(4), 857–871 (1971). <https://doi.org/10.2307/2528823>
20. Goyal, Y., Wu, Z., Ernst, J., Batra, D., Parikh, D., Lee, S.: Counterfactual visual explanations. In: *International Conference on Machine Learning*, pp. 2376–2384. PMLR (2019)
21. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: *Advances in Neural Information Processing Systems*, vol. 35, pp. 507–520 (2022)
22. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. *Data Min. Knowl. Disc.* 1–55 (2022)
23. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851 (2020)
24. Hothorn, T., Zeileis, A.: Predictive distribution modeling using transformation forests. *J. Comput. Graph. Stat.* **30**(4), 1181–1196 (2021). <https://doi.org/10.1080/10618600.2021.1872581>
25. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint [arXiv:1907.09615](https://arxiv.org/abs/1907.09615) (2019)
26. Kanamori, K., Takagi, T., Kobayashi, K., Arimura, H.: DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In: *IJCAI*, pp. 2855–2862 (2020)
27. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Comput. Surv.* **55**(5), 1–29 (2022)
28. Karimi, A.H., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 353–362 (2021)
29. Keane, M.T., Kenny, E.M., Delaney, E., Smyth, B.: If only we had better counterfactual explanations: five key deficits to rectify in the evaluation of counterfactual XAI techniques. arXiv preprint [arXiv:2103.01035](https://arxiv.org/abs/2103.01035) (2021)
30. Keane, M.T., Smyth, B.: Good counterfactuals and where to find them: a case-based technique for generating counterfactuals for explainable AI (XAI). In: Watson, I., Weber, R. (eds.) *Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, 8–12 June 2020, Proceedings 28*. pp. 163–178. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-58342-2\\_11](https://doi.org/10.1007/978-3-030-58342-2_11)
31. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: *Proceedings of the 2nd International Conference on Learning Representations* (2014). <https://hdl.handle.net/11245/1.434281>
32. König, G., Freiesleben, T., Grosse-Wentrup, M.: Improvement-focused causal recourse (ICR). In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 11847–11855 (2023)
33. López-Ibáñez, M., Paquete, L., Stützle, T.: Exploratory analysis of stochastic local search algorithms in biobjective optimization. In: Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M. (eds.) *Experimental Methods for the Analysis of Optimization Algorithms*, pp. 209–222. Springer, Cham (2010). [https://doi.org/10.1007/978-3-642-02538-9\\_9](https://doi.org/10.1007/978-3-642-02538-9_9)

34. Lyons, H., Velloso, E., Miller, T.: Conceptualising contestability: perspectives on contesting algorithmic decisions. *Proc. ACM Hum.-Comput. Interact.* **5**(CSCW1), 1–25 (2021)
35. Mahajan, D., Tan, C., Sharma, A.: Preserving causal constraints in counterfactual explanations for machine learning classifiers. arXiv preprint [arXiv:1912.03277](https://arxiv.org/abs/1912.03277) (2019)
36. Mittelstadt, B., Russell, C., Wachter, S.: Explaining explanations in AI. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 279–288. ACM (2019). <https://doi.org/10.1145/3287560.3287574>
37. Molnar, C.: *Interpretable Machine Learning*. 2nd edn. (2022). <https://christophm.github.io/interpretable-ml-book>
38. Nemirovsky, D., Thiebaut, N., Xu, Y., Gupta, A.: CounteRGAN: generating counterfactuals for real-time recourse and interpretability using residual GANs. In: *Uncertainty in Artificial Intelligence*, pp. 1488–1497. PMLR (2022)
39. Pawelczyk, M., Broelemann, K., Kasneci, G.: Learning model-agnostic counterfactual explanations for tabular data. In: *Proceedings of the Web Conference 2020*, pp. 3126–3132 (2020)
40. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: FACE: feasible and actionable counterfactual explanations. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 344–350 (2020)
41. Rezende, D., Mohamed, S.: Variational inference with normalizing flows. In: *Proceedings of the 32th International Conference on Machine Learning*, pp. 1530–1538. PMLR (2015)
42. Shi, T., Horvath, S.: Unsupervised learning with random forest predictors. *J. Comput. Graph. Stat.* **15**(1), 118–138 (2006)
43. Shwartz-Ziv, R., Armon, A.: Tabular data: deep learning is not all you need. *Inf. Fusion* **81**, 84–90 (2022)
44. Stepin, I., Alonso, J.M., Catala, A., Pereira-Fariña, M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access* **9**, 11974–12001 (2021)
45. Van Looveren, A., Klaise, J., Vacanti, G., Cobb, O.: Conditional generative models for counterfactual explanations. arXiv preprint [arXiv:2101.10123](https://arxiv.org/abs/2101.10123) (2021)
46. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
47. Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review. arXiv preprint [arXiv:2010.10596](https://arxiv.org/abs/2010.10596) (2020)
48. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL & Tech.* **31**, 841 (2017)
49. Watson, D.S., Blesch, K., Kapar, J., Wright, M.N.: Adversarial random forests for density estimation and generative modeling. In: *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, pp. 5357–5375. PMLR (2023)
50. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms—a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) *PPSN 1998*. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0056872>

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





# Causality-Aware Local Interpretable Model-Agnostic Explanations

Martina Cinquini<sup>(✉)</sup>  and Riccardo Guidotti<sup></sup>

University of Pisa, Pisa, Italy

`martina.cinquini@phd.unipi.it`, `riccardo.guidotti@unipi.it`

**Abstract.** A main drawback of eXplainable Artificial Intelligence (XAI) approaches is the feature independence assumption, hindering the study of potential variable dependencies. This leads to approximating black box behaviors by analyzing the effects on randomly generated feature values that may rarely occur in the original samples. This paper addresses this issue by integrating causal knowledge in an XAI method to enhance transparency and enable users to assess the quality of the generated explanations. Specifically, we propose a novel extension to a widely used local and model-agnostic explainer, which encodes explicit causal relationships within the data surrounding the instance being explained. Extensive experiments show that our approach overcomes the original method in terms of faithfully replicating the black-box model's mechanism and the consistency and reliability of the generated explanations.

**Keywords:** Causal Discovery · Interpretable Machine Learning · Causal Explanations · Post-hoc Explainability

## 1 Introduction

In past decades, the growing availability of data and computational power have allowed the development of sophisticated Machine Learning (ML) models that have provided a significant contribution to the progress of Artificial Intelligence (AI) in many real-world applications [29]. Despite their success, the surge in performance has often been achieved through increased model complexity, turning such approaches into black-box systems. The lack of a clear interpretation of the internal structure of ML models embodies a crucial weakness since it causes uncertainty regarding the way they operate and, ultimately, how they infer outcomes [21]. Indeed, opaque approaches may inadvertently obfuscate responsibility for any biases or produce wrong decisions learned from spurious correlations in training data [12]. To mitigate these risks, the scientific community's interest in eXplainable Artificial Intelligence (XAI) has been increasingly emerging in the past decade. According to the classification of XAI methodologies [12], we focus on post-hoc explainability. Given an AI decision system based on a black-box classifier and an instance to explain, post-hoc local explanation methods approximate the black-box behavior by learning an interpretable model in a

synthetic neighborhood of the instance under analysis generated randomly. However, particular combinations of feature values might be unrealistic, leading to implausible synthetic instances [2, 20]. This weakness emerges since these methods do not consider the local distribution features, the density of the class labels in the neighborhood [25], and, most importantly, the causal relationships among input features [22]. Such inability to disentangle correlation from causation can deliver sub-optimal or even erroneous explanations to decision-makers [31]. Moreover, causation is ubiquitous in Humans’ conception of their environment [6]. Human beings are extremely good at constructing mental decision models from a few data samples because people excel at generalizing data and thinking in a cause/effect manner. Consequently, integrating *causal knowledge* in XAI should be emphasized to attain a higher degree of interpretability and prevent procedures from failing in unexpected circumstances [3]. To this aim, Causal Discovery (CD) methods can map all the causal pathways to a variable and infer how different variables are related. Even partial knowledge of the causal structure of observational data could improve understanding of which input features black-box models have used to make their predictions, allowing for a higher degree of interpretability and more robust explanations.

In this paper, we propose CALIME for Causality-Aware Local Interpretable Model-Agnostic Explanations. The method extends LIME [30] by accounting for underlying causal relationships present in the data used by the black-box. To attain this purpose, we replace the synthetic data generation performed by LIME through random sampling with GENCDA [7], a synthetic dataset generator for tabular data that explicitly allows for encoding causal dependencies. We would like to emphasize that causal relationships are not typically known a priori, but we adopt a CD approach to discover nonlinear causalities among the variables and use them at generation time. Thus, the novelty of CALIME lies in the *explicit encoding of causal relationships* in the local synthetic neighborhood of the instance for which the explanation is required. We highlight that our proposal of the causality-aware explanation method can be easily adapted to extend and improve other model-agnostic explainers such as LORE [10], or SHAP [23]. Our experiments show that CALIME significantly improves over LIME both for the stability of explanations and the fidelity in mimicking the black-box.

The rest of this paper is organized as follows. Section 2 describes the state-of-the-art related to XAI and causality. Section 3 presents a formalization of the problem addressed and recalls basic concepts for understanding our proposal, which is detailed in Sect. 4. The experimental results are presented in Sect. 5, while the conclusions, as well as future works, are discussed in Sect. 6.

## 2 Related Works

LIME [30] is a model-agnostic method that returns local explanations as feature importance vectors. Further details are in Sect. 3 as it is the starting point for our proposal. While LIME stands out for its simplicity and effectiveness, it has several weak points. A significant limitation is its reliance on converting all data

into a binary format, which is assumed to be more interpretable for humans [9]. Additionally, its use of random perturbations to generate explanations can lead to inconsistencies, possibly producing varying explanations for the same input and prediction across different runs [35]. Furthermore, the reliability of additive explanations comes into question, especially when noisy interactions are introduced in the reference dataset used for generating explanations [8].

Over recent years, numerous researchers have analyzed such limitations and proposed several works to extend or improve them. For instance, KLIME [14] runs the K-Means clustering algorithm to partition the training data and then fit local models within each cluster instead of perturbation-based data generation around an instance being explained. In [16] is proposed LIME-SUP that approximates the original LIME better than KLIME by using supervised partitioning. Furthermore, KL-LIME [28] adopts the Kullback-Leibler divergence to explain Bayesian predictive models. Within this constraint, both the original task and the explanation model can be arbitrarily changed without losing the theoretical information interpretation of the projection for finding the explanation model. ALIME [33] presented modifications using an autoencoder as a better weighting function for the local surrogate models. In QLIME [4], the authors consider nonlinear relationships using a quadratic approximation. Another approach proposed in [32] utilizes a Conditional Tabular Generative Adversarial Network (CTGAN) to generate more realistic synthetic data for querying the model to be explained. Theoretically, GAN-like methods can learn possible dependencies. However, as empirically demonstrated in [7], these relationships are not directly represented, and there is no guarantee that they are followed in the data generation process. In [35] is proposed DLIME, a Deterministic Local Interpretable Model-Agnostic Explanations where random perturbations are replaced with hierarchical clustering to group the data. After that, a kNN is used to select the cluster where the instance to be explained belongs. Finally, in [36] is presented BAY-LIME, a Bayesian local surrogate model that exploits prior knowledge and Bayesian reasoning to improve both the consistency in repeated explanations of a single prediction and the robustness of kernel settings.

Despite considerable progress in LIME variants, a significant limitation remains unaddressed in the state-of-the-art approaches: they do not explicitly account for causal relationships. This gap represents a fundamental shortfall. Nonetheless, employing a causal framework for deriving explanations could lead to a more reliable and solid explanatory process [3]. Our research demonstrates that local surrogate models exhibit increased fidelity when trained within synthetic environments, taking causality into account.

XAI methods integrating causal knowledge are a recently challenging research area [26]. Among them, one of the most popular methods for Counterfactual Explanation (CE) shows how the prediction result changes with small perturbations to the input [34]. In [19], a focus is presented on the role of causality towards feasible counterfactual explanations requiring complete knowledge of the causal model. Another CE approach called Ordered Counterfactual Explanation is in [18]. It works under the assumption of linear causal relationships

and provides an optimal perturbation vector and the order of the features to be perturbed, respecting causal relationships.

Even though the explainers mentioned above account for causal relationships, to the best of our knowledge, no state-of-the-art XAI techniques incorporate any causal knowledge during the explanation extraction. Some indirectly account for causality through a latent representation or adopt a known causal graph, typically as a post-hoc filtering step [24]. Others consider causal relations between the input features and the outcome label but are not directly interested in the interactions among input features [17].

Hence, our method represents a novel post-hoc local explanation strategy by embedding causal connections into the explanation generation process. In particular, it does not necessitate pre-existing causal information but independently identifies causal relationships as part of the explanation extraction phase.

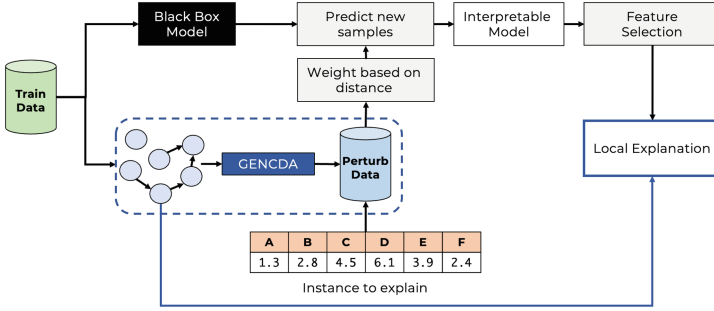
### 3 Background

This paper addresses the *black-box outcome explanation problem* [12]. A classifier  $b$  is *black-box* if its internal mechanisms are either hidden from the observer or, even if known, they remain incomprehensible to humans. Given  $b$  and an instance  $x$  classified by  $b$ , i.e.,  $b(x) = y$ , the black-box outcome explanation problem aims at explaining  $e$  belonging to a human-interpretable domain. In our work, we focus on feature importance modeling the explanation as a vector  $e = \{e_1, e_2, \dots, e_m\}$ , in which the value  $e_i \in e$  is the importance of the  $i^{\text{th}}$  feature for the decision made by  $b(x)$ . To understand the contribution of each feature, the sign and the magnitude of  $e_i$  are considered: if  $e_i < 0$ , the feature contributes negatively to the outcome  $y$ ; otherwise, the feature contributes positively. The magnitude represents how significant the feature’s contribution is to the prediction.

**LIME.** The main idea of LIME [30] is that the explanation may be derived locally from records generated randomly in the synthetic neighborhood  $Z$  of the instance  $x$  to be explained. LIME randomly draws samples and weights them w.r.t. a certain distance function  $\pi$  to capture the proximity with  $x$ . Then, a perturbed sample of instances  $Z$  is used to feed to the black-box  $b$  and obtain the classification probabilities  $b_p(Z)$  w.r.t. the class  $b(x) = y$ . Such  $b_p(Z)$  are combined with the weights  $W$  to train a linear regressor with Lasso regularization considering the top  $k$  most essential features. The coefficients of the linear regressor are returned as explanation  $e$ . A crucial weakness of LIME is the *randomness* of perturbations around the instance to be explained. To address this problem, we employ a data generation process that provides more realistic data respecting causal relationships.

**GENCDA.** In [7] is presented GENCDA, a synthetic data generator for tabular data that explicitly allows for encoding the causal structure among variables using NCDA, a boosted version of NCD [15]. Assuming there is no confounding, no selection bias, and no feedback among variables, NCD recovers the causal graph  $\mathcal{G}$  from the observational distribution by exploring a functional causal model in





**Fig. 1.** The high-level workflow of our framework, with the blue dashed line showing the difference in generating the synthetic neighborhood compared to LIME (Color figure online).

which effects are modeled as nonlinear functions of their causes and where the influence of the noise is restricted to be additive. GENCDA takes as input the *real* dataset  $X$  that has to be extended with *synthetic* data, the DAG  $\mathcal{G}$  extracted from  $X$  by NCDA and a set of distributions. It generates a synthetic dataset with variable dependencies that adhere to the causal relationships modeled in  $\mathcal{G}$ .

## 4 Causality-Aware LIME

This section describes our method for Causality-Aware Local Interpretable Model-agnostic Explanations (CALIME). Before introducing it, we provide a practical example to demonstrate the potential limitations of generating explanations without considering causal relations. Consider a dataset  $X$  that describes a bank’s customers requesting loans. This dataset includes features such as *age*, *education level*, *income*, and *number of weekly working hours*. Let  $b$  be the AI system based on a black-box adopted by the bank and used to grant loans to customers  $x \in X$ . Consider  $x = \{(age, 24), (edu\_level, high-school-5), (income, 800), (work\_hours, 20)\}$  a customer who was denied the loan. To offer additional explanations for the loan denial, the bank decides to apply the LIME algorithm to the instance  $x$ . A possible explanation for  $b(x)$  could be  $e = \{(age, 0.3), (edu\_level, 0.9), (income, 0.2), (work\_hours, 0)\}$ . Thus, according to  $e$ , it appears that the primary factor contributing to the loan denial is the low *education level*. By examining the neighborhood  $Z$  generated by LIME, we may come across several synthetic instances like  $z = \{(age, 24), (edu\_level, phd-8), (income, 900), (work\_hours, 20)\}$  where a higher education level is observed. If we had known the causal relationships among the customers in the training set of the black-box, we would have discovered that there is a relationship between *age* and *education level*, i.e.,  $age \rightarrow edu\_level$ , and that synthetic instances like  $z$  are implausible because they do not respect such relationship.

Therefore, the key idea of CALIME is to locally explain the predictions of any black-box model while considering the underlying causal relationships within the

**Algorithm 1:** CALIME( $x, b, X, k, N$ )

```

Input :  $x$  - instance to explain,  $b$  - classifier,  $X$  - reference dataset,  $k$  - nbr of
         features,  $N$  - nbr of samples
Output:  $e$  - features importance
1  $Z \leftarrow \emptyset, W \leftarrow \emptyset;$  // init. empty synth data and weights
2  $G \leftarrow \text{NCDA}(X);$  // extract DAG modeling causal relationships
3  $I \leftarrow \text{fit\_distributions}(G, X);$  // learn root distributions
4  $R \leftarrow \text{fit\_regressors}(G, X);$  // learn regressors
5 foreach  $i \in [1, \dots, N]$  do
6    $z \leftarrow \text{GENCDA\_sampling}(x, G, I, R);$  // causal permutations
7    $Z \leftarrow Z \cup \{z\};$  // add synthetic instance
8    $W \leftarrow W \cup \{\exp(-\frac{\pi(x, z)^2}{\sigma^2})\};$  // add weights
9  $e \leftarrow \text{solve\_Lasso}(Z, b_p(Z), W, k);$  // get coefficients
10 return  $e;$ 

```

data. The novelty of our proposal is actualized in *the generation of the neighborhood* around the instance to explain. Indeed, instead of random perturbation, CALIME uses GENCDA as a synthetic dataset generator for tabular data, which can discover nonlinear dependencies among the features and use them during the generation process. While the improvement of CALIME over LIME may seem straightforward, the ability to encode explicitly causal relationships provides a significant added value to explanations. In particular, respecting dependencies during the explanation process ensures that local explanations are based on plausible synthetic data and that the final explanations are more trustworthy and less subject to possible noise in the data. Figure 1 illustrates a high-level workflow depicting our proposal.

The pseudo-code of CALIME is reported on Algorithm 1, the main differences with LIME are highlighted in blue<sup>1</sup>. First, CALIME runs on  $X$  the CD algorithm NCDA and extracts the DAG  $G$  that describes the causal structure of  $X$  (line 2). After that, for each *root* variable  $j$  with respect to  $G$ , i.e., such that  $pa(j) = \emptyset$ , CALIME learns the best distribution that fits  $X^{(j)}$  and produces a set of distribution generators  $I$  (line 3). For each *dependent* variable  $j$  in  $G$ , i.e.,  $pa(j) \neq \emptyset$ , CALIME trains a regressor using as features  $X^{pa(j)}$  and as target  $X^{(j)}$ , and produces a set of regressor generators  $R$  (line 4). For each instance to generate, it runs GENCDA locally on  $x$ . Then, CALIME takes as input the instance  $x$ , the DAG  $G$ , the set of distribution generators for root variables  $I$ , and the set of regressor generators for dependent variables  $R$  (line 6). The GENCDA\_*sampling* randomly selects the features  $\{j_1, \dots, j_q\}$  to change among the root ones<sup>2</sup>. For each root feature, the corresponding data generator  $I_j$  generates a synthetic value  $z_j$ . After all the root features have been synthetically generated, CALIME checks the causal

<sup>1</sup> We highlight that the GENCDA algorithm does not explicitly appear in CALIME pseudo-code as it is decomposed among lines [2 – 4] and 6.

<sup>2</sup> The number of features to change  $q$  is randomly selected in  $[2, q]$ .

relationships modeled in  $G$ . For all dependent variables  $j$  such that  $pa(j)$  contains a variable that has been synthetically generated, the regressor generator  $R_j$  responsible for predicting the value of  $j$  is applied on  $z^{pa(j)}$  to generate the updated value for feature  $j$  by respecting the causal relationship captured in  $G$ .

To clarify how CALIME works, we recall the toy example presented at the beginning of this section. Through NCDA, CALIME discovers a DAG  $G$  indicating the causal relation  $age \rightarrow edu\_level$ . Therefore, it learns the best distributions to model *age*, *income*, and *work\_hours*. After that it learns a regressor  $R_{edu\_level}$  on  $\langle X^{(age)}, X^{(edu\_level)} \rangle$ . Let consider the instance to explain  $x = \{(age, 34), (edu\_level, 6.5), (income, 1000), (work\_hours, 35)\}$ . When a synthetic instance  $z$  has to be generated, then (i) *education level* can not be changed if *age* is not changed, and (ii) when *age* is also changed *education level* must be changed according to  $R_{edu\_level}$ . For instance, if  $z_{age} = 32$  then we can have  $z_{edu\_level} = R_{edu\_level} = phd-8$ . This way, the regressor will consider only synthetic customers with a higher *age* when the *education level* is higher.

## 5 Experiments

We report here the experiments carried out to validate CALIME<sup>3</sup>. First, we illustrate the datasets used, the classifiers, and the experimental setup. Then, we present the evaluation measures adopted. Lastly, we demonstrate that our proposal outperforms LIME in terms of fidelity, plausibility, and stability.

### 5.1 Datasets and Classifiers

We experimented with CALIME on multiple datasets from UCI Repository<sup>4</sup>, namely: **banknote**, **magic**, **wdbc**, **wine-red** and **statlog** which belong to diverse yet critical real-world applications. Table 1 (left) summarizes each dataset. The **banknote** dataset is a well-known benchmark data set for binary classification problems. The goal is to predict whether a banknote is authentic or fake based on the measured characteristics of digital images of each banknote. The **statlog** dataset is an image segmentation dataset where the instances were drawn randomly from 7 outdoor images. Each instance is a 3x3 region, and the images were hand-segmented to create a classification for every pixel. **wine-red** represents wines of different qualities with respect to physicochemical tests from red variants of the Portuguese Vinho Verde wine. The **wdbc** dataset is formed by features computed from a digitized image of a fine needle aspirate of a breast mass. They describe the characteristics of the cell nuclei found in the image. The **magic** dataset contains data to simulate high-energy gamma particles in a ground-based atmospheric Cherenkov telescope.

<sup>3</sup> Python code and datasets are available at <https://github.com/marti5ini/CALIME>.

<sup>4</sup> <https://archive.ics.uci.edu/ml/datasets/>.

**Table 1.** Dataset statistics and classifiers accuracy. We present the number of records ( $n$ ), the number of features ( $m$ ), the number of labels that the class can assume ( $l$ ), and the number of training records for black-box ( $X_b$ ), the number of explanations records ( $X_t$ ), and the number of records to evaluate the causal relations founded ( $X_c$ ). Additionally, we report Random Forests ( $RF$ ) and Neural Networks ( $NN$ ) accuracy.

	$n$	$m$	$l$	$ X_b $	$ X_c $	$ X_t $	$RF$	$NN$
<b>banknote</b>	1,372	4	2	890	382	100	.99	1.0
<b>statlog</b>	2,310	19	7	1,547	663	100	.98	.96
<b>magic</b>	19,020	11	2	13,244	5,676	100	.92	.85
<b>wdbc</b>	569	30	2	328	141	100	.95	.92
<b>wine-red</b>	1,159	11	6	358	203	154	.82	.70

We would like to highlight that, due to the nature of GENCDA, all these datasets are tabular datasets with instances represented as continuous features. As part of our future research direction, we plan to extend CALIME with a different CD approach capable of handling also categorical and/or discrete variables.

We split each dataset into three partitions:  $X_b$ , is the set of records to train a black-box model;  $X_c$ , is the set of records reserved for discovering the causal relationships;  $X_t$ , is the partition that contains the record to explain. We trained the following black-box models: Random Forest (RF), and Neural Network (NN) as implemented by *scikit-learn*<sup>5</sup>. For each black-box and dataset, we performed a random search for the best parameter setting<sup>6</sup>. Classification accuracy and partitioning sizes are shown in Table 1-(right).

## 5.2 Comparison with Related Works

We evaluate the effectiveness of the CALIME framework through a comparative analysis, contrasting it to several state-of-the-art proposals designed to outperform the limitations of LIME. We specifically select two baseline methods: F-LIME [32], which employs a Conditional Tabular Generative Adversarial Network to create more realistic synthetic data for model explanations, and D-LIME [35], a Deterministic Local Interpretable Model-Agnostic Explanations approach that replaces random perturbations with hierarchical clustering to group the data.

The rationale for selecting these specific methods is grounded in several key factors: i) similar to CALIME, they innovate by altering the neighborhood generation mechanism while preserving the core algorithmic structure; ii) they have been developed relatively recently; iii) the presence of accessible source code enhances reproducibility. For their implementation, we used the versions available in the respective repository<sup>7</sup>. The parameter settings followed the original paper.

<sup>5</sup> Black-boxes: <https://scikit-learn.org/>.

<sup>6</sup> Detailed information regarding the parameters used can be found in the repository.

<sup>7</sup> The F-LIME code can be accessed at <https://github.com/seansaito/Faster-LIME>, while D-LIME at [https://github.com/rehmanzafar/dlime\\_experiments](https://github.com/rehmanzafar/dlime_experiments).

### 5.3 Evaluation Measures

We evaluate the quality of the explanations returned to three criteria, i.e., *fidelity*, *plausibility*, and *stability* of the explanations.

**Fidelity.** One of the metrics most widely used in XAI to evaluate how good an explainer is at mimicking the black-box decisions is the *fidelity* [12]. There are different specializations of fidelity, depending on the type of explainer under analysis [12]. In our setting, we define the fidelity in terms of the coefficient of determination  $R^2$ :

$$R_x^2 = 1 - \frac{\sum_{i=1}^N (b(z_i) - r(z_i))^2}{\sum_{i=1}^N (b(z_i) - \bar{y})^2} \quad \text{with} \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N b(z_i)$$

where  $z_i \in Z$  is the synthetic neighborhood for a certain instance  $x$ , and  $r$  is the linear regressor with Lasso regularization trained on  $Z$ .  $R^2$  ranges in  $[0, 1]$  and a value of 1 indicates that the regression predictions perfectly fit the data.

**Plausibility.** We evaluate the plausibility of the explanations regarding the goodness of the synthetic datasets locally generated by LIME and CALIME by using the following metrics based on distance, outlierness, statistics, likelihood, and detection. In [27] is presented a set of functionalities that facilitates evaluating the quality of synthetic datasets. Our experiments exploit this framework to compare the synthetic data of the neighborhood  $Z$  with  $X_b$ .

*Average Minimum Distance Metric.* Given the local neighborhood  $Z$  generated around instance  $x$ , a synthetic instance  $z_i \in Z$  is plausible if it is not too much different from the most similar instance in the reference dataset  $X$ .

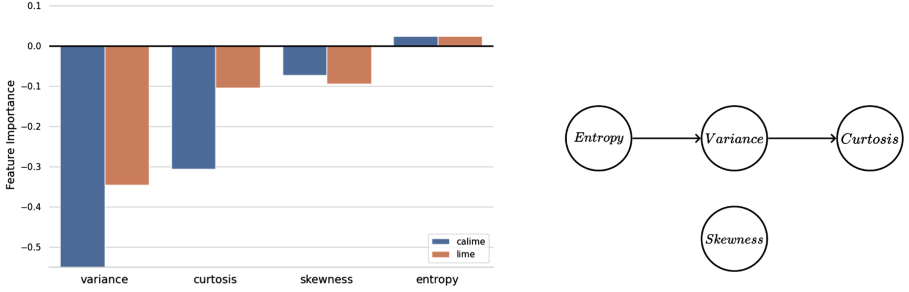
Hence, for a given explained instance  $x$ , we calculate the plausibility in terms of Average Minimum Distance (*AMD*):

$$AMD_x = \frac{1}{N} \sum_{i=1}^N d(z_i, \bar{x}) \quad \text{with} \quad \bar{x} = \arg \min_{x' \in X/\{x\}} d(z_i, x')$$

where the lower the *AMD*, the more plausible the instances in  $Z$ , the more reliable the explanation.

*Outlier Detection Metrics.* We also evaluate the plausibility of outliers in the synthetic neighborhood  $Z$ . The fewer outliers are in  $Z$  to  $X$ , the more reliable the explanation is. In particular, we estimate the number of outliers in  $Z$  by employing three outlier detection techniques<sup>8</sup>: Local Outlier Factor (LOF), Angle-Based Outlier Detection (ABOD), and Isolation Forest (IF) [5]. These three approaches return a value in  $[0, N]$ , indicating the number of outliers identified by the method. We report the Average Outlier Score (*AOS*) that combines the normalized scores of these indicators.

<sup>8</sup> LOF and IF: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors>;  
ABOD: <https://pyod.readthedocs.io>.



**Fig. 2.** (Left). Feature importance computed by CALIME and LIME. (Right). Causal graph of *banknote* inferred by a causal discovery method [15].

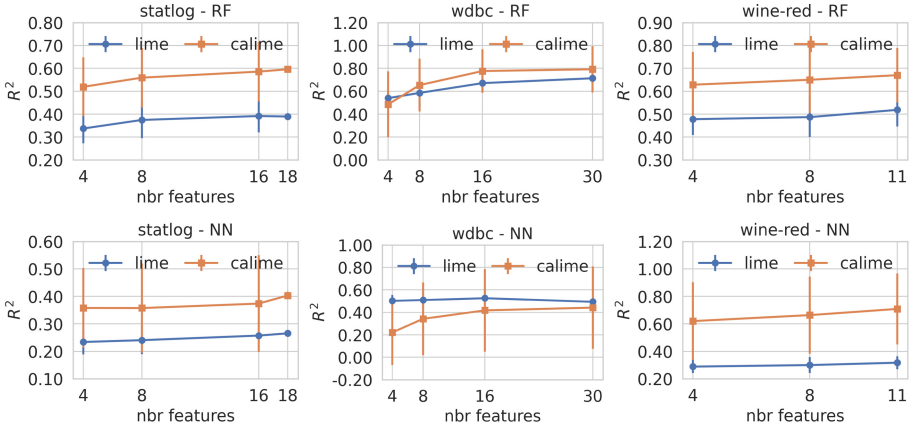
*Statistical Metrics.* To compare  $X$  with  $Z$ , we exploit the statistical measures outlined in SDV [27]. Specifically, we opted for the Gaussian Mixture Log Likelihood, (*GMLogLikelihood*), Inverted Kolmogorov-Smirnov D statistic, (*KSTest*), and Continuous Kullback-Leibler Divergence (*ContinuousKLDivergence*).

*GMLogLikelihood* fits multiple Gaussian Mixture models to the real data using different numbers of components and returns the average log-likelihood given to the synthetic data. *KSTest* performs the two-sample Kolmogorov-Smirnov test using the empirical Cumulative Distribution Function (CDF). The output for each column is one minus the KS Test D statistic, which indicates the maximum distance between the expected CDF and the observed CDF values. *ContinuousKLDivergence* is an information-based measure of disparity among probability distributions. The SDV framework approximates the KL divergence by binning the continuous values into categorical values and then computing the relative entropy. Afterward, normalize the value using  $1/(1 + D_{KL})$ . The metric is computed as the average across all the column pairs. We report the Average Statistical Metric (*ASM*) that aggregates these three indicators.

*Detection Metrics.* A way to test the plausibility of a synthetic dataset is to use a classification approach to assess how much the real data differs from synthetic ones [27]. The idea is to shuffle the real and synthetic data together, label them with flags indicating whether a specific record is real or synthetic, and cross-validate an ML classification model that tries to predict this flag [27]. The output of the metric is one minus the average Area Under the Receiver Operating Curve across all the cross-validation splits. We employ a Logistic detector and SVM as “discriminators” and report the Average Detection Metric (*ADM*).

**Stability.** To gain the users’ trust, explanation methods must guarantee stability across different explanations [13]. Indeed, the stability of explanations is a fundamental requirement for a trustworthy approach [8]. Let  $E = \{e_1, \dots, e_n\}$  be the set of explanations for the instances  $X = \{x_1, \dots, x_n\}$ . In line with [11], we asses the *stability* through the local Lipschitz estimation [1]:

$$LLE_x = \text{avg}_{x_i \in \mathcal{N}_x^k} \frac{\|e_i - e\|_2}{\|x_i - x\|_2}$$



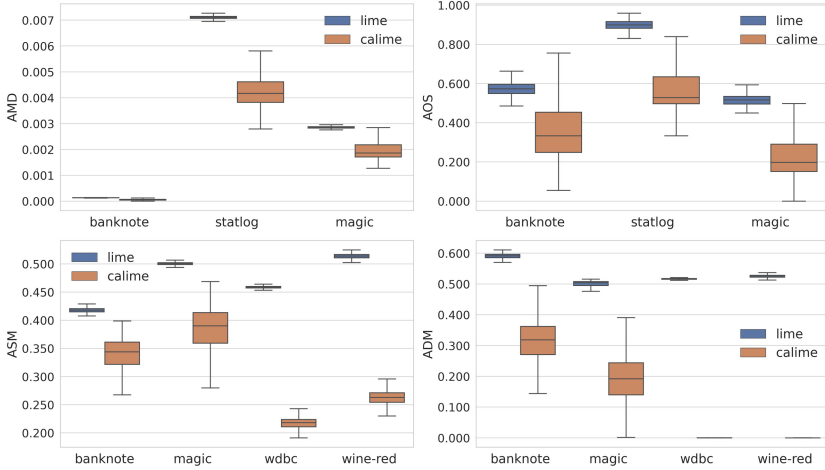
**Fig. 3.** Fidelity as  $R^2$  varying the number of features for `statlog`, `wdbc` and `wine-red`. Markers represent the mean values, while vertical bars are the standard deviations.

where  $x$  is the instance to explain and  $\mathcal{N}_x^k \subset X$  is the  $k$ -Nearest Neighborhood of  $x$  with the  $k$  neighbors selected from  $X_t$ . The lower the  $LLE$ , the higher the stability. In the plots illustrating the stability through  $LLE$ , besides the average value, we also report the minimum and maximum, using them as contingency bands. According to the literature [1], we used  $k = 5$ .

## 5.4 Results

We propose a qualitative comparison between CALIME and LIME by examining the explanations generated by both approaches for a specific instance within the `banknote` dataset (Fig. 2 - Left). It is important to note that CALIME is based on the causal framework illustrated in Fig. 2 on the right discovered using a causal discovery method [15], where *entropy* causes *variance*, which in turn influences *curtosis*<sup>9</sup>. This sequence mirrors the actual physical process whereby the textural features of a banknote determine its classification as genuine or counterfeit. We notice that the feature importance scores for *variance* and *curtosis* obtained through CALIME are significantly higher compared to those attained using LIME. This is due to the fact that the neighborhood of the explained record is generated by adhering to the causal relationships outlined in the causal graph. Conversely, LIME may not adequately capture this causal sequence, potentially undervaluing *variance* due to its dependence on correlation, which can be confounded by spurious relationships in the data. Hence, CALIME offers a more contextually relevant explanation by incorporating the causal relationships intrinsic to banknote verification processes. It ensures that model explanations are also meaningful, reflecting the dynamics involved in the real-world task.

<sup>9</sup> To avoid misinterpretation of these variables, it is important to note that their names are not intended to refer to their mathematical meanings.

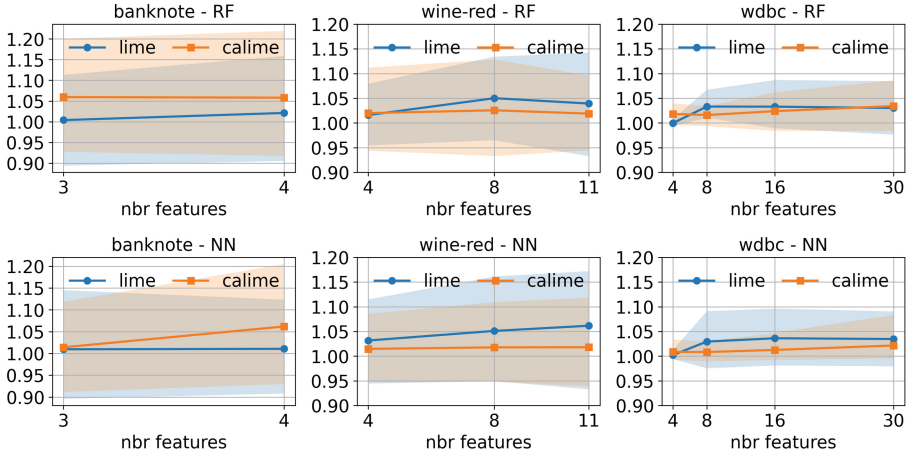


**Fig. 4.** Plausibility errors as  $AMD$ ,  $AOD$ ,  $ASM$ , and  $ADM$  in box plots aggregating the results across the scores obtained with different numbers of features. Best viewed in color. (Color figure online)

In Fig. 3 we show the fidelity as  $R^2$  of LIME and CALIME for `statlog`, `wdbc` and `wine-red` varying the number of features  $k$ . Results show that CALIME provides an improvement to LIME except `wdbc` for which the performance of CALIME is slightly worse than LIME. More in detail, for  $k = 30$ , LIME is more faithful for the NN classifier trained on `wdbc` since it achieves .44 in contrast to .41 for CALIME. For the standard deviation, we notice that CALIME is more stable than LIME. To summarize, CALIME provides explanations more trustworthy than those returned by LIME at the cost of a slightly larger variability in the fidelity of the local models.

In Fig. 4, we illustrate the plausibility of LIME and CALIME through box plots aggregating the results obtained for the different number of features  $k$  for the selected datasets considering simultaneously results for the different black-boxes. We remind the reader that all the scores observed, i.e.,  $AMD$ ,  $AOD$ ,  $ASM$ , and  $ADM$ , are “sort of” errors: lower values indicate higher plausibility scores. Analyzing the results, we observe that CALIME outperforms LIME in terms of plausibility for all the metrics analyzed. The enhancement in CALIME’s performance can be credited to the acquired knowledge of relationships among the dataset variables, which enables the generation of data points in the local neighborhood of  $x$  that closely resemble the original data. Concerning the  $AMD$  and  $AOS$  scores, we observe the results focusing the analysis on `banknote`, `magic` and `statlog`. In general, CALIME consistently produces a closer and more condensed neighborhood. In contrast, the randomly generated samples by LIME are further apart and exhibit lower density around the instance being explained. For the `banknote` dataset, we notice relatively similar results between the two methods, whereas the most noticeable difference, particularly in terms of  $AMD$





**Fig. 5.** Instability as  $LLE$  varying the number of features. Markers represent the mean values, while the contingency area highlights the minimum and maximum values.

and  $AOS$ , becomes evident when analyzing `statlog`. Regarding the  $ASM$  and  $AMD$  metrics, for which we provide insights concerning `banknote`, `magic`, `wdbc`, and `wine-red`, CALIME offers a more realistic and probable synthetic local neighborhood compared to LIME. Notably, the datasets that show the most significant improvement in LIME’s performance are `wdbc` and `wine-red`. Consistent with observations in fidelity, the greatest plausibility of CALIME correlates with a more diverse neighborhood, as evidenced by the larger box plots (e.g., for `magic`).

In Fig. 5, we illustrate the instability of LIME and CALIME as  $LLE$  (the lower, the better) for `banknote`, `wine-red` and `wdbc` varying the number of features  $k$ . The colored areas highlight the minimum and maximum values of  $LLE$  obtained by replacing the min and max operators in the previous formula. We notice that LIME is more stable than CALIME or comparable when the number of features perturbed by the neighborhood generation procedures is small, i.e.,  $k \leq 4$ . Conversely, CALIME is more resistant to noise than LIME when  $k > 4$ .

Finally, we show the comparison between CALIME, LIME, and two other state-of-the-art methods that replace the neighborhood generation in the original algorithm to achieve explanations with higher fidelity and plausibility. In Table 2, we provide the results obtained w.r.t. these metrics. Regardless of how the neighborhood is generated, all methods perform better than LIME. Our approach excels, especially in terms of  $AMD$ ,  $AOS$ , and  $R^2$  when compared to all other methods. About  $ADM$  and  $ASM$ , the results are nearly comparable between CALIME and F-LIME. This similarity may be attributed to the GAN-based generation used by F-LIME. In theory, GAN-like methods have the potential to capture possible dependencies. However, as empirically demonstrated in [7], these relationships are not explicitly represented, and there is no guarantee that they are faithfully followed in the data generation process.

**Table 2.** A comparative analysis of Fidelity and Plausibility between CALIME and baseline methods.

Datasets	Explainers	AMD ↓	AOS ↓	ASM ↓	ADM ↓	$R^2$ ↑
<b>banknote</b>	LIME	.188	459	.409	.583	.657
	CALIME	<b>.085</b>	<b>236</b>	.346	.412	<b>.661</b>
	F-LIME	.109	428	<b>.336</b>	<b>.337</b>	.643
	D-LIME	.193	672	.360	.502	.608
<b>wine-red</b>	LIME	.234	471	.518	.675	.489
	CALIME	<b>.092</b>	<b>380</b>	.287	.524	<b>.683</b>
	F-LIME	.213	400	<b>.106</b>	.510	.625
	D-LIME	.207	293	.173	<b>.503</b>	.611
<b>statlog</b>	LIME	.608	794	.353	.490	.398
	CALIME	<b>.487</b>	<b>349</b>	.265	.418	<b>.453</b>
	F-LIME	.553	441	<b>.216</b>	<b>.375</b>	.446
	D-LIME	.610	597	.280	.478	.413
<b>wdbc</b>	LIME	.340	453	.460	.516	.650
	CALIME	<b>.298</b>	<b>400</b>	<b>.250</b>	.244	<b>.726</b>
	F-LIME	.338	440	.380	<b>.248</b>	.715
	D-LIME	.393	490	.521	.251	.705

In summary, CALIME empirically exhibits better performance than LIME on the datasets and the black-boxes analyzed. However, one drawback of CALIME, at least concerning the current implementation, lies in its slower execution, primarily due to the additional time overhead associated with two specific tasks: *(i)* the extraction of the DAG and *(ii)* the learning of probability distributions for root variables and regressors to approximate dependent variables. For instance, while LIME can generate explanations in less than a second, CALIME necessitates one order of magnitude more time for completion.

## 6 Conclusion

We have presented the first proposal in the research area of post-hoc local model-agnostic explanation methods that *discovers* and *incorporates* causal relationships in the explanation extraction process. In particular, we have used GENCDA to sample synthetic data accounting for causal relationships. Empirical results suggest that CALIME can overcome the weaknesses of LIME concerning both the stability of the explanations and fidelity in mimicking the black-box.

From an application perspective, there is a growing demand for trustworthy and transparent AI approaches in high-impact domains such as financial services or healthcare. For instance, in medicine, this need arises from their possible applicability in many different areas, such as diagnostics and decision-making,

drug discovery, therapy planning, patient monitoring, and risk management. In these scenarios, due to mapping of explainability with causality, the exploitation of CALIME will allow users to make an informed decision on whether or not to rely on the system decisions and, consequently, strengthen their trust in it.

A limitation of our proposal is that adopting GENCDCA that in turn is based on NCDA, CALIME can only work on datasets composed of continuous features. We need to rely on CD approaches to simultaneously account for heterogeneous continuous and categorical datasets to overcome this drawback. For future research direction, it would be interesting to employ GENCDCA and, in general, the knowledge about causal relationships in the explanation extraction process of other model-agnostic explainers like SHAP [23] or LORE [10]. Indeed, the CALIME framework can be plugged into any model-agnostic explainer. Finally, to completely cover LIME applicability, we would like to study to which extent it is possible to employ causality awareness on data types different from tabular data, such as images and time series.

**Acknowledgments.** This work is partially supported by the EU NextGenerationEU program under the funding schemes PNRR-PE-AI FAIR (Future Artificial Intelligence Research), PNRR-SoBigData.it - Strengthening the Italian RI for Social Mining and Big Data Analytics - Prot. IR0000013, H2020-INFRAIA-2019-1: Res. Infr. G.A. 871042 *SoBigData++*, G.A. 761758 *Humane AI*, G.A. 952215 *TAILOR*, ERC-2018-ADG G.A. 834756 *XAI*, and CHIST-ERA-19-XAI-010 SAI.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Alvarez-Melis, D., Jaakkola, T.S.: Towards robust interpretability with self-explaining neural networks. In: NeurIPS, pp. 7786–7795 (2018)
2. Artelt, A., et al.: Evaluating robustness of counterfactual explanations. arXiv preprint [arxiv:2103.02354](https://arxiv.org/abs/2103.02354) (2021)
3. Beretta, I., Cinquini, M.: The importance of time in causal algorithmic recourse. In: Longo, L. (ed.) Explainable Artificial Intelligence. xAI 2023. Communications in Computer and Information Science, vol. 1901, pp. 283–298. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-44064-9\\_16](https://doi.org/10.1007/978-3-031-44064-9_16)
4. Bramhall, S., Horn, H., Tieu, M., Lohia, N.: Qlime-a quadratic local interpretable model-agnostic explanation approach. SMU Data Sci. Rev. **3**(1), 4 (2020)
5. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. ACM Comput. Surv. **41**(3), 1–58 (2009)
6. Chou, Y., Moreira, C., Bruza, P., Ouyang, C., Jorge, J.A.: Counterfactuals and causability in explainable artificial intelligence: theory, algorithms, and applications. Inf. Fusion **81**, 59–83 (2022)
7. Cinquini, M., Giannotti, F., Guidotti, R.: Boosting synthetic data generation with effective nonlinear causal discovery. In: CogMI, pp. 54–63. IEEE (2021)
8. Gosiewska, A., Biecek, P.: Do not trust additive explanations. arXiv preprint [arXiv:1903.11420](https://arxiv.org/abs/1903.11420) (2019)

9. Guidotti, R., Monreale, A., Cariaggi, L.: Investigating neighborhood generation methods for explanations of obscure image classifiers. In: Yang, Q., Zhou, Z.-H., Gong, Z., Zhang, M.-L., Huang, S.-J. (eds.) PAKDD 2019. LNCS (LNAI), vol. 11439, pp. 55–68. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-16148-4\\_5](https://doi.org/10.1007/978-3-030-16148-4_5)
10. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.* **34**(6), 14–23 (2019)
11. Guidotti, R., Monreale, A., Matwin, S., Pedreschi, D.: Black box explanation by learning image exemplars in the latent feature space. In: Brefeld, U., Fromont, E., Hotho, A., Knobbe, A., Maathuis, M., Robardet, C. (eds.) ECML PKDD 2019. LNCS (LNAI), vol. 11906, pp. 189–205. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-46150-8\\_12](https://doi.org/10.1007/978-3-030-46150-8_12)
12. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM Comput. Surv.* **51**(5), 93:1–93:42 (2019)
13. Guidotti, R., Ruggieri, S.: On the stability of interpretable models. In: IJCNN, pp. 1–8. IEEE (2019)
14. Hall, P., Gill, N., Kurka, M., Phan, W.: Machine learning interpretability with H2O driverless AI. H2O. AI (2017)
15. Hoyer, P.O., Janzing, D., Mooij, J.M., Peters, J., Schölkopf, B.: Nonlinear causal discovery with additive noise models. In: NIPS, pp. 689–696. Curran Associates, Inc. (2008)
16. Hu, L., Chen, J., Nair, V.N., Sudjianto, A.: Locally interpretable models and effects based on supervised partitioning (LIME-SUP). arXiv preprint [arXiv:1806.00663](https://arxiv.org/abs/1806.00663) (2018)
17. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint [arXiv:1907.09615](https://arxiv.org/abs/1907.09615) (2019)
18. Kanamori, K., Takagi, T., Kobayashi, K., Ike, Y., Uemura, K., Arimura, H.: Ordered counterfactual explanation by mixed-integer linear optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 11564–11574 (2021)
19. Karimi, A., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: FAccT, pp. 353–362. ACM (2021)
20. Laugel, T., Lesot, M., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: unjustified counterfactual explanations. In: IJCAI, pp. 2801–2807. ijcai.org (2019)
21. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: a review of machine learning interpretability methods. *Entropy* **23**(1), 18 (2021)
22. Longo, L., Goebel, R., Lecue, F., Kieseberg, P., Holzinger, A.: Explainable artificial intelligence: concepts, applications, research challenges and visions. In: Holzinger, A., Kieseberg, P., Tjoa, A.M., Weippl, E. (eds.) CD-MAKE 2020. LNCS, vol. 12279, pp. 1–16. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-57321-8\\_1](https://doi.org/10.1007/978-3-030-57321-8_1)
23. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: NIPS, pp. 4765–4774 (2017)
24. Martínez, Á.P., Marca, J.V.: Explaining visual models by causal attribution. In: ICCV Workshops, pp. 4167–4175. IEEE (2019)
25. Moradi, M., Samwald, M.: Post-hoc explanation of black-box classifiers using confident itemsets. *Exp. Syst. Appl.* **165**, 113941 (2021)

26. Moraffah, R., Karami, M., Guo, R., Raglin, A., Liu, H.: Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explor. Newslett.* **22**(1), 18–33 (2020)
27. Patki, N., Wedge, R., Veeramachaneni, K.: The synthetic data vault. In: 2016 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016, Montreal, 17–19 October 2016, pp. 399–410. IEEE (2016)
28. Peltola, T.: Local interpretable model-agnostic explanations of Bayesian predictive models via Kullback-Leibler projections. *arXiv preprint [arXiv:1810.02678](https://arxiv.org/abs/1810.02678)* (2018)
29. Preece, A.D.: Asking ‘why’ in AI: explainability of intelligent systems - perspectives and challenges. *Intell. Syst. Account. Finance Manag.* **25**(2), 63–72 (2018)
30. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why Should I Trust You?”: explaining the predictions of any classifier. In: *KDD*, pp. 1135–1144. ACM (2016)
31. Richens, J.G., Lee, C.M., Johri, S.: Improving the accuracy of medical diagnosis with causal machine learning. *Nat. Commun.* **11**(1), 1–9 (2020)
32. Saito, S., Chua, E., Capel, N., Hu, R.: Improving LIME robustness with smarter locality sampling. *arXiv preprint [arXiv:2006.12302](https://arxiv.org/abs/2006.12302)* (2020)
33. Shankaranarayana, S.M., Runje, D.: ALIME: autoencoder based approach for local interpretability. In: Yin, H., et al. (eds.) *IDEAL 2019*. LNCS, vol. 11871, pp. 454–463. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-33607-3\\_49](https://doi.org/10.1007/978-3-030-33607-3_49)
34. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017)
35. Zafar, M.R., Khan, N.: Deterministic local interpretable model-agnostic explanations for stable explainability. *Mach. Learn. Knowl. Extr.* **3**(3), 525–541 (2021)
36. Zhao, X., Huang, W., Huang, X., Robu, V., Flynn, D.: Baylime: Bayesian local interpretable model-agnostic explanations. In: *Uncertainty in Artificial Intelligence*, pp. 887–896. PMLR (2021)



# Evaluating the Faithfulness of Causality in Saliency-Based Explanations of Deep Learning Models for Temporal Colour Constancy

Matteo Rizzo<sup>1</sup>, Cristina Conati<sup>2</sup>, Daesik Jang<sup>3</sup>, and Hui Hu<sup>4</sup>

<sup>1</sup> Department of Environmental Sciences, Informatics, and Statistics, Ca' Foscari University of Venice, 30172 Mestre, VE, Italy

matteo.rizzo@unive.it

<sup>2</sup> Faculty of Science, Department of Computer Science, The University of British Columbia, 2366 Main Mall 201, Vancouver, BC V6T 1Z4, Canada

conati@cs.ubc.ca

<sup>3</sup> Huawei Vancouver, 4321 Still Creek Dr, Burnaby, BC, Canada

daesik.jang@huawei.com

<sup>4</sup> Huawei, Bei Jing Shi, Xi Cheng Qu, W Chang'an St, 11th Floor, Beijing Capital Times Square, No. 88 West Chang'an Avenue, Beijing 100031, China

huhui12@huawei.com

**Abstract.** The opacity of deep learning models constrains their debugging and improvement. Augmenting deep models with saliency-based strategies, such as attention, has been claimed to help better understand the decision-making process of black-box models. However, some recent works challenged the faithfulness of in-model saliency in Natural Language Processing (NLP), questioning the causality relationship between the highlights provided by attention weight and the model prediction. More generally, the adherence of attention weights to the actual decision-making process of the model, a property called faithfulness, was oppugned. We add to this discussion by evaluating the faithfulness of causality for in-model saliency applied to a video processing task for the first time, namely, temporal color constancy. We assess by adapting to our target task two tests for faithfulness from recent NLP literature, whose methodology we refine as part of our contributions. We show that attention does not offer causal faithfulness, while confidence, a particular type of in-model visual saliency, does.

**Keywords:** Explainability · Black-box · Faithfulness · Saliency · Attention

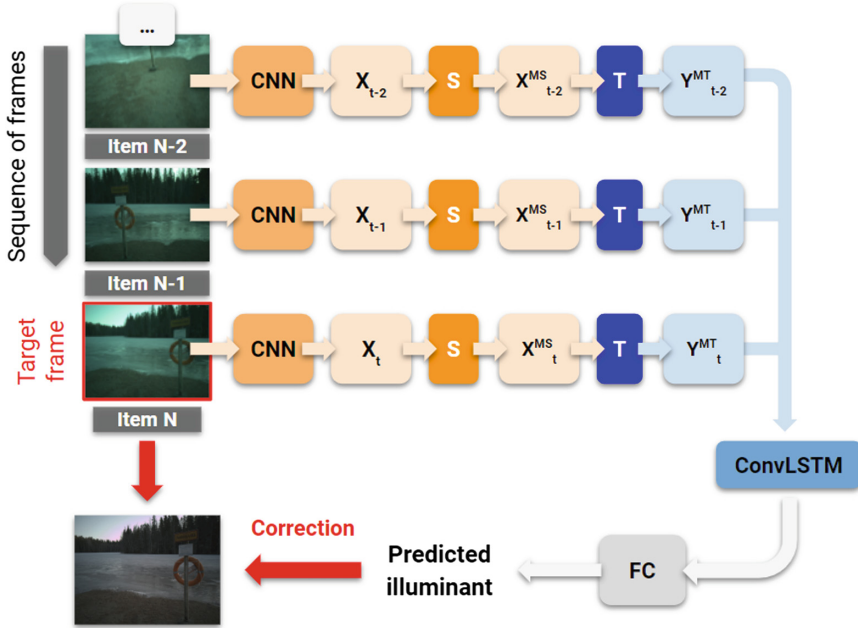
## 1 Introduction

Deep Learning (DL) models, while accurate in various tasks, remain primarily opaque, making their decision-making process challenging to comprehend. This opacity limits the evaluation of their generalizability and impedes model

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 125–142, 2024.

[https://doi.org/10.1007/978-3-031-63800-8\\_7](https://doi.org/10.1007/978-3-031-63800-8_7)



**Fig. 1.** Example of CNN+LSTM architecture for the TCC task.

debugging and improvement. Our study seeks to enhance the explainability of DL models, particularly for Sequential Data (SD). Prior research has mainly focused on language data in this context [1], offering primarily anecdotal and contradictory insights on how attention should be considered for explainability (*e.g.*, [2–4]). While explainability in non-SD models, like image object recognition, is better established [5], it is unclear how these findings apply to other models and tasks, especially in emerging video-related domains [6].

Our research extends these studies to the explainability of DL models for Temporal Color Constancy (TCC) in video data. TCC involves estimating the illuminant color of video frames, aiding in digital camera processing to enhance video quality by correcting color distortions [7, 8]. An example is shown in the left side of Fig. 1, where the green illuminant in the target frame (*i.e.*, the last in the sequence) is detected using information from the previous frames and used to perform the color correction. Current State-of-the-Art TCC methods, like TCCNet, use a CNN+LSTM (Convolutional Neural Network and Long Short Term Memory Network) architecture [8], but their operational understanding is limited, particularly regarding their performance strengths and weaknesses. This architecture is exemplified in Fig. 1. A convolutional network processes each frame in the sequence estimating feature maps ( $X$ ) weighted by a spacial saliency mask ( $S$ , either confidence or attention). The saliency-weighted feature maps ( $X^{MS}$ ) corresponding to each frame are then further weighted temporally ( $T$ , using either the mean of the confidence mask or learned temporal attention),

producing the inputs ( $Y^{MT}$ ) to the recurrent component of the architecture. Finally, a fully connected layer produces the final illuminant.

Our study focuses on enhancing the explainability of CNN+LSTM models for Temporal Color Constancy (TCC), aiming to provide developers insights into model features crucial for prediction, aiding in debugging and improvement. We explore in-model saliency techniques like “attention” [9], which modify the model architecture by adding mechanisms, such as additional neural network layers, to learn weights for each input component. Intuitively, these weights, visualized through saliency maps like heatmaps, help identify key input components for predictions, offering causal explanations. Attention methods, besides boosting State-of-the-Art (SoA) accuracy in fields like Natural Language Processing [10] and computer vision [11], have been used to decipher model decisions, highlighting critical parts of the input [12–14]. We also examine the concept of “confidence”, introduced in the FC4 method for single-frame color constancy [15]. FC4 estimates illuminant color in single-frame tasks based on a convolutional neural network and confidence-weighted pooling. The pooling strategy leverages “confidence”, a set of importance weights learned contextually to the illuminant estimate as an additional output channel. When proposing the FC4 method, the authors preliminarily examine the impact of confidence on accuracy and explainability for illuminate estimation, showing promising results. Unlike attention, confidence weights are learned alongside other feature maps, potentially revealing more about the model’s internal logic. This aspect, crucial for accuracy and explainability, has not been extensively explored in previous research.

Our primary objective is to advance the explainability of CNN+LSTM models in the context of Temporal Color Constancy (TCC), focusing on attention and confidence mechanisms within these architectures [8]. In this study, we concentrate on two pivotal aspects to determine the effectiveness of in-model explainability techniques. First, we ensure that introducing these mechanisms doesn’t detract from the original model’s accuracy. We aim to establish a causal relationship between the model’s saliency highlights and predictive outcomes.

The second and equally important aspect we examine is the faithfulness of the explanations generated by these techniques. The concept of faithfulness originates from the Natural Language Processing (NLP) community [16–18] and has been recently elaborated upon by Rizzo et al. [18]. They dissect the idea of an explanation into two fundamental elements: *evidence*, which pertains to information relating to the model in question, and *interpretation*, the human-derived semantic meaning attributed to this evidence. Consequently, an explanation is an inference drawn from interpreting specific evidence. An explanation’s faithfulness is thus gauged by how accurately this interpretation aligns with the model’s internal processing when the evidence is involved. We anticipate a *causal interpretation* of attention weights (our evidence) regarding the model’s output. In other words, we expect that high attention scores correlate with the high importance of the related input component for the prediction in a causal fashion. Our analysis of faithfulness focuses on this interpretation, which has been the intuition and norm since attention was first introduced [9]. However, addressing its faithfulness is vital for two reasons: firstly, it is foundational for examining other



essential properties of explanations (like robustness and understandability), and secondly, due to emerging skepticism about whether attention-based saliency maps truly provide accurate insights into a model’s learning process [2–4, 19]. While much past research depends on attention for generating explanations, recent studies have produced mixed results about the reliability of attention in crafting faithful explanations.

The absence of a definitive methodology to measure faithfulness in the literature presents a challenge [20]. However, the suite of tests proposed in previous studies offers a means to explore the potential connection between interpretations of attention weights and the inner workings of a model. While these tests cannot conclusively determine the faithfulness of an interpretation, they assist in verifying whether an interpretation is likely unfaithful. Therefore, this paper adapts and extends two of these tests to our TCC domain, aiming to evaluate whether attention and confidence can be reliably considered causal and faithful techniques for explainability.

Concerning the CNN+LSTM approach, we apply “attention”, “confidence”, and the combination of the two to the architecture’s spatial (S) component (CNN), temporal (T) component (LSTM), and both. Thus, we obtain nine models that we evaluate for accuracy and faithfulness. Regarding accuracy and explainability, the spatial and temporal dimensions of in-model saliency have not been compared extensively by previous literature. Thus, our evaluation aims to contribute to filling this gap. Our contributions are the following:

- We dive into the largely uncharted territory of explainability of in-model saliency explanations for video data, experimenting with the under-explored task of illuminant estimation from sequences of frames (*i.e.*, TCC).
- We analyze the faithfulness of causal interpretations of two different types of saliency (*i.e.*, attention and confidence) and three different dimensions of saliency (*i.e.*, spatial, temporal, and both combined). A formal comparison among these different types and dimensions of saliency is under-investigated in current literature.
- We extend two tests for faithfulness previously investigated only for NLP to the TCC scenario by applying a rigorous methodology<sup>1</sup>

## 2 Related Work

Prior studies on computational color constancy, mainly focusing on single images, have briefly touched upon attention for accuracy improvements but have not delved deeply into explainability [21, 22]. Our work also investigates the confidence method, first introduced by Hu et al. [15] for single-frame color constancy, where it enhanced accuracy and hinted at the potential for explainability.

We assess these in-model saliency methods based on the faithfulness of their causal interpretation. Following the framework by Rizzo et al. [18], we categorize saliency weights as *evidence*, their causal relation of importance to the model

<sup>1</sup> Code repository: <https://github.com/matteo-rizzo/saliency-faithfulness-eval>.

output as *interpretation*, and the highlighted input components as *explanation*. However, evaluations of saliency-based explanations’ faithfulness, particularly in video data, are scarce and often lack a clear distinction between evidence, interpretation, and explanation [2–4, 19].

In this paper, we extend to video data two tests proposed by Wiegrefe & Pinter [3] to evaluate the faithfulness of attention in NLP tasks by adapting them to our assessment of faithfulness of the causal interpretation of the saliency scores. The first test (WP1 from now on) is designed to assess whether or not attention weights have a relevant impact on task accuracy in the first place. The second test (WP2 from now on) tries to gauge whether attention weights embed information about the relationship among input timesteps. Details on these tests are provided in Sect. 4.

Before Wiegrefe & Pinter [3], Jain & Wallace [2] proposed two different tests to evaluate the faithfulness of attention in NLP tasks. One test compares with alternative measures of input feature importance, *e.g.* gradient-based measures, assuming that attention-based importance is faithful if the feature importance weights it generates highly correlate with those generated by the other measures. We do not look at this test because it relies on the unverified assumption that the alternative feature importance measures are faithful. The second test involves analyzing whether replacing learned attention weights with different distributions affects model prediction, assuming that if this change does not affect prediction, then weights are not involved in the decision process. Thus, they cannot provide faithful explanations of such a process. This test is complementary to the WP1 mentioned above in that it checks the importance of learning the weights through the attention mechanism, whereas WP1 corresponds that any weights have a role in the decision process in the first place.

The later tests conducted by Serrano & Smith [4] aim to understand how well attention weights represent the importance of the encoded input components by zeroing out sets of weights and seeing how this affects prediction. Their findings indicate that attention weights are poor indicators of the importance of encoded input components. However, their methodology is limited to plotting trends and lacks a quantitative assessment of whether a model passed or failed. These are common issues across the existing evaluations of faithfulness [2–4], which we address when applying the WP1 and WP2 tests from Wiegrefe & Pinter [3] to the TCC task.

### 3 Proposed Neural Architectures

To rigorously evaluate saliency faithfulness in Temporal Color Constancy (TCC), we experimented with nine distinct CNN+LSTM models encompassing three dimensions: Spatial (S), Temporal (T), and Spatio-Temporal (ST). Our study also investigated two saliency types, attention (A) and confidence (C), with a particular interest in their potential for enhancing explainability. Additionally, we explored a hybrid approach, denoted as CA, integrating confidence for spatial information and attention for temporal aspects, in line with their initial design purposes [9, 15].

Our CNN+LSTM architecture (depicted in Fig. 1) includes a spatial saliency module that processes each CNN-encoded frame  $X_i$ , learning a mask  $MS_i$ . The sequence of masked encoded frames  $X^{MS} = X \cdot MS$  is then input into the ConvLSTM, which is equipped with a temporal saliency mechanism learning a temporal mask  $MT_i$ . The output from this process is a series of temporally encoded timesteps  $Y_i$  weighted as  $Y^{MT} = Y \cdot MT$ . After processing through a Fully Connected (FC) layer, this output is utilized for illuminant prediction and subsequent color correction of the last frame in the sequence.

The implementation of attention modules, both spatial and temporal, was adapted from Meng et al. [23]. Spatial attention is learned via a three-layer CNN module that reduces the feature maps to one channel, employing batch normalization and ReLU activations in the first two layers, followed by a Sigmoid activation in the final layer. Temporal attention involves computing the Softmax of the output from two feed-forward neural networks, which are trained jointly with the rest of the system. At each timestep, the temporal attention mechanism considers every timestep in the encoded sequence  $X_i^{MS}$  and the previous hidden state  $H_{t-1}$ , resulting in a weighted sum of features from all frames fed into the ConvLSTM.

For confidence, spatially oriented as per its original conceptualization for single-frame computational color constancy [15], it is learned as an additional channel alongside feature maps and used to weigh encoded images. Temporal weights are derived by averaging the values of spatial confidence masks, which correlate with the accuracy of single-frame predictions [15].

Both attention and confidence can be visualized via heatmaps (Fig. 2), providing intuitive insights into the influential input features for model predictions. This study builds upon earlier TCC research employing CNN+LSTM models [8] and is a fundamental exploration of in-model saliency in neural networks. Future research will expand this analysis to more intricate deep-model designs like transformers.

## 4 Original Methodology of the Tests

The original methodology of the tests refers to “attention”, and is thus reported in these terms despite our analysis also involving “confidence” saliency. Faithfulness is investigated in terms of the interpretation that saliency scores have a causal relation of importance to the model output.

Test WP1 is designed to assess the role of attention in enhancing the accuracy of deep neural architectures for specific tasks and datasets. It particularly examines whether the attention mechanism contributes to more accurate predictions, a key factor in determining a model’s faithfulness in the decision-making process. The concept of faithfulness here refers to how well the attention mechanism reflects the model’s actual computational process in making decisions.

In WP1, the performance of a contextual model denoted as  $M^C$  is evaluated in two scenarios: (i) using its standard learned saliency weights ( $M_C^C$ ), and (ii) using an alternative version with randomly assigned uniform weights ( $M_U^C$ ).

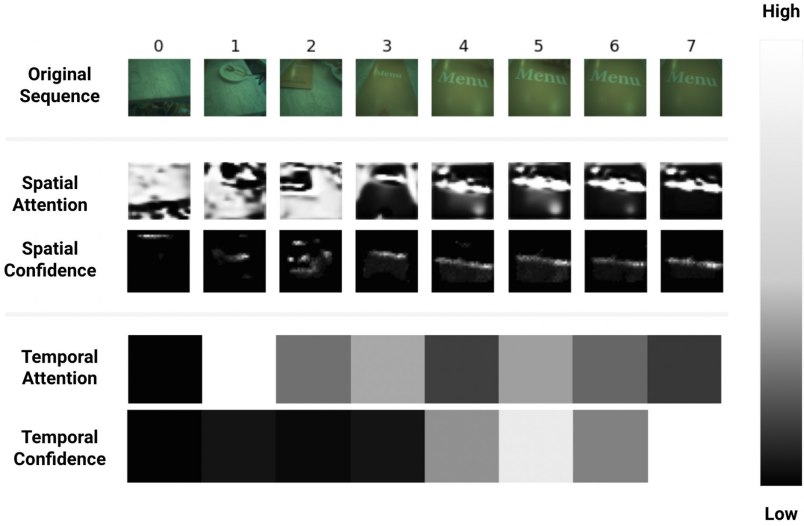


Fig. 2. Saliency heatmaps for attention and confidence.

The contextual model,  $M^C$ , typically includes a recurrent layer, such as LSTM, capable of understanding temporal relationships among input components, in contrast to a Non-Contextual (NC) model that would use linear layers instead of recurrent ones. The effectiveness of the attention mechanism is validated if  $M_C^C$  demonstrates superior accuracy compared to  $M_U^C$ . If  $M_C^C$  outperforms  $M_U^C$ , it implies that attention plays an active role in the model's decision-making, making it a candidate for further investigation regarding its faithfulness. Conversely, if  $M_C^C$  does not surpass  $M_U^C$ , attention may not significantly contribute to the decision-making process, thus questioning its faithfulness. This test establishes a necessary condition for faithfulness, setting the stage for the subsequent WP2 test.

Test WP2 delves deeper into the contextual nature of attention. It investigates whether the saliency weights learned by the attention mechanism encode contextual information about the input components. Contextual information here refers to understanding how different parts of the input relate to each other and their collective impact on the model's output. WP2 tests this by replacing the weights in a Non-Contextual model ( $M^{NC}$ )-one that does not naturally capture temporal or sequential relationships-with weights learned by a contextual model ( $M^C$ ). The aim is to see if introducing these contextual weights into a non-contextual setting enhances the model's decision-making process, as reflected by improved accuracy. This is measured by comparing the performance of  $M^{NC}$  with contextual weights ( $M_C^{NC}$ ) against both its original performance with non-contextual weights ( $M_{NC}^{NC}$ ) and the baseline uniform weights performance from WP1 ( $M_U^C$ ). A positive result in WP2 suggests that the attention mechanism is not merely learning random weights; instead, it captures and transfers valuable

contextual information from  $M^C$  to  $M^{NC}$ . This outcome reinforces the role of attention in highlighting crucial parts of the input, leveraging an understanding of the relationships among these components-i.e., the contextual information. The comparison is made fair by training the linear layers in  $M^{NC}$  alongside the other layers, ensuring an equitable basis for assessing accuracy between contextual and non-contextual models.

#### 4.1 Adaptation of the Original Tests

WP1 and WP2 were originally proposed for experimenting with binary text classification using a vanilla LSTM architecture. Thus, they must be adapted to our target TCC task and CNN+LSTM design.

WP1 is a model and task-agnostic test that requires minor adaptations. We independently generated the required random uniform distribution specified by the original methodology at inference time, both for the spatial and temporal saliency dimensions, using the same strategy for attention and confidence (i.e., the utility function offered by the PyTorch framework for DL v1.9.0). The uniformly distributed weights were then used to replace the model’s lessons. In the spatial and temporal cases, we froze to a random uniform distribution of the corresponding module’s saliency weights, leaving the other module’s unaltered. At the same time, we tweaked the saliency weights of both modules at once for the spatiotemporal scenario.

Concerning test WP2, we selectively replaced the convolutional and recurrent layers of the CNN+LSTM architecture based on which dimension, among spatial, temporal, and spatiotemporal, we were examining. This prevents the network from accessing information about the relationship between input components, such as pixels in the frames and timesteps in the sequences. In evaluating spatial saliency, we applied one linear layer in place of the convolutional component while keeping the recurrent component unaltered and vice versa to evaluate temporal saliency. When looking at spatiotemporal saliency, we replaced the convolutional and recurrent layers with linear ones.

## 5 Experimental Setup

We evaluated our models on the TCC dataset [8], which is the largest and most realistic dataset available for TCC<sup>2</sup>. It comprises 600 real-world videos recorded with a high-resolution mobile phone camera shooting  $1824 \times 1368$  sized pictures. The length of these videos ranges from 3 to 17 frames (7.3 on average, the median is 7.0, and the mode is 8.5). Ground truth information is present only for the last frame in each video (i.e., the shot frame) and was collected using a grey surface calibration target. Despite TCC being the largest dataset available,

<sup>2</sup> Of the other existing datasets for TCC, [24] is very small, [25] was specifically designed for experiments on AC bulb illumination and [26] features very low-resolution images which are not in line with the standards of modern consumer photography.

the number of sequences is relatively small for effectively training a DL model. Therefore, we perform data augmentation using the same procedure proposed in [8] to train TCCNet. Namely, the frames in the sequences were randomly rotated by an amount in the range  $[-30, +30]$  degrees and cropped to a proportion in the range of  $[0.8, 1.0]$  on the shorter dimension. The generated patches were flipped horizontally with a probability of 0.5. Data augmentation is achieved by dynamically applying these transformations at training time to each batch of sequences, which is the standard practice in PyTorch, our DL framework of reference.

All models have been trained using a mix of Tesla P100 and NVidia GeForce GTX 1080 Ti GPUs from local lab equipment and cloud services. It took about 24h for a model to complete a single learning procedure. The training of the saliency models was performed for 500 epochs using the RMSprop optimizer with batch size 1 and learning rate initially set to  $3e^{-5}$ . We opted for a hidden size equal to 128 and kernel size equal to 5, as suggested by the ablation study of TCCNet in [8]. The SqueezeNet convolutional backbone was initialized with the weights pretrained on ImageNet [27] provided by PyTorch<sup>3</sup>. The error  $\epsilon$  between the illuminant  $\hat{c}$  estimated by the saliency models and the ground truth  $c_{gt}$  has been computed using the angular error, a measure used in many works on computational color constancy and reported in Formula 1.

$$\epsilon_{\hat{c}, c_{gt}} = \arccos\left(\frac{\hat{c} \cdot c_{gt}}{\|\hat{c}\| \cdot \|c_{gt}\|}\right) \quad (1)$$

As in previous works on computational color constancy, the metrics we selected to evaluate model performance in terms of accuracy provide insights into the distribution of the angular errors across the test items. These metrics include the Mean Angular Error (MAE), the median, and the trimean (the weighted average of the median and upper and lower quartiles) across the test set, indicating how the models performed on average and accounting for outliers. We also report the MEA on the best 25th and worst 25th and 5th percentiles, showing how the model performed on easy, hard, and very hard inputs, respectively.

## 6 Method

To bolster the robustness of our faithfulness evaluations in Temporal Color Constancy (TCC), our study incorporated a four-fold cross-validation using diverse training-test splits of the TCC dataset [8]. This methodological choice was driven by the need to balance the size of training and testing samples, considering the overall size of our dataset. Consequently, our findings are presented as average results and standard deviations across these splits. A significant part of our analysis centered around the MAE, a metric selected to concisely represent the

<sup>3</sup> The pretrained models offered by PyTorch are available at <https://pytorch.org/docs/stable/torchvision/models.html>.

central tendency of angular errors in predicting illuminates (find more detailed metrics in the Appendix).

For WP1, our approach involved comparing the performance of models utilizing learned saliency ( $M_C^C$ ) with those employing frozen random uniform weights ( $M_U^C$ ). This comparison was conducted using paired t-tests, with p-values adjusted through the Benjamini-Hochberg method to account for multiple comparisons. In the context of WP2, we conducted ANOVA tests with MAE as the dependent variable. Factors in these tests included the dimension of saliency (spatial, temporal, or spatiotemporal), the type of saliency (attention, confidence, or combination), and the nature of weights used in inference (random uniform, contextual, or non-contextual). The results from these ANOVAs were further refined through post-hoc analysis using the Tukey-HSD method. Additionally, the magnitude of the effects in both the t-tests and ANOVAs was quantified using Cohen’s d value, providing a statistical measure of the size of the observed effects.

Our methodology also examined the divergence between sets of saliency weights, particularly when models of identical architecture with different saliency weights were compared. This divergence was measured using the Jensen-Shannon Divergence (JSD) for temporal saliency distributions and a combination of binary cross-entropy, structural similarity index, and intersection over union for spatial divergence. This combined approach allowed us to evaluate divergence at pixel, patch, and feature-map levels.

The interplay between saliency weights divergence and model accuracy becomes particularly pertinent when no significant difference in accuracy is observed between models. In such cases, we identified three distinct scenarios: (i) a significant discrepancy in accuracy regardless of saliency weight divergence, (ii) a minimal difference in accuracy accompanied by substantial divergence in saliency weights, and (iii) both minimal differences in accuracy and saliency divergence. In scenarios (i) and (ii), the absence of a notable accuracy difference likely indicates that saliency weights do not play a significant role in the model’s decision-making process. In contrast, scenario (iii) necessitates additional investigation to ascertain whether the observed pattern is due to the saliency not being faithful to its intended causal interpretation, potential model overfitting, or a ceiling effect resulting from the simplicity of the task.

## 7 Results

In this section, we discuss how our saliency-augmented models compare to the baseline in terms of accuracy. Then, we look at the faithfulness of the generated saliency maps concerning their causal interpretation. We apply the two selected tests for faithfulness (WP1 and WP2) to three types of saliency across three dimensions of a saliency-augmented CNN+LSTM architecture. The types of saliency are Confidence (C), Attention (A), and both combined (CA). The dimensions are Spatial (S), Temporal (T), and Spatiotemporal (ST).

## 7.1 Preliminary Accuracy Investigation

While making a neural model more transparent by modifying its architecture, we would like its accuracy to remain unaltered (or possibly to increase). With this premise in mind, we analyze the impact on the accuracy of augmenting a CNN+LSTM architecture with a saliency mechanism. Specifically, we examine how our nine proposed saliency models compare the MAE to a baseline CNN+LSTM without saliency mechanisms. Despite all models performing worse than the baseline regarding sheer numbers, this trend did not prove significant when running t-tests. The small effect sizes for attention spatiotemporal, attention temporal, and confidence temporal (A-ST, A-T, and C-T) indicate that these models are likely to be equivalent to the baseline in terms of accuracy. More details on these results are available in the Appendix.

## 7.2 Test WP1

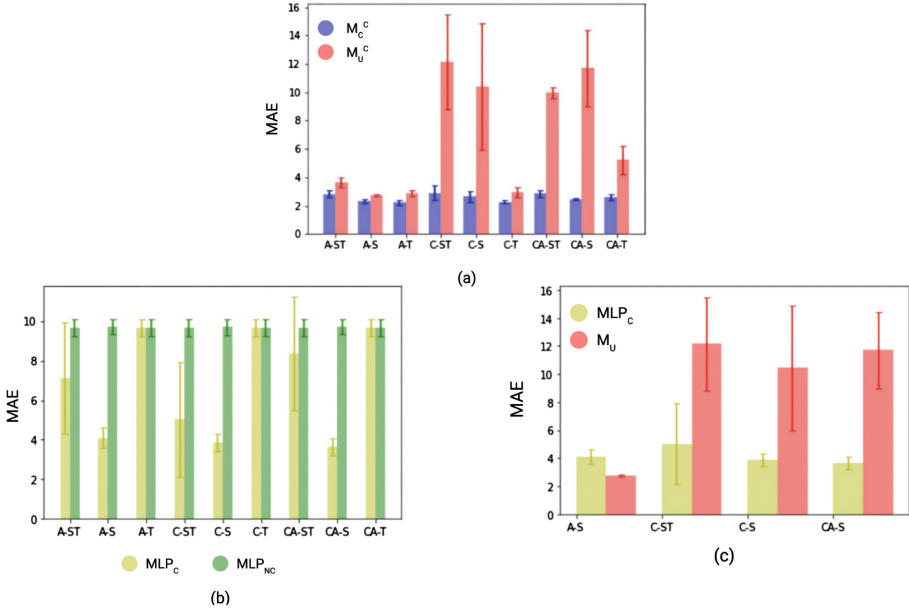
Figure 3a compares the MAE achieved by models using either random uniformly distributed saliency weights ( $M_U^C$ ) or saliency weights derived from learned model parameters ( $M_C^C$ ). In terms of sheer numbers, we observe that the error achieved by models using random, uniformly distributed weights is always higher than that achieved by models using weights derived from learned model parameters. This trend is particularly accentuated when spatial confidence is involved in the evaluation. Running t-tests followed by Benjamini-Hochberg adjustments for multiple comparisons confirm that the trend is statistically significant (p-value < 0.05) for each of the examined configurations. Moreover, the corresponding effect sizes are huge (Cohen’s  $d > 1$ ). Thus, all our nine proposed models pass test WP1. This means that the learned saliency scores carry valuable causal information to the model for achieving accurate predictions. Accuracy is sensitive to manipulation of the saliency distribution.

## 7.3 Test WP2

We remark that test WP2 is passed if the MAE achieved by the non-contextual model using saliency weights derived from a contextual model ( $M_C^{NC}$ ) is lower than (i) the MAE achieved by the non-contextual model using weights derived from its learned parameters ( $M_{NC}^{NC}$ ), and (ii) the MAE achieved by the contextual model using frozen uniformly distributed saliency weights ( $M_U^C$ ). We performed these two comparisons for each of the nine proposed saliency models as the contextual model of reference. Figure 3b presents the MAE values for comparison (i) for the considered saliency types and dimensions concerning the type of weights the model uses.

The bar chart shows that (i) holds for attention spatial, confidence spatial, confidence and attention combined spatial, and spatiotemporal (A-S, C-S, CA-S, C-ST). For the other configurations, we need to check if the lack of a significant difference could be due to low divergence in the saliency masks generated by the non-contextual model using learned saliency weights ( $M_{NC}^{NC}$ ) and





**Fig. 3.** Plot of Mean Average Error (MAE) values for Test WP1 (a) (models using either random uniformly distributed saliency weights  $M_U^C$  or saliency weights derived from learned model parameters  $M_C^C$ ) and Test WP2, comparison (i) (b) and (ii) (c) (respectively, MAE from non-contextual models using saliency weights derived from a contextual model  $M_C^{NC}$  is lower than (i) the MAE achieved by the non-contextual model using weights derived from its learned parameters  $M_{NC}^{NC}$ , and (ii) the MAE achieved by the contextual model using frozen uniformly distributed saliency weights  $M_U^C$ ).

the non-contextual model using saliency weights imposed from the contextual model ( $M_C^{NC}$ ). We look at the relationships between accuracy and the generated saliency masks for the temporal and spatiotemporal models and interpret them as discussed in Sect. 6. For all of the considered models, saliency divergence is high (*i.e.*,  $Div_{temp} > 0.7$ ,  $Div_{spat} > 125$ ), which suggests that the low difference in accuracy is not due to saliency weights being very similar, but instead to them not being involved in the decision-making process of the model in terms of causal interpretation. Therefore, we do not consider these models in (ii).

As shown by the plot in Fig. 3c, comparison (ii) holds for confidence spatiotemporal, confidence spatial, and confidence and attention combined spatial (C-ST, C-S, CA-S). Thus, these models pass the test WP2. On the other hand, comparison (ii) does not hold for attention spatial (A-S). In this case, the contextual architecture of the model has more impact on accuracy than the saliency mechanism.

		Attention (A)			Confidence (C)			Confidence + Attention (CA)		
		S	T	ST	S	T	ST	S	T	ST
<b>Accuracy</b>			*	*		*				
<b>Faithfulness</b>	<b>WP1</b>									
	<b>WP2</b>									

\* Accuracy close to baseline     Test passed     Test failed

**Fig. 4.** Summary table of the results of tests WP1 and WP2.

## 7.4 Discussion

Figure 4 summarizes the results of the accuracy analysis and the two tests for our nine models.

First, we note that the temporal dimension is present in all three top-performing model configurations, suggesting that this facet of saliency is essential for accuracy. However, no model appears faithful to our casual interpretation when leveraging the temporal dimension only. An intuitive justification for this phenomenon stems from observing that temporal saliency focuses on a few frames in a sequence. This means that a lot of possibly relevant spatial information is discarded. Thus, a model might learn not to actively use temporal saliency in its decision-making process to preserve accuracy.

Second, we observe that spatial confidence is present in all configurations that pass the two assessments, which suggests that this saliency dimension and type help support the faithfulness of the causal interpretation. This might be due to confidence scores being jointly learned with the other feature maps and thus more strictly connected to the inner decision-making process of the model. On the other hand, attention configurations never succeed in upholding faithfulness. The crucial difference between attention and confidence is that a separate ad hoc convolutional module learns the former while the latter is learned as an additional feature map. As a result, the attention model is more complex (*i.e.*, has a more significant number of trainable parameters,  $\sim 3$ ). It might be that attention networks get sufficiently complex to achieve high accuracy while ignoring saliency in their decision-making process. That is, there could be a trade-off between model complexity and the causal interpretation of saliency. A model could be complex enough to solve the task effectively, and introducing further parameters may result in a ceiling effect. In this case, the model could still fulfill the target task without fully leveraging its complexity and, thus, regardless of the learned saliency weights.

## 8 Conclusions and Future Work

In this study, we explored in-model saliency methods, particularly attention and confidence, in the novel context of video-based illuminant estimation, focusing on their faithfulness in influencing model predictions. This assessment covered three dimensions: spatial, temporal, and spatiotemporal. We adapted two tests from previous NLP research to evaluate the causal relationship between saliency scores and model predictions. We enhanced the methodology with statistical analysis and examination of saliency weight divergence. Our findings indicate that spatial and spatiotemporal confidence may be faithful to their causal interpretation, while attention models generally failed these tests. This aligns with previous research questioning the reliability of attention as an explanatory tool for causal interpretation. On the other hand, the promising results achieved by confidence shed light on the importance of how in-model saliency is integrated for driving faithful causality in explanations. Additionally, our accuracy analysis showed that temporal models tend to perform better.

However, our study has limitations, primarily its restriction to a single task and dataset, challenging the generalizability of our results. Future work will expand these assessments to a broader range of datasets and tasks, including those from NLP literature, and explore diverse model architectures like transformers and different attention methods (*e.g.*, roll out and flow [28]). We acknowledge the need for a clear threshold to distinguish a test’s failure due to insufficient divergence in saliency weights, which we aim to establish. Additionally, we plan to investigate other properties like robustness and plausibility. Despite these limitations, our research provides a foundational analysis of the faithfulness of in-model saliency in the TCC task and addresses methodological gaps in previous studies.

**Disclosure of Interests.** The authors declare no competing interests.

## A Extended Results of the Experiments

(See Tables 1, 2, 3, 4 and 5).

**Table 1.** Accuracy. Metrics concerning the angular error for models using no saliency (B), attention (A), confidence (C), and confidence and attention combined (CA).

Model	Saliency	Mean		Median		Trimean		Best 25%		Worst 25%		Worst 5%	
		Avg	Std dev	Avg	Std dev	Avg	Std dev	Avg	Std dev	Avg	Std dev	Avg	Std dev
B	–	2.28	0.24	1.50	0.19	1.72	0.23	0.46	0.06	5.32	0.58	6.60	1.13
A	ST	2.83	0.27	1.87	0.19	2.13	0.22	0.45	0.05	6.78	0.76	9.08	1.04
	S	2.32	0.14	1.66	0.05	1.81	0.05	0.45	0.04	5.34	0.56	6.49	1.16
	T	2.22	0.18	1.26	0.07	1.50	0.07	0.28	0.04	5.79	0.58	7.43	1.35
C	ST	2.90	0.52	2.24	0.50	2.35	0.51	0.64	0.15	6.44	1.05	8.00	1.19
	S	2.64	0.37	1.98	0.37	2.11	0.36	0.60	0.13	5.85	0.70	6.83	1.22
	T	2.28	0.08	1.64	0.11	1.81	0.07	0.49	0.06	5.07	0.30	6.37	0.10
CA	ST	2.84	0.27	2.05	0.21	2.27	0.22	0.59	0.12	6.44	0.65	7.94	1.52
	S	2.45	0.07	1.71	0.09	1.89	0.06	0.47	0.07	5.66	0.30	6.99	0.21
	T	2.60	0.22	1.77	0.22	1.98	0.22	0.52	0.10	6.04	0.43	7.62	0.57

**Table 2.** Test WP1. Metrics concerning the angular error for models using random uniformly distributed (R) versus learned (L) saliency weights.

Model	Saliency	Mean		Median		Trimean		Best 25%		Worst 25%		Worst 5%	
		Avg	Std dev	Avg	Std dev	Avg	Std dev	Avg	Std dev	Avg	Std dev	Avg	Std dev
B	–	2.28	0.24	1.50	0.19	1.72	0.23	0.46	0.06	5.32	0.58	6.60	1.13
A	ST (L)	2.83	0.27	1.87	0.19	2.13	0.22	0.45	0.05	6.78	0.76	9.08	1.04
	ST (R)	3.66	0.34	2.42	0.24	2.69	0.27	0.67	0.10	8.77	1.02	11.42	1.28
	S (L)	2.32	0.14	1.66	0.05	1.81	0.05	0.45	0.04	5.34	0.56	6.49	1.16
	S (R)	2.74	0.06	1.99	0.08	2.20	0.09	0.58	0.04	6.08	0.18	7.22	0.22
	T (L)	2.22	0.18	1.26	0.07	1.50	0.07	0.28	0.04	5.79	0.58	7.43	1.35
	T (R)	2.90	0.21	1.87	0.06	2.14	0.10	0.53	0.11	6.92	0.56	8.85	1.08
C	ST (L)	2.90	0.52	2.24	0.50	2.35	0.51	0.64	0.15	6.44	1.05	8.00	1.19
	ST (R)	12.16	3.34	11.06	2.59	11.33	2.86	2.94	1.32	23.34	6.67	25.80	8.61
	S (L)	2.64	0.37	1.98	0.37	2.11	0.36	0.60	0.13	5.85	0.70	6.83	1.22
	S (R)	10.43	4.47	9.76	5.31	10.00	5.10	3.93	2.32	18.00	5.85	20.13	5.91
	T (L)	2.28	0.08	1.64	0.11	1.81	0.07	0.49	0.06	5.07	0.30	6.37	0.10
	T (R)	2.94	0.38	2.15	0.28	2.31	0.29	0.66	0.11	6.57	0.99	8.29	1.37
CA	ST (L)	2.84	0.27	2.05	0.21	2.27	0.22	0.59	0.12	6.44	0.65	7.94	1.52
	ST (R)	9.97	0.38	9.08	0.81	9.49	0.41	3.08	0.13	18.36	1.37	20.38	1.86
	S (L)	2.45	0.07	1.71	0.09	1.89	0.06	0.47	0.07	5.66	0.30	6.99	0.21
	S (R)	11.70	2.71	9.68	2.65	10.57	2.96	2.72	0.78	23.11	4.41	25.83	4.22
	T (L)	2.60	0.22	1.77	0.22	1.98	0.22	0.52	0.10	6.04	0.43	7.62	0.57
	T (R)	5.23	1.00	3.75	0.91	4.13	1.03	0.84	0.07	12.04	2.14	14.77	2.93

**Table 3.** Test WP2. Metrics concerning the angular error for models using *attention*.

Saliency Type	Saliency	Mean		Median		Trimean		Best 25%		Worst 25%		Worst 5%	
		Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
Baseline	ST	9.89	0.44	9.74	0.30	9.53	0.55	2.63	0.64	17.86	0.75	19.48	1.18
	S	4.10	0.51	9.25	9.31	2.82	17.72	19.44	9.80	9.23	9.25	2.79	17.81
	T	9.91	0.45	9.99	0.55	9.64	0.57	2.69	0.46	17.75	0.69	19.29	0.77
Learned	ST	9.64	0.42	9.03	0.49	9.10	0.51	2.05	0.43	18.20	0.82	20.47	0.67
	S	9.70	0.40	9.35	0.51	9.21	0.46	2.56	0.23	17.99	0.86	19.64	0.45
	T	9.64	0.43	8.96	0.56	9.07	0.50	2.31	0.58	18.16	0.62	20.37	0.85
Contextual	ST	7.12	2.81	6.21	3.24	6.31	3.08	1.65	0.84	14.42	4.20	17.24	3.60
	S	4.10	0.51	3.18	0.25	3.40	0.35	1.05	0.08	8.79	1.40	11.01	1.72
	T	9.65	0.43	9.30	0.68	9.17	0.61	2.39	0.31	18.06	0.73	20.49	0.71

**Table 4.** Test WP2. Metrics concerning the angular error for models using *confidence*.

Saliency Type	Saliency	Mean		Median		Trimean		Best 25%		Worst 25%		Worst 5%	
		Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.	Avg.	Std Dev.
Baseline	ST	9.89	0.44	9.70	0.36	9.52	0.57	2.65	0.64	17.84	0.78	19.44	1.19
	S	19.35	9.83	9.49	9.44	2.89	17.65	19.47	0.22	8.22	0.91	9.59	1.01
	T	9.91	0.45	9.98	0.57	9.63	0.58	2.70	0.45	17.74	0.70	19.28	0.77
Learned	ST	9.64	0.42	9.10	0.44	9.12	0.50	2.42	0.50	18.09	0.78	20.51	0.74
	S	9.69	0.42	9.20	0.28	9.26	0.42	2.66	0.41	18.01	0.52	20.32	0.56
	T	9.65	0.42	9.01	0.53	9.06	0.50	2.34	0.56	18.10	0.67	20.33	1.03
Contextual	ST	5.03	2.89	4.14	3.11	4.33	3.02	1.09	0.80	10.51	4.58	13.09	5.09
	S	3.86	0.43	2.99	0.44	3.20	0.37	0.96	0.22	8.22	0.91	9.59	1.01
	T	9.65	0.42	9.30	0.66	9.19	0.59	2.35	0.40	18.12	0.91	20.43	0.74

**Table 5.** Test WP2. Metrics concerning the angular error for models using *confidence* as spatial saliency and *attention* as temporal saliency

Saliency Type	Saliency	Mean		Median		Trimean		Best 25%		Worst 25%		Worst 5%	
		Avg.	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev	Avg	Std Dev
Baseline	ST	9.89	0.44	9.71	0.39	9.53	0.57	2.67	0.63	17.82	0.79	19.38	1.19
	S	9.83	0.49	9.49	0.53	9.44	0.70	2.89	0.53	17.65	0.41	19.47	0.68
	T	9.91	0.45	9.99	0.56	9.64	0.58	2.69	0.46	17.75	0.69	19.29	0.77
Learned	ST	9.64	0.42	9.12	0.44	9.13	0.50	2.43	0.50	18.07	0.79	20.46	0.75
	S	9.69	0.39	9.16	0.60	9.17	0.49	2.60	0.34	18.04	0.82	19.76	0.54
	T	9.64	0.43	8.95	0.58	9.05	0.52	2.31	0.59	18.16	0.62	20.40	0.85
Contextual	ST	8.35	2.85	7.81	3.41	7.76	3.20	2.02	0.87	16.16	4.33	18.22	3.99
	S	3.64	0.45	2.73	0.45	2.97	0.46	0.78	0.13	7.97	0.71	9.73	0.86
	T	9.66	0.42	9.28	0.74	9.19	0.64	2.41	0.30	18.05	0.74	20.30	0.45

## References

1. Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A survey of the state of explainable AI for natural language processing. In: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China, December 2020, pp. 447–459. Association for Computational Linguistics (2020). <https://aclanthology.org/2020.aacl-main.46>
2. Jain, S., Wallace, B.C.: Attention is not explanation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, June 2019, vol. 1 (Long and Short Papers), pp. 3543–3556. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/N19-1357>. <https://www.aclweb.org/anthology/N19-1357>
3. Wiegrefe, S., Pinter, Y.: Attention is not explanation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, November 2019, pp. 11–20. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/D19-1002>. <https://www.aclweb.org/anthology/D19-1002>
4. Serrano, S., Smith, N.A.: Is attention interpretable? In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, July 2019, pp. 2931–2951. Association for Computational Linguistics (2019). <https://doi.org/10.18653/v1/P19-1282>. <https://www.aclweb.org/anthology/P19-1282>
5. Hohman, F., Kahng, M., Pienta, R., Chau, D.H.: Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Trans. Visualization Comput. Graph.* **25**(8), 2674–2693 (2018). ISSN 1941-0506. <https://doi.org/10.1109/tvcg.2018.2843369>
6. Hiley, L., Preece, A.D., Hicks, Y.A.: Explainable deep learning for video recognition tasks: a framework & recommendations. *ArXiv arxiv:1909.05667* (2019). <https://api.semanticscholar.org/CorpusID:202565462>
7. Ramanath, R., Snyder, W., Yoo, Y.J., Drew, M.: Color image processing pipeline. *IEEE Signal Process. Maga.* **22**(1), 34–43 (2005). ISSN 1558-0792. <https://doi.org/10.1109/msp.2005.1407713>

8. Qian, Y., Käpylä, J., Kämäräinen, J.K., Koskinen, S., Matas, J.: A benchmark for temporal color constancy (2020)
9. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). <http://arxiv.org/abs/1409.0473>
10. de Santana Correia, A., Colombini, E.L.: Attention, please! a survey of neural Attention Models in Deep Learning. [arXiv:2103.16775](https://arxiv.org/abs/2103.16775) [cs] (2021)
11. Guo, M.H., et al.: Attention mechanisms in computer vision: a survey (2021). <https://arxiv.org/abs/2111.07624>
12. Choi, E., Bahadori, M.T., Sun, J., Kulas, J., Schuetz, A., Stewart, W.: RETAIN: an interpretable predictive model for healthcare using reverse time attention mechanism. In: Advances in Neural Information Processing Systems, vol. 29 (2016). <https://proceedings.neurips.cc/paper/2016/hash/231141b34c82aa95e48810a9d1b33a79-Abstract.html>
13. Lei, T.: Interpretable neural models for natural language processing. Thesis, Massachusetts Institute of Technology (2017). <https://dspace.mit.edu/handle/1721.1/108990>
14. Mullenbach, J., Wiegrefe, S., Duke, J., Sun, J., Eisenstein, J.: Explainable prediction of medical codes from clinical text. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, Louisiana, June 2018, vol. 1 (Long Papers), pp. 1101–1111. Association for Computational Linguistics (2018). <https://doi.org/10.18653/v1/N18-1100>. <https://www.aclweb.org/anthology/N18-1100>
15. Hu, Y., Wang, B., Lin, S.: Fc<sup>4</sup>: fully convolutional color constancy with confidence-weighted pooling. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 330–339 (2017). <https://doi.org/10.1109/cvpr.2017.43>. ISSN: 1063-6919
16. Jacovi, A., Goldberg, Y.: Towards faithfully interpretable NLP systems: how should we define and evaluate faithfulness? In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4198–4205. Association for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.acl-main.386>. <https://aclanthology.org/2020.acl-main.386>
17. Jacovi, A., Goldberg, Y.: Aligning faithful interpretations with their social attribution. *Trans. Assoc. Comput. Linguist.* **9**, 294–310 (2021). ISSN 2307-387X. <https://doi.org/10.1162/tacl.a.00367>
18. Rizzo, M., Veneri, A., Albarelli, A., Lucchese, C., Nobile, M., Conati, C.: A theoretical framework for ai models explainability with application in biomedicine. In: 2023 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pp. 1–9 (2023). <https://doi.org/10.1109/CIBCB56990.2023.10264877>
19. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS 2018, Red Hook, NY, USA, pp. 9525–9536. Curran Associates Inc. (2018)
20. Chan, C.S., Kong, H., Liang, G.: A comparative study of faithfulness metrics for model interpretability methods. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Dublin, Ireland, May 2022, vol. 1: Long Papers, pp. 5029–5038. Association for Computational Linguistics (2022). <https://doi.org/10.18653/v1/2022.acl-long.345>

21. Yan, C., et al. STAT: spatial-temporal attention mechanism for video captioning. *IEEE Trans. Multimedia* **22**(1), 229–241 (2020). ISSN 1941-0077. <https://doi.org/10.1109/tmm.2019.2924576>
22. Zhang, Y., Xiong, N.N., Wei, Z., Yuan, X., Wang, J.: ADCC: an effective and intelligent attention dense color constancy system for studying images in smart cities. [arXiv:1911.07163](https://arxiv.org/abs/1911.07163) [cs] (2020)
23. Meng, L., et al.: Interpretable spatio-temporal attention for video action recognition. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 1513–1522 (2019). <https://doi.org/10.1109/iccvw.2019.00189>. ISSN: 2473-9944
24. Prinet, V., Lischinski, D., Werman, M.: Illuminant chromaticity from image sequences. In: 2013 IEEE International Conference on Computer Vision, pp. 3320–3327 (2013). <https://doi.org/10.1109/iccv.2013.412>
25. Yoo, J., Kim, J.: Dichromatic model based temporal color constancy for ac light sources. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12321–12330 (2019). <https://doi.org/10.1109/cvpr.2019.01261>
26. Ciurea, F., Funt, B.: A large image database for color constancy research. In: *Imaging Science and Technology Eleventh Color Imaging Conference*, pp. 160–164 (2003)
27. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: *CVPR 2009* (2009)
28. Abnar, S., Zuidema, W.: Quantifying attention flow in transformers. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197. Association for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.acl-main.385>



# CAGE: Causality-Aware Shapley Value for Global Explanations

Nils Ole Breuer<sup>1</sup> (✉), Andreas Sauter<sup>2</sup>, Majid Mohammadi<sup>2</sup>, and Erman Acar<sup>3</sup>

<sup>1</sup> GT-ARC gGmbH, Berlin, Germany  
nils.breuer@gt-arc.com

<sup>2</sup> Vrije Universiteit Amsterdam, Amsterdam, The Netherlands  
{a.sauter, m.mohammadi}@vu.nl

<sup>3</sup> ILLC and IviI, University of Amsterdam, Amsterdam, The Netherlands  
erman.acar@uva.nl

**Abstract.** As Artificial Intelligence (AI) is having more influence on our everyday lives, it becomes important that AI-based decisions are transparent and explainable. As a consequence, the field of eXplainable AI (or XAI) has become popular in recent years. One way to explain AI models is to elucidate the predictive importance of the input features for the AI model in general, also referred to as global explanations. Inspired by cooperative game theory, Shapley values offer a convenient way for quantifying the feature importance as explanations. However many methods based on Shapley values are built on the assumption of feature independence and often overlook causal relations of the features which could impact their importance for the ML model. Inspired by studies of explanations at the local level, we propose CAGE (Causally-Aware Shapley Values for Global Explanations). In particular, we introduce a novel sampling procedure for out-coalition features that respects the causal relations of the input features. We derive a practical approach that incorporates causal knowledge into global explanation and offers the possibility to interpret the predictive feature importance considering their causal relation. We evaluate our method on synthetic data and real-world data. The explanations from our approach suggest that they are not only more intuitive but also more faithful compared to previous global explanation methods.

**Keywords:** Explainable Artificial Intelligence · XAI · Shapley values · Global Explanation · Causality · Causal Explanations

## 1 Introduction

Explainable artificial intelligence (XAI) is a field of study in artificial intelligence (AI) research that complements complex machine learning (ML) models with comprehensible insights, facilitating human understanding and trust [15]. It endeavors to unravel the intricate decision-making processes of AI systems, providing clear, interpretable, and accessible explanations that favorably align with human cognition and reasoning. As AI technologies burgeon and permeate various sectors -from healthcare [22] and finance [38] to criminal justice [39]- the imperative for XAI is magnified, demanding that the



opaque “black-box” models are more transparent and the algorithmic decisions thereof are justified.

In this ever-evolving landscape, various research lines and methodologies have been proposed, each aspiring to shed light on different aspects of the workings of AI models [27]. Among these, Shapley value-based methods such as SHAP [19] and SAGE [5] have gained substantial traction, employing concepts from cooperative game theory to attribute locally (instance-based) and globally (model-based) the contribution of each feature to a model’s prediction, respectively. Despite their popularity (in particular the former), these methods, built on assumptions of feature independence, often overlook the nuanced causal relationships and interactions amongst features, potentially leading to oversimplified or misleading explanations.

Pivotal work [20] underscores that genuine explanations are intrinsically tied to causality, reflecting a philosophical viewpoint where explanations are crafted through counterfactual reasoning - envisaging alternative scenarios and assessing their impact on outcomes. Thus, causality emerges as a fundamental pillar in crafting meaningful and intuitive explanations [20, 21]. In such a pursuit, recent explorations such as Causal SHAP [10] and Asymmetric SHAP [7] have sought to infuse causality into local explanation frameworks, as they emerge as promising frontiers, endeavoring to intertwine causal reasoning with *local* explanation techniques. These methods are argued to reflect the human cognitive processes of causal inference, striving for explanations that resonate with innate human understanding and intuition while being grounded in strong mathematical foundations [20].

In this article, we introduce a method that incorporates a causal lens into Shapley-value based *global* explanations (i.e., SAGE), abbreviated by the acronym CAGE. Empowered by its capability to express complex causal relations between features, we show both theoretically and empirically that CAGE can alleviate the aforementioned deficiencies, and result in more faithful global explanations. More specifically, the main contributions of this article are as follows.

1. We introduce a model-agnostic causality-aware conceptual framework based on Shapley values for the global explanations i.e., CAGE. In particular, we establish a novel sampling procedure that respects the causal relations of input features;
2. We theoretically show that CAGE satisfies desirable causal properties; an indication that it is designed from first principles.
3. We carry out an empirical analysis with both synthetic and real-world data, concluding that explanations resulting from CAGE are more faithful compared to their causally agnostic counterparts.

In the remainder of this paper, we start by introducing core concepts and our notation. We then present CAGE in detail, show that it derives causally sound explanations, and present an algorithm to estimate its values. Furthermore, we apply CAGE to synthetic and real-world data to substantiate our claims. Finally, we discuss the most related works and provide an in-depth discussion about our results before we conclude. The code for our framework and experiments is available at <https://github.com/no-breuer/CausalGlobalExplanation>.

## 2 Preliminaries and Notation

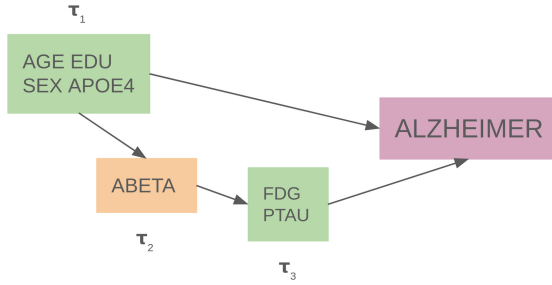
In this section, we introduce the key concepts and notations used throughout this paper. These are additive importance measures, Shapley values, and fundamental notions of causality.

### 2.1 Causal Models and Interventions

**Structural Causal Models (SCM).** We resort to *structural causal models (SCM)* to express causal relations formally. An SCM is a tuple  $M = (\mathbf{X}, \mathbf{F}, \mathbf{U}, \mathbf{P})$  of observed variables  $\mathbf{X} = \{X_1, \dots, X_n\}$ , unobserved exogenous variables  $\mathbf{U} = \{U_1, \dots, U_n\}$ , functional relations  $\mathbf{F}$  that define direct causal effects, and  $\mathbf{P}$  a set of pairwise independent distributions of exogenous variables. Each SCM induces a directed acyclic graph (DAG), where the direct causes of an endogenous variable are incoming edges (parents). Formally, each variable  $X_i$  is determined by  $f_i \in \mathbf{F}$  s.t.  $X_i \leftarrow f_i(\mathbf{Pa}_i, U_i)$ , where  $\mathbf{Pa}_i \subseteq \mathbf{X} \setminus \{X_i\}$  (parents) are the direct causes of  $X_i$ . From the conditional independence assumption a joint probability distribution  $P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \mathbf{Pa}_i)$  of the SCM can be inferred [23,29,30] which we will refer to as *observational distribution*.

**Causal Chain Graphs.** We use the notion of causal chain graphs [16] as a relaxation on the information encoded in causal graphs. Causal chain graphs represent a partial causal ordering of sets of variables (chain components) among which the causal relationships are not fully known. This means that variables are in the same component whenever they have a common confounder or mutual interaction between them (see Fig. 1 for an example).

**Interventions.** Interventions are purposefully modifying the values of variables to discern cause-and-effect relationships. Therefore, they are of vital importance for causal reasoning. An intervention is formally expressed by the so-called *do-operator*, denoted for  $\mathbf{Y} \subseteq \mathbf{X}$  as  $do(\mathbf{Y} = \mathbf{c})$  or  $do(\mathbf{Y})$  when the value is clear from the context. This forces one or more variables in an SCM to a particular value, effectively replacing all corresponding functions with this value. Graphically, this corresponds to pruning all incoming edges to that variable. Interventions result in a new joint distribution  $P(\mathbf{X} \setminus \mathbf{Y} | do(\mathbf{Y} = \mathbf{c}))$  [24] which we will refer to as the *post-interventional distribution*. If there are changes between the observational and post-interventional distributions, conclusions can be drawn about the causal influence of the intervention variable on the other variables [23,25].



**Fig. 1.** Causal chain graph that shows the partial causal ordering of the Alzheimer dataset *ADNI* [11] used later in the experiments. The chain graph consists of three components  $\tau_1$ ,  $\tau_2$ , and  $\tau_3$  that have a causal ordering. In the components, the complete causal relationships of the variables are not known. Variables in  $\tau_1$  and  $\tau_3$  are assumed to have common confounders (green) and  $\tau_2$  is assumed to have causal interaction (yellow). The target variable is marked in red. (Color figure online)

### 2.2 Shapley Additive Global Importance

In the domain of XAI, *additive importance measures* are of common practice since they provide model-agnostic explainability by dissecting contributions of individual features in the target AI models. Intuitively, feature importance can be interpreted as the amount of predictive power a feature provides for an AI model at hand. The main characteristic of these measures is that the individual feature importances  $\phi_i(v)$  sum up to the models’ overall prediction power w.r.t. a value function  $v$  [6]. Whenever  $v$  is clear from the context, we will only write  $\phi_i$ .

A broadly used instance of additive measures are the so-called *Shapley values*. In the context of XAI, the Shapley value of a feature  $i$  measures the marginal increase of the value function  $v$  of a model if that feature is considered in the prediction. The Shapley value of each feature  $i$ ,  $\phi_i$  is then the weighted sum of that increase over all possible permutations of all features  $D$  of a specific subset of features  $S$ . This is inspired by cooperative game theory and defined by [31] as follows:

$$\phi_i(v) = \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i\}} \binom{|D| - 1}{|S|}^{-1} (v(S \cup i) - v(S)), \tag{1}$$

where  $D$  is the set of all feature indices,  $S$  is a subset of  $D$  also called the “in-coalition” features, and  $i$  is the specific feature indices which is added to  $S$  and for which the importance is computed. The value function  $v$  only gets a subset of feature indices, but because we want to analyze the model that takes all features  $D$  as input, a way to sample the values of the missing feature indices  $\bar{S} = D \setminus S$  needs to be devised.

To derive *Shapley additive global importance (SAGE)* values for a model, Covert et al. [6] defines a value function that measures the change in the loss  $\mathcal{L}$  of a model, shown in Eq. 2:

$$v_f(S) = -\mathbb{E}[\mathcal{L}(\mathbb{E}[f(\mathbf{X})|\mathbf{X}_S], Y)], \tag{2}$$

where  $\mathbf{X}_S$  are the variables for the “in-coalition” features  $S$ ,  $\mathbf{X} = \mathbf{X}_S \cup \mathbf{X}_{\bar{S}}$  are all feature variables, and  $Y$  is the prediction target. To compute the prediction  $f(\mathbf{X})$  one must first sample  $\mathbf{X}_{\bar{S}}$  of the “out-coalition” features  $\bar{S}$ . The standard way [19] to do this is to sample  $\mathbf{X}_{\bar{S}}$  from a conditional distribution  $P(\mathbf{X}_{\bar{S}}|\mathbf{X}_S = \mathbf{x}_S)$ , where  $\mathbf{x}_S$  are the realized values of  $\mathbf{X}_S$ . This is done in the inner expected value where we marginalize the out-coalition features  $\bar{S}$ . This sampling procedure assumes feature independence which can lead to spurious explanations and misrepresenting feature dependencies [14].  $v_f(S)$ , therefore calculates the prediction quality if only the values of the features  $S$  are known by averaging over the features  $\bar{S}$  and this as an average over an entire data set. For the average over the entire data set the outer expected value is used. In Sect. 3 we describe how we overcome these shortcomings, i.e., the independence assumption, by considering causal models of the data.

### 3 Causality-Aware Global Explanations

In this section, we introduce our causality-aware global importance measure. In addition, we show that this measure has some desirable (causal) properties (cf. Theorem 1). And last, we provide an approximation algorithm for computing it.

#### 3.1 Global Causal Shapley Values

Considering the complexity of many real-world systems, it is unlikely that the independence assumption of [5] in the global explanation methods holds in general, hence it can lead to spurious explanations [14]. For that reason, we propose a global explanation method that considers causal dependencies when sampling out-coalition features, following the recent works on computing causality-inspired feature importance for local explanations [10, 12, 13]. For our sampling procedure, we assume a causal graph to be given. More specifically, we sample the out-coalition features  $\mathbf{X}_{\bar{S}}$  from a post-interventional distribution (after intervening on the known features of interest  $\mathbf{X}_S$ ) instead of a conditional distribution i.e.,  $P(\mathbf{X}_{\bar{S}}|do(\mathbf{X}_S = \mathbf{X}_S))$  instead of  $P(\mathbf{X}_{\bar{S}}|\mathbf{X}_S = \mathbf{X}_S)$ . This leads us to a sampling procedure from the post-interventional distribution, resulting in the following causal value function:

$$v_f(S) = -\mathbb{E}_{\mathbf{X}Y}[\mathcal{L}(\mathbb{E}_{\mathbf{X}_{\bar{S}}}[f(\mathbf{X})|do(\mathbf{X}_S = \mathbf{x}_S)], Y)], \quad (3)$$

which is determined by marginalizing the out-coalition features  $\bar{S}$  from the post-interventional distribution [10]:

$$\mathbb{E}[f(\mathbf{X})|do(\mathbf{X}_S = \mathbf{X}_S)] = \int f(\mathbf{X}_{\bar{S}}, \mathbf{X}_S)P(\mathbf{X}_{\bar{S}}|do(\mathbf{X}_S = \mathbf{x}_S))d\mathbf{X}_{\bar{S}}. \quad (4)$$

Through this intervention and by marginalizing the out-coalition features, we ensure the independence of the in-coalition features.

### 3.2 Properties of Global Causal Feature Importance

Our causal feature importance measure comes with a set of theoretical guarantees, that have been introduced in Jung et al. [13]. Intuitively, these are the desirable properties which ensure *causal soundness* of such measure. We present them below first, and then show that they are satisfied also in our global method CAGE.

**P1 Perfect assignment:** The global causal contributions are perfectly assigned if  $\mathbb{E}[\mathbb{I}(1)] - \mathbb{E}[\mathbb{I}(0)] = \sum_{i \in D} \phi_i$  where  $\mathbb{I}(1)$  and  $\mathbb{I}(0)$  correspond to the loss in Eq. (3) with intervention on all features and no intervention, respectively.

**P2 Causal irrelevance:** If  $X_i$  is causally irrelevant to  $Y$  for all  $\mathbf{W} \subseteq \mathbf{X} \setminus \{X_i\}$  s.t.  $\forall y, P(y \mid do(X_i, \mathbf{W})) = P(y \mid do(\mathbf{W}))$ , then  $\phi_i = 0$ , i.e., if a feature does not have any causal predictive power then CAGE value is 0.

**P3 Causal symmetry:** If  $X_i, X_j \in \mathbf{X}$  have the same causal contribution to the predictive power of  $Y$  for all  $\mathbf{W} \subseteq \mathbf{X} \setminus \{X_i, X_j\}$  s.t.  $\forall y, P(y \mid do(X_i, \mathbf{W})) = P(y \mid do(X_j, \mathbf{W}))$ , then  $\phi_i = \phi_j$ , i.e., if two features have the same causal predictive power then the features have the same CAGE value.

**P4 Causal approximation:** For any  $S \subseteq D$ :  $\forall i \in S$ ,  $\phi_i$  well approximates  $\mathbb{E}[Y \mid do(\mathbf{X}_S)]$  i.e.,  $\{\phi_i\}_{i=1}^n = \arg \min_{\{\phi'_i\}_{i=1}^n} \sum_{S \subseteq D} (\mathbb{E}[Y \mid do(\mathbf{X}_S)] - \sum_{i \in S} \phi'_i)^2 \omega(S)$  for some positive and bounded function  $\omega(S)$ .

Intuitively, P1 means that the sum of all causal feature contributions corresponds to the average treatment effect if we intervene on all features compared to no intervention. In particular, it captures how each feature contributes to the predictive power. P2 ensures that if a feature does not have any causal contribution to the predictive power then it has an importance value of zero. P3 means two features with the same causal predictive power have the same importance values. Posing the importance values as the solution to a weighted least square problem in P4 ensures that we can consider them as approximations of the causal effect.

**Theorem 1.** *CAGE is causally sound i.e., the derived values have properties P1 to P4.*

*Proof.* Let the value function  $v_f$  be defined as in Eq. (3).

To show perfect assignment (P1) i.e.,  $\sum_{i \in D} \phi_i = \mathbb{E}[\mathbb{I}(1)] - \mathbb{E}[\mathbb{I}(0)]$ , following [35] we can write

$$\phi_i(v_f) = \frac{1}{|D|!} \sum_{\pi \in \Pi(D)} \{v_f(\{i\} \cup Pre_\pi(i)) - v_f(Pre_\pi(i))\}$$

where  $\pi$  is a permutation from the set of all possible permutations of feature indices  $D$  and  $pre_\pi(i)$  is the predecessor of  $i$  in the permutation  $\pi$ . By summing all feature contributions we get

$$\begin{aligned}
\sum_{i=1}^{|D|} \phi_i(v_f) &= \frac{1}{|D|!} \sum_{\pi \in \Pi(D)} \sum_{i=1}^{|D|} \{v_f(\{i\} \cup \text{Pre}_\pi(i)) - v_f(\text{Pre}_\pi(i))\} \\
&= \frac{1}{|D|!} \sum_{\pi \in \Pi(D)} \{v_f(D) - v_f(\emptyset)\} \\
&= v_f(D) - v_f(\emptyset) \\
&= \mathbb{E}[-\mathcal{L}(\mathbb{E}[f(\mathbf{X})|\text{do}(\mathbf{X})], Y)] - \mathbb{E}[-\mathcal{L}(\mathbb{E}[f(\mathbf{X})|\text{do}(\emptyset)], Y)] \\
&= \mathbb{E}[\mathbb{I}(1)] - \mathbb{E}[\mathbb{I}(0)]
\end{aligned}$$

which shows the equality in P1.

To show causal irrelevance (P2), we assume  $X_i$  to have no causal contribution to the prediction of  $Y$  for all  $S \subseteq D \setminus \{i\}$ . Then, according to Eq. (3):

$$\begin{aligned}
v_f(S \cup \{i\}) &= -\mathbb{E}[\mathcal{L}(\mathbb{E}[f(\mathbf{X})|\text{do}(\mathbf{X}_S = \mathbf{x}_S, X_i = x_i)], Y)] \\
&= -\mathbb{E}[\mathcal{L}(\mathbb{E}[f(\mathbf{X})|\text{do}(\mathbf{X}_S = \mathbf{x}_S)], Y)] \\
&= v_f(S)
\end{aligned}$$

Hence, according to the definition of Shapley values (1)  $\phi_i = 0$ .

Although the proofs for P3 and P4 correspond to the ones in [13], we include them here for completeness and to provide an easy map to our notation. To show causal symmetry (P3) we assume the features  $X_i$  and  $X_j$  have the same causal predictive power. Therefore,

$$\begin{aligned}
\phi_i(v_f) &= \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i\}} \binom{|D| - 1}{|S|}^{-1} \{v_f(S \cup \{i\}) - v_f(S)\} \\
&= \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i, j\}} \binom{|D| - 1}{|S|}^{-1} \{v_f(S \cup \{i\}) - v_f(S)\} \\
&\quad + \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i, j\}} \binom{|D| - 1}{|S| + 1}^{-1} \{v_f(S \cup \{i, j\}) - v_f(S \cup \{j\})\} \\
&= \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i, j\}} \binom{|D| - 1}{|S|}^{-1} \{v_f(S \cup \{j\}) - v_f(S)\} \\
&\quad + \frac{1}{|D|} \sum_{S \subseteq D \setminus \{i, j\}} \binom{|D| - 1}{|S| + 1}^{-1} \{v_f(S \cup \{i, j\}) - v_f(S \cup \{i\})\} \\
&= \frac{1}{|D|} \sum_{S \subseteq D \setminus \{j\}} \binom{|D| - 1}{|S|}^{-1} \{v_f(S \cup \{j\}) - v_f(S)\} = \phi_j(v_f)
\end{aligned}$$

Causal approximation (P4) follows directly from [13], since the proof is on the level of the value function  $v_f$  the proof still holds for our global value function and our specific sampling procedure.  $\square$

By taking into account the causal structure of the features and the guarantees that Theorem 1 provides, we develop a method that also respects the properties of the additive global feature importance class. By interpreting the global importance of a feature by its causal contribution it has to the predictive performance our method closes a gap in explainability. Answering the questions to what extent a feature is a cause for an ML model to have a good performance differs significantly from previous Shapley-based explainability methods as they either do not measure causal contributions [5, 19] or they do not measure the contribution to the predictive performance [10, 13]. Additionally, it is worth mentioning that *uniqueness* directly follows from the fact that it is based on Shapley value, and since it satisfies soundness (as showed in [13]).

### 3.3 Computing Causal Shapley Values

Calculating the Shapley values for each feature  $X_i$  presents some practical challenges: (1) To compute the post-interventional distributions with our method the causal structure of the dataset must be known. (2) The post-interventional distribution must be transformed in an observational distribution to sample from it. (3) Computing exact Shapley values is a problem with exponential runtime because there are an exponential number of subsets  $S$  of features  $D$  over which we must iterate. In this Section, we introduce a pragmatic approach to handle these challenges to develop our global explanation method.

**Prior Knowledge on Causal Graphs.** A main assumption of computing causal Shapley values is that the causal structure is provided. This is a serious prerequisite since structural causal discovery is a challenging task itself. There are algorithms that are able to infer a structural causal model from data [36] and experiment-based approaches that identify the causal structure, but it is hardly realistic to infer a fully specified model with all possible confounders in general [32]. To alleviate this issue, we use causal chain graphs to calculate the feature importance. This allows us to have fewer assumptions on the causal structure of the features whenever the structure is partly unknown.

**Sampling Out-Coalition Features.** The second challenge is to estimate  $\mathbb{E}[f(\mathbf{X}) | do(\mathbf{X}_S = \mathbf{x}_S)]$  by sampling from  $P(\mathbf{X}_{\bar{S}} | do(\mathbf{X}_S = \mathbf{x}_S))$  according to Eq. (4), since we assume no interventional data. To tackle this challenge we resort to the factorization of the post-interventional distribution for causal chain graphs [10]:

$$\begin{aligned}
 P(\mathbf{X}_{\bar{S}} | do(\mathbf{X}_S = \mathbf{x}_S)) &= \prod_{\tau \in T_{\text{confounding}}} P(\mathbf{X}_{\tau \cap \bar{S}} | \mathbf{X}_{Pa_j \cap \bar{S}}, \mathbf{X}_{Pa_j \cap S}) \\
 &\times \prod_{\tau \in T_{\text{non-confounding}}} P(\mathbf{X}_{\tau \cap \bar{S}} | \mathbf{X}_{Pa_j \cap \bar{S}}, \mathbf{X}_{Pa_j \cap S}, \mathbf{X}_{\tau \cap S}) \quad (5)
 \end{aligned}$$

Equation (5) makes the distinction between confounded and not confounded chain graph components  $\tau$ . For chain components with confounded variables, the first part of the Equation is used. If the dependencies in a component are only due to mutual interactions between the variables the second part should be used. In contrast to Eq. (4),

we can now use discrete marginalization, since the causal graphs fulfill the Markov conditions.

**Approximation Algorithm.** Lastly, we combine the above insights to derive an algorithm that computes causality-aware Shapley additive importance measures for global explanations. Structurally, our algorithm follows the commonly used approach to approximate global importance values [6], which guarantees that the algorithm converges to the true  $\phi_i$  values in the limit. The novelty of our algorithm lies in the way we sample the out-coalition features. Instead of sampling them from the conditional, observational distribution, we sample values that adhere to the causal structure of the data as described above. The pseudocode of the overall algorithm can be seen in Algorithm 1.

The algorithm requires various inputs including the dataset, a partial causal order represented as a causal chain graph, and information about confounded components or interactions in the chain graph. The algorithm works in such a way that it computes the average of many samples (Line 2) of the expression  $\mathcal{L}(\mathbb{E}[f(\mathbf{X})|do(\mathbf{X}_{S \cup i} = \mathbf{x}_{S \cup i})], Y) - \mathcal{L}(\mathbb{E}[f(\mathbf{X})|do(\mathbf{X}_S = \mathbf{x}_S)], Y)$  (Line 22) which corresponds to  $v(S \cup i) - v(S)$  in Eq. (1).

During each iteration, a data instance and a feature permutation are randomly chosen (Line 3), initiating the additive process (Line 6). This process involves incrementally adding the next feature  $j$  of the permutation to the in-coalition features  $S$  (Line 7). The pivotal CAGE causal sampling procedure commences in Lines 9 and 10, where a batch of size  $M$  is drawn, followed by iterating over each component of the causal chain graph  $\tau$  in their causal ordering.  $|T|$  denotes the number of components in the causal chain graph.

If the features in a component  $\tau$  are confounded then each data point  $x_i^m$  can be drawn independently (Line 13). If all feature dependencies in a chain component are induced by mutual interaction we use Gibbs sampling [9] to draw the features  $\mathbf{x}_{\tau_i \cap \bar{S}}^m$  (Line 16). All sampled missing feature values  $\mathbf{x}_S^m$  are then used in Line 19 for prediction. These sampling procedures were introduced in Eq. (5) and proved by [10]. The impact of the additional feature  $j$  on the predictive performance represented by the difference in the loss with and without feature  $j$  (Line 21, 22) is then added to the cumulative CAGE value in Line 23.

## 4 Experiments

To evaluate our causal explanation framework and to compare it with other approaches, we will conduct several experiments. First, we will perform experiments on synthetic datasets. Then, we will apply our framework to a real-world example. We compare our global causality-aware explanation framework with the existing global explanation method *SAGE*.



**Algorithm 1:** Approximation algorithm for CAGE values.

**Input:** data  $\{\mathbf{x}^k, y^k\}_{k=0}^K$ , model  $f$ , loss  $\mathcal{L}$ ,  $N$  (outer samples),  $M$  (inner samples), causal chain graph  $G$ , confounding, feature indices  $D$  with dimension  $d$

**Output:** shapley values  $\frac{\phi_1}{N}, \dots, \frac{\phi_d}{N}$

```

1  $\phi_1 = \dots = \phi_d = 0$ 
2 for  $i = 1$  to  $N$  do
3   Sample  $(\mathbf{x}, y)$  from  $\{\mathbf{x}^k, y^k\}_{k=0}^K$  and permutation  $\pi$  of  $D$ 
4    $S = \emptyset$ 
5    $lossPrev = \mathcal{L}(\frac{1}{K} \sum_{k=1}^K f(\mathbf{x}^k), y)$ 
6   for  $j = 1$  to  $d$  do
7      $S = S \cup \{\pi[j]\}$ 
8      $\hat{y} = 0$ 
9     for  $m = 1$  to  $M$  do
10      for  $t = 1$  to  $|T|$  do
11        if  $confounding(\tau_t)$  then
12          for  $l \in \tau_t \cap \bar{S}$  do
13             $x_l^m \sim P(X_l | \mathbf{X}_{P_{a_t} \cap \bar{S}}, \mathbf{X}_{P_{a_t} \cap S})$ 
14          end
15        else
16           $\mathbf{x}_{\tau_t \cap \bar{S}}^m \sim P(\mathbf{X}_{\tau \cap \bar{S}} | \mathbf{X}_{P_{a_t} \cap \bar{S}}, \mathbf{X}_{P_{a_t} \cap S}, \mathbf{X}_{\tau \cap S})$ 
17        end
18      end
19       $\hat{y} = \hat{y} + f(\mathbf{x}_S, \mathbf{x}_{\bar{S}}^m)$ 
20    end
21     $loss = l(\frac{\hat{y}}{M}, y)$ 
22     $\Delta = lossPrev - loss$ 
23     $\phi_\pi[j] = \phi_\pi[j] + \Delta$ 
24     $lossPrev = loss$ 
25  end
26 end
27 return  $\frac{\phi_1}{N}, \dots, \frac{\phi_d}{N}$ 
28
```

## 4.1 Experiments on Synthetic Data

*Experimental Setup.* To ensure that we can assess which features are most important we conducted experiments with synthetic datasets. For these datasets, the data-generating process including its causal structure is completely known. We created three datasets with different causal structures. The first dataset only consists of independent direct causes that have the same influence on the target variable. The second dataset is of Markovian nature in the sense that we have one variable that is completely determined by its parents. Therefore, the variable is conditionally independent of its non-descendants, given its parents [8]. The third dataset is a mixed model consisting of both causal structures from above. This means there are causal dependencies between some variables but also direct independent variables. All three datasets are generated from

structural causal models where the variables are sampled as linear combinations of the parents and pairwise independent noise terms. The exact specification of the SCMs can be found in Appendix A. The graphs in Figs. 2a–2c show the causal graphs induced by the SCMs. Furthermore, they show the topological ordering translated into chain graphs as the causal knowledge for our experiments.

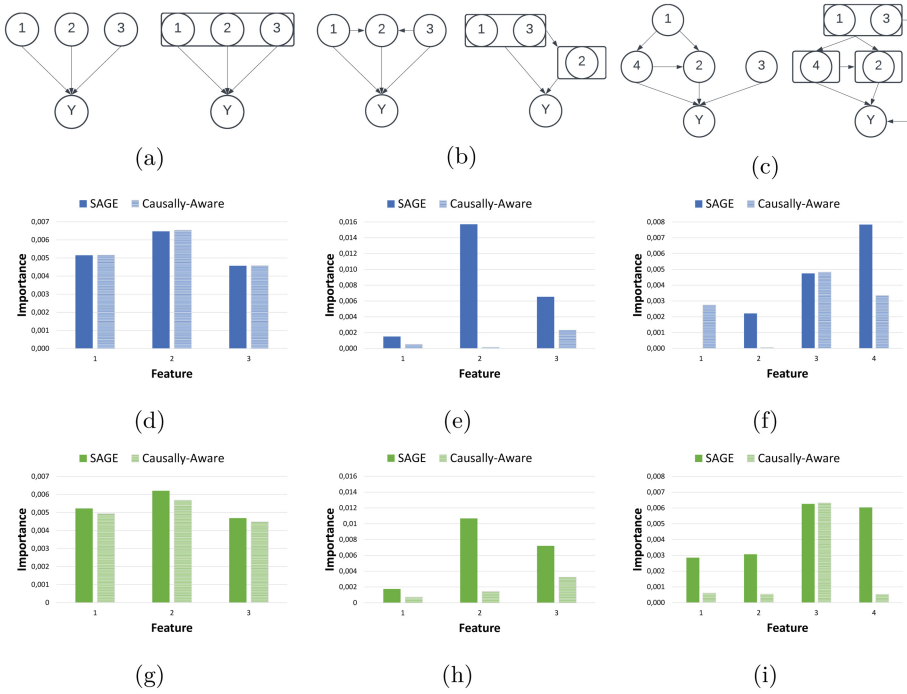
For each dataset, we fit a linear regression model and a simple multi-layer perceptron (MLP). To implement both model types we used the *scikit-learn* library with default values. Then we apply conventional SAGE and our causality-aware global explanation framework and compare the explanations.

*Results.* Figure 2 shows the explanatory results of the causal structures for both methods, the linear regression model (Figs. 2d, 2e, 2f) and the MLP (Figs. 2g, 2h, 2i). In the plots, the striped bars depict our causal explanation method the solid bars depict the results when applying SAGE.

SAGE explains the feature importance for the linear regression models solely on how the target variable is built. Variables that have the highest coefficient in the deterministic function of  $Y$  get the highest importance. In contrast, our causally aware framework takes the causal structure into account. It assigns the importance of features based on their causal contribution to the target variable. For example in the Markovian dataset, even though variable 2 has the highest coefficient in the linear model of  $Y$  it gets assigned the lowest importance score because it can be completely explained by variables 1 and 3. This also applies to the Mixed-data structure. If there are independent features that do not have any causal relation with other features then the importance of that feature is the same for both explanation methods.

In general, the following characteristics can be observed in the explanations for the linear regression models: First, if there is a variable that can be completely explained by other variables, i.e. the causal structure is clear, then this variable does not get any importance. This is in line with the causal irrelevance property introduced in Sect. 3.2. Second, if there are independent variables that are direct causes of the target then these variables have the same importance in both frameworks. Third, variables that are causes and effects at the same time get a reduction in importance but not a total deletion of importance.

The explanations of the MLP show a more nuanced picture. For the Markovian dataset, SAGE assigns feature importance similar to the linear regression model but for the mixed model dataset, it does not assign importance according to the linear model of  $Y$ . Our causally-aware method also shows different feature importance compared to the linear regression model. Features that can be completely explained by other features receive reduced feature importance but not a complete deletion of importance. For example, in the Markovian data experiment variable 2 gets reduced in importance but still gets assigned some importance, even though variable 2 is just a linear combination of variable 1 and 3. Furthermore, in the mixed-model data variables which are effects of other variables are reduced in importance. However, the root cause, variable 1, only gets small importance even though it has a high impact on the target by being the cause of variables 2 and 4. Nevertheless, the experiment with independent features shows the same results. We can see that the MLP is only able to use some of the causal informa-



**Fig. 2.** Results and data-generating causal structures for our experiments. The first row (Figs. 2a, 2b, 2c) show the true causal structure (left) of the data-generating SCMs and the corresponding causal chain graphs we use for the explanation (right). The second row (Figs. 2d, 2e, 2f, blue) shows the importance values determined for the linear regression models that were trained and evaluated on the causal structures above. The third row (Figs. 2g, 2h, 2i, green) shows the same information, but for the MLP models. The solid bars show values coming from SAGE, and the striped bars show values of our causally-aware global explanation method.

tion that is provided but the relationship of using the causal structure for explanation is weakened.

## 4.2 Explanations on Alzheimer Data

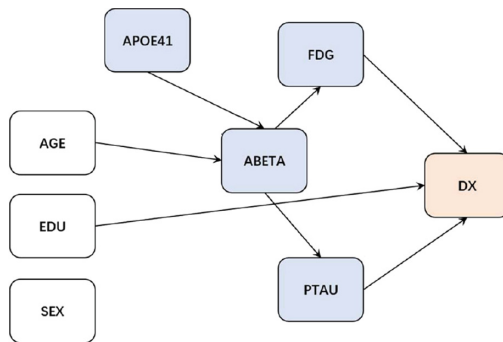
Following the promising results of the synthetic datasets, in this Section we explore the distinctions between the explanations SAGE derives and our causality-aware explanations when applied to a real-world dataset.

**Data.** To apply our framework to a real-world dataset it is necessary that we know the causal structure or at least a partial causal ordering of the features. For this experiment, we chose the Alzheimer’s Disease Neuroimaging Initiative (ADNI) dataset (adni.loni.usc.edu) because the causal structure has been investigated for this dataset [11]. The ADNI collects data from researchers to investigate the progression of Alzheimer’s disease. The data includes MRI images, genetics, cognitive tests, and

biomarkers as predictors of the disease. For this experiment, we do not use all features that the dataset offers but only the biomarkers fludeoxyglucose (*FDG*), amyloid beta (*ABETA*), phosphorylated tau (*PTAU*), and the number of apolipoprotein alleles (*APOE4*) additionally the age, gender, and education level as features, for simplified analysis are added.

*Experimental Setup.* We define a binary classification problem to use these seven features and predict if a person has Alzheimer’s or not. For this classification problem, we build two models. First a simple multilayer perceptron (MLP) with five layers (*layer sizes*= 64, 128, 128, 64, 32) and *Adam* optimization. Second a Random Forest with 200 trees. After cleaning and normalizing the dataset it consists of 1500 instances which we split into 25% test and 75% training sets (Details on data handling can be found in the code repository). The trained models have an accuracy of 85% for the MLP and 88% for the Random Forest on the test set.

We analyze the causal structure of the features by referring to the research of [32]. This study contrasts various causal structure discovery (CSD) algorithms and compares the resulting structures to a gold standard graph shown in Fig. 3. For our experiment, we use this gold standard graph which is based on biological and medical studies on Alzheimer’s risk factors. Based on the gold standard, we define the partial causal ordering from this graph as the topological ordering: [(*AGE*, *EDU*, *SEX*, *APOE4*), (*ABETA*), (*FDG*, *PTAU*)], to show the effectiveness of our approach when only the partial causal structure is provided. Additionally, we assume confounding in the first and the third chain graph components. The resulting causal chain graph for our experiment is shown in Fig. 1.

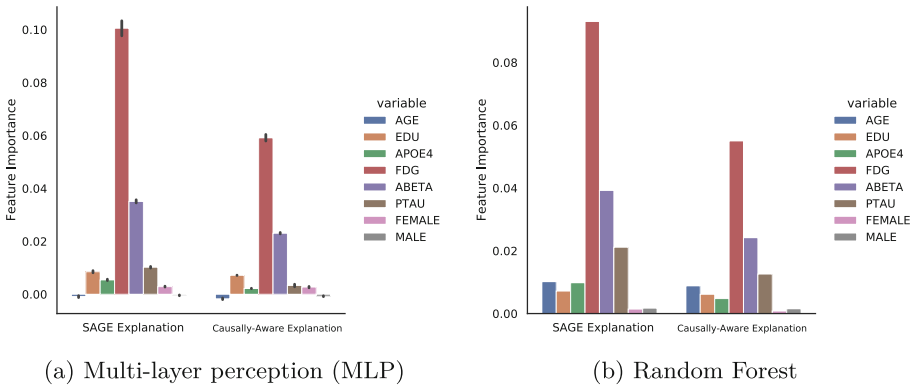


**Fig. 3.** Gold Standard Graph from [32]. The gold standard graph shows the causal relations between the seven features and the binary target variable *DX*. Blue nodes are biomarkers and white nodes are personal information about patients. From that, we derive the causal chain graph in Fig. 1.

*Results.* Figure 4 depicts, the results of the experiment with the MLP (Fig. 4a), and the results of the Random Forest (Fig. 4b). The left bars show the explanation of *SAGE* and the right bars the explanation of our causality-aware explanation framework. When

comparing the two ML methods, only minor differences can be observed. More specifically, there are two dissimilarities in which the models have different feature importance rankings. These two are the feature *EDU* (education) and the feature *APOE4*. Both are more important for the Random Forest classifier. There are other differences in the importance but they do not have an effect on the importance ranking. One noticeable is that the importance of *PTAU* is higher for the Random Forest.

When comparing the two explanatory methods the differences are minor. There are not as extreme differences in the ranking as there are with the synthetic data. However, there are mentionable changes. For example, we observe that the importance of features that are effects of other features, e.g. *PTAU* and *FDG* are reduced more compared to other features in the causality-aware explanations.



**Fig. 4.** Importance values of the ADNI data experiment. The left plot shows the feature importances for the MLP model and the right plot the feature importance of Random Forest. For each plot, the left set of bars shows the importance determined by SAGE and the right bars show the importances for our causality-aware global explanation method.

In Sect. 6, we provide a detailed analysis and discussion of the possible reasons for the observed differences between the frameworks and discuss the results in more detail.

## 5 Related Work

**Local Explanation.** Local explanations gravitate towards elucidating individual predictions and unraveling the distinctive importance attributed to features for an individual instance [19, 26]. A noteworthy methodology leveraging local explanation is the utilization of Shapley values [31]. Innovatively adapted from game theory, Shapley values have been employed to measure feature contribution towards a model’s output, as showcased in frameworks like SHAP [19]. Since its inception, SHAP has evolved, adapting to a diverse array of tasks and explanatory objectives through various extensions and modifications. Notable among these are *KernelSHAP* [4, 19], *TreeSHAP* [18], and *LossSHAP* [5, 18]. KernelSHAP, a model-agnostic explainable method, is appreciated

for its adaptability across numerous model types. In contrast, TreeSHAP is specially tailored for tree models, offering dedicated insights into tree-based model interpretations. LossSHAP diverges in its explanatory focus. Rather than adhering to traditional approaches, it emphasizes the influence of features based on evaluation metrics. For instance, it evaluates the importance of a feature by analyzing its impact on specific evaluation criteria such as the mean squared error in regression contexts.

**Causal Local Explanation.** An evolution in local explanations is witnessed in the integration of causality, fostering a more nuanced and reliable interpretation. Aas et al. [1] extend the *KernelSHAP* method so that it can handle highly correlated features. Another line of studies investigates how feature dependencies and *Shapley values* can be interpreted from a causal perspective. Frye et al. [7] present *Asymmetric Shapley values* where they incorporate causal knowledge by only allowing possible permutations of features that comply with the causal structure of the input features when computing *Shapley values*. Janzing et al. [12] tackle the question of how to deal with out-coalition features in *SHAP*-based methods. They replace conditional sampling in the *Shapley value* computation with conditioning by intervention with do-calculus. Similarly, Hesses et al. [10] and Jung et al. [13] use do-calculus to compute the causal contribution of a feature to the models' prediction. *Shapley Flow* is a *Shapley value*-based method that explains model predictions from a causal perspective. The authors suggest not to assign importance to variables in the causal graph but to assign importance to the edges of the causal graph [37]. Another interpretation of causal feature importance is given by abductive explanations which generate a minimal subset of features that are sufficient for the prediction [2]. This interpretation is closely linked to the causal strength quantification notion of [3].

**Global Explanation.** Global explanations pivot towards explaining the entire model mechanism by, for instance, providing the most important features for the model to make a prediction. SAGE (Shapley Additive Global Explanations) [6] emerges as a quintessential global explanation methodology. SAGE introduces additive importance measures as a similar class of methods like additive feature attribution methods [19]. In this class, the importance of a feature is defined as the predictive power that it contributes rather than the absolute effect it has on the prediction. This means that *SAGE* measures if a feature makes a prediction more or less correct, according to evaluation metrics, whereas *SHAP*-based methods measure the pure change that features have on the prediction. *SAGE* is, therefore, the global equivalent of *LossSHAP*. For an in-depth review of global XAI methods, we refer to [28].

## 6 Discussion

In our study, we embarked on an exploration of causal global explanation methods. We hypothesized that these methods, grounded in causal foundations, assign importance to features in a manner that is more congruent with their actual causal contributions towards model predictions, as opposed to the *SAGE* framework which lacks a causal basis.

Our results with linear regression on synthetic datasets substantiate this hypothesis. By leveraging causal knowledge, our method assigns importance to features more intuitively, yielding explanations that closely mirror actual causal feature contributions to predict the target variable. This precision is attributable to the availability of structured causal models (SCMs), clarifying feature constructions and contributions. Remarkably, the proposed method assigns minimal importance to features that significantly contribute to the target but are the effects of other features, aligning the explanations with actual causal contributions to the predictive power.

Interestingly, our causal framework exhibited a propensity to assign significance to root causes within the SCM, even in scenarios where these root causes did not exert direct influences on the target. This observation is pivotal, aligning with human cognitive patterns in causality attribution, and echoes theories suggesting humans evaluate each event within a causal chain based on its impact on the outcome [17, 33, 34]. The crediting causality hypothesis suggests that all events in a causal chain are evaluated on how much they change the outcome. This leads to the fact that in simpler mechanisms the root cause is often given as one of the main causes [34].

A salient observation is the causal method's tendency to attribute reduced absolute importance values compared to the traditional SAGE framework. This discrepancy may stem from algorithmic calculations and the dataset's structural composition, necessitating cautious inter-framework comparisons. Furthermore, the causal sampling method, by reducing outliers, may contribute to lower absolute importance values.

We additionally evaluated our method on a real-world example. As expected the results of this experiment were not so clear as for the synthetic examples. As a preliminary, it should be noted that Alzheimer's is still a rather unexplored disease and the underlying mechanisms are not fully understood. This is also how the gold standard graph, which is used as a basis for causal knowledge, can be classified. The graph is based on biological and medical observational analyses of the ADNI dataset in which the true causal mechanisms are not fully known. This means that the possibility of unobserved confounding needs to be assumed.

Nevertheless, we discuss our results for the real-world data based on the characteristics we have developed for the synthetic data. A repeating pattern that can be observed both in the synthetic data experiments and in the real-world application is that features that are solely effects of other features have a reduced feature importance. This is in line with the characteristics of our causality-aware explanation method that we put forward above. However, the concentration of feature importance on the root cause cannot be observed in real-world data experiments. We attribute this to these possible reasons: First, the model is not able to learn and use the causal structure. Models like MLP and Random Forests are high-dimensional ML models that only learn statistical correlations. Augmenting their explanation with causal knowledge does not necessarily mean that the models actually rely on it. Second, we do not provide complete causal ordering but only partial causal ordering with chain graphs. This means that some causal knowledge is lost and cannot be exploited for the explanation. Signs supporting these arguments can be observed in the explanations of the MLP with mixed-model synthetic data in Sect. 4.1. There, where the causal relations are more complicated, we observe the patterns of the real-world data application.

The analysis of the results thus reveals some interesting aspects. For explaining simple models like linear regression models the causal structure can be used to almost exactly represent the causal contributions of features. If the models become more high-dimensional this capability becomes less. Due to the fact that high-dimensional models tend to learn merely statistical correlations, we suspect that the causal information we provide is lost at the global level of explanation.

The primary challenge of our global causal explanation method lies in requiring a predefined causal structure for features, a difficult task as determining causality itself is an ongoing research area [36]. While we utilize causal chain graphs, their practicality diminishes with an increasing number of features, complicating the division into chain components. This complexity was evident in our ADNI dataset experiment, where unclear causal structures and minimal causal effect strengths made the results and their interpretations ambiguous. Users must scrutinize the causal structure's origin, effect strengths, and feature classifications in chain components for valid interpretations.

To the best of our knowledge, our work is the first one that incorporates causal knowledge into global Shapley - value-based explanation methodologies. Preliminary comparisons with local explanation methods [1, 13] indicate a consensus, underscoring the enhancement of explanatory accuracy and coherence when causal structures are incorporated. The proposed method demonstrated similar results and improvements, justifying our results on global-level explanations.

## 7 Conclusion

In this paper, we propose CAGE, a causality-aware global additive explanation framework based on Shapley values. We show that it is able to generate explanations that align with desirable causal properties, and outlined an algorithm for estimating its values. To this end, we introduced a novel sampling procedure for out-coalition features that respects their causal relation. Most notably, in contrast to previous global explanation approaches, our approach takes away the burden of the independence assumption among input features. Application of CAGE to both synthetic and real-world datasets shows that CAGE respects the causal relations of input features while explaining predictive models. We argue that this leads to more intuitive and faithful explanations of AI.

In future work, causal explanation methods based on Shapley values should investigate how the strong prerequisite that the causal structure of the features must be given can be overcome. This is the basis for the widespread use of causal explanation methods. Important points of orientation for this could be studies that investigate causal reasoning and causal learning under uncertainty or partially confounded settings.

**Acknowledgements.** Nils Ole Breuer works in the gemeinwohlorientierter KI-Anwendungen (Go-KI) project (Offenes Innovationslabor KI zur Förderung gemeinwohlorientierter KI-Anwendungen), funded by the German Federal Ministry of Labour and Social Affairs (BMAS) under the funding reference number DK1.00.00032.21.



This research was partially funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, <https://hybridintelligence-centre.nl>, grant number 024.004.022.

**Disclosure of Interest.** We have no conflicts of interest to disclose.

## A Data - Generating Causal Models

These are the structural causal models used for the data-generation process of the synthetic data experiments used in Sect. 4.1.

### A.1 Direct-Cause structure

The SCM that induced the graph in Fig. 2a:

$$\begin{aligned} N &= \mathcal{N}(0, 1) \\ X_1 &= \mathcal{N}(0, 1) \\ X_3 &= \mathcal{N}(0, 1) \\ X_2 &= \mathcal{N}(0, 1) \\ Y &= X_1 + X_2 + X_3 + N_Y \end{aligned}$$

### A.2 Markovian Structure

The SCM that induced the graph in Fig. 2b:

$$\begin{aligned} N &= \mathcal{N}(0, 1) \\ X_1 &= \mathcal{N}(1.5, 1) \\ X_3 &= \mathcal{N}(0.5, 2) \\ X_2 &= X_1 + X_3 + N_{V_2} \\ Y &= X_1 + 2X_2 + X_3 + N_Y \end{aligned}$$

### A.3 Mixed structure

The SCM that induced the graph in Fig. 2c:

$$\begin{aligned} N &= \mathcal{N}(0, 1) \\ X_1 &= \mathcal{N}(1.5, 1) \\ X_3 &= \mathcal{N}(0.5, 2) \\ X_4 &= X_1 + N_{V_4} \\ X_2 &= X_1 + X_4 + N_{V_2} \\ Y &= 0.3X_2 + X_3 + 2X_4 + N_Y \end{aligned}$$

## References

1. Aas, K., Jullum, M., Løland, A.: Explaining individual predictions when features are dependent: more accurate approximations to Shapley values. *Artif. Intell.* **298**, 103502 (2021)
2. Biradar, G., Izza, Y., Lobo, E., Viswanathan, V., Zick, Y.: Axiomatic aggregations of abductive explanations. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, pp. 11096–11104 (2024)
3. Chockler, H., Halpern, J.Y.: Responsibility and blame: a structural-model approach. *J. Artif. Intel. Res.* **22**, 93–115 (2004)
4. Covert, I., Lee, S.I.: Improving Kernelshap: practical Shapley value estimation using linear regression. In: *International Conference on Artificial Intelligence and Statistics*, pp. 3457–3465. PMLR (2021)
5. Covert, I., Lundberg, S., Lee, S.I.: Feature removal is a unifying principle for model explanation methods. arXiv preprint [arXiv:2011.03623](https://arxiv.org/abs/2011.03623) (2020)
6. Covert, I., Lundberg, S.M., Lee, S.I.: Understanding global feature contributions with additive importance measures. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 17212–17223 (2020)
7. Frye, C., Rowat, C., Feige, I.: Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 1229–1239 (2020)
8. Geiger, D., Pearl, J.: On the logic of causal models. In: *Machine Intelligence and Pattern Recognition*, vol. 9, pp. 3–14. Elsevier (1990)
9. Gelfand, A.E.: Gibbs sampling. *J. Am. Stat. Assoc.* **95**(452), 1300–1304 (2000)
10. Heskes, T., Sijben, E., Bucur, I.G., Claassen, T.: Causal Shapley values: exploiting causal knowledge to explain individual predictions of complex models. *Adv. Neural. Inf. Process. Syst.* **33**, 4778–4789 (2020)
11. Jack, C.R., Jr., et al.: The Alzheimer’s disease neuroimaging initiative (ADNI): MRI methods. *J. Magnet. Resonance Imaging Off. J. Int. Soc. Magnet. Resonance Med.* **27**(4), 685–691 (2008)
12. Janzing, D., Minorics, L., Blöbaum, P.: Feature relevance quantification in explainable AI: a causal problem. In: *International Conference on Artificial Intelligence and Statistics*, pp. 2907–2916. PMLR (2020)
13. Jung, Y., Kasiviswanathan, S., Tian, J., Janzing, D., Blöbaum, P., Bareinboim, E.: On measuring causal contributions via do-interventions. In: *International Conference on Machine Learning*, pp. 10476–10501. PMLR (2022)
14. Kumar, I.E., Venkatasubramanian, S., Scheidegger, C., Friedler, S.: Problems with Shapley-value-based explanations as feature importance measures. In: *International Conference on Machine Learning*, pp. 5491–5500. PMLR (2020)
15. Langer, M., et al.: What do we want from explainable artificial intelligence (XAI)?-a stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research. *Artif. Intell.* **296**, 103473 (2021)
16. Lauritzen, S.L., Wermuth, N.: Graphical models for associations between variables, some of which are qualitative and some quantitative. *Ann. Stat.* 31–57 (1989)
17. Lewis, D.: Causation. *J. Philos.* **70**(17), 556–567 (1973)
18. Lundberg, S.M., et al.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 56–67 (2020)
19. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
20. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019)

21. Mittelstadt, B., Russell, C., Wachter, S.: Explaining explanations in AI. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 279–288 (2019)
22. Paul, D., Sanap, G., Shenoy, S., Kalyane, D., Kalia, K., Tekade, R.K.: Artificial intelligence in drug discovery and development. *Drug Discovery Today* **26**(1), 80 (2021)
23. Pearl, J.: Causal diagrams for empirical research. *Biometrika* **82**(4), 669–688 (1995)
24. Pearl, J.: The do-calculus revisited. arXiv preprint [arXiv:1210.4852](https://arxiv.org/abs/1210.4852) (2012)
25. Peters, J., Janzing, D., Schölkopf, B.: *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, Cambridge (2017)
26. Ribeiro, M.T., Singh, S., Guestrin, C.: “why should i trust you?” explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
27. Saeed, W., Omlin, C.: Explainable AI (XAI): a systematic meta-survey of current challenges and future opportunities. *Knowl.-Based Syst.* **263**, 110273 (2023)
28. Saleem, R., Yuan, B., Kurugollu, F., Anjum, A., Liu, L.: Explaining deep neural networks: a survey on the global interpretation methods. *Neurocomputing* **513**, 165–180 (2022)
29. Schölkopf, B.: Causality for machine learning. In: *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 765–804 (2022)
30. Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward causal representation learning. *Proc. IEEE* **109**(5), 612–634 (2021)
31. Shapley, L.S., et al.: A value for n-person games (1953)
32. Shen, X., Ma, S., Vemuri, P., Simon, G.: Challenges and opportunities with causal discovery algorithms: application to Alzheimer’s pathophysiology. *Sci. Rep.* **10**(1), 2975 (2020)
33. Sloman, S.: *Causal Models: How People Think About the World and its Alternatives*. Oxford University Press, Oxford (2005)
34. Spellman, B.A.: Crediting causality. *J. Exp. Psychol. Gen.* **126**(4), 323 (1997)
35. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **41**, 647–665 (2014)
36. Vowels, M.J., Camgoz, N.C., Bowden, R.: D’ya like dags? A survey on structure learning and causal discovery. *ACM Comput. Surv.* **55**(4), 1–36 (2022)
37. Wang, J., Wiens, J., Lundberg, S.: Shapley flow: a graph-based approach to interpreting model predictions. In: *International Conference on Artificial Intelligence and Statistics*, pp. 721–729. PMLR (2021)
38. Wang, Y., Yan, G.: Survey on the application of deep learning in algorithmic trading. *Data Sci. Finan. Econ.* **1**(4), 345–361 (2021)
39. Završnik, A.: Algorithmic justice: Algorithms and big data in criminal justice settings. *Eur. J. Criminol.* **18**(5), 623–642 (2021)

**Fairness, Trust, Privacy, Security,  
Accountability and Actionability  
in eXplainable AI**



# Exploring the Reliability of SHAP Values in Reinforcement Learning

Raphael C. Engelhardt<sup>1</sup> <sup>(✉)</sup>, Moritz Lange<sup>2</sup> , Laurenz Wiskott<sup>2</sup> ,  
and Wolfgang Konen<sup>1</sup> 

<sup>1</sup> Faculty of Computer Science and Engineering Science, Cologne Institute of  
Computer Science, TH Köln, Gumannsbach, Germany  
{Raphael.Engelhardt,Wolfgang.Konen}@th-koeln.de

<sup>2</sup> Institute for Neural Computation, Faculty of Computer Science, Ruhr-University  
Bochum, Bochum, Germany  
{Moritz.Lange,Laurenz.Wiskott}@ini.rub.de

**Abstract.** Explainable artificial intelligence (XAI) is an increasingly important research field, fueled by the need for reliability and accountability in applications. For reinforcement learning (RL), achieving explainability is particularly challenging because agent decisions depend on the context of a trajectory, which makes data temporal and non-i.i.d. In the field of XAI, Shapley values and SHAP in particular are among the most widely used techniques. In this work, we investigate how SHAP performs in explaining RL models, especially in multidimensional action spaces that other XAI-for-RL methods struggle with. In particular, we make three contributions: (1) We investigate how design choices of the SHAP approach affect SHAP accuracy for RL models. We investigate the size of the so-called background data that is utilized to represent absent features, as well as the selection method with which the background data is formed. We find that SHAP for RL requires only modest amounts of background data and that clustering is preferred over sampling as a selection method. (2) Additionally, we analyze how SHAP-based feature importance relates to overall agent performance (return). We find that while feature importance is often correlated to agent performance, notable exceptions occur, especially for environments that are sensitive or fragile in the sense that small changes in actions may lead to catastrophic failure. However, since a significant correlation is found in the majority of the investigated environments, SHAP proves to be a valuable XAI tool for RL with multidimensional, continuous actions. (3) Illustratively, we show the time evolution of SHAP values and caution against misinterpreting sharp changes therein.

**Keywords:** Reinforcement learning · Explainability · Shapley values · SHAP · XAI

## 1 Introduction

The issue of explainability in artificial intelligence (XAI) has been of increasing importance during the last years and was often cited [1,2] as one of the main

challenges when applying AI to real-world scenarios, especially in safety-critical fields. As a consequence, a variety of methods have been developed with the goal of increasing insights into opaque AI models. One of these methods is the SHAP framework [3], rooted in mathematical game theory [4].

XAI for reinforcement learning (RL), and in particular SHAP for RL, can be more challenging than XAI for supervised machine learning (ML). This is due to the temporal and non-i.i.d. nature of the data in RL, where decisions depend on the state of the environment. In the context of RL, agents with multidimensional actions pose a particular challenge for XAI, and this article examines in particular what contribution SHAP can make to this challenge.

## 1.1 Shapley Values

Named after Lloyd Shapley, Shapley values give a solution to the problem of fairly distributing a given payout among the cooperative players of a game [4]. In short, the Shapley value corresponds to a player’s marginal contribution to the possible coalitions or the expected performance gain when said player joins a coalition. Given a game with a set  $\mathcal{N}$  of  $n = |\mathcal{N}|$  cooperative players and a function  $v$  assigning a value to each coalition, player  $j$ ’s added contribution to coalitions  $\mathcal{S}$  is given by

$$\phi_j = \frac{1}{n} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{j\}} \binom{n-1}{|\mathcal{S}|}^{-1} (v(\mathcal{S} \cup \{j\}) - v(\mathcal{S})) \quad (1)$$

It can be mathematically shown that Shapley values are the only method with a variety of desirable properties (efficiency, symmetry, dummy, and additivity) that lead to a payout distribution that can be called “fair” [5]. For the exact computation,  $2^{n-1}$  values of coalitions containing a specific player must be compared to  $2^{n-1}$  values of coalitions without the player, hence the cost is exponential in the number  $n$  of players.

## 1.2 Shapley Values for ML – SHAP

In the transition from game theory to machine learning, the prediction of an ML model takes the place of the value function, while the input features take the role of the single players. For large numbers of features, the exact computation of Shapley values suffers from combinatorial explosion and is generally not feasible. The framework SHAP [3] offers a variety of different approximation methods (one of them being KernelSHAP, which will be explained in more detail in Sect. 4).

The framework has seen a remarkable success, has been expanded with different approximation methods optimized for certain ML models as well as visualization tools, and has often been cited as the go-to approach for model-agnostic explainability of ML models.

When used to explain ML decisions, the first examples are typically the explanation of single decisions (think of the often-used example of a denied

bank-loan) or the discovery of general trends in classification examples. When applied to RL, the prediction of the ML model is the action performed by the agent in the environment, while the features are the observables accessible to the agent. The words “features” and “observations” can be used interchangeably; the terms “prediction” and “action” are also used as synonyms in the following. In this setting, the SHAP values are the contribution of each feature towards the model’s prediction, so that the RL agent’s action is the sum of its average action<sup>1</sup> and the SHAP values of all observables. We note in passing, that taking the RL agent’s action as the value function  $v$  in Eq. (1) for the SHAP procedure is not the only possibility. Alternatively, one could also take the episode return as a possible value function  $v$ . However, since this has higher computational demands, it was not considered in this work, but left for future work instead.

Compared to supervised ML classification and regression tasks, where the data can usually be assumed to be i.i.d., RL has structural differences. Given the interaction between agent and environment, RL data are usually non-i.i.d.: Decisions depend on the state of the environment and the overall success is determined by a sequence of profitable actions. In addition, complex environments require the agent to perform multidimensional actions at each timestep. As a result, each action dimension has its own set of SHAP values. Aggregating this multitude of values and ensuring the meaningfulness of SHAP for RL is a non-trivial task.

### 1.3 Contributions

The main contributions of this work are summarized as follows: (1) We empirically test the effect of the quantity of background data<sup>2</sup> and the method of background data selection on the computed SHAP values, and relate the results to the computation time. This evaluates the robustness of the approximation method and might help practitioners to better find the appropriate compromise between precision and computation time. (2) We expand the known definition of SHAP feature importance to the case of multidimensional actions and evaluate the computed feature importance on the RL task. (3) We interpret the time evolution of SHAP values throughout the episode in the context of the agent’s actions.

In Sect. 2, we discuss the related work on SHAP for RL. Section 3 describes the RL environments we use as benchmark. In the three follow-up Sect. 4, 5, 6, we describe our experiments and discuss results regarding the three main objectives given above. Finally, we draw a conclusion and hint at possible future work in Sect. 7.

---

<sup>1</sup> When using KernelSHAP, the average action is the average model prediction on the background data (see Sect. 4.1).

<sup>2</sup> The background data are used by KernelSHAP to fill in data for features that are absent in the currently investigated feature coalition. This will be explained in more detail in Sect. 4.1.

## 2 Related Work

With the success of AI and ML, which was often made possible with the help of complex deep learning models, the last years have seen a growing interest in XAI [6–9]. Important explanation techniques for ML in general (mostly classification and regression) are post-hoc explanations, like SHAP [3], LIME [10], or LRP [11], and self-explainable models like linear models or decision trees (DT) [12].

In the following, we focus on explainability for RL, which is often more challenging than for supervised ML classification or regression. This is due to the reasons already mentioned in Sect. 1.2 (multiple steps contribute to overall return; multiple actions or even multiple agents make it harder to find out which of the model outputs is responsible for reaching a high performance). Explainability in RL has been the topic of several reviews [2, 13, 14]. According to the review of Hickling et al. [13], the most common XAI approaches in RL are similar to the general ML case: either DTs as explainable surrogates for more complex DRL models [15–17] or post-hoc explanations via SHAP [5, 18–20] (or LIME or LRP).

DTs are often used to mimic simple (but not trivial) DRL agents (e.g. less than 10 inputs, single-dimensional action space) as was shown in [15–17, 21, 22]. If DTs are successful, they deliver explanations through human-understandable rules. However, for more complex environments (e.g. the MuJoCo environments studied in this work with 8–27 observables and multidimensional, continuous actions), it can be difficult or impossible to find simple DTs and large DTs are no longer interpretable.

In those cases, many works resort to SHAP-based post-hoc explanations [13], as they can be aggregated across multiple actions, or evaluated for many input dimensions or for multi-agent RL (MARL). Heuillet et al. [5] use SHAP for MARL environments where the contribution of a specific agent to the global reward is measured via Shapley values. Their task is not to explain the actions of a single DRL agent, but to evaluate each agent’s contribution. Another interesting approach using SHAP in conjunction with RL is the one described by Sequeira and Gervasio [23]. To the goal of gaining more insights into trained DRL agents, the authors compute different human-inspired “interestingness dimensions”, e.g., “confidence” and “riskiness” from interaction data obtained by evaluating RL agents in their respective environment for multiple episodes. Among other analyses, they use the SHAP framework to investigate how features influence the different “interestingness dimensions” on a global level or to better understand local sudden changes throughout the episodes. Rizzo et al. [19] use SHAP to explain a single agent for a traffic light control system. SHAP values indicate which inputs are important in certain states. This is similar to our approach, but they use SHAP only for a single-dimensional action space and for illustrating the decisions made at certain timesteps. Zhang et al. [20] use RL and SHAP for power system control. Liessner et al. [18] study a simple car control environment where an agent follows a street lane and has to obey certain speed limits. They present a so called RL-SHAP diagram where the agent’s inputs are placed on the y-axis, color-coded by the SHAP values, while the x-axis shows the distance



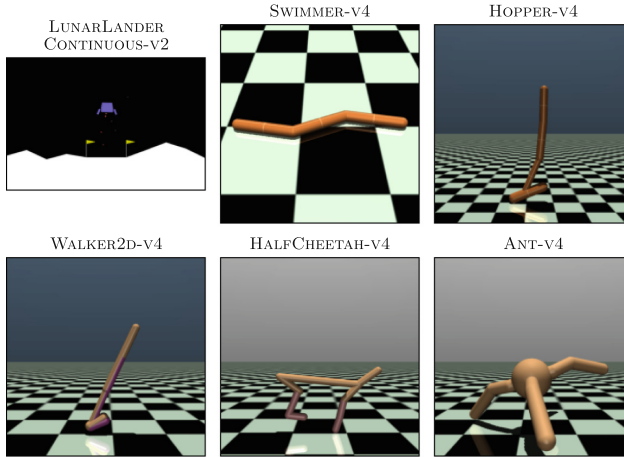


Fig. 1. Renderings of the benchmark environments

traveled. This method provides great visual understanding for this simple problem, but is harder to apply to more complex problems with multidimensional action spaces.

### 3 Benchmark Environments

We conduct our experiments using a variety of RL environments from the Gymnasium [24] suite (see Table 1 and Fig. 1), in particular LunarLanderContinuous and a set of MuJoCo tasks. In LunarLanderContinuous the goal is to operate the main and lateral engines of the lander in order to safely land in a predefined area of the lunar surface. The MuJoCo tasks all have the goal of fast directed locomotion of simulated robots in various different shapes and therefore with different numbers of joints and actuators. All these environments share properties relevant for our investigations: the observations are multidimensional and continuous, as are the actions (corresponding to thrust of the engines in case of LunarLanderContinuous or torque applied to the joints of MuJoCo robots).

For the training of DRL agents, we rely on Stable Baselines3 [25]. Agents are trained using either TD3 [26] or TQC [27] in order to achieve state-of-the-art (as reported on <https://huggingface.co/sb3>) high-performance and low-fluctuation results. We used the hyperparameters from RL Baselines3 Zoo [28].

## 4 Experiment 1: Dependency of KernelSHAP on Background Data

The research question of this experiment is whether the size and selection procedure (sampling or clustering) of the background dataset are critical for the

**Table 1.** Environments used as benchmark

Environment	observation dim.	action dim.	DRL agent	$\bar{R}$ : Performance in 100 episodes
LunarLanderCont.-v2	8	2	TQC	278.54 $\pm$ 29.29
Swimmer-v4	8	2	TD3	353.50 $\pm$ 2.41
Hopper-v4	11	3	TQC	3659.89 $\pm$ 6.30
Walker2d-v4	17	6	TD3	4470.01 $\pm$ 13.47
HalfCheetah-v4	17	6	TQC	12098.48 $\pm$ 107.42
Ant-v4	27	8	TD3	5966.68 $\pm$ 809.69

accuracy of the SHAP values, especially in high-dimensional environments. To clarify the meaning of “background data”, we briefly summarize the KernelSHAP approximation method.

#### 4.1 KernelSHAP and Background Data

The KernelSHAP method approximates the features’ impact by observing the output of the model when switching a feature from “absent” to “present”. To compute the SHAP values of an observation  $\mathbf{x}$  in  $J$ -dimensional space, first a number  $M$  of coalitions are sampled. Each coalition is represented by a vector  $\mathbf{z} \in \{0, 1\}^J$ , where 0 means that the feature  $j$  is absent and 1 that the feature is part of the coalition. A transformation function  $h$  translates these encodings  $\mathbf{z}$  to valid inputs for the model: while present features keep their actual value  $h(z_j) = x_j$ , absent features are replaced by values drawn randomly from the background data,  $h(z_j) \prec \mathbf{B}_j$ . This is the point at which the background data  $\mathbf{B}$  come into play. They can be understood as matrix with  $J$  columns (features) and  $N_b$  rows.  $N_b$  is the size of the background dataset.  $\mathbf{B}$  is constructed once prior to all KernelSHAP computations, by either sampling or clustering from a larger reservoir (see Sect. 4.2 for details). In the last step, a linear model

$$g(\mathbf{z}) = \phi_0 + \sum_{j=1}^J \phi_j z_j \quad (2)$$

is fitted to minimize the squared differences  $(f(h(\mathbf{z})) - g(\mathbf{z}))^2$  for all  $M$  coalitions, where  $f(h(\mathbf{z}))$  is the model output for a given input  $h(\mathbf{z})$ . The squared differences are weighted by the SHAP kernel (Theorem 2 in [3]), which assigns higher weights to coalitions with few and to coalitions with many present features. The coefficients of the linear model  $g$  are the SHAP values  $\phi_j$ . As a consequence, the computed SHAP values are not only a function of the model, but also of the background data (as well as stochasticity).

It is recommended to reduce the size of large background datasets by sampling or clustering. To empirically study the impact of the background dataset’s size and the effect of sampling or clustering, we propose the following experiment.

## 4.2 Experimental Setup

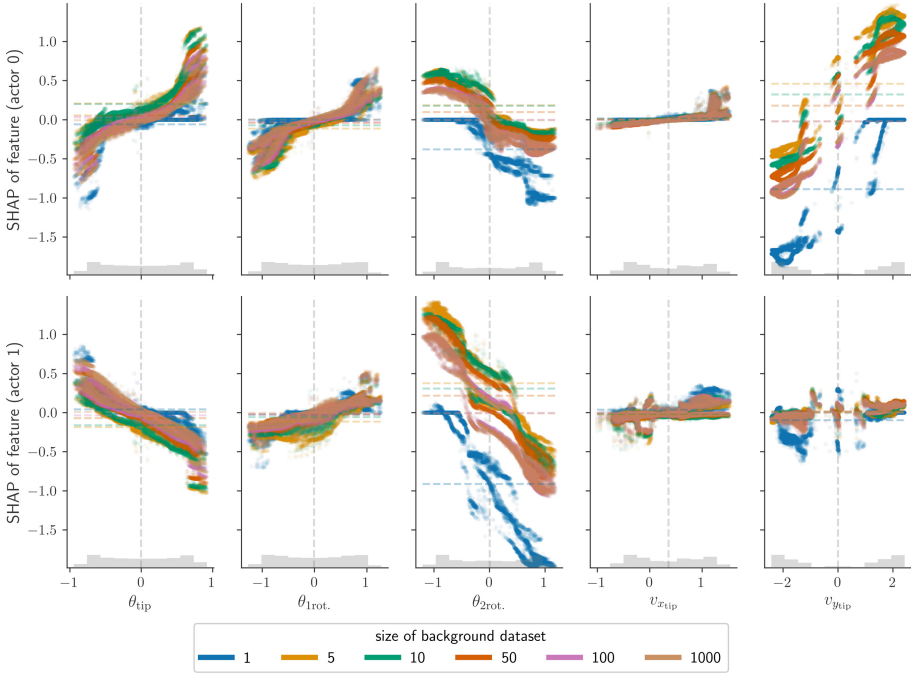
First, the background dataset is filled with samples from 10 episodes of the RL agent, leading to 10 000 samples for the MuJoCo environments and about 1700 for LunarLanderContinuous, where each sample is a point in the observation space. This dataset is then reduced to size  $N_b$  either by sampling or using the KMeans algorithm to produce  $N_b$  cluster centers. Based on these  $N_b$  background data, we now compute SHAP values for a fixed set of  $N_e = 1000$  samples drawn from data logged during a number of different evaluation episodes. The consistency of SHAP values computed with background data of various sizes is qualitatively visualized by the dependency plots in Fig. 2 and quantitatively evaluated in Fig. 3 by computing the root mean squared error (RMSE), measuring the difference between SHAP values with  $N_b \leq 100$  background data and approximately “true” SHAP values based on a much larger dataset with  $N_b = 1000$  background data as reference. The SHAP values corresponding to each action dimension are normalized by the standard deviation of the actions  $\sigma_a$  to make them comparable across the action dimensions. The RMSE is then computed across all features  $j$  and actions  $a$ .

This approach is tested on the variety of simulated control tasks with multidimensional, continuous observation and action spaces from Sect. 3. We run tests with increasing number of background data points  $N_b \in \{1, 5, 10, 20, 50, 100, 1000\}$  and two different selection methods (sampling or KMeans-clustering). For better statistics each run is repeated five times.

## 4.3 Robustness of KernelSHAP

Our results show a remarkable robustness of KernelSHAP. Visually inspecting the resulting dependency plots (Fig. 2), we note that only the values based on 1 background sample stand out. Differences between other distributions are often hardly noticeable to the naked eye. The RMSE of the respective set of SHAP values w.r.t. the one based on the largest set of background data ( $N_b = 1000$ ) gives a quantitative measurement. The linear arrangement of measurements in the log-log-plots of Fig. 3 suggests a power-law-relationship between the number of background samples and the error. The parameters of this law seem to be rather consistent across different RL-tasks. This evaluation also shows, that clustering leads to noticeable smaller errors than sampling. The results show little variance across multiple repetitions, as indicated by the very small error bars, and the qualitative results are consistent across all investigated benchmark tasks.

For all environments, when using selection method *sampling*, the power-law relationship follows a  $1/\sqrt{N_b}$  power-law remarkably well. This can be understood from the law of large numbers: If a measurement with i.i.d. fluctuations is repeated  $N$  times, the error in all averages shrinks by a factor of  $1/\sqrt{N}$ . The number  $N_b$  of background data puts an upper bound on the number of i.i.d. samples. With *clustering* we get a higher power law because the cluster centers are better representatives of the underlying data distribution.

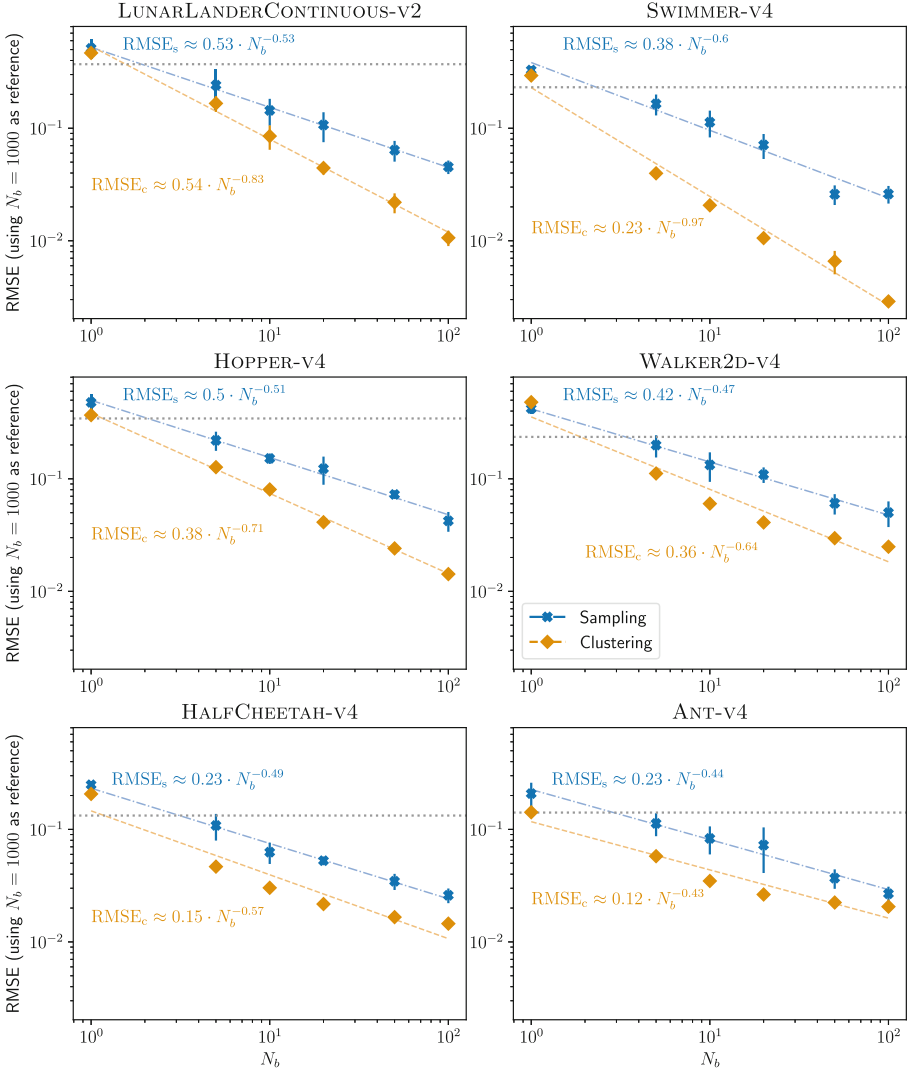


**Fig. 2.** SHAP dependency plots of 5 out of the 8 features of Swimmer, computed using differently sized background data (encoded by color) obtained by sampling. Colored dotted lines signify the average SHAP value of each feature. The histograms on the abscissa show the features’ distribution.

The horizontal line in each plot gives an idea of the upper tolerable limit for the error. As a measure, we here use the standard deviation of the SHAP values, normalized by  $\sigma_a$ . If the error is substantially smaller than the horizontal line, the given number  $N_b$  of background data should be sufficient. Figure 3 shows that this is the case for  $N_b \geq 5$ .

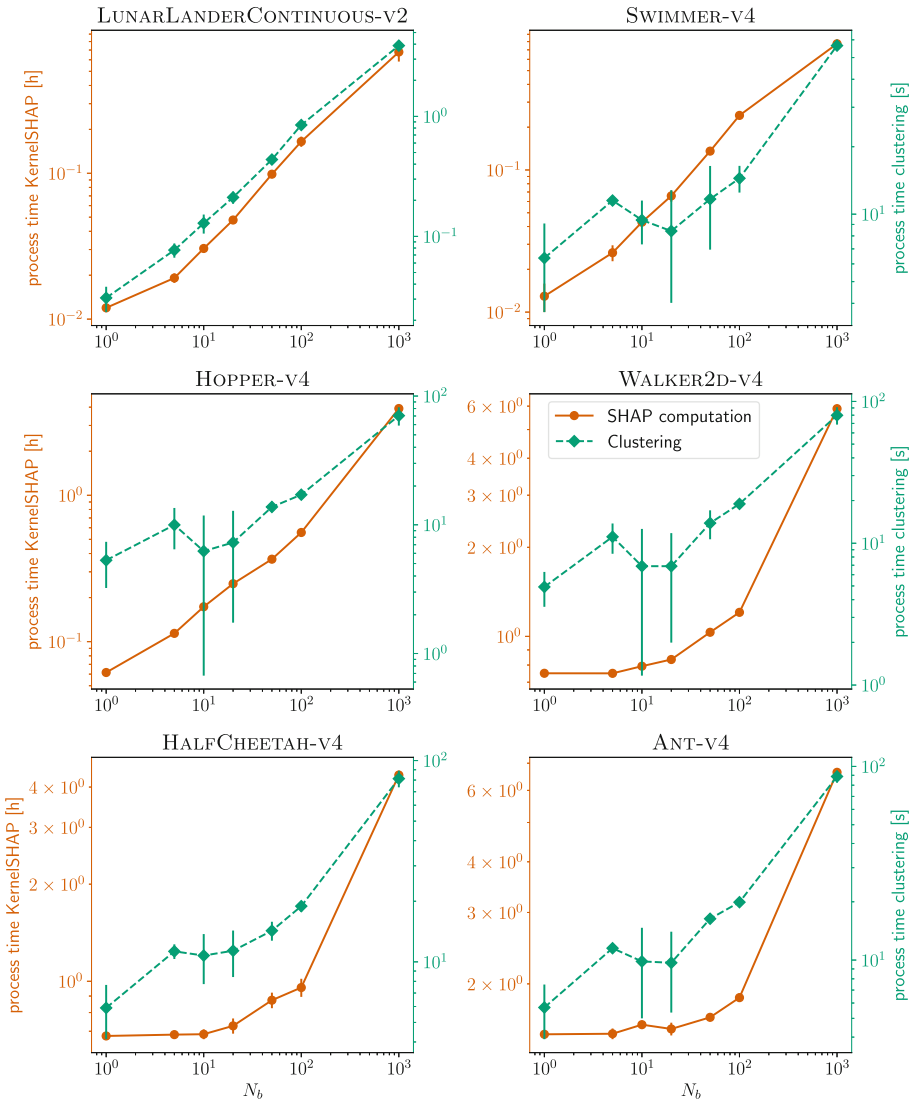
The rapidly decreasing gains in precision when adding more samples to the background dataset is especially relevant when put in context of increased computational costs. Figure 4 shows the process time for computing SHAP values of  $N_e = 1000$  samples based on differently sized background data. When searching for a trade-off between precision and computation time, this increase in compute should carefully be taken into consideration, especially in conjunction with the rapidly decreasing error. Since the overhead of KMeans clustering is negligible compared to the computational costs of SHAP value approximation (tens of seconds vs hours), this method of background data reduction is generally preferable, given the smaller errors.

In general, the outcome of this experiment is quite surprising: Even for the environments with many observation and action dimensions (meaning that the estimation of marginal distributions for the “missing” features requires high-



**Fig. 3.** RMSE of SHAP values computed with different numbers  $N_b$  of background samples (using SHAP computed on the largest sample  $N_b = 1000$  as reference) and two dataset reduction methods: sampling (dash-dotted blue line) and clustering (dashed orange line). The error bars mark  $\pm 1\sigma$  of five repetitions. The dotted horizontal line marks the threshold described in the main text. (Color figure online)

dimensional integrals), a relatively small number of samples or cluster centers is sufficient to reach reasonable accuracy.



**Fig. 4.** Computational costs ( $\mu \pm \sigma$  across five repetitions) of KernelSHAP computing SHAP values of  $N_e = 1000$  samples, based on differently sized background dataset  $N_b$  and the computational overhead of clustering with  $N_b$  cluster centers. Note that the costs of KernelSHAP shown on the left y-axis are measured in hours, while the costs for clustering shown on the right y-axis are measured in seconds. Thus, clustering costs are generally negligible. The costs for sampling instead of clustering are in the order of magnitude  $1 \times 10^{-4}$  s to  $1 \times 10^{-3}$  s and therefore completely negligible.

## 5 Experiment 2: Empirical Evaluation of SHAP-Based Feature Importance

SHAP is a method rooted in game-theory we use for attributing a certain action difference to the single elements of the observation vector (features). The action difference is the difference between the actual action of the agent, given an observation, and the average action. Based on this attribution, the importance of the single features can be defined. In the following experiment, we investigate how the computed global feature importance correlates to the feature importance in the RL task.

### 5.1 Generalized Feature Importance

Although the SHAP value for feature  $j$  and action  $a$  measures how this specific feature influences this specific action, it is not a priori clear whether the importance for a single action implies a similar importance for the RL performance of the agent (overall return from a RL episode). Molnar [7, Chap.9.6.5] suggests establishing a connection between SHAP values for a dataset with  $N$  instances and feature importance as

$$\text{FI}_{j,a} = \frac{1}{N} \sum_{n=1}^N |\phi_{j,a}^{(n)}|, \quad (3)$$

given by averaging the absolute SHAP values  $|\phi_j|$  of a feature  $j$  over the instances. Since in more complex RL tasks the agent has to perform multidimensional actions, the definition of a global feature importance is not obvious. The importance of a single feature can be very different for different elements of the action vector, as can be seen for example in Fig. 2, where the feature  $v_{y_{\text{tip}}}$  has the biggest impact on the first dimension of the action vector, while having only a marginal effect on the second one (rather flat distribution in the SHAP dependency plot). In addition, the different elements  $a_i$  of the action vector  $\mathbf{a} \in \mathbb{R}^A$  can have very different ranges. This would have a strong impact on the associated SHAP values according to their definition based on the difference between the actual action and average action. To mitigate these problems, normalizing the feature importances for each dimension by the standard deviation of the actions along the specific dimension and averaging them seems a natural extension of Eq. (3) to multidimensional actions. We therefore generalize feature importance to the case of multidimensional actions by defining

$$\text{FI}_j = \frac{1}{A} \sum_{i=1}^A \frac{\text{FI}_{j,a_i}}{\sigma(a_i)} \quad (4)$$

for a task with  $A$  action dimensions. Is the feature importance  $\text{FI}_j$  correlated to the change in agent performance (cumulative reward) when feature  $j$  is removed from the observations? This research question is investigated in the following experiment.

## 5.2 Experimental Setup

We assess the research question, using the six different RL tasks described in Table 1, with the following procedure:

1. The agents are evaluated in their respective environment for 10 episodes.
2. A KernelSHAP explainer is set up using  $N_b = 1000$  background data samples.
3. SHAP values are computed for  $N_e = 1000$  samples drawn randomly from 10 different evaluation episodes.
4. The feature importances  $FI_j$  are computed according to Eq (4).
5. The agent is evaluated again for 100 episodes, this time being “blinded” w.r.t. observation  $j$ . Observation  $j$  is substituted by its average value of the evaluation samples from step 3. The resulting average return of the agent blinded w.r.t. observation  $j$  is denoted by  $\bar{R}_{\setminus j}$ .

## 5.3 Performance Drop Vs. Feature Importance

Plotting  $\bar{R}_{\setminus j}$ , the performance of the agent blinded w.r.t. observable  $j$ , against feature importance  $FI_j$  in Fig. 5 shows the general correlation between the two measurements. Table 2 contains the results in succinct form.

**Table 2.** Summary of the correlation between SHAP feature importance and performance drop of partially-blinded agent

Environment	Pearson $r$	$R^2$
LunarLanderCont.-v2	-0.829	0.687
Swimmer-v4	-0.909	0.826
Hopper-v4	-0.0364	0.00133
Walker2d-v4	-0.687	0.472
HalfCheetah-v4	-0.530	0.281
Ant-v4	-0.557	0.310

The results show, with the notable exception of Hopper, a correlation between a feature’s importance in predicting an action and the agent’s performance when that feature is absent. While this correlation is especially prominent in the “simpler” environments Swimmer and LunarLanderContinuous, more complex environments show a weaker correlation: a general trend is still visible, but there are many examples where features with lower FI lead to stronger decreases in performance and vice versa. The results suggest interpreting the FI as computed by SHAP with caution.

The environment Hopper stands out, as apparently every single feature is crucial to the agent’s success. Omitting any feature leads to almost the same drastic decrease in performance. The feature importance therefore has almost no correlation with  $\bar{R}_{\setminus j}$ . To investigate whether the crucial role of every single



observable is a property of this trained agent or an intrinsic property of the environment, training was repeated with partially-blinded TQC agents. During training and evaluation of each of these agents, one observable is set to zero. Since such experiments require training multiple agents ex novo, and are therefore rather time-consuming, we performed this experiment only for the outstanding case of Hopper. Figure 6 shows the relation between  $FI_j$ ,  $\bar{R}_{\setminus j}$ , and  $\bar{R}_{\setminus j}^{(\text{retrain})}$ , the performance of agents newly trained without the specific feature  $j$ . When training new agents partially blinded ab initio w.r.t. one observable, the performance increases notably. While for no feature the performance reaches the fully-observable threshold, in most cases (except for  $\theta_{\text{thigh}}$ ,  $\theta_{\text{torso}}$ , and  $\omega_{\text{foot}}$ ) the performance lies at least around 3000. The feature importance of the agent, trained with all observables accessible, cannot be expected to correlate with the performance of newly-trained, partially-blinded agents.

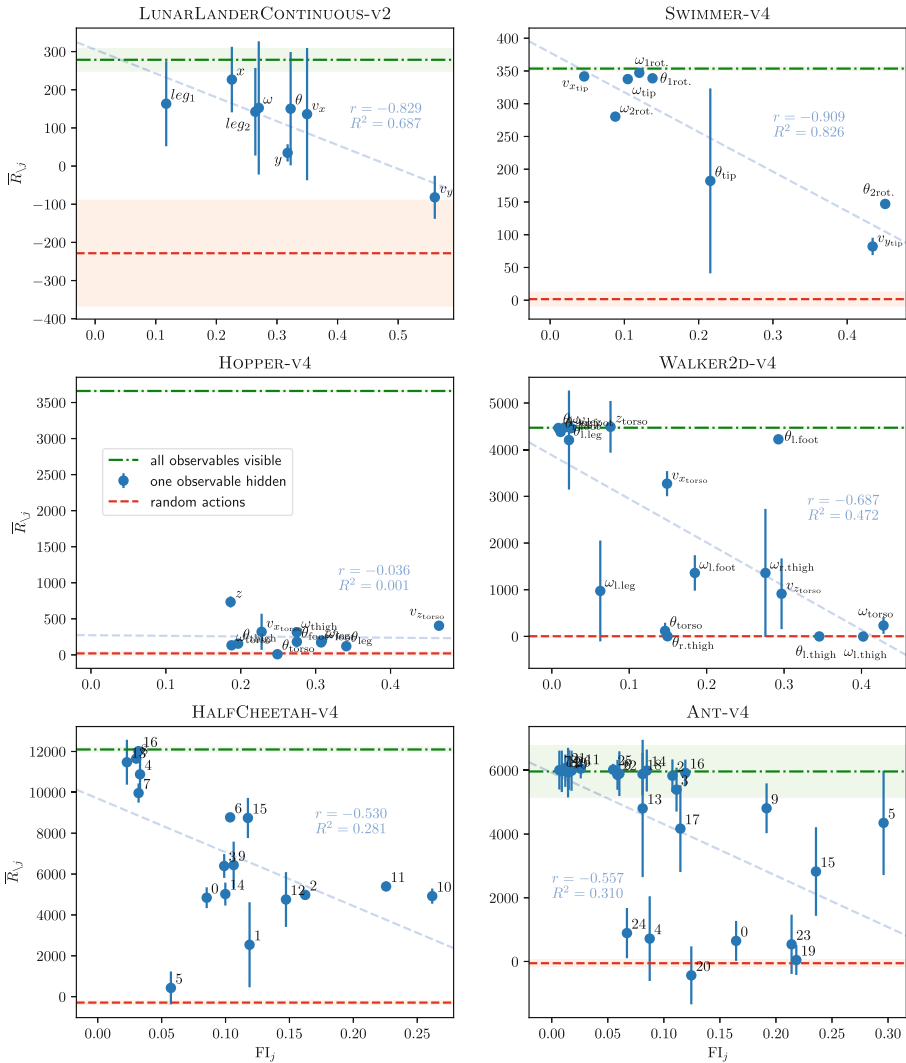
By visually inspecting episode renderings<sup>3</sup> corresponding to low feature importance and low performance for the partially-blinded agent (points in the lower left part of the plots in Fig. 5), one common behavior emerges: These episodes are always characterized by catastrophic failure of the agent falling over (e.g., the ant falling on its back) or by premature termination due to reaching a state defined by the environment as “unhealthy” (one or more observables leaving a predefined range). The hidden observables are apparently crucial for keeping the simulated robot in a safe state, even if they have been assigned a relatively low feature importance.

It should also be noted, that the process of omitting a feature can usually not be applied iteratively: Omitting several of the features *together*, that hidden individually have little impact on performance, often leads to a complete breakdown of performance.

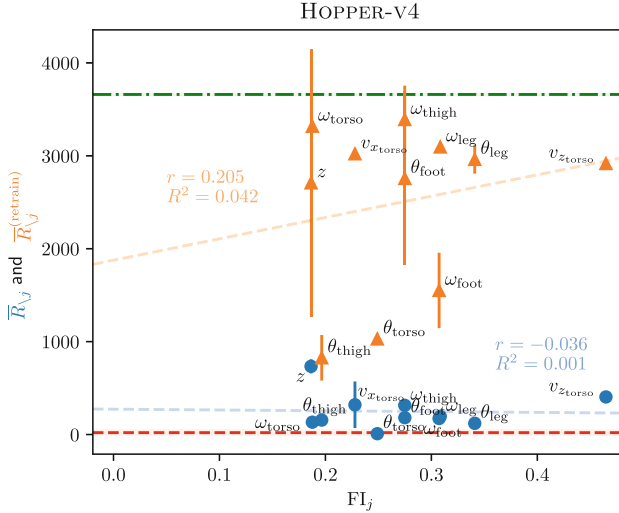
It is also worth noting that omitting some features has a greater impact on the *consistency* of the agent’s performance than others. This is most prominent in the case of Swimmer, where omitting a feature generally leads to a very consistent performance. If the agent is blinded w.r.t. feature  $\theta_{\text{tip}}$  instead, it can play either very successful or very unsuccessful episodes. This is shown in Fig. 7 where the distributions of returns are represented by violin plots and point clouds. Omitting feature  $\theta_{\text{tip}}$  leads to a strongly bimodal distribution. This can be explained with a peculiarity of the Swimmer environment: For coordinated movement, the rear rotor has to make a movement to the opposite side of the front tip. In absence of the front tip angle  $\theta_{\text{tip}}$ , the agent still observes the front tip’s angular velocity  $\omega_{\text{tip}}$ , which assumes the value 0 at either side, but the agent has to guess which side it is. If it guesses correctly, it receives a good return, otherwise the movement comes to a complete standstill.<sup>4</sup>

<sup>3</sup> The renderings of the five best and five worst episodes can be accessed on the Github repository [https://github.com/RaphaelEngelhardt/xai\\_shap4rl](https://github.com/RaphaelEngelhardt/xai_shap4rl).

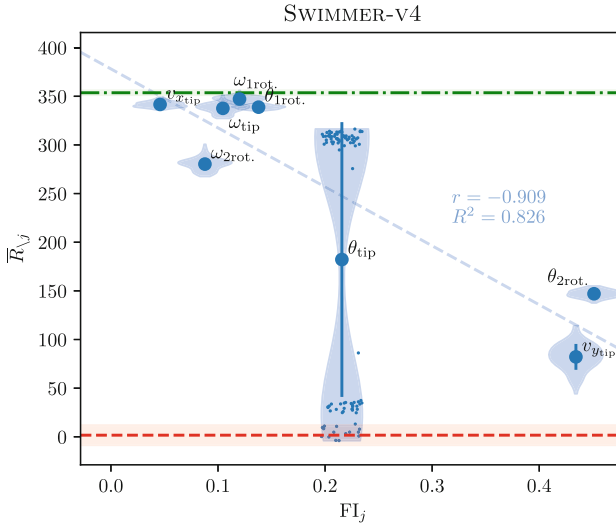
<sup>4</sup> Example videos of the best and the worst Swimmer episodes are accessible on the repository as supplementary material.



**Fig. 5.** Relation between SHAP-based feature importance (abscissa) of an observable and cumulative reward of an agent blinded w.r.t. said observable (ordinate). Error bars show  $\pm 1\sigma$  of the 100 evaluation episodes. The dash-dotted green horizontal line denotes the average performance of the agent when all observables are fully accessible, the green shaded area shows  $\pm 1\sigma$  over the 100 evaluation episodes. Analogously, the dashed red line and area mark the lower baseline, i.e. the return of an agent that acts randomly at each timestep. The dashed blue line is a linear fit to the data with correlation coefficient  $r$  and coefficient of determination  $R^2$ . (Color figure online)



**Fig. 6.** Version of Fig. 5 (Hopper), but with extended results: The orange triangles show the achievable returns when retraining partially-blinded agents from scratch.

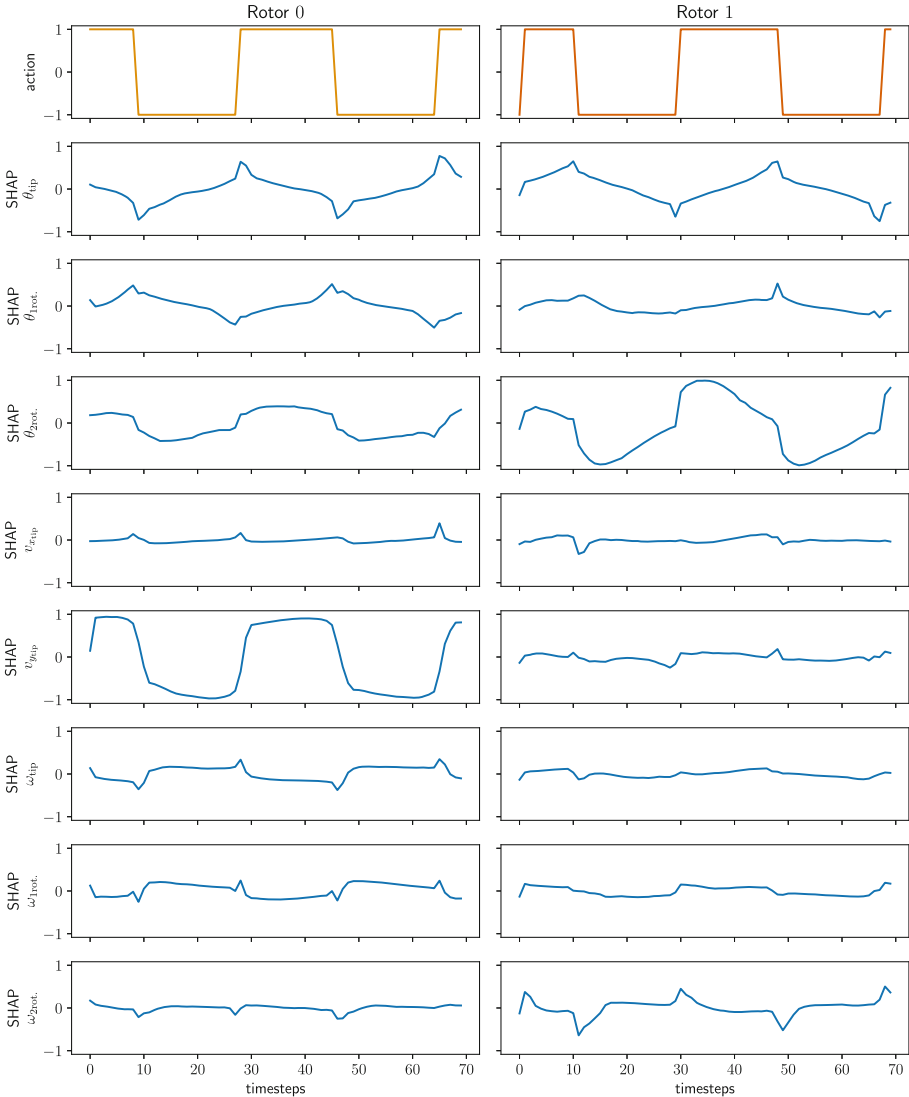


**Fig. 7.** Version of Fig. 5 (Swimmer) with added distributions of returns. While the distributions are generally centered, leading to small standard deviations, omission of  $\theta_{tip}$  leads to a bimodal distribution.

## 6 Interpretation of SHAP Time Dependency in RL

An advantage of using SHAP as XAI method in RL is that SHAP assigns a contribution of each feature to each action dimension at each timestep of an

episode. This provides a rich dataset which can be used to gain insights. As an example, Fig. 8 shows the time dependencies of the SHAP values for each feature and action dimension (the two rotors in case of Swimmer). This has some similarity to the RL-SHAP diagram introduced in [18].



**Fig. 8.** Time series of roughly two oscillations of Swimmer. Shown is the evolution of actions (first row) and SHAP values of the eight observables (rows 2–9) for the two rotors (columns).

We see for example that  $v_{y_{\text{tip}}}$  has a large contribution to rotor 0, but not to rotor 1. Likewise,  $\theta_{2_{\text{rot.}}}$  has a large contribution to rotor 1, but a much smaller contribution to rotor 0. It might be tempting to relate steep falls or rises in the SHAP value (e.g. for  $v_{y_{\text{tip}}}$  around timesteps 10 and 25, respectively) to drastic changes in importance, that is,  $v_{y_{\text{tip}}}$  initially appears to have a major positive effect on rotor 0 up to timestep 10 and thereafter a major negative effect. However, with this example, we would like to point out that such an interpretation is wrong. A SHAP value has always to be seen in connection with the ML prediction that it models, in our case the variable *action* 0 shown in the first row and column of Fig. 8. If this target value exhibits a jump, the overall SHAP values will also show the same jump. As a consequence, a high-contributing SHAP value needs to have a similar jump. The right interpretation is: Whatever the target value of *action* 0 is, feature  $v_{y_{\text{tip}}}$  has a large contribution to it of the *same* sign.

## 7 Conclusion and Outlook

This work analyzes the reliability of SHAP values as a XAI method for complex RL environments with multidimensional actions. A positive aspect of SHAP is its applicability as XAI method also in the case of multidimensional actions, whereas interpretable DTs are hard or impossible to build in such cases.

We have examined the SHAP accuracy as a function of background data size and found it to be surprisingly robust even if only a small size is used in order to reduce computational costs. We found KMeans clustering to be the preferable background data selection method.

Furthermore, we have generalized the SHAP-based feature importance to RL of multidimensional actions. While the SHAP value measures the contribution of a feature to a specific action, the feature importance expresses the importance of the feature in general, regardless of the action dimension. Given this generalized feature importance, we have investigated how well this importance is correlated to the agent’s performance (the return). We showed that often there is a clear correlation, while also exceptions exist, most notably in the case of Hopper, where every left-out feature leads to drastic performance breakdowns, irrespective of whether it had high or low feature importance.

Currently, we can only point out these two distinctive cases; finding the reason for these distinct behaviors is left for future research. A possible reason might be SHAP’s inability to handle interactions between features. Inspecting the cases with surprising breakdown, we can speculate that this happens more likely for unstable environments with a higher sudden-failure probability (Hopper, can fall down, irreversible) than for more stable environments (Swimmer, cannot fall).

We believe that these findings make an important contribution to the reliability of SHAP-based explainability in RL. The results presented in this work also contain useful insights for XAI practitioners in RL. In the future, we plan to investigate the reasons why SHAP-based importance sometimes does not correlate with agent performance. Furthermore, we plan to examine alternative value functions for SHAP, e.g. episode returns as outlined in Sect. 1.2.

**Acknowledgments.** This research was supported by the research training group “Dataninja” (Trustworthy AI for Seamless Problem Solving: Next Generation Intelligence Joins Robust Data Analysis) funded by the German federal state of North Rhine-Westphalia.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (XAI): a survey. arXiv preprint (2020). <https://doi.org/10.48550/arXiv.2006.11371>
2. Vouros, G.A.: Explainable deep reinforcement learning: state of the art and challenges. *ACM Comput. Surv.* **55**(5), 1–39 (2022). <https://doi.org/10.1145/3527448>
3. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates Inc., Red Hook (2017)
4. Shapley, L.S.: A value for n-person games. In: Kuhn, H.W., Tucker, A.W. (eds.) *Contributions to the Theory of Games (AM-28)*, vol. II, pp. 307–318. Princeton University Press, Princeton (1953). <https://doi.org/10.1515/9781400881970-018>
5. Heuillet, A., Couthouis, F., Díaz-Rodríguez, N.: Collective explainable AI: explaining cooperative strategies and agent contribution in multiagent reinforcement learning with Shapley values. *IEEE Comput. Intell. Mag.* **17**(1), 59–71 (2022). <https://doi.org/10.1109/MCI.2021.3129959>
6. Holzinger, A.: From machine learning to explainable AI. In: 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), pp. 55–66. IEEE (2018). <https://doi.org/10.1109/DISA.2018.8490530>
7. Molnar, C.: *Interpretable Machine Learning*, 2 edn. (2022). <https://christophm.github.io/interpretable-ml-book>. Accessed 15 Mar 2024
8. Nauta, M., et al.: From anecdotal evidence to quantitative evaluation methods: a systematic review on evaluating explainable AI. *ACM Comput. Surv.* **55**(13s) (2023). <https://doi.org/10.1145/3583558>
9. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.R.: Explaining deep neural networks and beyond: a review of methods and applications. *Proc. IEEE* **109**(3), 247–278 (2021). <https://doi.org/10.1109/JPROC.2021.3060483>
10. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should I trust you?”: explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2016*, pp. 1135–1144. Association for Computing Machinery, New York (2016). <https://doi.org/10.1145/2939672.2939778>
11. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**(7), e0130140 (2015). <https://doi.org/10.1371/journal.pone.0130140>

12. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**(5), 206–215 (2019). <https://doi.org/10.1038/s42256-019-0048-x>
13. Hickling, T., Zenati, A., Aouf, N., Spencer, P.: Explainability in deep reinforcement learning: a review into current methods and applications. *ACM Comput. Surv.* **56**(5), 1–35 (2023). <https://doi.org/10.1145/3623377>
14. Wells, L., Bednarz, T.: Explainable AI and reinforcement learning - a systematic review of current approaches and trends. *Front. Artif. Intell.* **4**, 550030 (2021). <https://doi.org/10.3389/frai.2021.550030>
15. Dhebar, Y., Deb, K., Nagesh Rao, S., Zhu, L., Filev, D.: Toward interpretable-AI policies using evolutionary nonlinear decision trees for discrete-action systems. *IEEE Trans. Cybern.* **54**(1), 50–62 (2024). <https://doi.org/10.1109/TCYB.2022.3180664>
16. Ding, Z., Hernandez-Leal, P., Ding, G.W., Li, C., Huang, R.: CDT: cascading decision trees for explainable reinforcement learning. arXiv preprint (2020). <https://doi.org/10.48550/arXiv.2011.07553>
17. Liu, G., Sun, X., Schulte, O., Poupart, P.: Learning tree interpretation from object representation for deep reinforcement learning. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems*, vol. 34, pp. 19622–19636. Curran Associates Inc., Red Hook (2021)
18. Liessner, R., Dohmen, J., Wiering, M.: Explainable reinforcement learning for longitudinal control. In: *Proceedings of the 13th International Conference on Agents and Artificial Intelligence ICAART*, vol. 2, pp. 874–881. SciTePress, Setúbal (2021). <https://doi.org/10.5220/0010256208740881>
19. Rizzo, S.G., Vantini, G., Chawla, S.: Reinforcement learning with explainability for traffic signal control. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3567–3572. IEEE (2019). <https://doi.org/10.1109/ITSC.2019.8917519>
20. Zhang, K., Zhang, J., Xu, P.D., Gao, T., Gao, D.W.: Explainable AI in deep reinforcement learning models for power system emergency control. *IEEE Trans. Comput. Social Syst.* **9**(2), 419–427 (2022). <https://doi.org/10.1109/TCSS.2021.3096824>
21. Engelhardt, R.C., Oedingen, M., Lange, M., Wiskott, L., Konen, W.: Iterative oblique decision trees deliver explainable rl models. *Algorithms* **16**(6), 282 (2023). <https://doi.org/10.3390/a16060282>
22. Custode, L.L.: Evolutionary optimization of decision trees for interpretable reinforcement learning. Ph.D. thesis, Università degli studi di Trento (2023). <https://doi.org/10.15168/11572.375447>
23. Sequeira, P., Gervasio, M.: IXDRL: a novel explainable deep reinforcement learning toolkit based on analyses of interestingness. In: Longo, L. (ed.) *Explainable Artificial Intelligence*. pp. 373–396. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-44064-9\\_20](https://doi.org/10.1007/978-3-031-44064-9_20)
24. Towers, M., et al.: *Gymnasium* (2023). <https://doi.org/10.5281/zenodo.8127026>
25. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **22**(268), 1–8 (2021)
26. Fujimoto, S., van Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 80, pp. 1587–1596. PMLR (2018)

27. Kuznetsov, A., Shvechikov, P., Grishin, A., Vetrov, D.: Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 5556–5566. PMLR (2020)
28. Raffin, A.: RL Baselines3 Zoo. <https://github.com/DLR-RM/rl-baselines3-zoo>. Accessed 15 Mar 2024





# Categorical Foundation of Explainable AI: A Unifying Theory

Francesco Giannini<sup>1,7</sup> , Stefano Fioravanti<sup>2</sup> , Pietro Barbiero<sup>3</sup> ,  
Alberto Tonda<sup>4</sup> , Pietro Liò<sup>5</sup> , and Elena Di Lavore<sup>6</sup> 

<sup>1</sup> Consorzio Interuniversitario Nazionale per l'Informatica, Roma, Italy

<sup>2</sup> Charles University Prague, Prague, Czechia

<sup>3</sup> Università della Svizzera Italiana, Lugano, Switzerland  
barbip@usi.ch

<sup>4</sup> University of Paris, Paris, France

<sup>5</sup> University of Cambridge, Cambridge, UK

<sup>6</sup> University of Pisa, Pisa, Italy

<sup>7</sup> Scuola Normale Superiore, Pisa, Italy

**Abstract.** Explainable AI (XAI) aims to address the human need for safe and reliable AI systems. However, numerous surveys emphasize the absence of a sound mathematical formalization of key XAI notions—remarkably including the term “*explanation*”, which still lacks a precise definition. To bridge this gap, this paper introduces a unifying mathematical framework allowing the rigorous definition of key XAI notions and processes, using the well-funded formalism of Category theory. In particular, we show that the introduced framework allows us to: (i) model existing learning schemes and architectures in both XAI and AI in general, (ii) formally define the term “*explanation*”, (iii) establish a theoretical basis for XAI taxonomies, and (iv) analyze commonly overlooked aspects of explaining methods. As a consequence, the proposed categorical framework represents a significant step towards a sound theoretical foundation of explainable AI by providing an unambiguous language to describe and model concepts, algorithms, and systems, thus also promoting research in XAI and collaboration between researchers from diverse fields, such as computer science, cognitive science, and abstract mathematics.

**Keywords:** Explainable AI · Category Theory · XAI Foundations and Taxonomies

## 1 Introduction

Explainable AI (XAI) research aims to address the human need for accurate and trustworthy AI through the design of interpretable AI models and algorithms able to explain uninterpretable AI models [4]. Some of these methods are so

---

F. Giannini, S. Fioravanti and P. Barbiero—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 185–206, 2024.

[https://doi.org/10.1007/978-3-031-63800-8\\_10](https://doi.org/10.1007/978-3-031-63800-8_10)

effective that their impact now deeply affects other research disciplines such as medicine [39], physics [10, 71], and even pure mathematics [14].

A considerable number of works attempted to describe key methods and notions in this fast-growing literature [2, 4, 13, 19, 34, 37, 61, 82]. However, none of these works are grounded on a solid and unifying theory of explainability, but they rather rely on qualitative descriptions, preventing them from drawing truly universal conclusions. Current surveys acknowledge this problem and grumble that key fundamental notions of explainable AI still lack a formal definition, and that the field as a whole is missing a unifying and sound formalism [2, 61]: The notion of “*explanation*” represents a pivotal example as it still lacks a proper mathematical formalization. The followings represent an example of some of the best definitions currently available in literature:

*“An explanation is an answer to a ‘why?’ question.”* [56];

*“An explanation is additional meta information, generated by an external algorithm or by the machine learning model itself, to describe the feature importance or relevance of an input instance towards a particular output classification.”* [13];

*“An explanation is the process of describing one or more facts, such that it facilitates the understanding of aspects related to said facts (by a human consumer).”* [61].

As the interest for XAI methods rises inside and outside academic environments, the need for a sound formalization and encompassing taxonomy of the field grows quickly, as a prerequisite for welcoming a wider audience and fostering theoretically grounded research. Moreover, the formalization of XAI concepts poses a distinctive challenge, given the diverse contributions from various disciplines such as computer science, psychology, philosophy, and mathematics. While this multidisciplinary approach enhances the richness of the field by incorporating diverse perspectives, it also presents a unique challenge. The disparate backgrounds of researchers introduce a broad spectrum of languages and logical frameworks, potentially becoming a barrier to mutual understanding. In light of these complexities, XAI requires a theoretical framework that serves two crucial functions: the formalization of concepts and the unification of the field through a language that accommodates the diverse contributions spanning different disciplines.

**Key Innovation.** To address this, we propose a theoretical framework allowing a unified, comprehensive and rigorous formalization of foundational XAI notions and processes (Sect. 3). To the best of the authors’ knowledge, this is the first work investigating this research direction in the XAI field. To this end, we formalize XAI notions using the language of Category theory. This choice is justified by two principal reasons: firstly, Category theory serves as an abstract language designed to encompass and harmonize diverse formal logic-mathematical systems, enabling the incorporation of all existing XAI contributions under a unified formalism; secondly, Category theory constitutes a robust mathematical framework with a focus on *processes*, facilitating the description of the inherent properties of XAI models by design. For these reasons, Category theory is

widely used in theoretical computer science [1, 72, 77, 78, 83], and, more recently, in AI [3, 11, 60, 74, 76].

**Contributions.** In particular, we show that our categorical framework enables us to: model existing learning schemes and architectures (Sect. 4.1), formally define the term “explanation” (Sect. 4.2), establish a theoretical basis for XAI taxonomies (Sect. 4.3), and analyze commonly overlooked aspects of explaining methods (Sect. 4.4).

## 2 Explainable AI Theory: Requirements

In order to build a sound and exhaustive theoretical framework for explainable AI, we identified a set of fundamental notions, including objects and processes, to be modelled and a proper language to formalize them. To represent XAI algorithms and their dynamics, we rely on Category Theory [21], as it provides a mathematical framework specifically designed to analyze processes and their dynamics (Sect. 2.1). In addition, to properly define notions—such as “explanation”—we rely on Institution Theory [32], as it provides an abstract framework to express formal languages’ syntax and semantics (Sect. 2.2).

### 2.1 Category Theory: A Framework for (X)AI Processes

(X)AI processes all share three basic properties: (i) they map (multiple) inputs to (multiple) outputs via a composition of parametric operations (see *monoidal categories*), (ii) they update the parameters of such operations based on some error function (see *feedback categories*), and (iii) they keep updating such parameters over time until convergence (see *cartesian streams*). In the following paragraphs we recall some basic notions of category theory which will allow us a proper formalization of these (X)AI properties.

*A Primer on Categories: Objects and Morphisms.* Intuitively, a category is simply a collection of objects and morphisms satisfying specific composition rules.

**Definition 1** ([21]). *A category  $\mathcal{C}$  consists of a class of objects  $\mathcal{C}^o$  and, for every  $X, Y \in \mathcal{C}^o$ , a set of morphisms  $\text{hom}(X, Y)$  with input type  $X$  and output type  $Y$ . A morphism  $f \in \text{hom}(X, Y)$  is written  $f: X \rightarrow Y$ , and for all morphisms  $f: X \rightarrow Y$  and  $g: Y \rightarrow Z$  there is a composite morphism  $f; g: X \rightarrow Z$ , with composition being associative. Moreover, for each  $X \in \mathcal{C}^o$  there is an identity morphism  $\mathbb{1}_X \in \text{hom}(X, X)$  that makes composition unital, i.e.  $f; \mathbb{1}_Y = f = \mathbb{1}_X; f$ .*

*Example 1.* **Set**, whose objects are *sets* and morphisms are *functions*, and **Vec**, whose objects are *vector spaces* and morphisms are *linear maps*, are well-known examples of categories.

Different categories can be connected using special operators called *functors* i.e., mappings of objects and morphisms from one category to another (preserving compositions and identity morphisms). For instance, there is a functor  $\mathcal{F}$  from **Vec** to **Set** that simply ignores the vector space structure.

*Monoidal Categories: Compose Multi-input/output Processes.* In this work we are mainly interested in *monoidal categories* as they offer a sound formalism for processes with multiple inputs and outputs [8,27]. Monoidal categories [53] are categories with additional structure, namely a monoidal product  $\times$  and a unit element  $U$ , enabling the composition of morphisms in parallel (cf. A.1). Notably, monoidal categories allow for a graphical representation of processes using *string diagrams* [40]. String diagrams enable a more intuitive reasoning over equational theories, and we will use them throughout the paper to provide illustrative, yet formal, definitions of XAI processes. The Coherence Theorem for monoidal categories [53] guarantees that string diagrams are a sound and complete syntax for monoidal categories. Thus all coherence equations for monoidal categories correspond to continuous deformations of string diagrams. For instance, given  $f: X \rightarrow Y$  and  $g: Y \rightarrow Z$ , the morphisms  $f; g: X \rightarrow Z$  and  $\mathbb{1}_X$  are represented as

$$X \xrightarrow{\boxed{f}} Y \xrightarrow{\boxed{g}} Z \quad X \xrightarrow{\mathbb{1}_X} X \quad \text{the equation } f; \mathbb{1}_Y = f = \mathbb{1}_X; f$$

as  $X \xrightarrow{\boxed{f}} Y = X \xrightarrow{\boxed{f}} Y = X \xrightarrow{\boxed{f}} Y$ , the morphism  $h$  with multiple inputs  $X_1, X_2, X_3$  and outputs  $Y_1, Y_2$  (left), and the parallel composition of two morphisms  $f_1: X_1 \rightarrow Y_1$  and  $f_2: X_2 \rightarrow Y_2$  (right) can be represented as follows:



*Feedback Categories: Update Processes’ State.* A common technique in machine learning involves the update of the parameters of a function, based on the *feedback* of a loss function. To model this process, we can use *feedback monoidal categories*.

**Definition 2** ([16,43]). A feedback monoidal category is a symmetric (cf. A.2) monoidal category  $\mathcal{C}$  endowed with an operator  $\odot_S: \text{hom}(X \times S, Y \times S) \rightarrow \text{hom}(X, Y)$  for all  $X, Y, S$  in  $\mathcal{C}^o$ , which satisfies a set of axioms<sup>1</sup>.

Given a morphism  $f: X \times S \rightarrow Y \times S$ , the string diagram of the feedback operation  $\odot_S$ , such that  $\odot_S(f): X \rightarrow Y$ , can be represented as



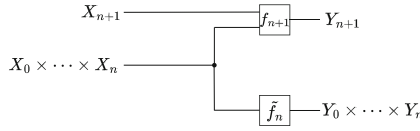
*Cartesian Streams: Dynamic Update of Processes Over Time.* In learning processes, optimizing the loss function often involves a sequence of feedback iterations. Following [76], we use *Cartesian streams* (cf. A.3) to model this kind of processes. Cartesian streams form a feedback monoidal category that is also *Cartesian*, i.e. that is equipped, for every object  $X$ , with morphisms  $\nu_X: X \rightarrow X \times X$  and  $\epsilon_X: X \rightarrow U$  that make it possible to copy and discard objects (cf. A.2). This makes them the ideal category to formalize (possibly infinite) streams of objects and morphisms.

<sup>1</sup> The full list of the axioms is reported in A.3.

**Definition 3** ([76, 85]). Let  $\mathcal{C}$  be a Cartesian category. We call  $\mathbf{Stream}_{\mathcal{C}}$  the category of Cartesian streams over  $\mathcal{C}$ , whose objects  $\mathbb{X} = (X_0, X_1, \dots)$  are countable lists of objects in  $\mathcal{C}$ , and given  $\mathbb{X}, \mathbb{Y} \in \mathbf{Stream}_{\mathcal{C}}^o$ , the set of morphisms  $\text{hom}(\mathbb{X}, \mathbb{Y})$  is the set of all  $\mathbb{f}: \mathbb{X} \rightarrow \mathbb{Y}$ , where  $\mathbb{f} = (f_0, f_1, \dots)$  is a family of morphisms in  $\mathcal{C}$ , with  $f_n: X_0 \times \dots \times X_n \rightarrow Y_n$ , for  $n \in \mathbb{N}$ .

In Cartesian streams, a morphism  $f_n$  represents a process that receives a new input  $X_n$  and produces an output  $Y_n$  at time step  $n$ . We can compute the outputs until time  $n$  by combining  $f_0, \dots, f_n$  to get  $\tilde{f}_n: X_0 \times \dots \times X_n \rightarrow Y_0 \times \dots \times Y_n$  as follows:

- $\tilde{f}_0 := f_0$
- $\tilde{f}_{n+1} := (\mathbb{1}_{X_{n+1}} \times \nu_{X_0 \times \dots \times X_n}); (f_{n+1} \times \tilde{f}_n)$



We denote by  $X^{\mathbb{N}}$  the object  $\mathbb{X} \in \mathbf{Stream}_{\mathcal{C}}^o$  such that  $\mathbb{X} = (X, X, \dots)$ , for some  $X \in \mathcal{C}^o$ . Notably, Cartesian streams form a feedback monoidal category [15] and thus are capable to model dynamic processes with feedback—such as learning processes, as we will show in Examples 3 and 4.

## 2.2 Institution Theory: A Framework for Explanations

In order to provide a formal definition of what is commonly dubbed as an “explanation”, we need a mathematical framework which has enough expressive power to subsume the broad meaning of this term. For this reason, we rely on institution theory [32], which will allow us to characterize the notion of “explanation” in a unified scheme in Sect. 4.2. Institution theory offers an ideal platform for formalizing explanations—whether expressed through symbolic languages or semantic-based models—as it enables a thorough analysis of both the structure (syntax) and meaning (semantics) of explanations across diverse languages [80], thus facilitating a deeper understanding of their nature.

More rigorously, an institution  $I$  consists of (i) a category  $\mathbf{Sign}_I$  whose objects are signatures (i.e. vocabularies of symbols), (ii) a functor  $\mathit{Sen}: \mathbf{Sign}_I \mapsto \mathbf{Set}$  which provides sets of well-formed expressions ( $\Sigma$ -sentences) for each signature  $\Sigma \in \mathbf{Sign}_I^o$ , and (iii) a functor  $\mathit{Mod}: (\mathbf{Sign}_I)^{op} \mapsto \mathbf{Set}^2$  that assigns a semantic interpretation (i.e. a world) to the symbols in each signature [31].

*Example 2* First-Order Logic (FOL), where the category of signatures is given by sets of relations as objects and arity-preserving functions as morphisms, is a typical example of institution. Sentences and models are defined by standard FOL formulas and structures.

<sup>2</sup> Given a category  $\mathcal{C}$ ,  $(\mathcal{C})^{op}$  denotes its *opposite* category, which is formed by reversing its morphisms [53], but keeping the same objects  $\mathcal{C}^o$ .

### 3 Categorical Framework of Explainable AI

We use feedback monoidal categories and institutions to formalize fundamental (X)AI notions. To this end, we first introduce the definition of “abstract learning process” (Sect. 3.1) as a morphisms’ composition in free feedback monoidal categories, and then we describe a functor instantiating this concept in the concrete feedback monoidal category of  $\text{Stream}_{\text{Set}}$  (Sect. 3.2). Intuitively, a *free* category serves as a template for a class of categories (e.g., feedback monoidals). To generate a free category, we just need to specify a set of objects and morphisms as generators. Then we can realize “concrete” instances of a free category  $F$  through a functor from  $F$  to another category  $C$  that preserves the axioms of  $F$  (cf. A.4).

#### 3.1 Abstract Learning Processes

We formalize the abstract concept of an (explaining) learning process drawing inspiration from [11, 76, 90]. At a high level, learning can be characterized as an *iterative process with feedback*. This process involves a function (known as *model* or *explainer* in a XAI method) which updates its internal states (e.g., a set of *parameters*) guided by some feedback from the environment (often managed by an *optimizer*). Hence, to properly define a learning process at abstract level, we rely on the following components:

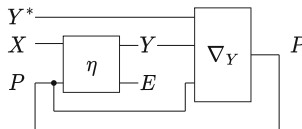
- the objects  $X, Y, Y^*, P$ , and  $E$  representing input, output, supervision, parameter, and explanation types;
- the *model/explainer* morphism  $\eta : X \times P \rightarrow Y \times E$  which produces predictions in  $Y$  and explanations in  $E$ ;
- the *optimizer* morphism  $\nabla_Y : Y^* \times Y \times P \rightarrow P$  producing updated parameters in  $P$  given supervisions in  $Y^*$ , model/explainer predictions in  $Y$  and parameters in  $P$ .

**Definition 4.** *XLearn* is the free feedback Cartesian category generated by the objects  $X, Y, Y^*, P, E$  and by the morphisms  $\eta : X \times P \rightarrow Y \times E$  and  $\nabla_Y : Y^* \times Y \times P \rightarrow P$ .

As a result, an abstract learning process can be defined by the composition of morphisms in *XLearn*.

**Definition 5.** An abstract learning process is the morphism  $\alpha$  in *XLearn* given by the following morphisms’ composition:

$$\alpha = \circlearrowleft_P ((\mathbb{1}_{Y^* \times X} \times \nu_P); (\mathbb{1}_{Y^*} \times \eta \times \mathbb{1}_P); (\mathbb{1}_{Y^* \times Y} \times \epsilon_E \times \mathbb{1}_P); \nabla_Y)$$



### 3.2 Concrete Learning and Explaining Processes

The free category  $\mathbf{XLearn}$  allows us to highlight the key features of learning processes from an abstract perspective. However, we can instantiate explaining learning processes in “concrete” forms using a feedback functor from the free category  $\mathbf{XLearn}$  to the category of Cartesian streams over  $\mathbf{Set}$ , i.e.  $\mathbf{Stream}_{\mathbf{Set}}$ . This functor can establish a mapping from our abstract construction to any concrete setting, involving diverse explainers (e.g., decision trees, logistic regression), input data types (e.g., images, text), supervisions, outputs, parameters, or explanations. Achieving this mapping requires the definition of a specific functor we call *translator*.

**Definition 6.** *An agent translator is a feedback Cartesian functor<sup>3</sup>  $\mathcal{T} : \mathbf{XLearn} \rightarrow \mathbf{Stream}_{\mathbf{Set}}$ .*

Among translators, we distinguish two significant classes: those that instantiate learning processes and those that instantiate explaining learning processes (Definitions 7 and 8 respectively). Intuitively, a concrete learning process is an instance of an abstract learning process whose model does not provide any explanation while in a concrete explaining learning process it does output an (non-empty) explanation.

**Definition 7.** *Given an agent translator  $\mathcal{T}$  with  $\mathcal{T}(E) = \{*\}^{\mathbb{N}}$ , where  $\{*\}$  is a singleton set. A learning process (LP) is the image  $\mathcal{T}(\alpha)$ , being  $\alpha$  the abstract learning process.*

The set  $\{*\}^{\mathbb{N}}$  denotes the neutral element of the monoidal product in  $\mathbf{Stream}_{\mathbf{Set}}$  and conveys the absence of explanations. In this case,  $\mathcal{T}(\eta)$  will be simply called *model*, and we will remove the explicit dependence on  $\{*\}$  in the output space as  $\mathcal{T}(Y) \times \{*\}^{\mathbb{N}} \cong \mathcal{T}(Y)$ . To instantiate explaining learning processes instead, we introduce two distinct types of translators: the semantic and the syntactic translator. This choice is motivated by the fundamental elements of institution theory, namely sentences and models: Sentences correspond to well-formed *syntactic* expressions, while models capture the *semantic* interpretations of these sentences [32]. We refer to concrete instances of both syntactic and semantic explaining learning processes as “explaining learning processes” (XLP).

**Definition 8.** *Let  $\mathcal{T}$  be an agent translator,  $I$  an institution and  $\alpha$  the abstract learning process. The image  $\mathcal{T}(\alpha)$  is said a syntactic explaining learning process if  $\mathcal{T}(E) = \mathit{Sen}(\Sigma)^{\mathbb{N}}$  and a semantic explaining learning process if  $\mathcal{T}(E) = \mathit{Mod}(\Sigma)^{\mathbb{N}}$ , for some signature  $\Sigma$  of  $I$ .*

The high degree of generality in this formalization enables the definition of any real-world learning setting and learning architecture (to the author knowledge). Indeed, using Cartesian streams as the co-domain of translators, we can

---

<sup>3</sup> *Feedback functors* are mappings between feedback categories that preserve the structure and axioms of feedback categories.

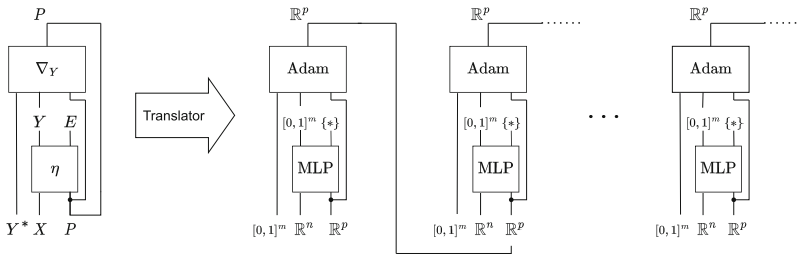
effectively model a wide range of learning use cases, including (but not limited to) those involving iterative feedback. To simplify the notation, in the following sections we use the shortcuts:  $\mathcal{X} = \mathcal{T}(X)$ ,  $\mathcal{Y} = \mathcal{T}(Y)$ ,  $\mathcal{Y}^* = \mathcal{T}(Y^*)$ ,  $\mathcal{P} = \mathcal{P}$ ,  $\mathcal{E} = \mathcal{T}(E)$ ,  $\hat{\eta} = \mathcal{T}(\eta)$ ,  $\hat{\nabla}_{\mathcal{Y}} = \mathcal{T}(\nabla_Y)$ , and  $\mathcal{Y} = \mathcal{Y}^*$  when not specified otherwise.

## 4 Impact on XAI and Key Findings

### 4.1 Finding #1: Our Framework Models Existing Learning Schemes and Architectures

As a proof of concept, in the following examples we show how the proposed categorical framework makes it possible to capture the structure of some popular learning algorithms such as supervised learning of neural networks.

*Example 3.* Classic supervised learning of a multi-layer perceptron (MLP, [69]) defined on  $n$ -dimensional feature vectors and  $m$  classes, can be obtained as an instance of an abstract learning process by means of the translator functor defined as follows (see Fig. 1):  $\mathcal{X} = (\mathbb{R}^n)^{\mathbb{N}}$ ,  $\mathcal{Y} = \mathcal{Y}^* = ([0, 1]^m)^{\mathbb{N}}$ ,  $\hat{\eta}_i$  being the same MLP for all  $i$ ,  $\mathcal{P}$  the space of the MLP parameters, e.g.  $\mathcal{P} = (\mathbb{R}^p)^{\mathbb{N}}$ ,  $\hat{\nabla}_{\mathcal{Y}_i}$  being e.g. the Adam optimizer [46], and  $\mathcal{E} = \{*\}^{\mathbb{N}}$ .



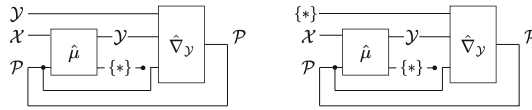
**Fig. 1.** On the left the string diagram representation of the abstract learning process. The translator functor maps each object and morphism into  $\text{Stream}_{\text{Set}}$ . On the right side of the arrow we see an unfolding of the resulting concrete learning process.

In Example 3 we model the MLP by fixing the components of the morphism  $\hat{\eta}$  to have constant values  $\hat{\eta}_i = \text{MLP}$  (and independent of the first  $i - 1$  inputs), so as the components of the input, output, and parameters are statically the same. This is also the case in the majority of the standard instances of concrete learning processes. However, the definition is a way more general and can be used to instantiate a broader class of learning processes and architectures including e.g. recurrent neural networks [36, 38] and transformers [86]. In these scenarios, the neural functions may become actually dependent on previous inputs, capturing the input stream. Additionally, we can also model learning settings where the model’s architecture changes over time, as in neural architecture search [22].

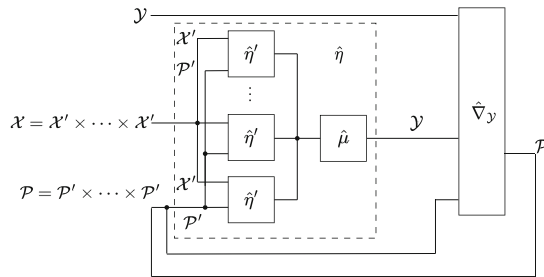


*Example 4.* A classical Neural Architecture Search algorithm [22] is an instance of an abstract learning process whose translator functor is defined as follows:  $\mathcal{X} = (\mathbb{R}^n)^{\mathbb{N}}$ ,  $\mathcal{Y} = \mathcal{Y}^* = ([0, 1]^m)^{\mathbb{N}}$ ,  $\hat{\eta}_i = \text{MLP}_i$  being a different neural architecture for every step,  $\mathcal{P}$  the space of the MLPs parameters, e.g.  $\mathcal{P} = (\mathbb{R}^p)^{\mathbb{N}}$ ,  $\hat{\nabla}_{\mathcal{Y}_i}$  being the Adam optimizer, and  $\mathcal{E} = \{*\}^{\mathbb{N}}$ .

To the best of our knowledge, the proposed formalism is general enough to potentially encompass any known learning process providing or not providing explanations. Indeed, the objects  $X, Y, Y^*, E$  can be instantiated through a suitable translator functor to have the desired characteristics, e.g.  $\mathcal{T}(Y)$  could include the explanations space, making the explanations optimizable, or  $\mathcal{T}(Y^*) = \{*\}^{\mathbb{N}}$ , to instantiate an unsupervised learning setting. As a proof of concept of the flexibility of our framework, we conclude this section with few more examples on how other common learning schemes can be easily instantiated (see Figs. 2 and 3).



**Fig. 2.** From left to right the instantiation of a classic supervised and unsupervised learning scheme, respectively. We keep the instance of the LPs fold for simplicity.



**Fig. 3.** Instantiation of a federated learning scheme. We keep the instance of the LP fold for simplicity.

### 4.2 Finding #2: Our Framework Enables a Formal Definition of “explanation”

The proposed theoretical framework allows us to provide the first formal definition of the term “explanation”, which is the fundamental notion at the base of explainable AI. Moreover, we opt for a definition that highlights a natural distinction between different forms of explanations, by relying on institution theory, which allows a straightforward characterization of syntactic and semantic explanations. While both forms of explanations are prevalent in the current

XAI literature, their distinction is often overlooked, thus limiting a deep understanding of the true nature and intrinsic limitations of a given explanation.

**Definition 9.** *Given an institution  $I$ , an object  $\Sigma$  of  $\text{Sign}_I$ , and a concrete explainer  $\hat{\eta} = \mathcal{T}(\eta) : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y} \times \mathcal{E}$ , an explanation  $\mathcal{E} = \mathcal{T}(E)$  in a language  $\Sigma$  is a set of  $\Sigma$ -sentences (syntactic explanation) or a model of a set of  $\Sigma$ -sentences (semantic explanation).*

We immediately follow up our definition with a concrete example to make it more tangible.

*Example 5.* Let  $I_{PL}$  be the institution of Propositional Logic and  $\Sigma$  a signature of  $I_{PL}$  with propositional variables  $\{x_{flies}, x_{animal}, x_{plane}, x_{bird}, \dots\} \subseteq \Sigma$  and with the standard connectives of Boolean Logic, i.e.  $\neg, \wedge, \vee, \rightarrow$ . For instance,  $\hat{\eta}$  could be an explainer aiming at predicting an output in  $\mathcal{Y} = \{x_{plane}, x_{bird}\}$  given an input in  $\mathcal{X} = \{x_{flies}, x_{animal}\}$ . Then a syntactic explanation could consist of a  $\Sigma$ -sentence like  $\varepsilon = x_{flies} \wedge \neg x_{animal} \rightarrow x_{plane}$ , a semantic explanation could be any truth-assignment to the variables in  $\varepsilon$ .

As a remark, we notice that Definition 9 generalizes and formalizes common (informal) definitions used in the literature, like the examples we cited in the Introduction from [13, 56, 61]. Our definition of explanation incorporates all these notions, as  $\Sigma$ -sentences and their models can provide any form of statement related to the explaining learning process and its inputs/outputs. This encompasses additional meta information and feature relevance [13], description of facts related to a learning process [61], or insights into why a specific output is obtained from a given input [56]. Furthermore, [80] and [32] proved how the semantics of “truth” is invariant under change of signature. This means that we can safely use signature morphisms to switch from one “notation” to another, inducing consistent syntactic changes in a  $\Sigma$ -sentence without conditioning the “meaning” or the “conclusion” of the sentence [32]. As a result, signature morphisms can translate a certain explanation between different signatures, hence paving the way to study “communication” as well as “understanding” between XLPs.

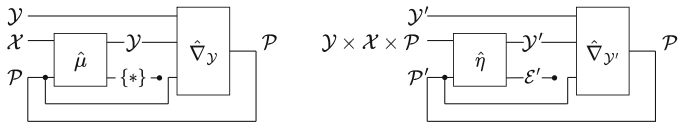
### 4.3 Finding #3: Our Framework Provides a Theoretical Foundation for XAI Taxonomies

Using our categorical constructions, we can develop a theory-grounded taxonomy of XAI methods that goes beyond current ones and catalogues existing approaches in a systematic and rigorous manner. We recognize the importance of such a foundation due to the ongoing debates and challenges faced by current taxonomies in comprehensively encompassing all existing XAI methods. Indeed, existing approaches in the XAI literature have only provided subjective viewpoints on the field, distinguishing methods according to controversial criteria such as: the scope/methodology/usage of explanations [13]; the how/what/why of explanations [61]; the relationship between explainers and systems being explained,

e.g., intrinsic/post-hoc, model-specific/agnostic, local/global [58]; or the specific data types, such as images [48], graphs [50], natural language [12], and tabular data [17]. However, these taxonomies lack a solid and grounded motivation and rely primarily on subjective preferences. As a result, they are unable to draw universal conclusions and provide a general understanding of the field. On the contrary, our taxonomy aims to fill this gap by providing a comprehensive classification of XAI methods grounded in a formal mathematical theory. As an example, we use the proposed categorical framework to explicitly describe the following macro-categories of XAI methods [13, 58, 61]. In case the explaining learning scheme involves more than one concrete process, we make use of two translator functors  $\mathcal{T}$  and  $\mathcal{T}'$ , and we refer to the objects of the latter using prime, e.g.  $\mathcal{T}'(Y) = \mathcal{Y}'$ . We keep  $\mathcal{Y} = \mathcal{Y}^*$  for simplicity.

*Post-hoc and Intrinsic.* XAI surveys currently distinguish between intrinsic and post-hoc explainers. Informally, the key difference is that intrinsic XAI methods evolve model parameters and explanations at the same time, whereas post-hoc methods extract explanations from pre-trained models [13, 58].

**Post-hoc Explainer.** Given a trained LP model  $\hat{\mu} : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y}$ , a post-hoc explainer is an XLP explainer such that  $\hat{\eta} : \mathcal{X}' \times \mathcal{P}' \rightarrow \mathcal{Y}' \times \mathcal{E}'$ , with  $\mathcal{X}' = \mathcal{Y} \times \mathcal{X} \times \mathcal{P}$ :



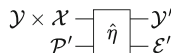
**Intrinsic Explainer.** An intrinsic explainer is an XLP explainer  $\hat{\eta}$  whose input objects are parameters  $\mathcal{P}'$  and a set of entries of a database  $\mathcal{X}'$ :



Common intrinsic explainers are logic/rule-based [5–7, 26, 54, 71, 91], linear [18, 35, 59, 70, 81, 87], and prototype-based [24, 44] approaches; while well-known post-hoc explainers include saliency maps [73, 75], surrogate models [52, 64, 66], and concept-based approaches [23, 29, 30, 45].

*Model-Agnostic and Model-Specific.* Intuitively, model-agnostic explainers extract explanations independently from the architecture and/or parameters of the model being explained, whereas model-specific explainers depend on the architecture and/or parameters of the model.

**Model-Agnostic Explainer.** Given an LP model  $\hat{\mu} : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y}$ , a model-agnostic explainer is an XLP explainer  $\hat{\eta} : \mathcal{X}' \times \mathcal{P}' \rightarrow \mathcal{Y}' \times \mathcal{E}'$ , such that  $\mathcal{X}' = \mathcal{Y} \times \mathcal{X}$ .



**Model-Specific Explainer.** A model-specific explainer simply differentiates from a model-agnostic, as the XLP explainer  $\hat{\eta} : \mathcal{X}' \times \mathcal{P}' \rightarrow \mathcal{Y}' \times \mathcal{E}'$  has  $\mathcal{X}' = \mathcal{Y} \times \mathcal{X} \times \mathcal{P}$ .

$$\mathcal{Y} \times \mathcal{X} \times \mathcal{P} \xrightarrow[\mathcal{P}']{\hat{\eta}} \mathcal{Y}' \times \mathcal{E}'$$

Typical examples of model-agnostic explainers include surrogate models [52, 64, 66] and some concept-based approaches [29, 30, 45]. Among renowned model-specific explainers instead we can include all model-intrinsic explainers [58] and some post-hoc explainers such as saliency maps [73, 75] as they can only explain gradient-based systems.

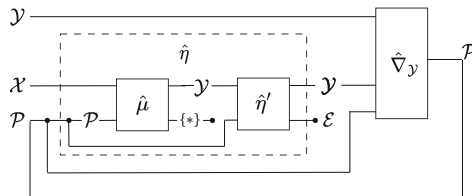
*Forward and Backward.* Another relevant difference among XAI methods for gradient-based models is whether the explainer relies on the upcoming parameters [62, 92, 93] or the gradient of the loss on the parameters in the learning optimizer [73, 75].

**Forward-Based explainer.** Given a gradient-based LP model  $\hat{\mu} : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y}$ , a forward-based explainer is an XLP explainer  $\hat{\eta} : \mathcal{X}' \times \mathcal{P}' \rightarrow \mathcal{Y}' \times \mathcal{E}'$  with  $\mathcal{X}' = \mathcal{X}'' \times \mathcal{P}$ .

$$\mathcal{X}'' \times \mathcal{P} \xrightarrow[\mathcal{P}']{\hat{\eta}} \mathcal{Y}' \times \mathcal{E}'$$

**Backward-Based Explainer.** Given a gradient-based LP model  $\hat{\mu} : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y}$  and an optimizer  $\hat{\nabla}_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \times \mathcal{P} \rightarrow \mathcal{P}$ , a backward-based explainer is an XLP explainer  $\hat{\eta} : \mathcal{X}' \times \mathcal{P}' \rightarrow \mathcal{Y}' \times \mathcal{E}'$  where,  $\mathcal{X}' = \mathcal{X}'' \times h(\mathcal{P})$ , being  $h(\mathcal{P}) = \frac{\partial \mathcal{L}(\mathcal{Y}, \mathcal{Y})}{\partial \mathcal{P}}$  the gradient of the loss function  $\mathcal{L}$  on  $\mathcal{P}$ .

*A Case Study: Concept Bottleneck Models.* Concept bottleneck models (CBM) [47] are recent XAI architectures which first predict a set of human understandable objects called “concepts” and then use these concepts to solve downstream classification tasks. Our framework allows to formally define these advanced XAI structures as follows: A CBM is an XLP such that  $\hat{\eta}$  is composed of a concept predictor  $\hat{\mu} : \mathcal{X} \times \mathcal{P} \rightarrow \mathcal{Y}$  and a task predictor  $\hat{\eta}' : \mathcal{Y} \times \mathcal{P} \rightarrow \mathcal{Y} \times \mathcal{E}$ ,  $\mathcal{Y}$  is the set of classes and  $\mathcal{P} = \mathcal{P}' \times \mathcal{P}''$  is the product of the parameter space of the two models.

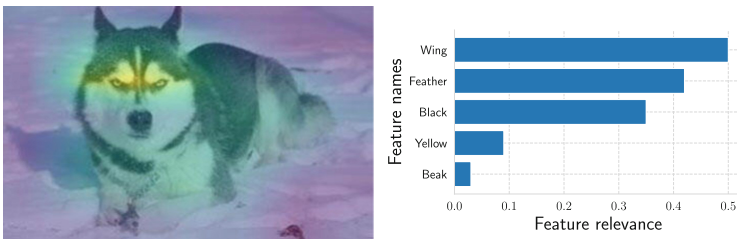


Overall these examples give a taste of the flexibility and expressive power of our categorical framework demonstrating how it can successfully encompass existing XAI approaches and taxonomies.

#### 4.4 Finding #4: Our Framework Emphasizes Commonly Overlooked Aspects of Explanations

Current XAI taxonomies often neglect the distinction between syntactic and semantic approaches. On the contrary, our Definition 8 provides a natural distinction between these two forms of XLPs, thus introducing a novel perspective to analyze XAI methods. On the one hand, syntactic approaches work on the structure of symbolic languages and operate on  $\Sigma$ -sentences. Notable examples of syntactic approaches are proof systems such as natural deduction [63] and sequent calculi [79] which are designed to operate on formal languages such as first-order logic. On the other hand, semantic approaches provide explanations related to the meaning or interpretation of sentences as a model of a language. Most XAI techniques actually fall into this class of methods as semantic explanations establish a direct connection with specific problem domains [42, 52, 64, 75]. We discuss the relation between a syntactic and a semantic technique with a couple of concrete examples, by emphasizing the connections between XAI methods that often slip unnoticed.

*Example 6.* The Gradient-weighted Class Activation Mapping (Grad-CAM), [75] is a classic example of a semantic (backward) XLP, whose institution can be defined as a fragment of FOL with all the signatures' objects consisting of a single predicate and a finite set of constants. Intuitively, in a classification task the constants represent the pixels of an image and the relation represents the saliency degree of each pixel for the class prediction. Sentences and models are defined as in FOL by using the provided signature. A general syntactic explanation in this institution can be easily expressed by taking a signature  $\Sigma = \{S, p_1, p_2, \dots\}$  with  $S$  the unary saliency predicate and  $p_i$  the constant for the  $i$ -th pixel. Assuming to collect the most “salient” pixels in the set  $SalPix$ , the syntactic explanation may be expressed by:  $\bigwedge_{p \in SalPix} S(p)$ . Figure 4–left instead represents a semantic explanation where Grad-CAM interprets predicates and constants in the syntactic formula assigning them a meaning (i.e., concrete values).



**Fig. 4.** (left) Saliency map. (right) Feature importance.

*Example 7.* Another classic example of semantic XLPs are feature importance methods [89], such as LIME, [65]. As saliency maps, LIME relies on an institution whose signatures consist on a set of constants  $f_i$  (each for every considered feature) and a single predicate  $R$  to express their relevance. Syntactic explanations have the form  $\bigwedge_{f \in \text{ImpFeat}} R(f)$  where the set  $\text{ImpFeat}$  collects the most “relevant” features for a task. Figure 4–right shows an example of a semantic explanation of LIME, where the length of each bar represents the relevance of the corresponding feature.

*Comparing the Expressive Power of Explanations.* The Grad-CAM and LIME examples offer the ideal setting to show how our framework can formally assess the expressive power of different forms of explanations, drawing connections between apparently different XLPs. Indeed, provided that both Grad-CAM and LIME signatures contain the same number of pixels/features, these two forms of explanations syntactically possess the same expressive power (while differing in their semantic models and generating algorithms for the explanations). Consequently, we can convert any saliency map into an equivalent feature importance representation and vice versa (using an arity-preserving function), without compromising their meaning/truth value [80]. This is generally true whenever it is possible to define an arity-preserving mapping between different signatures in the same institution. This observation underscores an often overlooked aspect in XAI literature: evaluating the expressive power of explanations requires comparing their signatures and syntax, more than the way these explanations are visualized. User studies that compare different forms of visualization essentially assess the visualization’s expressive power, which relates to human understanding, rather than the expressive power of an explanation itself.

*Limitations of Semantic Explanations.* The connection between a semantic explanation and a specific context (e.g., an image) may stimulate human imagination, but it often limits the scope and robustness of the explanation, hindering human understanding in the long run [29, 68]. On the contrary, symbolic languages such as first-order logic or natural language are preferable for conveying meaningful messages as explanations as suggested by [41, 55]. For this reason, a promising but often overlooked research direction consists in accurately lifting semantic explanations into a symbolic language in order to provide a syntactic explanation [6, 7, 9, 33, 49, 66]. By recognizing the importance of this distinction, our formalization can provide a suitable basis to gain deeper insights into the limitations of different forms of explanations.

## 5 Discussion

*Significance and Relations with Other XAI Foundational Works.* The explainable AI research field is growing at a considerable rate [57] and it now has a concrete impact on other research disciplines [10, 14, 39] as well as on the deployment of AI technologies [20, 51, 88]. This rising interest in explainable AI increases the

need for a sound foundation and taxonomy of the field [2, 61]. Indeed, existing reviews and taxonomies significantly contribute in: (i) describing and clustering the key methods and trends in the XAI literature [58], (ii) proposing the first qualitative definitions for the core XAI terminology [13], (iii) relating XAI methodologies, aims, and terminology with other scientific disciplines [28, 56], and (iv) identifying the key knowledge gaps and research opportunities [13]. However, most of these works acknowledge the need for a more sound mathematical foundation and formalism to support the field. Our framework arises to fill this gap. In particular our methodology formalizes key XAI notions for the first time, using the category of Cartesian streams and the category of signatures. Our work also draws from [76] who propose Cartesian streams to model gradient-based learning, and [11] who model gradient-based learning using the category of lenses [67]. The categorical formalisms of lenses and streams are closely related [15]. Intuitively, lenses can be used to encode one-stage processes, while streams can encode processes with time indexed by natural numbers, i.e. providing a more suitable description of the dynamic process of learning. However, our work opens to more general AI systems which are not necessarily gradient-based by generalizing the category of lenses with Cartesian streams.

*Limitations.* In this work we face the challenging task of formalizing (previously informal) notions such as “explanation”, while acknowledging the ongoing debate over their meaning, not only within the AI community but also in philosophy, epistemology, and psychology. Our formalization offers a robust theory-grounded foundation for explainable AI, but it does require readers to engage with abstract categorical structures. However, embracing this initial challenge brings a substantial payoff by enabling us to achieve a comprehensive and unified theory of the field, encompassing all the pertinent instantiations of XAI notions, structures, explanations, and paradigms.

*Conclusion.* This work presents the first formal theory of explainable AI. In particular, we formalized key notions and processes that were still lacking a rigorous definition. We then show that our categorical framework enables us to: (i) model existing learning schemes and architectures, (ii) formally define the term “explanation”, (iii) establish a theoretical basis for XAI taxonomies, and (iv) emphasize commonly overlooked aspects of XAI methods, like the comparison between syntactic and semantics explanations. Through this work, we provide a first answer to the pressing need for a sound foundation and formalism in XAI as advocated by the current literature [2, 61]. While our taxonomy provides guidance to navigate the field, our formalism strengthens the reputation of explainable AI encouraging a safe and ethical deployment of AI technologies. We think that this work may contribute in paving the way for new research directions in XAI, including the exploration of previously overlooked theoretical side of explainable AI, and the mathematical definition of other foundational XAI notions, like “understandability” and “trustworthiness”.

**Acknowledgement.** This paper was supported by: TAILOR, the FWF project P33878 “Equations in Universal Algebra”, HumanE-AI-Net projects funded by EU Horizon 2020 under GA No 952215 and No 952026, EU Horizon 2020 under GA No 848077, Horizon-MSCA-2021 under GA No 101073307, the SNF project “TRUST-ME” No 205121L-214991. This work has been also supported by the Partnership Extended PE00000013 - “FAIR - Future Artificial Intelligence Research” - Spoke 1 “Human-centered AI”.

**Disclosure of Interests.** The authors have no competing interests.

## A Elements of Category Theory

### A.1 Monoidal Categories

The process interpretation of monoidal categories [8,27] sees morphisms in monoidal categories as modelling processes with multiple inputs and multiple outputs. Monoidal categories also provide an intuitive syntax for them through string diagrams [40]. The coherence theorem for monoidal categories [53] ensures that string diagrams are a sound and complete syntax for them and thus all coherence equations for monoidal categories correspond to continuous deformations of string diagrams. One of the main advantages of string diagrams is that they make reasoning with equational theories more intuitive.

**Definition 1** ([21]). *A category  $\mathcal{C}$  is given by a class of objects  $\mathcal{C}^\circ$  and, for every two objects  $X, Y \in \mathcal{C}^\circ$ , a set of morphisms  $\text{hom}(X, Y)$  with input type  $X$  and output type  $Y$ . A morphism  $f \in \text{hom}(X, Y)$  is written  $f: X \rightarrow Y$ . For all morphisms  $f: X \rightarrow Y$  and morphisms  $g: Y \rightarrow Z$  there is a composite morphism  $f; g: X \rightarrow Z$ . For each object  $X \in \mathcal{C}^\circ$  there is an identity morphism  $\mathbb{1}_X \in \text{hom}(X, X)$ , which represents the process that “does nothing” to the input and just returns it as it is. Composition needs to be associative, i.e. there is no ambiguity in writing  $f; g; h$ , and unital, i.e.  $f; \mathbb{1}_Y = f = \mathbb{1}_X; f$ .*

Monoidal categories [53] are categories endowed with extra structure, a monoidal product and a monoidal unit, that allows morphisms to be composed *in parallel*. The monoidal product is a functor  $\times: \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$  that associates to two processes,  $f_1: X_1 \rightarrow Y_1$  and  $f_2: X_2 \rightarrow Y_2$ , their parallel composition  $f_1 \times f_2: X_1 \times X_2 \rightarrow Y_1 \times Y_2$ . The monoidal unit is an object  $U \in \mathcal{C}^\circ$ , which represents the “absence of inputs or outputs” and needs to satisfy  $X \times U \cong X \cong U \times X$ , for each  $X \in \mathcal{C}^\circ$ . For this reason, this object is often not drawn in string diagrams and a morphism  $s: U \rightarrow Y$ , or  $t: X \rightarrow U$ , is represented as a box with no inputs, or no outputs.

### A.2 Cartesian and Symmetric Monoidal Categories

A symmetric monoidal structure on a category is required to satisfy some coherence conditions [53], which ensure that string diagrams are a sound and complete syntax for symmetric monoidal categories [40]. Like functors are mappings



between categories that preserve their structure, *symmetric monoidal functors* are mappings between symmetric monoidal categories that preserve the structure and axioms of symmetric monoidal categories.

Some symmetric monoidal categories have additional structure that allows resources to be copied and discarded [25]. These are called *Cartesian categories*.

### A.3 Feedback Monoidal Categories

*Feedback monoidal functors* are mappings between feedback monoidal categories that preserve the structure and axioms of feedback monoidal categories.

Feedback monoidal categories are the *syntax* for processes with feedback loops. When the monoidal structure of a feedback monoidal category is cartesian, we call it *feedback cartesian category*. Their *semantics* can be given by monoidal streams [15]. In cartesian categories, these have an explicit description. We refer to them as cartesian streams, but they have appeared in the literature multiple times under the name of “stateful morphism sequences” [76] and “causal stream functions” [84].

### A.4 Free Categories

We generate “abstract” categories using the notion of *free category* [53]. Intuitively, a free category serves as a template for a class of categories (e.g., feedback monoidals). To generate a free category, we just need to specify a set of objects and morphisms generators. Then we can realize “concrete” instances of a free category  $F$  using a functor from  $F$  to another category  $C$  that preserves the axioms of  $F$ . If such a functor exists then  $C$  is of the same type of  $F$  (e.g., the image of a free feedback monoidal category via a feedback functor is a feedback monoidal category).

### A.5 Institutions

An *institution*  $I$  is constituted by:

- (i) a category  $\text{Sign}_I$  whose objects are signatures (i.e. vocabularies of symbols);
- (ii) a functor  $\text{Sen} : \text{Sign}_I \mapsto \text{Set}$  providing sets of well-formed expressions ( $\Sigma$ -sentences) for each signature  $\Sigma \in \text{Sign}_I^o$ ;
- (iii) a functor  $\text{Mod} : \text{Sign}_I^{op} \mapsto \text{Set}$  providing semantic interpretations, i.e. worlds.

Furthermore, Satisfaction is then a parametrized relation  $\models_\Sigma$  between  $\text{Mod}(\Sigma)$  and  $\text{Sen}(\Sigma)$ , such that for all signature morphism  $\rho : \Sigma \mapsto \Sigma'$ ,  $\Sigma'$ -model  $M'$ , and any  $\Sigma$ -sentence  $e$ ,

$$M' \models_\Sigma \rho(e) \text{ iff } \rho(M') \models_{\Sigma'} e$$

where  $\rho(e)$  abbreviates  $\text{Sen}(\rho)(e)$  and  $\rho(M')$  stands for  $\text{Mod}(\rho)(e)$ .

## References

1. Abramsky, S., Coecke, B.: A categorical semantics of quantum protocols. In: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, pp. 415–425 (2004)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE Access* **6**, 52138–52160 (2018)
3. Aguinaldo, A., Regli, W.: A graphical model-based representation for classical ai plans using category theory. In: ICAPS 2021 Workshop on Explainable AI Planning (2021)
4. Arrieta, A.B., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020)
5. Barbiero, P., et al.: Interpretable neural-symbolic concept reasoning. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) Proceedings of the 40th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 202, pp. 1801–1825. PMLR (2023). <https://proceedings.mlr.press/v202/barbiero23a.html>
6. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC Press, Boca Raton (1984)
7. Ciravegna, G., Barbiero, P., Giannini, F., Gori, M., Lió, P., Maggini, M., Melacci, S.: Logic explained networks. *Artif. Intell.* **314**, 103822 (2023)
8. Coecke, B., Kissinger, A.: Picturing Quantum Processes - A first course in Quantum Theory and Diagrammatic Reasoning. Cambridge University Press, Cambridge (2017)
9. Costa, F., Ouyang, S., Dolog, P., Lawlor, A.: Automatic generation of natural language explanations. In: Proceedings of the 23rd International Conference on Intelligent User Interfaces Companion, pp. 1–2 (2018)
10. Cranmer, M.D., Xu, R., Battaglia, P., Ho, S.: Learning symbolic physics with graph networks. arXiv preprint [arXiv:1909.05862](https://arxiv.org/abs/1909.05862) (2019)
11. Cruttwell, G.S.H., Gavranović, B., Ghani, N., Wilson, P., Zanasi, F.: Categorical foundations of gradient-based learning. In: ESOP 2022. LNCS, vol. 13240, pp. 1–28. Springer, Cham (2022). [https://doi.org/10.1007/978-3-030-99336-8\\_1](https://doi.org/10.1007/978-3-030-99336-8_1)
12. Danilevsky, M., Qian, K., Aharonov, R., Katsis, Y., Kawas, B., Sen, P.: A survey of the state of explainable AI for natural language processing. arXiv preprint [arXiv:2010.00711](https://arxiv.org/abs/2010.00711) (2020)
13. Das, A., Rad, P.: Opportunities and challenges in explainable artificial intelligence (xai): a survey. ArXiv [arxiv:2006.11371](https://arxiv.org/abs/2006.11371) (2020)
14. Davies, A., et al.: Advancing mathematics by guiding human intuition with AI. *Nature* **600**(7887), 70–74 (2021)
15. Di Lavore, E., de Felice, G., Román, M.: Monoidal streams for dataflow programming. In: Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science. Association for Computing Machinery, New York (2022), <https://doi.org/10.1145/3531130.3533365>
16. Di Lavore, E., Gianola, A., Román, M., Sabadini, N., Sobociński, P.: A canonical algebra of open transition systems. In: Salaün, G., Wijs, A. (eds.) FACS 2021. LNCS, vol. 13077, pp. 63–81. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-90636-8\\_4](https://doi.org/10.1007/978-3-030-90636-8_4)
17. Di Martino, F., Delmastro, F.: Explainable AI for clinical and remote health applications: a survey on tabular and time series data. *Artif. Intell. Rev.* 1–55 (2022)

18. Doshi-Velez, F., Wallace, B.C., Adams, R.: Graph-sparse lda: a topic model with structured sparsity. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
19. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: a survey. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 0210–0215. IEEE (2018)
20. Durán, J.M., Jongsma, K.R.: Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical AI. *J. Med. Ethics* **47**(5), 329–335 (2021)
21. Eilenberg, S., MacLane, S.: General theory of natural equivalences. *Trans. Am. Math. Soc.* **58**(2), 231–294 (1945)
22. Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: a survey. *J. Mach. Learn. Res.* **20**(1), 1997–2017 (2019)
23. Espinosa Zarlenga, M., et al.: Concept embedding models: beyond the accuracy-explainability trade-off. *Adv. Neural. Inf. Process. Syst.* **35**, 21400–21413 (2022)
24. Fix, E., Hodges, J.L.: Discriminatory analysis. nonparametric discrimination: Consistency properties. *Int. Stat. Rev./Revue Internationale de Statistique* **57**(3), 238–247 (1989)
25. Fox, T.: Coalgebras and cartesian categories. *Comm. Algebra* **4**(7), 665–667 (1976)
26. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. *Ann. Appl. Stat.* 916–954 (2008)
27. Fritz, T.: A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Adv. Math.* **370**, 107239 (2020)
28. Geiger, A., Potts, C., Icard, T.: Causal abstraction for faithful model interpretation. arXiv preprint [arXiv:2301.04709](https://arxiv.org/abs/2301.04709) (2023)
29. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 3681–3688 (2019)
30. Ghorbani, A., Wexler, J., Zou, J., Kim, B.: Towards automatic concept-based explanations. arXiv preprint [arXiv:1902.03129](https://arxiv.org/abs/1902.03129) (2019)
31. Goguen, J.: What is a concept? In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS-ConceptStruct 2005. LNCS (LNAI), vol. 3596, pp. 52–77. Springer, Heidelberg (2005). [https://doi.org/10.1007/11524564\\_4](https://doi.org/10.1007/11524564_4)
32. Goguen, J.A., Burstall, R.M.: Institutions: abstract model theory for specification and programming. *J. ACM (JACM)* **39**(1), 95–146 (1992)
33. Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., Giannotti, F.: Local rule-based explanations of black box decision systems. arXiv preprint [arXiv:1805.10820](https://arxiv.org/abs/1805.10820) (2018)
34. Gunning, D., Stefik, M., Choi, J., Miller, T., Stumpf, S., Yang, G.Z.: Xai-explainable artificial intelligence. *Sci. Rob.* **4**(37), eaay7120 (2019)
35. Hastie, T.J.: Generalized additive models. In: *Statistical Models in S*, pp. 249–307. Routledge (2017)
36. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
37. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for explainable ai: challenges and prospects. arXiv preprint [arXiv:1812.04608](https://arxiv.org/abs/1812.04608) (2018)
38. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.* **79**(8), 2554–2558 (1982)
39. Jiménez-Luna, J., Grisoni, F., Schneider, G.: Drug discovery with explainable artificial intelligence. *Nat. Mach. Intell.* **2**(10), 573–584 (2020)


40. Joyal, A., Street, R.: The geometry of tensor calculus, i. *Adv. Math.* **88**(1), 55–112 (1991)
41. Kahneman, D.: *Thinking, Fast and Slow*. Macmillan, New York (2011)
42. Karasmanoglou, A., Antonakakis, M., Zervakis, M.: Heatmap-based explanation of yolov5 object detection with layer-wise relevance propagation. In: *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1–6. IEEE (2022)
43. Katis, P., Sabadini, N., Walters, R.F.C.: Feedback, trace and fixed-point semantics. *RAIRO-Theor. Inf. Appl.* **36**(2), 181–194 (2002)
44. Kaufmann, L.: Clustering by means of medoids. In: *Proceedings of Statistical Data Analysis Based on the L1 Norm Conference*, Neuchatel, 1987, pp. 405–416 (1987)
45. Kim, B., Khanna, R., Koyejo, O.O.: Examples are not enough, learn to criticize! criticism for interpretability. *Adv. Neural Inf. Process. Syst.* **29** (2016)
46. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)* (2014)
47. Koh, P.W., et al.: Concept bottleneck models. In: *International Conference on Machine Learning*, pp. 5338–5348. PMLR (2020)
48. Kulkarni, A., Shivananda, A., Sharma, N.R.: Explainable AI for computer vision. In: *Computer Vision Projects with PyTorch*, pp. 325–340. Springer, Heidelberg (2022). [https://doi.org/10.1007/978-1-4842-8273-1\\_10](https://doi.org/10.1007/978-1-4842-8273-1_10)
49. Letham, B., Rudin, C., McCormick, T.H., Madigan, D., et al.: Interpretable classifiers using rules and bayesian analysis: building a better stroke prediction model. *Ann. Appl. Stat.* **9**(3), 1350–1371 (2015)
50. Li, Y., Zhou, J., Verma, S., Chen, F.: A survey of explainable graph neural networks: Taxonomy and evaluation metrics. *arXiv preprint [arXiv:2207.12599](https://arxiv.org/abs/2207.12599)* (2022)
51. Lo Piano, S.: Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. *Human. Social Sci. Commun.* **7**(1), 1–7 (2020)
52. Lundberg, S., Lee, S.I.: A unified approach to interpreting model predictions. *arXiv preprint [arXiv:1705.07874](https://arxiv.org/abs/1705.07874)* (2017)
53. Mac Lane, S.: *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer, New York (1978). <https://doi.org/10.1007/978-1-4757-4721-8>
54. Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., De Raedt, L.: Deep-problog: neural probabilistic logic programming. *Adv. Neural Inf. Process. Syst.* **31** (2018)
55. Marcus, G.: The next decade in AI: four steps towards robust artificial intelligence. *arXiv preprint [arXiv:2002.06177](https://arxiv.org/abs/2002.06177)* (2020)
56. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019)
57. Minh, D., Wang, H.X., Li, Y.F., Nguyen, T.N.: Explainable artificial intelligence: a comprehensive review. *Artif. Intell. Rev.* **55**(5), 3503–3568 (2022)
58. Molnar, C.: *Interpretable machine learning* (2020). <https://www.lulu.com/>
59. Nelder, J.A., Wedderburn, R.W.: Generalized linear models. *J. Roy. Stat. Soc.: Ser. A (Gen.)* **135**(3), 370–384 (1972)
60. Ong, E., Veličković, P.: Learnable commutative monoids for graph neural networks. *arXiv preprint [arXiv:2212.08541](https://arxiv.org/abs/2212.08541)* (2022)
61. Palacio, S., Lucieri, A., Munir, M., Ahmed, S., Hees, J., Dengel, A.: Xai handbook: towards a unified framework for explainable AI. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3766–3775 (2021)
62. Petsiuk, V., Das, A., Saenko, K.: Rise: randomized input sampling for explanation of black-box models. *arXiv preprint [arXiv:1806.07421](https://arxiv.org/abs/1806.07421)* (2018)

63. Prawitz, D.: *Natural Deduction: A Proof-Theoretical Study*. Courier Dover Publications, Mineola (2006)
64. Ribeiro, M.T., Singh, S., Guestrin, C.: “why should i trust you?” explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016)
65. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. arXiv preprint [arXiv:1606.05386](https://arxiv.org/abs/1606.05386) (2016)
66. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: high-precision model-agnostic explanations. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32 (2018)
67. Riley, M.: Categories of optics. arXiv preprint [arXiv:1809.00738](https://arxiv.org/abs/1809.00738) (2018)
68. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **1**(5), 206–215 (2019)
69. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science (1985)
70. Santosa, F., Symes, W.W.: Linear inversion of band-limited reflection seismograms. *SIAM J. Sci. Stat. Comput.* **7**(4), 1307–1330 (1986)
71. Schmidt, M., Lipson, H.: Distilling free-form natural laws from experimental data. *Science* **324**(5923), 81–85 (2009)
72. Selinger, P.: Control categories and duality: on the categorical semantics of the lambda-mu calculus. *Math. Struct. Comput. Sci.* **11**, 207–260 (2001)
73. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626 (2017)
74. Shiebler, D., Gavranović, B., Wilson, P.: Category theory in machine learning. arXiv preprint [arXiv:2106.07032](https://arxiv.org/abs/2106.07032) (2021)
75. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv preprint [arXiv:1312.6034](https://arxiv.org/abs/1312.6034) (2013)
76. Sprunger, D., Katsumata, S.: Differentiable causal computations via delayed trace. In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, 24–27 June 2019*, pp. 1–12. IEEE (2019). <https://doi.org/10.1109/LICS.2019.8785670>
77. Stein, D., Staton, S.: Compositional semantics for probabilistic programs with exact conditioning. In: *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–13 (2021). <https://doi.org/10.1109/LICS52264.2021.9470552>
78. Swan, J., Nivel, E., Kant, N., Hedges, J., Atkinson, T., Steunebrink, B.: A compositional framework. In: *The Road to General Intelligence*, pp. 73–90. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-08020-3\\_9](https://doi.org/10.1007/978-3-031-08020-3_9)
79. Takeuti, G.: *Proof Theory*, vol. 81. Courier Corporation, Mineola (2013)
80. Tarski, A.: The semantic conception of truth: and the foundations of semantics. *Phil. Phenomenol. Res.* **4**(3), 341–376 (1944)
81. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **58**(1), 267–288 (1996)
82. Tjoa, E., Guan, C.: A survey on explainable artificial intelligence (xai): toward medical xai. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(11), 4793–4813 (2020)

83. Turi, D., Plotkin, G.D.: Towards a mathematical operational semantics. In: Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science, pp. 280–291 (1997)
84. Uustalu, T., Vene, V.: The essence of dataflow programming. In: Yi, K. (ed.) APLAS 2005. LNCS, vol. 3780, pp. 2–18. Springer, Heidelberg (2005). [https://doi.org/10.1007/11575467\\_2](https://doi.org/10.1007/11575467_2)
85. Uustalu, T., Vene, V.: Comonadic notions of computation. *Electron. Notes Theor. Comput. Sci.* **203**(5), 263–284 (2008)
86. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
87. Verhulst, P.F.: Resherches mathematiques sur la loi d'accroissement de la population. *Nouveaux memoires de l'academie royale des sciences* **18**, 1–41 (1845)
88. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017)
89. Wei, P., Lu, Z., Song, J.: Variable importance analysis: a comprehensive review. *Reliabil. Eng. Syst. Saf.* **142**, 399–432 (2015)
90. Wilson, P., Zanasi, F.: Reverse derivative ascent: a categorical approach to learning Boolean circuits. *Electron. Proc. Theor. Comput. Sci.* **333**, 247–260 (2021)
91. Yang, H., Rudin, C., Seltzer, M.: Scalable bayesian rule lists. In: International Conference on Machine Learning, pp. 3921–3930. PMLR (2017)
92. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10590-1\\_53](https://doi.org/10.1007/978-3-319-10590-1_53)
93. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: prediction difference analysis. arXiv preprint [arXiv:1702.04595](https://arxiv.org/abs/1702.04595) (2017)



# Investigating Calibrated Classification Scores Through the Lens of Interpretability

Alireza Torabian and Ruth Urner<sup>(✉)</sup> 

EECS Department, York University, Toronto, Canada  
{talireza,uruth}@yorku.ca

**Abstract.** Calibration is a frequently invoked concept when useful label probability estimates are required on top of classification accuracy. A calibrated model is a function whose values correctly reflect underlying label probabilities. Calibration in itself however does not imply classification accuracy, nor human interpretable estimates, nor is it straightforward to verify calibration from finite data. There is a plethora of evaluation metrics (and loss functions) that each assess a specific aspect of a calibration model. In this work, we initiate an axiomatic study of the notion of calibration. We catalogue desirable properties of calibrated models as well as corresponding evaluation metrics and analyze their feasibility and correspondences. We complement this analysis with an empirical evaluation, comparing common calibration methods to employing a simple, interpretable decision tree.

**Keywords:** Calibration · Axiomatic analysis · Evaluation measures

## 1 Introduction

In many applications it is important that a classification model not only has high accuracy but that a user is also provided with a reliable estimate of confidence in the predicted label. Calibration is a concept that is often invoked to provide such confidence estimates to a user. As such, calibration is a notion that is *inherently aimed at human interpretation*. In binary classification, a perfectly calibrated model  $f$  provides the guarantee that if it predicts  $f(x) = p$  on some instance  $x$ , then *among the set of all instances on which  $f$  assigns this value  $p$*  the probability of label 1 is indeed  $p$  (and the probability of label 0 thus  $1 - p$ ).

While calibration is generally considered useful, we would argue that in many cases, even if achieved, it is doomed to fail at its original goal of providing insight to a human user: for most suitably complex classification models, a human user that observes  $f(x) = p$  has no notion of the set of all instances on which  $f$  also outputs  $p$ . The promise given by calibration is thus meaningless.

In this work we investigate which (additional) properties would actually yield human understandable calibration scores. We take an axiomatic (or property based) approach and start by outlining formal desiderata for a calibrated model.

The first and obvious property is high classification accuracy, which is not implied by calibration. Second, a property that is often implicitly aimed at in the context of calibration is that the predictor actually approximates the regression function  $\eta(x) = \mathbb{P}[y = 1|x]$  of the data-generating process [25]. This is however also not implied by calibration. We then propose three properties that relate to interpretability: 1) that the pre-images (or level sets)  $f^{-1}(r)$  of the model are identifiable to a user, 2) that the range of values that the model outputs is not too large, and 3) that the model is monotonic with respect to the underlying regression function. Section 3 starts with a detailed discussion and motivation for these suggested desiderata. In that section we also formally analyze the interplay between these (initially strictly phrased) properties.

Since a learned model can usually not be expected to satisfy properties such as optimal accuracy or calibration perfectly, in Sect. 4 we then move to outlining relaxations of our desiderata in form of measures of distance from the properties. The discussion and analysis in that section focuses on measures at the population level of a data-generating process. We analyze how simple operations on a predictor, which may improve its calibration, affect these measures.

In the last Sect. 5, we deal with empirical, finite data based versions of these measures. We again start by outlining and discussing these empirical measures, most of which are from the literature on calibration. Our experiments on a variety of real world datasets then evaluate them on a simple, inherently interpretable model for calibration, namely a decision tree, and compare its performance to three other, not necessarily interpretable standard calibration methods. The goal of this section is to take a model for which we can control our interpretability criteria (the pre-images  $f^{-1}(r)$  are here the leaves of the tree and thus interpretable, and the number of these can be set by a user), and assess how this interpretable model compares to other methods in terms of other performance measures.

In summary, we systematically outline and analyze the interplay between desirable properties, evaluation measures and sometimes implicit objectives on three levels: on an idealized level as axioms (or deterministic properties), on the distributional level as probabilistic metrics, and on the empirical level as measures to be estimated from finite data. While the first level is aimed at capturing the aleatoric uncertainty in the data generation, and the second defines measures of how well a predictor reflects this uncertainty, the last level integrates the epistemic uncertainty, namely how to estimate these qualities from samples. Our work sheds light on the role of interpretability in the context of calibration, which we find essential for calibration to be meaningful and thus useful to users.

**Overview on Related Work.** Calibration is a well established notion with studies on this concept dating back decades [6, 9, 15]. Summarizing this rich body of literature is beyond the scope of this manuscript, but recent surveys provide an overview on the concept of calibration, common methods aimed at achieving it and popular evaluation metrics [24, 26]. With the advent of increasingly powerful yet opaque machine learning models, the concept of calibration has enjoyed renewed interest and research activity in recent years [1, 2, 8, 12].



Methods to obtain calibration broadly fall into two categories: post-processing an existing model or directly training in a way that promotes calibration in the learned model. Platt Scaling (PS) [20] and Isotonic Regression (IS) [27] (which we include in our experiments) are two well established methods in the former category. Another class of commonly used post-processing methods, for which formal guarantees also exist, is re-calibration based on binning [11, 16, 19, 25]. To directly promote calibration, training by optimizing a *proper loss* is often recommended. Proper losses are minimized by the data-generating distribution’s regression function. Very recent work has analyzed when this actually leads to calibrated models [2].

A major challenge with understanding how to obtain calibrated models is the lack of clear, commonly accepted criteria for “how uncalibrated” a model is. There are a variety of studies that aim to address this inherent ambiguity both from a practical point of view, by systematically developing and comparing evaluation methods [8, 21, 24] and from a theoretical perspective by formally establishing failure modes and success guarantees [1].

While some recent studies point out contributions of calibration *for model interpretability* [23], we are not aware of a systematic analysis of the interpretability *of calibration* itself, which is the focus of this work.

## 2 Formal Setup

*Binary Classification.* We consider the standard setup of statistical learning: We let  $X$  denote a feature space and  $Y = \{0, 1\}$  the label space. The data generation is modelled as a distribution  $D$  over  $X \times Y$ . We use  $D_X$  to denote the marginal of  $D$  over  $X$ . We use  $\text{supp}(\cdot)$  to denote the support of a distribution. With slight abuse of notation, for a distribution  $D$  over  $X \times Y$ , we will often write  $\text{supp}(D)$  to also refer to the support  $\text{supp}(D_X)$  of the marginal  $D_X$ . Further, we let  $\eta_D : X \rightarrow [0, 1]$  denote the *regression function* of the distribution  $D$ :

$$\eta_D(x) = \mathbb{P}_{(x',y) \sim D}[y = 1 | x' = x]$$

A *predictor* is a function  $f : X \rightarrow \mathbb{R}$  that assigns every instance a real valued score. Given a data generating distribution  $D$ , we let  $\text{range}_D(f)$  denote the *effective range* of the predictor, namely the smallest set  $R$  such that with probability 1 over drawing  $x \sim D_X$ , we have  $f(x) \in R$ . For discrete distributions, we can alternatively define  $\text{range}_D(f) := \{f(x) | x \in \text{supp}(D_X)\}$ . For simplicity, we will usually use statements such as “for all  $x \in \text{supp}(D_X)$ ” instead of “with probability 1 over  $D_X$ ”, and “there exist  $x \in \text{supp}(D_X)$ ” instead of “with probability greater than 0 over  $D_X$ ”. These concepts are equivalent for discrete distributions (and under some continuity assumptions on functions in the non-discrete case). The above substitutions can be made for more general cases.

We define the *cells generated by predictor  $f$*  as the subsets of  $X$  on which  $f$  is constant, i.e., the pre-images under  $f$  of the values in  $\text{range}_D(f)$ ; a predictor  $f$  thus partitions  $X$  into cells.

A *classifier* is a function  $h : X \rightarrow Y$  that assigns every feature vector a class label. For binary classification, it is common to threshold some predictor for this. Given  $f : X \rightarrow \mathbb{R}$ , we define the classifier induced by  $f$  with threshold  $\theta$  as

$$f_\theta(x) = \mathbb{1}[f(x) \geq \theta]$$

where  $\mathbb{1}[\cdot]$  denotes the indicator function. We use  $\mathcal{F} = \mathbb{R}^X$  to denote the set of all (measurable) predictors. Predictors  $f$  are evaluated by means of a *loss function*  $\ell : \mathcal{F} \times X \times Y \rightarrow \mathbb{R}$ , where  $\ell(f, x, y)$  indicates the quality of prediction  $f(x)$  given observed label  $y$ . The goal is to achieve low *expected loss*

$$\mathcal{L}_D(f) = \mathbb{E}_{(x,y) \sim D}[\ell(f, x, y)].$$

The *binary loss* (or *0/1-loss*) is the standard evaluation metric for classifiers  $\ell^{0/1}(h, x, y) = \mathbb{1}[h(x) \neq y]$ . The *Bayes classifier* is a classifier with minimal expected binary loss, denoted by  $\text{opt}_D^{0/1}$ , the *Bayes loss* of  $D$ .

*Calibration.* In many applications, it is desirable to not only achieve low classification loss (that is high accuracy), but to have a predictor that accurately reflects *probabilities* of the label events. The notion of *calibration* defines such a property; namely, that the predicted value  $f(x)$  accurately reflects the probability of seeing label 1 among all instances that are given value  $f(x)$  [6, 9, 15, 24].

**Definition 1.** A predictor  $f : X \rightarrow [0, 1]$  is calibrated if for all  $x \in X$  we have:

$$f(x) = \mathbb{E}_{(x', y') \sim D}[y' \mid f(x') = f(x)]$$

Predictors are rarely expected to be perfectly calibrated as in the above definition. There are a variety of notions to measure “degrees of miscalibration” both with respect to the underlying distribution and empirically as observed on a dataset. We outline some of these in later parts of this work (see Sects. 4 and 5). We note here that, due to the conditioning on the level sets of the predictor in the definition of calibration, there is no straightforward way of measuring miscalibration, in particular not as an expectation over a pointwise defined loss function which would depend only on a predictor  $f$  and an observation  $(x, y)$ .

### 3 Desiderata for Calibration

We now list some formal requirements for predictors that are aimed to be calibrated. The goal here is to make often implicit motivations explicit and formal. The first, obvious, requirement (first item in the list) is calibration itself as defined in Definition 1. However, such a predictor should often have additional qualities that are not subsumed by the notion of calibration. Of course it is still desirable (if not imperative) that the predictor allows to be thresholded into a classifier with high accuracy (second item in the list). Moreover, the hope behind calibration is often that the predictor  $f$  will actually be a good representation

of data-generating distribution’s regression function  $\eta_D$  (third item). Neither of these latter two requirements is implied by calibration. We will formally elaborate on this in Sect. 3.1 below.

Furthermore, we would argue that the concept of calibration is inherently aimed at *aiding human interpretation*. The intent of providing a probability estimate rather than simply outputting the most likely label is to provide a human user with a better way to gauge the certainty with which the user should expect to see a certain label. However for this estimate to be meaningful to a human user, *the user needs to have a notion of the pool of instances that also received this particular estimate*. That is, if a calibrated model outputs  $f(x) = 0.7$ , the human user needs a notion of the set  $f^{-1}(0.7) = \{x \in X | f(x) = 0.7\}$ , among which this user is now promised that 70% of instances will have label 1. Note that in this case calibration does not imply that 70% of instances with this exact (or similar) feature vector  $x$  will have label 1. Thus, the mere statement  $f(x) = 0.7$  (even from a calibrated predictor) does not provide insight into the data generating process. The fourth item in the list below captures these considerations: the cells induced by a calibrated predictor should be interpretable to a human user and there shouldn’t be too many of such cells.

The fifth and last item in the list of requirements below also aims at human interpretability. If a user observes the predicted values on two input instances  $f(x)$  and  $f(x')$ , say  $f(x) = 0.57$  and  $f(x') = 0.89$ , the most meaningful insight might be that the first instance  $x$  is less likely to have label 1 than the second instance  $x'$ , (based on observing that  $f(x) < f(x')$ ). The exact values (0.57 and 0.89) may not be as easy to make sense of. However, this type of pairwise comparison is valid only if the predictor is point-wise monotonic with respect to the data-generating distribution’s regression function.

*Formal Requirements.* We let  $f : X \rightarrow \mathbb{R}$  denote a predictor and  $D$  be a distribution over  $X \times Y$ . The list below summarizes our desiderata for  $f$ :

1. **Calibration.**  $f$  is perfectly calibrated (see Definition 1):

$$\forall r \in \text{range}_D(f) : \mathbb{E}_{(x,y) \sim D}[y | f(x) = r] = r.$$

2. **Classification accuracy.** Thresholding on  $f$  yields an optimal classifier:

$$\exists \theta \in \mathbb{R} : \mathcal{L}_D^{0/1}(f_\theta) = \text{opt}_D^{0/1}.$$

3. **Approximating the regression function:**  $f$  perfectly approximates  $\eta_D$ :

$$\forall x \in \text{supp}(D) : f(x) = \eta_D(x).$$

4. **Interpretability:** The cells induced by  $f$ , that is the pre-images  $f^{-1}(r) := \{x \in X | f(x) = r\}$  for  $r \in \text{range}_D(f)$ , are meaningful to a human user. Moreover, there are relatively few induced cells. That is  $|\{f^{-1}(r) | r \in \text{range}_D(f)\}| = |\text{range}_D(f)|$  is small.

5. **Monotonicity:** Predictor  $f$  generates probability estimates that are monotonic with respect to the regression function  $\eta_D$ , that is:

$$\forall x_i, x_j \in \text{supp}(D) : (\eta_D(x_i) - \eta_D(x_j)) \cdot (f(x_i) - f(x_j)) \geq 0$$

If equality holds only when  $\eta_D(x_i) = \eta_D(x_j)$ , we call  $f$  *strictly monotonic* with respect to  $\eta_D$ .

We will start by analyzing these strictly phrased properties. In Sect. 4 below, we will introduce and investigate probabilistic relaxations of these properties.

### 3.1 Interplay of Strict Properties

We start our analysis by investigating relationships, implications and compatibilities between the above desiderata. At first glance, it might appear as if calibration is a stronger requirement than the existence of a threshold for optimally accurate classification. However, it is not difficult to see, and generally known [24], that calibration is actually a property that is independent of accuracy. A predictor can be perfectly calibrated while effectively useless for classification. And conversely a predictor can be highly accurate while not being calibrated at all.

**Observation 1** *Calibration does not imply optimal classification accuracy and optimal classification accuracy does not imply calibration.*

*Proof.* Consider a one-dimensional feature space,  $X = \mathbb{R}$ , and a distribution  $D$  that has marginal mass distributed uniformly on two points,  $D_X(-1) = D_X(+1) = 0.5$ , with a deterministic regression function  $\eta_D(x) = \mathbf{1}[x \geq 0]$ . Now the constant predictor  $f(x) = 0.5$  is perfectly calibrated, but any threshold  $\theta \in \mathbb{R}$  will result in worst possible classification loss  $\mathcal{L}_D^{0/1}(f_\theta) = 0.5$ . On the other hand, a predictor  $g$  with  $g(x) = 0.5 - \epsilon$  for  $x < 0$  and  $g(x) = 0.5 + \epsilon$  for  $x \geq 0$  for any  $\epsilon > 0$  admits a threshold (namely  $\theta = 0.5$ ) such that the resulting classifier  $g_\theta$  has perfect classification loss  $\mathcal{L}_D^{0/1}(g_\theta) = 0$  while not being calibrated.  $\square$

Of course, the regression function  $\eta_D$  is always a predictor (albeit usually an unknown one) that is both perfectly calibrated and optimally accurate (by definition, with threshold  $\theta = 0.5$ ). However, we now show that in most cases (except for distributions where the regression function is overly simple) it is not the only predictor that enjoys these two qualities. This then means that these two properties together (calibration and possibility of optimal classification accuracy) do not imply that the regression function  $\eta_D$  is well approximated.

**Theorem 2.** *There exist predictors  $f$  different from  $\eta_D$  (with positive probability) satisfying both perfect calibration and optimal classification accuracy if and only if one of the sets  $(\text{range}_D(\eta_D) \cap [0, 0.5))$  and  $(\text{range}_D(\eta_D) \cap [0.5, 1])$  has size at least 2 (that is, if and only if a Bayes optimal predictor outputs both labels and the effective range of  $\eta_D$  has size at least 3; or a Bayes optimal predictor outputs only one label and the effective range of  $\eta_D$  has size at least 2).*

*Proof.* Let's assume that at least one of the sets  $\text{range}_D(\eta_D) \cap [0, 0.5)$  and  $\text{range}_D(\eta_D) \cap [0.5, 1]$  has size at least 2. Without loss of generality we can assume that there exist  $\eta_1, \eta_2 \in \text{range}_D(\eta_D)$ , with  $\eta_1, \eta_2 < 0.5$  and  $\eta_1 \neq \eta_2$ . Let's denote regions where the regression function takes on these values by  $X_1 = \eta_D^{-1}(\eta_1) \subseteq X$ , and  $X_2 = \eta_D^{-1}(\eta_2) \subseteq X$ . By definition of the effective range, these sets have positive probability under  $D_X$ . Now consider the predictor

$$f(x) = \begin{cases} \mathbb{E}_{(x',y) \sim D}[y \mid x' \in (X_1 \cup X_2)] & \text{if } x \in (X_1 \cup X_2) \\ \eta_D(x) & \text{if } x \notin (X_1 \cup X_2). \end{cases}$$

By construction, this predictor, thresholded at 0.5 has the same classification loss as  $\eta_D$  (namely  $\text{opt}_D^{0/1}$ ) while being different from  $\eta_D$  with positive probability.

Conversely, assume that there exists a predictor  $f$  that is perfectly calibrated and achieves Bayes loss with some threshold  $\theta \in [0, 1]$ , but is not identical to  $\eta_D$  (meaning the functions differ with positive probability with respect to  $D_X$ ). Since  $\mathcal{L}_D^{0/1}(f_\theta) = \text{opt}_D^{0/1}$ , the sets  $f^{-1}([0, \theta]) \cap \text{supp}(D)$  and  $\eta_D^{-1}([0, 0.5]) \cap \text{supp}(D)$  must be identical and the sets  $f^{-1}([\theta, 1]) \cap \text{supp}(D)$  and  $\eta_D^{-1}([0.5, 1]) \cap \text{supp}(D)$  must be identical. Now if  $\eta_D$  was constant on both of these sets, then the only way for  $f$  to be calibrated would be to also take on that same constant values (and thus  $f$  would be identical to  $\eta_D$ ). Thus, if  $f$  differs from  $\eta_D$  in the support of  $D_X$  while being calibrated, then  $\eta_D$  is not constant on at least one of  $\eta_D^{-1}([0.5, 1]) \cap \text{supp}(D)$  or  $\eta_D^{-1}([0, 0.5]) \cap \text{supp}(D)$ , which implies that at least one of  $\text{range}_D(\eta_D) \cap [0, 0.5)$  and  $\text{range}_D(\eta_D) \cap [0.5, 1]$  has size at least 2.  $\square$

**Corollary 1.** *Perfect calibration and optimal classification accuracy together do not imply perfect approximation of  $\eta_D$ .*

Rather than calibration, strict monotonicity is a property that is closely related to both optimal classification accuracy and approximation of  $\eta_D$ .

**Observation 3** *Strict monotonicity implies optimal classification accuracy.*

*Proof.* Consider a predictor  $f$  and assume that  $f$  satisfies strict monotonicity with respect to  $\eta_D$ . Using the threshold 0.5 on the regression function  $\eta_D$ , we can split the set  $\text{supp}(D)$  into two disjoint subsets  $X_- := \{x \in \text{supp}(D) : \eta_D(x) < 0.5\}$  and  $X_+ = \{x \in \text{supp}(D) : \eta_D(x) \geq 0.5\}$ . Let  $f_{X_-} := \{f(x) : x \in X_-\}$  and  $f_{X_+} := \{f(x) : x \in X_+\}$  be the ranges of values that  $f$  takes on  $X_-$  and  $X_+$  respectively. For any  $x_i$  from  $X_-$  and any  $x_j$  from  $X_+$ ,  $f(x_i) < f(x_j)$  since  $\eta_D(x_i) < \eta_D(x_j)$  and  $f$  is strictly monotonic. This shows that any member of  $f_{X_-}$  is smaller than any member of  $f_{X_+}$ . Therefore,  $\inf(f_{X_+}) \geq \sup(f_{X_-})$ . Thus  $\theta = (\inf(f_{X_+}) + \sup(f_{X_-}))/2$  is a threshold on  $f$  that achieves Bayes loss.  $\square$

**Theorem 4.** *A predictor  $f$  perfectly approximates the regression function  $\eta_D$  if and only if it is perfectly calibrated and strictly monotonic with respect to  $\eta_D$ .*

*Proof.* If  $f$  perfectly approximates  $\eta_D$  (that is, they are identical with probability 1 over  $D_X$ ), then  $f$  is obviously strictly monotonic and perfectly calibrated.

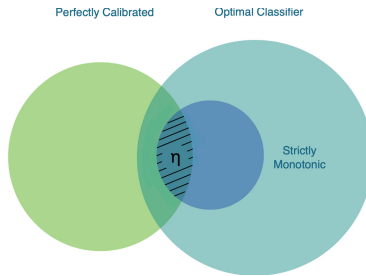
Now we will argue that a predictor that is strictly monotonic with respect to  $\eta_D$  and calibrated, also perfectly approximates  $\eta_D$ . Equivalently, we show that if  $f$  is strictly monotonic, but does not perfectly approximate  $\eta_D$ , then  $f$  is not calibrated. So let's assume  $f$  is strictly monotonic but not perfectly approximating  $\eta_D$ . Then there exists an  $x' \in \text{supp}(D)$  with  $f(x') \neq \eta_D(x')$ . Let  $S := \{x \in \text{supp}(D) : f(x) = f(x')\}$ . Since  $f$  is strictly monotonic, we have  $\eta_D(x) = \eta_D(x')$  for all  $x \in S$ . Therefore,

$$\mathbb{E}_{(x,y) \sim D} [y \mid f(x) = f(x')] = \mathbb{E}_{(x,y) \sim D} [y \mid x \in S] = \eta_D(x').$$

Since  $\eta_D(x') \neq f(x')$ ,  $\mathbb{E}_{(x,y) \sim D} [y \mid f(x) = f(x')] \neq f(x')$ , thus  $f$  is not calibrated.  $\square$

**Corollary 2.** *Neither calibration nor strict monotonicity alone implies perfect approximation of  $\eta_D$ .*

In Fig. 1 below, we illustrate the relationship between the properties.



**Fig. 1.** Interplay of calibration desiderata. The intersection of strictly monotonic and perfectly calibrated predictors only contains the regression function  $\eta_D$  (and functions that agree with  $\eta_D$  with probability 1 over  $D_X$ ).

## 4 Relaxed Desiderata for Calibration

A predictor  $f$  that is learned from finite samples is unlikely to fulfill the desiderata precisely. Thus, we now outline relaxed, probabilistic versions of our five desirable properties, or measures of how much the properties are violated. We here focus on population level measures (rather than possible estimates from finite samples, some of which we discuss in Sect. 5). It is important to note that the interpretability of the pre-images of a predictor is not a property that is quantifiable by means of a mathematical definition. Therefore, we focus in this section on quantifying the size of the effective range as an aspect of interpretability, and propose a novel, distribution based measure for this.

1. **Calibration:** Degree of calibration is measured by the  $L_p$ -norm expected calibration error [11]:

$$CE_{p,D}(f) = \left( \mathbb{E}_{(x,y) \sim D} [|f(x) - \mathbb{E}_{(x',y') \sim D} [y' \mid f(x') = f(x)]|^p] \right)^{1/p}$$

2. **Classification accuracy:** The quality of classification is measured by the standard expected classification loss for a thresholded predictor  $f_\theta$ :

$$\mathcal{L}_D^{0/1}(f_\theta) = \mathbb{E}_{(x,y) \sim D} \mathbb{1}[f_\theta(x) \neq y]$$

3. **Approximating the regression function:** To assess whether the predictor  $f$  is effectively approximating the regression function, we use the Mean Squared Error (MSE) [5, 25] :

$$MSE_D(f) = \mathbb{E}_{(x,y) \sim D} [(y - f(x))^2]$$

Note that MSE is the expectation over a *proper loss*, namely the quadratic loss  $\ell^2(f, x, y) = (f(x) - y)^2$ , and is thus minimized (over all predictors) by the regression function  $\eta_D$  [2]. It can thus be viewed as a distance from  $\eta_D$ .

4. **Interpretability:** To relax the strict measure of counting cells induced by  $f$  (ie.  $|\text{range}_D(f)|$  being small), we introduce the *Probabilistic Count (PC)*, as a novel measure that quantifies the size of a predictor’s range, taking into account the data-generating distribution. For distribution  $D$  over  $X \times Y$ , we define the *probabilistic count* of predictor  $f : X \rightarrow \mathbb{R}$  with respect to  $D$  as:

$$PC_D(f) = \frac{1}{\mathbb{P}_{x,x' \sim D_X} [f(x) = f(x')]}.$$

We show that  $PC_D(f)$  is always at most  $|\text{range}_D(f)|$ . Appendix Sect. A contains this result (Theorem 10) and further illustrations of this measure.

5. **Monotonicity:** Kendall’s  $\tau$  (tau) coefficient is a measure of the monotonicity of finite samples [22]. For any set of samples  $(x_1, y_1), \dots, (x_n, y_n)$ , any pair of samples  $(x_i, y_i)$  and  $(x_j, y_j)$ , where  $i < j$ , are discordant if  $(x_i - x_j) \cdot (y_i - y_j) < 0$ . Kendall’s Tau coefficient is defined as:

$$\tau = 1 - \frac{2 \times \text{number of discordant pairs}}{\binom{n}{2}}$$

Kendall’s  $\tau$  coefficient is in the range  $-1 \leq \tau \leq 1$ .  $\tau = 1$  represents perfect agreement between the ranking of two variables, and  $\tau = -1$  represents perfect disagreement, i.e. one ranking is the reverse of the other. This coefficient can be used to measure monotonicity, but it doesn’t consider the ties to measure strict monotonicity. We now introduce a probabilistic version of this coefficient to measure the monotonicity of two random variables, namely  $\eta_D(x)$  and  $f(x)$ . We define the *probabilistic Kendall’s Tau coefficient* for predictor  $f : X \rightarrow \mathbb{R}$  with respect to the distribution  $D$  over  $X \times Y$  as:

$$KT_D(f) = 1 - 2 \times \mathbb{P}_{x,x' \sim D_X} [(\eta_D(x) - \eta_D(x')) \cdot (f(x) - f(x')) < 0 \mid x \neq x']$$

In the remainder of this section we explore the effects of two intuitive operations that contribute to model interpretability: cell merging (with score averaging) and readjusting scores by label averaging in a cell. Through the analysis of these operations and their effects on the measures from the list above, we aim to clarify when a model can be simplified in a way that may increase its interpretability, without compromising other properties that are desirable for calibrated models. Table 1 summarizes the results of this section. The arrows indicate whether a measure can increase, decrease and remain the same through the operation.

**Table 1.** Implications of cell merging and score averaging on the measures. The arrows and equality signs represent the possible outcomes for each measure.

	$CE_{p,D}$	$\mathcal{L}_D^{0/1}$	$MSE_D$	$PC_D$	$KT_D$
Cell merging along with averaging scores	$\downarrow=$ (Thm. 6)	$\uparrow\downarrow=$ (Obs. 7)	$\uparrow\downarrow=$ (Obs. 7)	$\downarrow=^a$ (Thm. 5)	$\uparrow\downarrow=$ (Obs. 7)
Average label assigning	$\downarrow=$ (Thm. 8)	$\downarrow=$ (Thm. 8)	$\downarrow=$ (Thm. 8)	$\downarrow=$ (Thm.8)	$\uparrow\downarrow=$ (Obs. 9)

<sup>a</sup> The conclusion for  $PC_D$  holds for any new score for the cell according to Theorem 5.

### 4.1 Analysis of Cell Merging

One aspect of interpretability of a predictor is the size of its effective range. When two cells are combined in that a joint value is assigned to all points from the two cells, the size of the effective range decreases. This can thus be viewed as a simple operation which will make the predictor more amenable to human interpretation.

**Definition 2 (Cell merge with score averaging).** *Let  $D$  be a distribution over  $X \times Y$ ,  $f : X \rightarrow \mathbb{R}$  be a predictor and let  $r_1, r_2 \in \text{range}_D(f)$  be two values in the effective range of  $f$ . We say that predictor  $g : X \rightarrow \mathbb{R}$  is obtained by  $(r_1, r_2)$ -cell merge of  $f$  if  $g$  satisfies:*

$$g(x) := \begin{cases} r & \text{if } f(x) = r_1 \text{ or } f(x) = r_2 \\ f(x) & \text{otherwise,} \end{cases}$$

for some  $r \in \mathbb{R}$ . We say that  $g$  is obtained by  $(r_1, r_2)$ -cell merge of  $f$  with score averaging if  $r = \frac{r_1 \cdot \mathbb{P}_{x \sim D_X}[f(x)=r_1] + r_2 \cdot \mathbb{P}_{x \sim D_X}[f(x)=r_2]}{\mathbb{P}_{x \sim D_X}[f(x)=r_1] + \mathbb{P}_{x \sim D_X}[f(x)=r_2]}$ .

We start by showing that cell merging always decreases the probabilistic count (whether the score is averaged or an arbitrary new score is chosen).

**Theorem 5.** *Let  $D$  be a distribution over  $X \times Y$ ,  $f, g : X \rightarrow \mathbb{R}$  be predictors and  $r_1, r_2 \in \text{range}_D(f)$ . If  $g$  is obtained by  $(r_1, r_2)$ -cell merge of  $f$ , then*

$$PC_D(g) \leq PC_D(f).$$



*Proof.* To prove  $\text{PC}_D(g) \leq \text{PC}_D(f)$ , it suffices to show that

$$\begin{aligned} \frac{1}{\text{PC}_D(g)} - \frac{1}{\text{PC}_D(f)} &= \mathbb{P}_{x,x' \sim D_X} [g(x) = g(x')] - \mathbb{P}_{x,x' \sim D_X} [f(x) = f(x')] \\ &= \mathbb{E}_{x,x' \sim D_X} [\mathbb{1} [g(x) = g(x')] - \mathbb{1} [f(x) = f(x')]] \geq 0 \end{aligned}$$

For any  $x, x' \in \text{supp}(D)$ ,  $\mathbb{1} [g(x) = g(x')]$  and  $\mathbb{1} [f(x) = f(x')]$  have the same value (either both are 1 or 0), except when both  $x$  and  $x'$  belong to cells where  $f$  assigns values  $r_1, r_2$  or  $r$ , since these are the only different cells between  $f$  and  $g$ . For all  $x$  and  $x'$  in these cells,  $g(x) = g(x') = r$ , thus  $\mathbb{1} [g(x) = g(x')] = 1$ . Therefore we have  $\mathbb{1} [g(x) = g(x')] - \mathbb{1} [f(x) = f(x')] \geq 0$  for all  $x, x' \in \text{supp}(D)$ .  $\square$

We next investigate the effect of cell merging on the  $L_p$  norm calibration error  $\text{CE}_{p,D}$ . We show  $\text{CE}_{p,D}$  can only decrease if cells are merged with score averaging.

**Theorem 6.** *Let  $D$  be a distribution over  $X \times Y$ ,  $f, g : X \rightarrow \mathbb{R}$  be predictors and  $r_1, r_2 \in \text{range}_D(f)$ . If  $g$  is obtained by  $(r_1, r_2)$ -cell merge of  $f$  with score averaging, then we have*

$$\text{CE}_{p,D}(g) \leq \text{CE}_{p,D}(f).$$

*Proof.* First note that for any predictor  $h : X \rightarrow \mathbb{R}$  and all  $x \in X$ , we have

$$\begin{aligned} |h(x) - \mathbb{E}_{(x',y') \sim D} [y' \mid h(x') = h(x)]| &= |h(x) - \mathbb{E}_{x' \sim D_X} [\eta_D(x') \mid h(x') = h(x)]| \\ &= |\mathbb{E}_{x' \sim D_X} [h(x) - \eta_D(x') \mid h(x') = h(x)]| \\ &= |\mathbb{E}_{x' \sim D_X} [h(x') - \eta_D(x') \mid h(x') = h(x)]| \end{aligned}$$

where the last inequality holds since the expectation is conditioned on any  $x'$  such that  $h(x') = h(x)$ . Thus we get

$$\text{CE}_{p,D}(h) = \left( \mathbb{E}_{x \sim D_X} [|\mathbb{E}_{x' \sim D_X} [h(x') - \eta_D(x') \mid h(x') = h(x)]|^p] \right)^{1/p} \quad (1)$$

for any predictor  $h$ . Now note that if  $r_1 = r_2$  then  $f = g$ , and  $\text{CE}_{p,D}(f) = \text{CE}_{p,D}(g)$ . Otherwise, according to Eq. 1 above applied to  $f$  and  $g$ :

$$\begin{aligned} \text{CE}_{p,D}(f)^p - \text{CE}_{p,D}(g)^p &= \mathbb{E}_{x \sim D_X} [|\mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = f(x)]|^p] \\ &\quad - \mathbb{E}_{x \sim D_X} [|\mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = g(x)]|^p] \\ &= \mathbb{E}_{x \sim D_X} [|\mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = f(x)]|^p] \\ &\quad - |\mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = g(x)]|^p \end{aligned}$$

The cells that  $f$  and  $g$  have different expectations in the previous term are the ones with  $f(x)$  equals to  $r_1$  or  $r_2$  or  $r$ . All members in these three cells are in the same cell in the range of  $g$  with  $g(x) = r$ . So,

$$\begin{aligned} \text{CE}_{p,D}(f)^p - \text{CE}_{p,D}(g)^p &= |\mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = r_1]|^p \cdot \mathbb{P}_{x \sim D_X} [f(x) = r_1] \\ &\quad + |\mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = r_2]|^p \cdot \mathbb{P}_{x \sim D_X} [f(x) = r_2] \\ &\quad + |\mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = r]|^p \cdot \mathbb{P}_{x \sim D_X} [f(x) = r] \\ &\quad - |\mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = r]|^p \cdot \mathbb{P}_{x \sim D_X} [g(x) = r] \end{aligned}$$

For the rest of the proof, we use the following notations:

$$\begin{aligned}
 e_1 &:= \mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = r_1] \\
 e_2 &:= \mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = r_2] \\
 e_3 &:= \mathbb{E}_{x' \sim D_X} [f(x') - \eta_D(x') \mid f(x') = r] \\
 e' &:= \mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = r] \\
 w_1 &:= \mathbb{P}_{x \sim D_X} [f(x) = r_1] \\
 w_2 &:= \mathbb{P}_{x \sim D_X} [f(x) = r_2] \\
 w_3 &:= \mathbb{P}_{x \sim D_X} [f(x) = r] \\
 w' &:= \mathbb{P}_{x \sim D_X} [g(x) = r] = w_1 + w_2 + w_3
 \end{aligned}$$

The latter equality holds since  $g(x) = r$  if and only if  $f(x) \in \{r_1, r_2, r\}$ . Now:

$$\text{CE}_{p,D}(f)^p - \text{CE}_{p,D}(g)^p = |e_1|^p \cdot w_1 + |e_2|^p \cdot w_2 + |e_3|^p \cdot w_3 - |e'|^p \cdot (w_1 + w_2 + w_3) \tag{2}$$

Now we use the law of total expectation to rewrite  $e'$  as  $e_1$ ,  $e_2$ , and  $e_3$ :

$$\begin{aligned}
 e' &= [\mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = r, f(x') = r_1] \cdot w_1 \\
 &\quad + \mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = r, f(x') = r_2] \cdot w_2 \\
 &\quad + \mathbb{E}_{x' \sim D_X} [g(x') - \eta_D(x') \mid g(x') = r, f(x') = r] \cdot w_3] / (w_1 + w_2 + w_3) \\
 &= [(r - \mathbb{E}_{x' \sim D_X} [\eta_D(x') \mid f(x') = r_1]) \cdot w_1 \\
 &\quad + (r - \mathbb{E}_{x' \sim D_X} [\eta_D(x') \mid f(x') = r_2]) \cdot w_2 \\
 &\quad + (r - \mathbb{E}_{x' \sim D_X} [\eta_D(x') \mid f(x') = r]) \cdot w_3] / (w_1 + w_2 + w_3) \\
 &= [(r + e_1 - r_1) \cdot w_1 + (r + e_2 - r_2) \cdot w_2 + (e_3) \cdot w_3] / (w_1 + w_2 + w_3) \\
 &= [e_1 \cdot w_1 + e_2 \cdot w_2 + e_3 \cdot w_3] / (w_1 + w_2 + w_3)
 \end{aligned}$$

$$\begin{aligned}
 \Rightarrow |e'|^p &= \left| \frac{e_1 \cdot w_1 + e_2 \cdot w_2 + e_3 \cdot w_3}{w_1 + w_2 + w_3} \right|^p \\
 &\leq \left[ \frac{|e_1| \cdot w_1 + |e_2| \cdot w_2 + |e_3| \cdot w_3}{w_1 + w_2 + w_3} \right]^p = [|e_1| \cdot w'_1 + |e_2| \cdot w'_2 + |e_3| \cdot w'_3]^p,
 \end{aligned}$$

in which  $w'_i = w_i / (w_1 + w_2 + w_3)$ . So,  $w'_1 + w'_2 + w'_3 = 1$ . Function  $|\cdot|^p$  is a convex function for any  $p \in \mathbb{N}$ . Therefore, according to Jensen's inequality [14]:

$$\begin{aligned}
 &[|e_1| \cdot w'_1 + |e_2| \cdot w'_2 + |e_3| \cdot w'_3]^p \leq |e_1|^p \cdot w'_1 + |e_2|^p \cdot w'_2 + |e_3|^p \cdot w'_3 \\
 \Rightarrow &|e'|^p \leq \frac{|e_1|^p \cdot w_1 + |e_2|^p \cdot w_2 + |e_3|^p \cdot w_3}{w_1 + w_2 + w_3} \\
 \Rightarrow &\text{CE}_{p,D}(f)^p \geq \text{CE}_{p,D}(g)^p \quad (\text{using Equation 2}) \\
 \Rightarrow &\text{CE}_{p,D}(f) \geq \text{CE}_{p,D}(g).
 \end{aligned}$$

□

The following two observations demonstrate the impact of merging cells on classification loss, mean squared error and the probabilistic Kendall’s Tau.

**Observation 7** *If  $g$  is obtained by  $(r_1, r_2)$ -cell merge of  $f$  with score averaging (under the conditions of Definition 2), then  $\mathcal{L}_D^{0/1}(g)$  may be smaller, larger or equal to  $\mathcal{L}_D^{0/1}(f)$ , and the same holds for  $\text{MSE}_D(g)$  and  $\text{KT}_D(g)$  in comparison with  $\text{MSE}_D(f)$  and  $\text{KT}_D(f)$  respectively.*

*Proof. Binary Loss:* Let  $f$  be a predictor and let  $a$  and  $b$  be two cells generated by  $f$  with  $f(x) = 0.4$  for all  $x \in a$  and  $f(x) = 0.8$  for all  $x \in b$ . Let  $D$  be a distribution whose marginal assigns the same probability to these cells  $D_X(a) = D_X(b) = 0.1$ . Further, let’s assume that there are two additional, heavier cells  $c$  and  $d$ , with  $D_X(c) = D_X(d) = 0.4$ , and  $f(x) = 0.45$  while  $\eta_D(x) = 0$  for all  $x \in c$  and  $f(x) = 0.55$  while  $\eta_D(x) = 1$  for all  $x \in d$ . Thus, independently of the regression function’s values in the lighter cells  $a$  and  $b$ , the best classification threshold for  $f$  will be any  $\theta \in (0.45, 0.55)$ , say  $\theta = 0.5$ .

Let  $g$  be obtained from  $f$  by a  $(0.4, 0.8)$ -cell merge. Then  $g(x) = 0.6$  for all  $x \in a \cup b$ . Since  $g(x) = f(x)$  for all  $x \in c \cup d$  (the heavier cells),  $g$  will also be optimally thresholded with any  $\theta \in (0.45, 0.55)$ , thus with optimal threshold, say  $\theta = 0.5$ , for both  $g$  and  $f$  we get  $g_\theta(x) = 1 \neq 0 = f_\theta(x)$  for all  $x \in a$ . With slight abuse of notation, we let  $\eta(a) = \mathbb{P}_{(x,y) \sim D}[y = 1 \mid x \in a]$  denote the probability of label 1 generated conditioned on cell  $a$ . If  $\eta(a) < 0.5$ , then  $\mathcal{L}_D^{0/1}(g) > \mathcal{L}_D^{0/1}(f)$ , if  $\eta(a) > 0.5$ , then  $\mathcal{L}_D^{0/1}(g) < \mathcal{L}_D^{0/1}(f)$  and if  $\eta(a) = 0.5$ , then  $\mathcal{L}_D^{0/1}(g) = \mathcal{L}_D^{0/1}(f)$ .

*Kendall’s Tau Coefficient:* Let’s consider the same scenario as above, but now with the four cells  $a, b, c$  and  $d$  having equal probability weight, say  $D_X(a) = D_X(b) = D_X(c) = D_X(d) = 0.25$ . As above, we denote the conditional label probabilities in these cells by  $\eta_D(a)$ ,  $\eta_D(b)$ ,  $\eta_D(c)$ , and  $\eta_D(d)$  respectively. If  $\eta_D(a) < \eta_D(c) < \eta_D(d) < \eta_D(b)$ , then the scores assigned by  $f$  are monotonic with respect to  $\eta_D$ , while the scores of  $g$  are not. We thus get  $1 = \text{KT}_D(f) > \text{KT}_D(g)$ . In case  $\eta_D(c) < \eta_D(b) < \eta_D(a) < \eta_D(d)$ , the scores of  $g$  are monotonic, but the scores of  $f$  are not. Thus  $1 = \text{KT}_D(g) > \text{KT}_D(f)$ . Finally, if  $\eta_D$  is a constant function, then the cell merge does not change the Kendall’s Tau.

*Mean Squared Error:* To show that the same phenomena can occur for the MSE, let’s consider a scenario where the predictor  $f$  assigns value 0 to all points in a cell  $a$  and value 1 to all points in a cell  $b$ , and let’s assume  $D_X(a) = D_X(b)$ . Upon merging them, their combined score becomes 0.5. When  $\eta(a) = 0$  and  $\eta(b) = 1$ , the MSE, conditioned on these cells, increases from 0 to 0.25. Conversely,  $\eta(a) = 1$  and  $\eta(b) = 0$ , the mean squared error decreases from 1 to 0.25. If  $\eta(a) = 0.25$  and  $\eta(b) = 0.75$ , then the mean squared error remains the same at 0.25. □

## 4.2 Analysis of Average Label Assignment

We now analyze another operation, where the score for every cell of predictor  $f$  is replaced with the true label average in that cell. We say  $\bar{f}_D : X \rightarrow [0, 1]$  with

$$\bar{f}_D(x) := \mathbb{E}_{(x',y') \sim D}[y' \mid f(x) = f(x')].$$

is obtained by average label assignment with respect to distribution  $D$  from  $f$ . This true average label with respect to the data-generating distribution is typically not available to user, however might be (approximately) estimated from samples.

**Theorem 8.** For distribution  $D$  over  $X \times Y$  and any predictor  $f : X \rightarrow \mathbb{R}$  the predictor  $\bar{f}_D(x)$  obtained by average label assignment with respect to distribution  $D$  from  $f$  satisfies the following for  $\theta = 0.5$ :

$$\begin{aligned} \text{CE}_{p,D}(\bar{f}_D) &= 0, & \text{MSE}_D(\bar{f}_D) &\leq \text{MSE}_D(f), \\ \mathcal{L}_D^{0/1}((\bar{f}_D)_\theta) &\leq \mathcal{L}_D^{0/1}(f_\theta), & \text{PC}_D(\bar{f}_D) &\leq \text{PC}_D(f). \end{aligned}$$

*Proof.* First note that  $\text{CE}_{p,D}(\bar{f}_D) = 0$  is immediate from the definition. Now let  $\text{range}_D(f) = \{s_1^f, s_2^f, \dots, s_n^f\}$  and let  $\{b_1^f, b_2^f, \dots, b_n^f\}$  be the corresponding cells generated by  $f$ . So, for any predictor  $f : X \rightarrow \mathbb{R}$ ,

$$\begin{aligned} \text{MSE}_D(f) &= \mathbb{E}_{(x,y) \sim D}[(y - f(x))^2] \\ &= \sum_{i \in [1,n]} \mathbb{E}_{(x,y) \sim D}[(y - f(x))^2 \mid x \in b_i^f] \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \\ &= \sum_{i \in [1,n]} \mathbb{E}_{(x,y) \sim D}[(y - s_i^f)^2 \mid x \in b_i^f] \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \\ &= \sum_{i \in [1,n]} (\mathbb{E}_{(x,y) \sim D}[y^2 \mid x \in b_i^f] - 2s_i^f \mathbb{E}_{(x,y) \sim D}[y \mid x \in b_i^f] + (s_i^f)^2) \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \end{aligned}$$

Using  $\bar{y}_i^f$  as  $\mathbb{E}_{(x,y) \sim D}[y \mid x \in b_i^f]$  for any  $i \in [1, n]$  and the identity  $\mathbb{E}[X^2] = \text{Var}(X) + (\mathbb{E}[X])^2$ , we rewrite the expression as:

$$\begin{aligned} \text{MSE}_D(f) &= \sum_{i \in [1,n]} (\text{Var}_{(x,y) \sim D}[y \mid x \in b_i^f] + (\bar{y}_i^f)^2 - 2s_i^f \bar{y}_i^f + (s_i^f)^2) \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \\ &= \sum_{i \in [1,n]} (\text{Var}_{(x,y) \sim D}[y \mid x \in b_i^f] + (\bar{y}_i^f - s_i^f)^2) \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \end{aligned}$$

The values of  $f$  and  $\bar{f}_D$  are different, while the cells of  $f$  are a refinement of those of  $\bar{f}_D$ , i.e., any two elements from  $\text{supp}(D)$  from the same cell of  $f$  are also in the same cell of  $\bar{f}_D$ . So,  $\forall i \in [1, n], b_i^{\bar{f}_D} = b_i^f, \bar{y}_i^{\bar{f}_D} = \bar{y}_i^f$ , and  $\text{Var}_{(x,y) \sim D}[y \mid x \in b_i^{\bar{f}_D}] = \text{Var}_{(x,y) \sim D}[y \mid x \in b_i^f]$ . Also, according to the definition of  $\bar{f}_D$ ,  $\forall i \in [1, n], s_i^{\bar{f}_D} = \bar{y}_i^{\bar{f}_D}$ . So we complete the proof for the MSE by:

$$\begin{aligned} \text{MSE}_D(f) - \text{MSE}_D(\bar{f}_D) &= \sum_{i \in [1,n]} ((\bar{y}_i^f - s_i^f)^2 - (\bar{y}_i^{\bar{f}_D} - s_i^{\bar{f}_D})^2) \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \\ &= \sum_{i \in [1,n]} (\bar{y}_i^f - s_i^f)^2 \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \geq 0 \end{aligned}$$

Now we rewrite the classification loss of predictor  $f$  using the  $n$  cells:

$$\begin{aligned} \mathcal{L}_D^{0/1}(f_\theta) &= \mathbb{E}_{(x,y)\sim D} \mathbb{1}[f_\theta(x) \neq y] \\ &= \sum_{i \in [1,n]} \mathbb{E}_{(x,y)\sim D} [\mathbb{1}[f_\theta(x) \neq y] \mid x \in b_i^f] \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f] \\ &= \sum_{i \in [1,n]} \mathbb{E}_{(x,y)\sim D} [\mathbb{1}[\mathbb{1}[f(x) \geq \theta] \neq y] \mid x \in b_i^f] \cdot \mathbb{P}_{x \sim D_X}[x \in b_i^f]. \end{aligned}$$

Let's consider the expectation part of this expression for one arbitrary cell:

$$\begin{aligned} &\mathbb{E}_{(x,y)\sim D} [\mathbb{1}[\mathbb{1}[f(x) \geq \theta] \neq y] \mid x \in b_i^f] \\ &= \mathbb{E}_{x \sim D_X} [\mathbb{1}[f(x) < \theta] \cdot \eta_D(x) + \mathbb{1}[f(x) \geq \theta] \cdot (1 - \eta_D(x)) \mid x \in b_i^f] \end{aligned} \quad (3)$$

The classification on cell changes if and only if the label assigned to that cell (with  $\theta = 0.5$ ) changes. The label on the whole cell is constant. Without loss of generality suppose cell  $b_i^f$  is labelled 0 under  $f$  and labelled 1 under  $\bar{f}_D$ . Consequently,  $s_i^f < 0.5$  and  $\mathbb{E}_{(x,y)\sim D}[y \mid x \in b_i^f] \geq 0.5$ , which means  $\mathbb{E}_{x \sim D_X}[\eta_D(x) \mid x \in b_i^f] \geq 0.5$ . Now we rewrite Eq. 3 for both predictors  $f$  and  $\bar{f}_D$ :

$$\begin{aligned} \mathbb{E}_{(x,y)\sim D} [\mathbb{1}[\mathbb{1}[f(x) \geq \theta] \neq y] \mid x \in b_i^f] &= \mathbb{E}_{x \sim D_X} [\eta_D(x) \mid x \in b_i^f] \\ \mathbb{E}_{(x,y)\sim D} [\mathbb{1}[\mathbb{1}[\bar{f}_D(x) \geq \theta] \neq y] \mid x \in b_i^f] &= \mathbb{E}_{x \sim D_X} [1 - \eta_D(x) \mid x \in b_i^f] \end{aligned}$$

Since  $\mathbb{E}_{x \sim D_X} [\eta_D(x) \mid x \in b_i^f] \geq 0.5$ :

$$\begin{aligned} \mathbb{E}_{x \sim D_X} [1 - \eta_D(x) \mid x \in b_i^f] &\leq \mathbb{E}_{x \sim D_X} [\eta_D(x) \mid x \in b_i^f] \implies \\ \mathbb{E}_{(x,y)\sim D} [\mathbb{1}[\mathbb{1}[f(x) \geq \theta] \neq y] \mid x \in b_i^f] &\geq \mathbb{E}_{(x,y)\sim D} [\mathbb{1}[\mathbb{1}[\bar{f}_D(x) \geq \theta] \neq y] \mid x \in b_i^f]. \end{aligned}$$

Thus the classification loss on any arbitrary cell for predictor  $f$  is greater than or equal to the loss for  $\bar{f}_D(x)$ , completing our proof that  $\mathcal{L}_D^{0/1}((\bar{f})_\theta) \leq \mathcal{L}_D^{0/1}(f_\theta)$ .

When the initial scores are replaced with the average of true labels, it is possible for two cells of  $f$  to have equal new scores. In such cases, these cells are merged, and by Theorem 5, it follows that  $\text{PC}_D(\bar{f}_D) \leq \text{PC}_D(f)$ .  $\square$

Next we present an analogous result for monotonicity.

**Observation 9** *Let  $\bar{f}_D(x)$  be the predictor obtained by average label assignment with respect to some distribution  $D$  from some predictor  $f$ . Then the probabilistic Kendall's Tau coefficient of  $\bar{f}_D(x)$  may be smaller, larger or equal to the Kendall's Tau coefficient of  $f$  with respect to  $D$ .*

*Proof.* Let's consider a predictor  $f$  that generates only two distinct values  $r_a$  and  $r_b$ , say  $r_a < r_b$  and let  $a$  and  $b$  be the corresponding cells. As before, we denote the expected label in these cells by  $\eta_D(a)$  and  $\eta_D(b)$ .

We first consider the case that both cells  $a$  and  $b$  contains four distinct domain points, and the values of  $\eta_D$  on the four points in cell  $a$  are three times 0.1, and once 1, and 0.2 for all four points in  $b$ . Thus  $\eta_D(a) = 0.325$  and  $\eta_D(b) = 0.2$ , and

$$\text{KT}_D(f) = 1 - 2 \cdot \frac{8}{56} = \frac{5}{7}, \quad \text{KT}_D(\bar{f}_D) = 1 - 2 \cdot \frac{24}{56} = \frac{1}{7}.$$

Thus, in this case, substituting the scores with the average true labels has weakened the monotonicity of the predictor.

Now consider the same scenario with the only difference being that the values of  $\eta_D$  for the points in region  $a$  are now 0.1, twice 0.3, and 1. In this case the monotonicity, as measured by the probabilistic Kendall's Tau, has improved:

$$\text{KT}_D(f) = 1 - 2 \cdot \frac{24}{56} = \frac{1}{7}, \quad \text{KT}_D(\bar{f}_D) = 1 - 2 \cdot \frac{8}{56} = \frac{5}{7}.$$

Lastly, if the regression function is a constant function, then  $\text{KT}_D(f) = \text{KT}_D(\bar{f}_D)$ .  $\square$

## 5 Experimental Evaluation of Decision Tree Based Models

In our experimental evaluation we compare standard methods for calibration to a simple model that is inherently interpretable, namely a Decision Tree (DT). Note that for a decision tree, the induced cells are inherently interpretable, and it is also straight-forward to control the number of cells, thus this methods straightforwardly satisfies our basic requirements for interpretability. Our goal is then to determine how this simple interpretable method compares to non-interpretable standard methods, in terms of our remaining desiderata, namely calibration, classification accuracy, approximation of the regression function and monotonicity. We include two standard methods for calibration through post-processing, namely Platt Scaling (PS) [20] and Isotonic Regression [27] in our comparison. A support vector machine (SVM) is first trained as the base model for Platt Scaling and Isotonic Regression. Additionally we compare to another tree based calibration model, Probability Calibration Tree (PCT) [17].

**Evaluation Metrics Employed.** Since some of our desiderata, namely calibration, approximation of the regression function and monotonicity directly depend on the unknown values of the regression function  $\eta_D$ , there is no immediate way to assess these from finite data. We employ commonly used metrics that we list below, as well as a novel metric that we introduce. For classification accuracy we evaluate the empirical binary loss [4]; for calibration we evaluate the (empirical) Expected calibration error (ECE) [18]; for approximation of the regression function we evaluate the Root Mean Square Error (RMSE) [13], and for monotonicity we evaluate the Area under the ROC curve (AUC) [3]. Another metric for calibration that we evaluate is the Area under the Validity Curve (AUC<sub>V</sub>)

[11]. And in addition to these metrics from the literature, we introduce a novel calibration metric that we term Probability Deviation Error (PDE).

For the definitions of these metrics below, we let  $x_i \in X$  and  $y_i \in \{0, 1\}$  denote the features and label of a single sample.  $D^{(n)}$  denotes the collection of  $n$  samples:

$$D^{(n)} := ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$$

*Classification (0/1)-Loss.* In our experiments, we utilize the threshold  $\theta = 0.5$  for

$$\mathcal{L}_n^{0/1}(f_\theta) = \frac{1}{n} \sum_{i=1}^n \begin{cases} (y_i)^2 & \text{if } f(x_i) \leq \theta \\ (1 - y_i)^2 & \text{otherwise.} \end{cases}$$

*Root Mean Square Error (RMSE)* [13]

$$\mathcal{L}_n^{\text{RMSE}}(f) = \left( \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2 \right)^{1/2}$$

*Area Under the ROC Curve (AUC)* [3] Given a predictor  $f$ , using different thresholds  $\theta \in [0, 1]$ , we obtain classifiers  $f_\theta$  with increasing True Positive and False Positive Rates (TPR and FPR) over the sample points. These pairs of rates yield curve (where TPR is viewed as a function of FPR), and AUC is defined as the area under this curve. This is standard metric to evaluate the monotonicity of a predictor with respect to the regression function. If the model generates wrong scores (in terms of pointwise probability estimates), but in the correct order (that  $f$  is monotonic with respect to  $\eta_D$ ), then AUC is still high.

*Expected Calibration Error (ECE).* This criterion compares the average predicted scores and the average of true labels with respect to a given set of bins  $b_1, b_2, \dots, b_B$ , where the bins form a partition of the space or dataset [8, 12, 18]. We let  $w_i$  denote the fraction of data points contained in bin  $b_i$ . ECE is then defined as follows:

$$\text{ECE}_p := \left( \sum_{i=1}^B w_i \cdot \text{PCE}(b_i)^p \right)^{1/p}.$$

where PCE is the *Partition Calibration Error*, which is the difference between the average values generated by  $f$  and the average of labels in a bin:

$$\text{PCE}(b_i) := \frac{|\sum_{j=1}^n (f(x_j) - y_j) \mathbb{1}[x_j \in b_i]|}{\sum_{j=1}^n \mathbb{1}[x_j \in b_i]} = \left| \frac{1}{|b_i|} \sum_{x_j \in b_i} f(x_j) - \frac{1}{|b_i|} \sum_{x_j \in b_i} y_j \right|$$

When no binning is provided, uniform mass binning is employed, that is the produced scores are sorted and allocated into a fixed number  $B$  of equally weighted bins. The resulting criterion is denoted by  $\text{ECE}_{B,p} := \left( \frac{1}{B} \sum_{i=1}^B \text{PCE}(b_i)^p \right)^{1/p}$ . While ECE is widely used to evaluate calibration models [1, 8, 10, 19], it can be

a problematic measure when the bins used do not correspond to the actual cells of the predictor  $f$ . ECE then effectively evaluates a different predictor, namely the predictor that results from  $f$  when the scores are averaged in each given bin. We discuss how this can result in distorted conclusion in Appendix Sect. B.

*Probability Deviation Error (PDE).* To address the issues of ECE (see appendix Sect. B), we propose a new metric which we term Probability Deviation Error (PDE). The PDE compares the point-wise scores with the average label in each bin, thereby fixing the problem associated with the ECE. On predefined bins  $b_1, b_2, \dots, b_B$  with weights  $w_1, w_2, \dots, w_B$ , the  $L_p$  norm PDE is defined as follows:

$$\text{PDE}_p := \left( \sum_{i=1}^B w_i \cdot \text{PPD}(b_i)^p \right)^{1/p}.$$

where PPD, or *Partition Probability Deviation* is the average difference between point-wise scores generated by  $f$  in a bin and the label average in the bin:

$$\text{PPD}(b_i) := \frac{\sum_{j=1}^n |f(x_j) - \hat{y}_i| \mathbb{1}[x_j \in b_i]}{\sum_{j=1}^n \mathbb{1}[x_j \in b_i]} = \frac{1}{|b_i|} \sum_{x_j \in b_i} |f(x_j) - \hat{y}_i|$$

where  $\hat{y}_i = \frac{1}{|b_i|} \sum_{x_k \in b_i} y_k$  is the average label in bin  $b_i$ . If no partition into bins is given, as for ECE, uniform mass binning with  $B$  bins is used by default. In this case, we denote the criterion as  $\text{PDE}_{B,p} := \left( \frac{1}{B} \sum_{i=1}^B \text{PPD}(b_i)^p \right)^{1/p}$ . We illustrate that this metric better reflects quality of calibration than ECE, by empirically comparing these two measures on synthetically generated data (see Appendix Sect. C).

*Area Under the Validity Curve (AUC<sub>V</sub>)* This metric has recently been proposed to evaluate calibration [11]. We first define the *validity function*  $V : \mathbb{R}^X \times [0, 1] \rightarrow [0, 1]$  that assigns to each threshold  $\epsilon \in [0, 1]$  the probability mass of the area where predictor  $f$  is  $\epsilon$ -valid as measured by the  $L_1$  norm calibration error:

$$V(f, \epsilon) = \mathbb{P}_{(x,y) \sim D} [ |f(x) - \mathbb{E}_{(x',y') \sim D} [y' \mid f(x') = f(x)]| \leq \epsilon ]$$

This function generates a curve, the *validity curve*, whose integral over  $[0, 1]$  is the metric  $\text{AUC}_V(f)$  [11]. Its relation to the  $L_1$  norm calibration error for  $f : X \rightarrow [0, 1]$  has been shown to satisfy  $\text{CE}_1(f) = 1 - \text{AUC}_V(f)$  [11].

With finite data  $D^{(n)}$ , the validity function  $V$  is estimated as follows [11]:

$$\hat{V}(f, \epsilon) = \frac{1}{n} \sum_{i=1}^n \mathbb{1} \left[ |f(x_i) - \hat{\mathbb{E}}_{(x,y) \sim D^{(n)}} [y \mid f(x) = f(x_i)]| \leq \epsilon \right]$$

where  $\hat{\mathbb{E}}_{(x,y) \sim D^{(n)}} [y \mid f(x) = p] := \frac{\sum_{i=1}^n y_i \mathbb{1}[f(x_i)=p]}{\sum_{i=1}^n \mathbb{1}[f(x_i)=p]}$ . This latter empirical expectation counts the number of samples with the same score. But on a finite dataset no two (or very few) points may have the same score. The measure thus requires



an appropriate binning method. Prior work [11] involved averaging scores over the uniform mass bins as in ECE, effectively evaluating a different predictor.

We used a different method to estimate the validity function with finite number of samples, based on the K-nearest neighbors (KNN). K-nearest neighbor based  $AUC_V$  estimation is the area under the following estimated validity curve:

$$\hat{V}_{\text{KNN}}(f, \epsilon) = \frac{1}{n} \sum_{i=1}^n \mathbb{1} \left[ |f(x_i) - \hat{\mathbb{E}}_{\text{KNN}(x,y) \sim D^{(n)}} [y | f(x) = f(x_i)]| \leq \epsilon \right],$$

where the empirical expectation estimation is

$$\hat{\mathbb{E}}_{\text{KNN}(x,y) \sim D^{(n)}} [y | f(x) = p] := \frac{\sum_{i=1}^n y_i \mathbb{1} [f(x_i) \in \text{k-nn}(p)]}{k}$$

with  $\text{k-nn}(p)$  being the set of  $k$  samples with the closest  $f$ -scores to  $p$ .

Our method takes into account the scores that predictor  $f$  assigns to each datapoint, instead of relying on the average scores over different bins. We thus avoid evaluating a modified version of the model. We denote the KNN based  $AUC_V$  estimation by  $AUC_{V,\text{KNN}}$  and employed this in our experiments.

**Datasets.** We used 36 datasets for binary and multi-class classification tasks. All 36 datasets are from UCI [7] and their properties are summarized in Table 2.

**Table 2.** Real world datasets used in our experiments.

Dataset	Instances	Attributes	Classes	Dataset	Instances	Attributes	Classes
audiology	226	69	24	mice-protein	1080	80	8
bank-marketing	41188	19	2	new-thyroid	215	5	3
bankruptcy	10503	64	2	news-popularity	39644	59	2
car-evaluation	1728	6	4	nursery	12960	8	5
cervical-cancer	858	32	2	optdigits	5620	64	10
colposcopy	287	62	2	page-blocks	5473	10	5
credit-rating	690	15	2	pendigits	10992	16	10
cylinder-bands	512	39	2	phishing	1353	10	3
german-credit	1000	20	2	pima-diabetes	768	8	2
hand-postures	78095	39	2	segment	2310	20	7
htru2	17898	8	2	shuttle	58000	9	7
iris	150	4	3	sick	3772	29	2
kr-vs-kp	3196	36	2	spambase	4601	57	2
mfeat-factors	2000	216	10	taiwan-credit	30000	23	2
mfeat-fourier	2000	76	10	tic-tac-toe	958	9	2
mfeat-karhunen	2000	64	10	vote	435	16	2
mfeat-morph	2000	6	10	vowel	990	14	10
mfeat-pixel	2000	240	10	yeast	1484	8	10

**Results.** We evaluate the listed metrics of the four calibration methods on all real 36 world datasets. For each dataset and method, we average over 10 repetitions, and each time randomly partitioning the samples into training, calibration, and test sets (40.5%, 49.5% and 10% respectively). An SVM with Gaussian kernel is the base model for PS and IR post-processing calibration methods. Training of the tree-based calibration models (PCT and DT) involved a cost-complexity post-pruning step. The optimal cost-complexity parameter is found via 5-fold cross-validation on the calibration set. Afterwards, the whole calibration set is used to train the calibration model and the model is pruned using the found optimal factor. To evaluate the models with L1-norm-ECE, we use uniform-mass binning with 32 bins. For L1-norm-PDE, we used the leaves of the tree for models PCT and DT. For DT this corresponds to the cells generated by the predictor. There are no meaningful cells for PS and IR, thus no PDE is reported.

**Table 3.** The calibration methods are compared using the above metrics. We report the number of times each method is the top performer. The numbers in each column do not add up to the number of datasets as multiple models may have won simultaneously. The pre-fixed bins for metric PDE are the leaves generated by PCT and DT. We use  $k = 10$  for  $AUC_{V,KNN}$ .

Method	RMSE	0/1-loss	AUC	$ECE_{B=32,p=1}$	$PDE_{p=1}$	$AUC_{V,KNN}$
PS	21	22	27	7	–	5
IR	25	24	28	14	–	9
PCT	<b>31</b>	<b>31</b>	<b>32</b>	<b>26</b>	19	<b>15</b>
<b>DT</b>	24	24	19	21	<b>36</b>	7

Table 3 summarizes the results over the 36 datasets. For each of the four methods and six metrics, we report how often the method obtained the best score. We count cases of ties towards all winning methods (which is why the columns in that table don’t always sum up to 36). Note that the simple decision tree (DT) is the only predictor evaluated here that can be considered interpretable. The other three methods produce infinitely many different scores (their effective range is infinite), and a user can not reasonably be expected to have a notion of the shapes of the resulting cells. The summary shows that in our experiments the simple decision tree is a predictor that performs similarly well as PS and IR on all metrics and performs best among all four methods in terms of PDE. The PCT method outperforms DT on most metrics. However we would argue that the overall performance of DT is a worth-while trade-off for interpretability.

## 6 Concluding Discussion

The goal of this work is to provide a systematic framework for understanding different aspects of calibration and to highlight the importance of taking interpretability into account when promoting calibration. Calibration is a

notion that is inherently aimed at providing users with better understanding of label certainty. Our axiomatic framing and analysis highlight which aspects of a calibrated predictor can improve human comprehension of the provided scores (namely interpretable cells, not too large number of cells and monotonicity with respect to the regression function of the data-generating process), and show how some aspects fulfill other important purposes (accuracy, and point-wise approximation of the regression function). The three levels of analysis (axioms/properties, distributional measures of distance from these and empirical measures) further clarify the higher level concepts that frequently cited empirical measures are aimed at.

Providing confidence scores to end users without a way of clearly communicating the meaning and range of validity of these scores might pose more risks in terms of effects of downstream decisions than not providing any confidence scores at all (for example when a high confidence score instills a false sense of certainty). We hope that our work contributes to and will inspire more investigations into interpretability for calibrated scores.

**Acknowledgments.** This research was funded by an NSERC discovery grant.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## A Exploring the Probabilistic Count (PC)

In this section, we exhibit characteristics of our newly introduced measure, the probabilistic count (PC). We first establish a connection between PC and the true size of the effective range. Subsequently, we provide some illustrative examples.

**Theorem 10.** *For distribution  $D$  over  $X \times Y$  and predictor  $f : X \rightarrow \mathbb{R}$ , we have  $\text{PC}_D(f) \leq |\text{range}_D(f)|$ . Equality  $\text{PC}_D(f) = |\text{range}_D(f)|$  holds if and only if all cells of  $f$  have the same probability.*

*Proof.* Assume  $\text{range}_D(f) = n = \{r_1, r_2, \dots, r_n\}$ , and  $p_i := \mathbb{P}_{x \sim D_X}[f(x) = r_i]$ .

$$\begin{aligned} \frac{1}{\text{PC}_D(f)} &= \mathbb{P}_{x, x' \sim D_X}[f(x) = f(x')] \\ &= \sum_{i=1}^n [\mathbb{P}_{x, x' \sim D_X}[f(x) = f(x') | f(x) = r_i] \cdot \mathbb{P}_{x, x' \sim D_X}[f(x) = r_i]] \\ &= \sum_{i=1}^n [\mathbb{P}_{x' \sim D_X}[f(x') = r_i] \cdot \mathbb{P}_{x \sim D_X}[f(x) = r_i]] = \sum_{i=1}^n p_i^2. \end{aligned}$$

We define two vectors with size  $n$  as  $u = (p_1, \dots, p_n)$  and  $v = (1, \dots, 1)$ . Using Cauchy-Schwarz inequality  $|\langle u, v \rangle|^2 \leq |\langle u, u \rangle| \cdot |\langle v, v \rangle|$  and the equality holds if and only if  $u$  and  $v$  are parallel. With this we get  $|\langle u, v \rangle|^2 = (\sum_{i=1}^n p_i)^2 \leq$

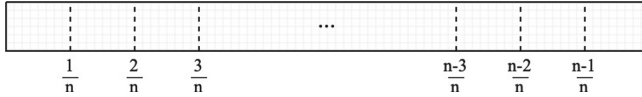
$(\sum_{i=1}^n p_i^2) \cdot n$  which implies  $\sum_{i=1}^n p_i^2 \geq \frac{1}{n}$ . So,  $PC_D(f) \leq n$ . The equality holds if and only if  $(p_1, \dots, p_n)$  and  $(1, \dots, 1)$  are parallel which means that all  $p_i$ s are equal.  $\square$

The probabilistic count depends on the number of cells and their probability weights. We here present a series of examples of this metric. In each example, the cells created in the range of  $f$  from  $\text{supp}(D)$  are visualized in a bar. The length of each partition represents its probability in the distribution  $D$ .

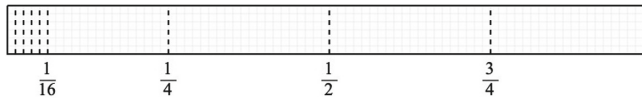
PC is not monotonic with the actual number of cells. The function in Fig. 3 generates 9 cells while its PC is 4.28. If we compare this predictor with a function that generates 5 balanced cells, which leads to PC equals to 5, we can show that the predictor in Fig. 3 has less PC while it has more cells (Figs. 2, 4, 5 and 6).

### B Critiquing the Expected Calibration Error

The empirical expected calibration error (ECE) is a metric that is frequently employed to measure calibration [1, 10, 19]. It averages scores within each bin, rather than evaluating the individual scores, which we show makes it less accurate. When a bin contains both overconfident and underconfident scores, they average out, making the performance seem better than it is, as illustrated in Example 1.

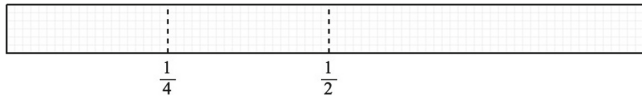


**Fig. 2.** Probabilistic count example on  $n$  cells with the same weight; in this case  $PC_D(f) = n$  is the number of cells.

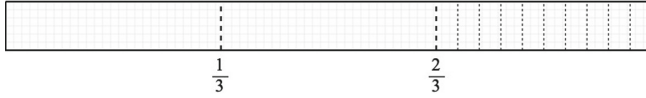


**Fig. 3.** Probabilistic count example on 9 cells including 5 small cells; the cells with small weights do not have much effect on the probabilistic count; while we have 9 cells,  $PC_D(f)$  is close to 4; this shows that PC emphasizes the number of significant cells.

$$PC_D(f) = \frac{1}{\frac{1}{4} \cdot 3 + \frac{1}{80} \cdot 2 + \frac{1}{80} \cdot 5} \approx 4.08.$$

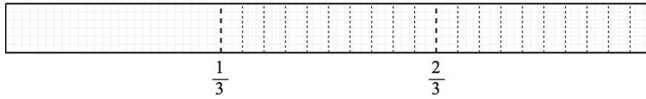


**Fig. 4.** Probabilistic count example on three cells with different weights;  $PC_D(f) = \frac{1}{\frac{1}{4} \cdot 2 + \frac{1}{2}} \approx 2.66$ .



**Fig. 5.** Probabilistic count example on 12 cells including 10 cells distributed on a third; for three equally weighted cells, the probabilistic count is 3; we have split the third partition into ten small cells with the same weights.  $PC_D(f) = \frac{1}{\frac{1}{3}^2 \cdot 2 + \frac{1}{30}^2 \cdot 10} \approx 4.28$ .

*Example 1.* Consider a predictor  $f$  evaluated with a partition that contains a bin  $b$  with  $f(x_i) = 0.35$  for half the samples, and  $f(x_i) = 0.65$  for the other half. Assume that on this bin the regression function satisfies  $\eta_D(x) = 0.5$  for all  $x$ , and that there are sufficiently many samples from the bin that the empirical average is close to 0.5. Now when evaluating the ECE on this bin, the result would be  $|\frac{0.35+0.65}{2} - 0.5| = 0$ , indicating flawless performance of  $f$  in terms of calibration, which is not correct. In contrast, the probability deviation error (PDE) takes individual scores into account. For the same bin, the PDE evaluates to  $\frac{|0.35-0.5|+|0.65-0.5|}{2} = 0.15$ , which corresponds to the correct calibration error.



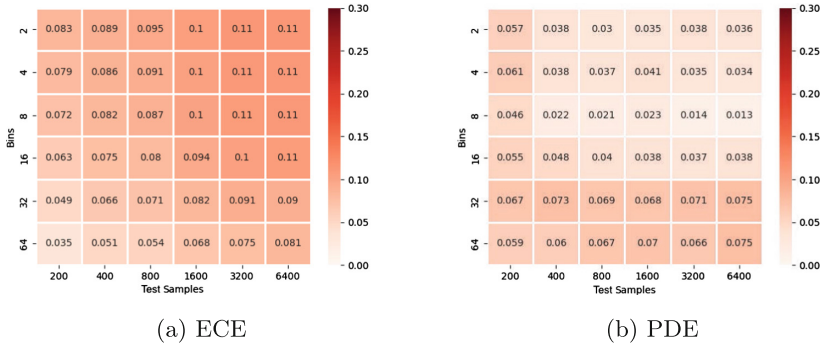
**Fig. 6.** Probabilistic count example on 21 cells including 20 cells distributed on two thirds; we have split each of the second and the third cells into ten small cells with the same weights;  $PC_D(f) = \frac{1}{\frac{1}{3}^2 + \frac{1}{30}^2 \cdot 20} = 7.5$ .

### C Empirically Motivating the Probability Deviation Error

To motivate our proposed measure PDE beyond the Example 1 above, we empirically compare PDE with ECE on a large synthetic dataset. We compare their bias, where the bias of a calibration metric  $\mu$  for predictor  $f : X \rightarrow [0, 1]$  over  $n$  samples  $D^{(n)}$  with respect to the distribution  $D$  is defined as ([12]):

$$\text{Bias}(D, D^{(n)}, \mu) := \frac{1}{m} \sum_{i=1}^m \mu(D^{(n)}(f)) - \frac{1}{n} \sum_{i=1}^n (|f(x_i) - \eta_D(x_i)|),$$

where  $m$  is the number of experiments, each over datasets of size  $n$ . We used  $m = 10$ . Since the regression function is essential to assess bias, we synthetically generated 27,500 samples, generating labels according to a predefined regression function. Thus, we have access to  $\eta_D(x_i)$  for our generated points. We used different sizes of test sets (generated with the same procedure), and Uniform-mass binning for ECE and PDE with a different number of bins ranging from 2 to 64. Our analysis indicates that PDE has mostly lower bias than ECE, provided there are enough samples per bin, see Fig. 7 below.



**Fig. 7.** Evaluating the bias of ECE and PDE, with the latter exhibiting larger bias in most cases, especially with larger numbers of test samples.







## References

- Blasiok, J., Gopalan, P., Hu, L., Nakkiran, P.: A unifying theory of distance from calibration. In: Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC, pp. 1727–1740. ACM (2023)
- Blasiok, J., Gopalan, P., Hu, L., Nakkiran, P.: When does optimizing a proper loss yield calibration? In: Advances in Neural Information Processing Systems, vol. 36. NeurIPS (2023)
- Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recogn.* **30**(7), 1145–1159 (1997)
- Branco, P., Torgo, L., Ribeiro, R.P.: A survey of predictive modelling under imbalanced distributions. arXiv [arXiv:1505.01658](https://arxiv.org/abs/1505.01658) (2015)
- Brier, G.W.: Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.* **78**(1), 1–3 (1950)
- Dawid, A.P.: The well-calibrated Bayesian. *J. Am. Stat. Assoc.* **77**, 605–610 (1982)
- Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
- Famiglini, L., Campagner, A., Cabitza, F.: Towards a rigorous calibration assessment framework: advancements in metrics, methods, and use. In: 26th European Conference on Artificial Intelligence ECAI. *Frontiers in Artificial Intelligence and Applications*, vol. 372, pp. 645–652. IOS Press (2023)
- Foster, D.P., Vohra, R.V.: Asymptotic calibration. *Biometrika* **85**(2), 379–390 (1998)
- Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. arXiv [arXiv:1706.04599](https://arxiv.org/abs/1706.04599) (2017)
- Gupta, C., Ramdas, A.: Distribution-free calibration guarantees for histogram binning without sample splitting. In: Proceedings of the 38th International Conference on Machine Learning ICML, pp. 3942–3952. PMLR (2021)
- Huang, S., et al.: MBCT: tree-based feature-aware binning for individual uncertainty calibration. In: Proceedings of the ACM Web Conference 2022, pp. 2236–2246. Association for Computing Machinery (2022)
- Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. *Int. J. Forecast.* **22**(4), 679–688 (2006)

14. Jensen, J.L.W.V.: Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica* **30**(none), 175 – 193 (1906)
15. Kakade, S.M., Foster, D.P.: Deterministic calibration and nash equilibrium. *J. Comput. Syst. Sci.* **74**(1), 115–130 (2008)
16. Kumar, A., Liang, P.S., Ma, T.: Verified uncertainty calibration. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. (2019)
17. Leathart, T., Frank, E., Holmes, G., Pfahringer, B.: Probability calibration trees. *CoRR* [arXiv:1808.00111](https://arxiv.org/abs/1808.00111) (2018)
18. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Binary classifier calibration: non-parametric approach. *arXiv preprint* [arXiv:1401.3390](https://arxiv.org/abs/1401.3390) (2014)
19. Naeini, M.P., Cooper, G.F., Hauskrecht, M.: Obtaining well calibrated probabilities using Bayesian binning. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2901–2907. AAAI Press (2015)
20. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Adv. Large Margin Classif.* **10**, 61–74 (2000)
21. Posocco, N., Bonnefoy, A.: Estimating expected calibration errors. In: Farkaš, I., Masulli, P., Otte, S., Wermter, S. (eds.) *ICANN 2021*. LNCS, vol. 12894, pp. 139–150. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86380-7\\_12](https://doi.org/10.1007/978-3-030-86380-7_12)
22. Puka, L.: Kendall’s Tau. In: Lovric, M. (ed.) *International Encyclopedia of Statistical Science*, pp. 713–715. Springer, Berlin Heidelberg (2011). [https://doi.org/10.1007/978-3-642-04898-2\\_324](https://doi.org/10.1007/978-3-642-04898-2_324)
23. Scafarto, G., Posocco, N., Bonnefoy, A.: Calibrate to interpret. In: Amini, MR., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G. (eds.) *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD. Lecture Notes in Computer Science*, vol. 13713, pp. 340–355. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-26387-3\\_21](https://doi.org/10.1007/978-3-031-26387-3_21)
24. Silva Filho, T., Song, H., Perelló Nieto, M., Santos-Rodríguez, R., Kull, M., Flach, P.: Classifier calibration: a survey on how to assess and improve predicted class probabilities. *Mach. Learn.* **112**, 1–50 (2023)
25. Sun, Z., Song, D., III, A.O.H.: Minimum-risk recalibration of classifiers. In: *Advances in Neural Information Processing Systems*, vol. 36. *NeurIPS* (2023)
26. Wang, C.: Calibration in deep learning: A survey of the state-of-the-art. *arXiv preprint* [arXiv:2308.01222](https://arxiv.org/abs/2308.01222) (2024)
27. Zadrozny, B., Elkan, C.: Transforming classifier scores into accurate multiclass probability estimates. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699. Association for Computing Machinery (2002)



# XentricAI: A Gesture Sensing Calibration Approach Through Explainable and User-Centric AI

Sarah Seifi<sup>1,2</sup>  , Tobias Sukianto<sup>1,3</sup> , Maximilian Strobel<sup>2</sup>, Cecilia Carbonelli<sup>2</sup> , Lorenzo Servadei<sup>1</sup> , and Robert Wille<sup>1,4</sup> 

<sup>1</sup> Technical University of Munich, Arcisstraße 21, 80333 Munich, Germany  
sarah.seifi@tum.de

<sup>2</sup> Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany

<sup>3</sup> Johannes Kepler University Linz, Altenbergerstraße 69, 4040 Linz, Austria

<sup>4</sup> Software Competence Center Hagenberg GmbH (SCCH), Softwarepark 32a, 4232 Hagenberg, Austria

**Abstract.** Gesture recognition systems offering contactless human-machine interaction have diverse applications, from smart homes to healthcare. However, they often face challenges from unexpected changes in user behavior and a lack of explainability, especially concerning fields like medical diagnosis or security systems. To address these issues, we introduce a novel approach that exploits advances in Explainable Artificial Intelligence (AI) and Experience Replay techniques for human-centric AI in radar-based gesture sensing. Our contributions include model calibration via Transfer Learning using Experience Replay and feedback on anomalous gestures through feature analysis with Explainable AI. Experimental results show improved accuracy, low forgetting rate, and enhanced user engagement, suggesting the potential for fostering trust in AI technology. The model calibration leads to an average accuracy improvement of 5.4% with respect to the uncalibrated model. Furthermore, leveraging the Explainable AI feedback to enhance gesture execution yields a 38.1% average accuracy improvement compared to unguided user behavior.

**Keywords:** Explainable AI · User-Centric AI · Gesture Recognition

## 1 Introduction

In today's fast-paced technological landscape, gesture recognition systems have become a hygienic, contactless solution for enabling human-machine interaction. By offering a seamless interface between individuals and technology, these systems find valuable utilization in diverse domains, such as smart homes, medical applications, and security systems [1–3]. However, the performance of such systems often faces a significant challenge – the noticeable disparity between the data used to train the models and the real-world scenarios in which they are deployed [4]. This discrepancy results in a distributional shift that can lead to reduced accuracy and unexpected misclassifications in



the field, consequently undermining the usability and reliability of these gesture recognition systems. To address these limitations, this paper presents an innovative approach combining Explainable Artificial Intelligence (XAI) with Experience Replay (ER) for a user-centric AI approach, which we call XentricAI.

This synergetic approach aims to overcome the current limitations of gesture recognition systems by creating a more user-centric framework robust towards distributional shifts.

Within the framework of this study, we introduce several contributions to the field of gesture recognition and AI-driven user interaction, which together constitute the building blocks of XentricAI:

- **User-Centric Model Recalibration** - An approach involving end-users adapting AI models to their unique gesture execution patterns. This improves real-world accuracy and enhances user engagement.
- **Anomalous Gesture Feedback through Feature Analysis** - Integration of the XAI method SHAP [5] gives users insights into the factors influencing AI model decisions. This allows users to adapt their gestures for improved recognition, enhancing the transparency and effectiveness of user-AI interaction.

The proposed XentricAI algorithm represents a user-centered solution to the challenges of gesture recognition systems, ultimately paving the way for more robust and user-friendly AI-driven interactions.

## 2 Background and Related Work

This section sets the context by addressing AI model explainability challenges and the significance of transparency in user interactions. It also analyzes existing model calibration methods, forming the basis for the approaches presented in the following sections.

### 2.1 User-Centric XAI Techniques

Current neural network architectures often lack explainability and operate as complex “black boxes”, complicating understanding for both experts and users [6, 7].

Understanding and transparency are crucial because they foster trust in users and, as a result, make them use Machine Learning (ML) models. For ML experts, comprehending the model’s decision-making process aids in diagnosing and rectifying issues. However, for domain experts and regular users, it is even more vital when, e.g., relying on a recommendation from a model for a critical decision, such as medical diagnosis or financial planning [8, 9]. Users impacted by AI model predictions need transparent reasoning about the models’ decision-making to ensure fair decisions [10].

However, achieving this transparency is a challenge, especially in gesture-sensing, where providing feedback to the user still needs to be explored. Existing XAI methods often need more empirical validation through user studies [7, 11].

Nevertheless, some user-centered XAI approaches have been proposed in other fields by providing algorithm visualizations, user interfaces, and toolkits. For instance, an

interactive visual analytics system, enabling data scientists to understand feature impacts on predictions and investigate individual data points' reasons for prediction outcomes, was proposed in [12]. In another work, graph neural network predictions were explained to domain experts in drug repurposing using visual explanations at each design layer [13]. By designing an explainable diagnostic tool for intensive care, a research group evaluated their theory-driven conceptual framework linking different XAI explanation facilities to user reasoning goals [14].

While promising for enhancing explainability, it is important to recognize the susceptibility of XAI, particularly in the context of vision and vision-language tasks, to potential adversarial attacks and vulnerabilities that may undermine their reliability and robustness [15, 16].

A key difference to [12–14], where they concentrate on providing black-box explanations to domain or ML experts, is that our work proposes a user-centric-based AI solution in which the regular user is actively involved. Additionally, XentricAI can explain model misclassifications to users via XAI techniques, encouraging more effective inputs and sensible interaction. Physical features enhance decision transparency, empowering users to adapt gestures for improved model detection accuracy.

## 2.2 Gesture Sensing Model Calibration Using Experience Replay

Calibrating a ML model to a user is essential in user-centric AI, aligning technical capabilities with practical usability, enhancing user experience, and addressing the performance gap between training and real-world scenarios. This process tailors the model to the user while preserving their privacy. Transfer Learning (TL) addresses this by repurposing pre-trained models for new tasks [17].

One research group presents a gesture recognition algorithm of electromyographic data between five gestures using an ensemble of classical ML models [18]. Model calibration is executed by retraining the model with user data. However, the model performance on the initial train dataset is overlooked. A ML model might suffer from catastrophic forgetting, which refers to the performance drop or forgetting rate in previously learned knowledge when applying it to new unseen data [19] or, in the case of this study, a new user.

Our approach combines user retraining with ER, where a small subset of data instances from the initial train dataset is sampled and added to the new, unseen training data for optimal adaptation while minimizing the forgetting rate [20]. The forgetting rate refers to the rate at which the model forgets or replaces old experiences (meaning the performance on the data it initially was trained on) with new ones as it learns from different tasks or domains after retraining. Unlike prior work assuming similar user gesture execution patterns [21, 22], our method accommodates varied execution styles, including speed and proximity to the radar. As a result, this broader perspective better calibrates models under significant distributional changes.

Our paper introduces contributions that merge XAI and TL for a human-centric AI approach while also pioneering ER for model calibration in radar-based gesture sensing, accommodating significant distributional shifts.

### 3 XAI for User-Centric and Customized Gesture Sensing

This section presents XentricAI, combining XAI techniques with customized gesture sensing.

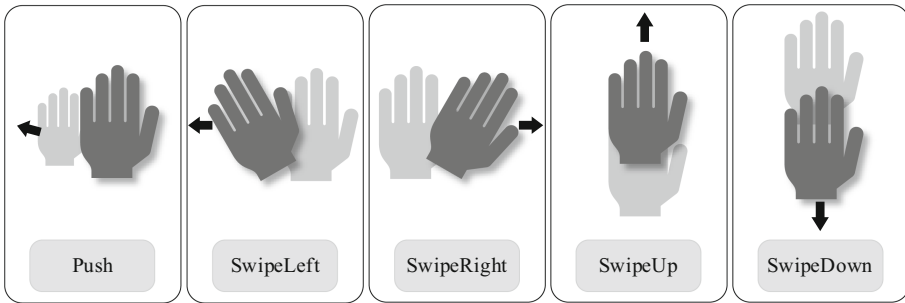
To this end, we build upon a previously proposed gesture sensing algorithm and intuitive radar features [23]. After revisiting these elements, we elaborate on the two components of XentricAI. We explain the ER-based model calibration and then conclude by utilizing XAI methods to explain anomalous gestures.

#### 3.1 Gesture Sensing Algorithm and Feature Design

Here, we explain the neural network architecture, the feature design, and the radar preprocessing. As these ideas mainly stem from previous research, they are not classified as novel contributions.

**Radar Preprocessing and Intuitive Feature Design.** Incorporating intuitive physical meanings into AI systems is a step toward making the decision-making of black-box models more transparent, relatable, and accessible. Since this work is centered on user-centric AI, instead of relying on heavy 2D radar processing, we adopt the lightweight radar processing algorithm introduced in [23]. There, the hand movements are distinguished via a time series of radio-frequency scattering characteristics, i.e., *radial distance* (Range), *radial velocity* (Doppler), *horizontal angle* (Azimuth), *vertical angle* (Elevation), and *signal magnitude* (Peak). The radar preprocessing algorithm first identifies the hand as the closest target to the radar and then extracts the mentioned features. This is done by transforming the raw radar data via fast-time processing into range profiles and then applying a peak search on this data. Once the hand's range bin is identified, slow-time processing is conducted along the hand's range bin. The resulting Doppler profile is then averaged across all antennas, and the highest signal within the respective Doppler bin is extracted, providing information about the radial velocity. Simultaneously, the amplitude of the signal is indicative of its magnitude. Finally, the processing includes the estimation of horizontal and vertical angles related to the detected Doppler bin, thus providing insight into the hand's position and movement in both the horizontal and vertical dimensions. These features are used as inputs for the gesture sensing network. For more details regarding the preprocessing, we refer to [23].

**Gesture Sensing Network Architecture.** In this study, a tiny Recurrent Neural Network (RNN) architecture detects and classifies gestures. This network consists of a Gated Recurrent Unit (GRU) layer with 16 units to learn from time-series data and a dense layer with six neurons and Softmax activation to distinguish between five gestures and the *Background* class. The five gestures were *Swipe Left*, *Swipe Right*, *Swipe Up*, *Swipe Down*, and *Push*, as can be seen in Fig. 1. The inputs for the network are data sequences of the five extracted features. One data sequence comprises 22 frames, while the label of the last frame is used as the label of the entire sequence. Two consecutive sequences are shifted by one frame. TL is applied to the model without freezing any layers.



**Fig. 1.** Schematic illustration of the recorded gesture executions.

### 3.2 Model Calibration Using Experience Replay

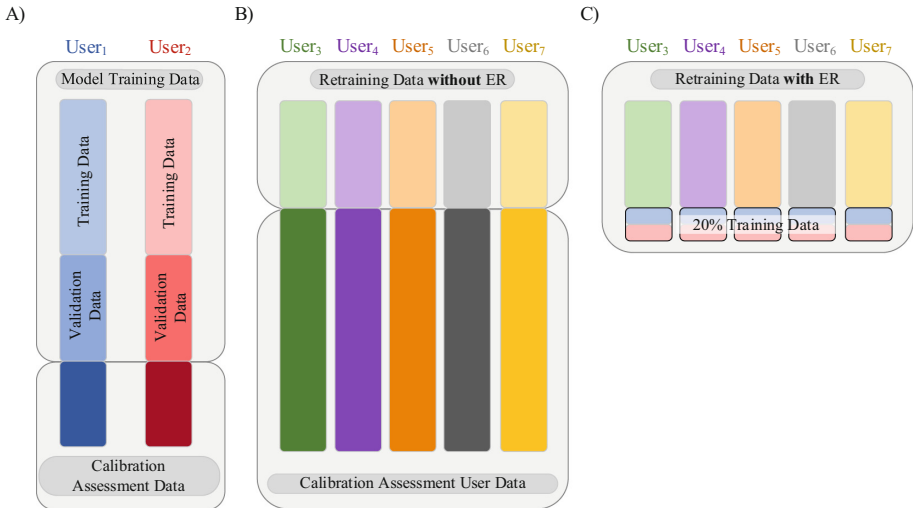
For model calibration, we engage the user in recording a set of gestures, capturing their gesture execution patterns (highlighted as  $User_3$  to  $User_7$  in Fig. 2). The user dataset is divided into two independent, temporally non-overlapping parts. As illustrated in Fig. 2 B), one part of the user dataset is used for model calibration without ER and is named *retraining data without ER*. To perform model calibration with ER, 20% randomly selected recordings from the initial training dataset (based on the data from  $User_1$  and  $User_2$  as shown in panel A)) are added to the retraining data without ER. This novel dataset is named *retraining data with ER* (s. panel C)). The model is then retrained with this combined dataset, enabling adaptation to the user's unique gesture patterns while eliminating the risk of catastrophic forgetting.

Two assessment datasets are created to assess the calibrated model and determine the forgetting rate of the calibrated model with and without ER. For the first assessment dataset, a segment of the dataset from  $User_1$  and  $User_2$  is deliberately withheld from the training process. This segment is preserved exclusively for evaluating the extent of forgetting and is named *calibration assessment data* (s. panel A)). The second assessment dataset is called *calibration assessment user data* (s. panel C)). It is based on the second part of the user dataset, consisting of recordings independent from the model retraining dataset.

### 3.3 Anomalous Gesture Detection and Characterization

To further enhance the user-centric design of AI systems, we also want to make the model's decision-making process more transparent after adapting the model to the user behavior. We want to achieve this by providing the user with an explanation of what caused deviations in the event of anomalous gestures while keeping the user in the loop. This explanation mechanism leverages the physically interpretable meaning of the features and an XAI method, namely SHAP.

In the subsequent section, anomalous gestures are defined, followed by a comprehensive description of the mechanism employed in XentricAI to explain such anomalies, which is further illustrated in Algorithm 1 in the appendix.



**Fig. 2.** Illustration of the different datasets and their functions. A) Most of the data of two users is used for model training and is split into training and validation data. A small fraction of  $User_1$  and  $User_2$  data is used for model calibration evaluation. It is named *calibration assessment data*. B) The data of the remaining five users is used to calibrate the model without ER (*retraining data without ER*) as well to assess the calibrated model (*calibration assessment user data*). C) For model calibration with ER, 20% randomly selected recordings from the training data are added to the retraining data. The resulting dataset is called *retraining data with ER*.

**Anomalous Gesture Detection.** Firstly, there are cases of undetected gestures where the system fails to generate any prediction for a particular gesture instance. Secondly, mixed-class predictions are encountered when a single gesture is associated with predictions corresponding to multiple, different gesture classes. Lastly, our analysis includes instances of sparse predictions in which a gesture classification is marked by intermittently occurring predictions at the level of individual frames. This phenomenon suggests an unevenness or irregularity in the model’s classification process, with specific frames being assigned predictions while others still need to be addressed. These anomalous gesture patterns collectively reveal complexities in the model’s performance and provide valuable insights into areas where refinement is required. By identifying and understanding these anomalies, we take significant steps toward enhancing the reliability and interpretability of the model’s gesture recognition outcomes.

**Anomalous Gesture Characterization.** By employing the XAI method SHAP, anomalous gestures are characterized, granting the user insights into the model’s mispredictions. SHAP is a widely utilized model-agnostic method for explaining the output of ML models [5]. It uses the concept of *Shapley values* from cooperative game theory to determine the contribution of each feature to the model prediction [24]. These values fairly distribute feature contributions by averaging marginal effects in all possible feature coalitions. The algorithm calculates the average marginal contribution of each feature value across all possible coalitions, which are combinations of present or absent features. To do this, it first generates predictions for different coalitions with and without

the analyzed feature, then takes the difference between those predictions to calculate the marginal contribution of the feature. This process is repeated for all features, and the resulting values are the SHAP Values (SVs), which are estimates of the *Shapley values*, representing each feature's importance on the model's prediction. This is defined as:

$$g(z') = \phi_0 + \sum_{j=1}^M \phi_j z'_j \quad (1)$$

with  $g$  being the explanation model,  $z' \in \{0, 1\}^M$  is the coalition vector,  $\phi_j \in \mathbb{R}$  is the SV for a feature  $j$ , and  $M$  the maximum coalition size.

The SHAP algorithm provides explanations on a local as well as a global level which is achieved by averaging over all absolute local explanations:

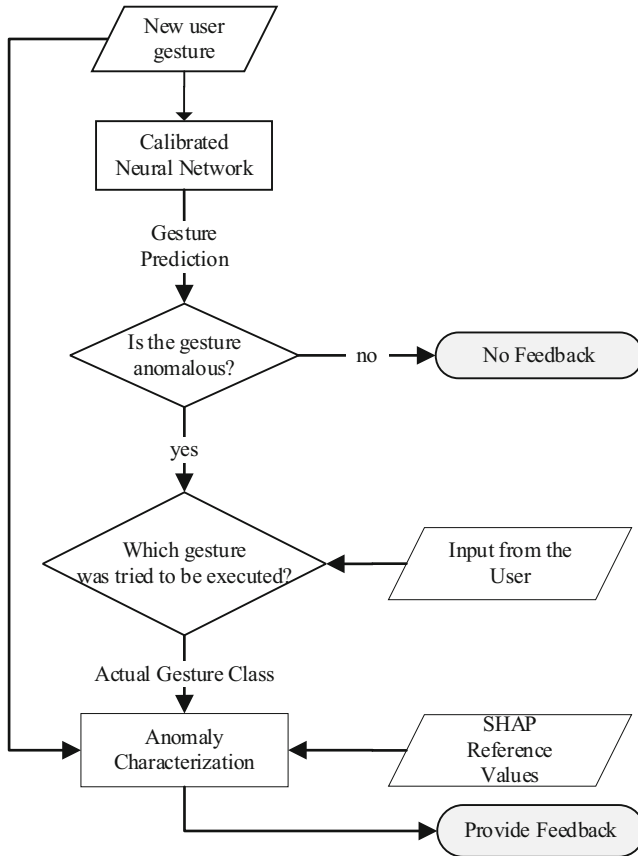
$$I_j = \frac{1}{n} \sum_{i=1}^n |\phi_j^i| \quad (2)$$

with  $I_j$  being the global explanation for feature  $j$  and  $n$  being the number of samples in the dataset.

Our methodology consists of two principal phases, depicted in a schematic manner in Fig. 2:

1. **Initialization Phase:** This phase sets the foundation for gesture characterization. For each gesture class, the global SVs of  $n$  nominal gestures are calculated. Using a thresholding mechanism, the range of acceptable SVs, referred to as SHAP Reference Values (SRVs), for each feature is deduced. Specifically, the upper and lower thresholds are determined by considering the minimum and maximum SVs across the nominal gestures. To capture the relationships between feature importances, we gauge whether the ordering of feature importance changes between nominal and anomalous gestures. This involves calculating a median threshold using the median of the upper and lower thresholds. The slope between consecutive features of this median threshold aids in analyzing the alteration in feature importance ordering. This initialization phase is performed once and prepares the model for subsequent utilization in characterizing future anomalous gestures.
2. **Explanation Phase:** In this phase, each anomalous gesture is characterized. Local and global SVs are computed for the anomalous gesture. The system asks for user input regarding the intended gesture class. Using the user input, the anomalous SVs and the SRVs are leveraged to provide feedback to the user, aiding in understanding and refining the execution of the gesture.

SVs at each time step were calculated by training the *GradientExplainer* from SHAP, which extends the *IntegratedGradients* method [25].



**Fig. 3.** Flowchart for gesture characterization and feedback generation based on user input.

## 4 Experiments

In this section, we first review the implementation settings. Afterward, we show the model calibration results. Then, we evaluate to what extent the feedback to the user improved their gesture execution and, combined with the model calibration, led to improved accuracy.

### 4.1 Implementation Settings

In the implementation, we used TensorFlow v2.9.1<sup>TM</sup>. We used the 6 Core Intel® Core i7-9850H CPU as a processing unit. The experiments were conducted over five users and averaged over three experiments per user.

**Measurement Setup and Dataset Collection.** For this work, Infineon Technologies' XENSIV™ BGT60LTR13C 60 GHz FMCW radar has been used. The radar system was configured with an operational frequency range spanning from 58.5 GHz to 62.5 GHz, resulting in a range resolution of 37.5 mm. The radar system's capabilities extended to a maximum resolvable range of 1.2 m. For signal transmission, the radar employed a burst configuration comprising 32 chirps per burst with a frame rate of 33 Hz.

For the data set collection, seven persons performed five different gestures in five locations in a field view of  $\pm 45^\circ$  with a distance to the radar of  $\leq 1$  m.

Within one gesture recording with a length of approximately 3 s or 100 frames, one gesture was performed with an average duration of 0.3 s or ten frames. All non-gesture frames within one recording were labeled as *Background*. A more detailed description of the label refinement algorithm can be found in [23]. The users were asked to fully extend their arms for a nominal gesture execution.

The dataset of the five individual users included gestures with fast and slow execution, i.e., approximately 0.1 s and 3 s, and a partially extended arm leading to a more significant distance towards the radar.

**Model Architecture, Training and Evaluation.** As previously mentioned, the gesture sensing model consists of two layers. The first layer has 16 nodes and a ReLU activation function, while the output layer has five nodes (corresponding to the number of classes) and a softmax activation function.

During training, the model uses the Adam optimizer, a learning rate of 0.001, the sparse categorical cross-entropy loss function, and a batch size of 32 for 100 epochs. The data of two individuals were used for model training and validation (s. Fig. 2 A)). This model is referred to as *default model*. The remaining five persons were seen as individual users on whose data the model needs to be recalibrated. The user dataset was split into two individual, temporally non-overlapping and independent parts: One part was used for model calibration whereas the second part was used for model calibration assessment, as previously highlighted in Sect. 3.2 and Fig. 2 B)-C).

For model performance measures we utilized the accuracy metric, particularly suited for datasets with balanced classes.

**XAI Training.** As a first step, we train the explanation model using SHAP's Gradient-Explainer. For this, the previously trained gesture sensing model as well as the training data are utilized. The trained explanation model is then used to estimate the SVs of each feature at each time step. Using  $n$  nominal gestures, the SRVs are determined to enable the anomalous gesture characterization.



## 4.2 Experimental Results

**Model Calibration Using ER.** Table 1 shows the model calibration results with and without ER compared to the uncalibrated default model and the forgetting rate.

Calibrating the model with user data resulted in performance improvement of 1.2% without ER and 5.4% in case of ER, signifying the model’s successful adaptation to user-specific gesture behavior and posture. Furthermore, incorporating retraining with ER effectively maintains a low forgetting rate of 0.7% on previous tasks, compared to a higher forgetting rate of 10.1% without ER, thus preserving the model’s capacity for generalization.

**Table 1.** Model calibration results.

	Default Model	Model w/o ER	Model with ER
User Data Accuracy [%]	83.5	84.7	<b>88.9</b>
$\Delta$ Accuracy [%]	–	+1.2	<b>+5.4</b>
Unseen Train Data Accuracy	92.9	82.8	<b>92.2</b>
Forgetting Rate [%]	–	10.1	<b>0.7</b>

**Anomalous Gesture Characterization.** This part provides exemplary results of an anomalous gesture characterization using the proposed explainable mechanism. In Fig. 4 the results of the SHAP feedback algorithm are shown while executing the gesture *Swipe Left* at a normal pace, panels A) and B), vs. at a fast pace, panels C) and D).

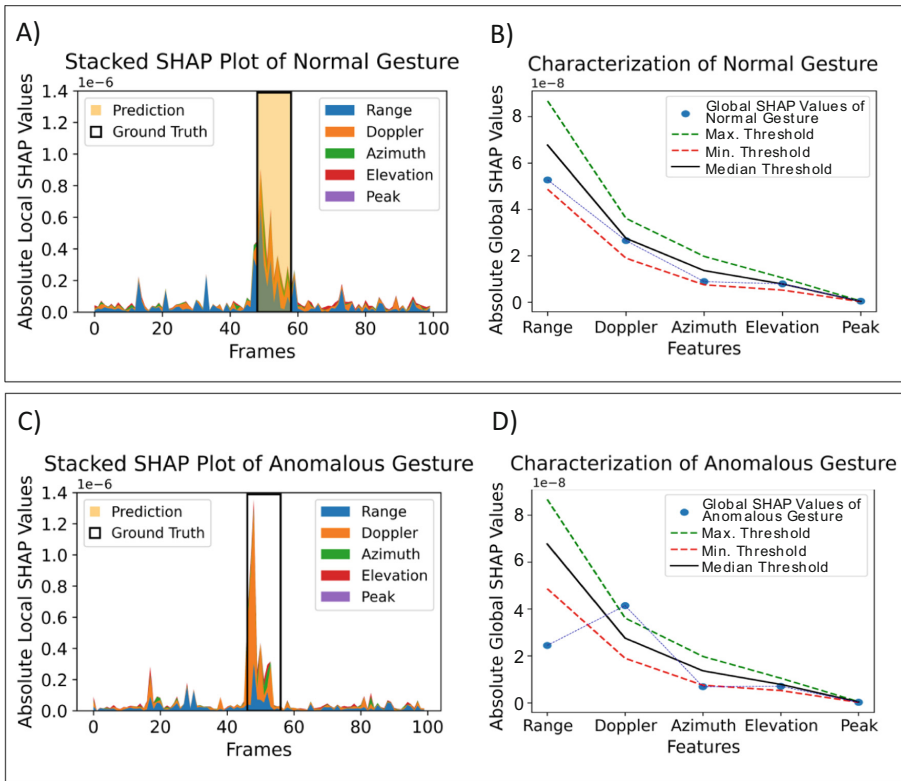
It should be noted that while model calibration aims to robustify the model towards speed and range variations of the user, some gestures still exhibit anomalies. As shown in Fig. 4, the fast *Swipe Left* gesture goes undetected, and we intend to address this issue by offering feedback and, hence, providing transparency.

Panels A) and C) show an exemplary stacked plot of each feature’s absolute local SVs at every timestep. The period during which the gesture was performed is visually represented as a black box named “Ground Truth”. The model prediction is then highlighted in a color corresponding to the gesture class. Panel A) displays an accurate gesture detection, while in contrast, panel C) shows that the gesture remained undetected. Panels B) and D) show the retrieved minimum, maximum, and median thresholds using the proposed thresholding mechanism and the user input about the actual gesture class.

In nominal gestures, the feature *Range* has the highest influence, followed by *Doppler*, as seen in panel B). Whether *Swipe Left/Right* or *Swipe Up/Down* was executed, either *Azimuth* follows in the ranking or *Elevation*. It should be noted that *Peak* had little to no influence, which indicates that it does not contribute towards the model output.

Using the median threshold, it becomes clear that an important rule needs to be followed: The relationship between Range and Doppler should have a downward slope. We can observe that this is not the case for the anomalous gesture. Additionally, the global SVs for both the Range and Doppler features significantly deviate from the acceptable range, indicating the possible cause of the misprediction. Consequently, feedback can now be provided to the user: The speed of the hand movement affected how well the gesture was recognized, causing an unusual and substantial impact on the model’s results and, hence, requiring corrective adjustment.

It should be noted that deviations of the global SVs from the nominal thresholds were observed for all gesture classes and users in the case of an anomaly. No significant difference in the results between the different gesture types was observed. The gesture execution location also did not affect the outcome of XentricAI. Feedback was provided based on the type of deviation.



**Fig. 4.** Characterization of a normal vs. an anomalous gesture (normal vs. fast-paced *SwipeLeft* gesture execution) using the SHAP feedback algorithm.

Table 2 shows the model performance improvement after the users received gesture adjustment suggestions. Our results indicate the efficacy of our innovative user-centric AI techniques, which not only increase model performance through calibration but also maintain a negligible forgetting rate. Additionally, the SHAP engine’s feedback helps the user understand the underlying reasons for model misbehavior and improves gesture execution. This approach brings many advantages, encompassing improved model accuracy, heightened user engagement, and a more intuitive user-AI interaction.

Importantly, our methodology exhibits flexibility and can be readily extended to encompass different features, neural network architectures, and sensing modalities, offering an adaptable framework for enhancing user-centric AI across various applications.

**Table 2.** Enhanced Model Performance with SHAP feedback.

	Before Feedback	After Feedback	Improvement
Accuracy [%]	48.9	87.0	38.1

## 5 Conclusion

This work introduced XentricAI, a novel approach that merges XAI and TL using ER for a more human-centric AI paradigm in radar-based gesture sensing. By combining user-specific calibration and XAI, we enhance model accuracy and provide users with insights into model decisions, improving their gesture execution. Our model calibration approach leads to an average accuracy increase of 5.4% with a low forgetting rate of 0.7%. Providing gesture adjustment suggestions based on XAI, an average enhanced model accuracy performance of 38.1% was achieved. Our contributions address the challenges of model adaptation to user behavior and provide intuitive explanations for model behavior deviations. This approach has the potential to bridge the gap between model performance and user satisfaction in real-world applications of gesture recognition.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## Appendix

---

**Algorithm 1.** SHAP Gesture Characterization.
 

---

**Input:** Dataset with  $C$  gesture classes  $D = \{D_1, \dots, D_C\}$  where  $D_i = \{X_1, \dots, X_M\}$  with  $M$  data samples  $X_i$

```

1: Initialization Phase {
2: Initialize empty dictionaries: shap reference values
3: for each gesture in  $D_i$  do
4:   for each  $X_i$  in  $D_i$  do
5:     Initialize empty lists: nominal_feature_shap
6:     for each nominal_gesture in  $X_i$  do
7:       Calculate global_shap_value for each feature
8:       Append to nominal_feature_shap
9:     end for
10:   end for
11:   Compute lower_threshold =  $\min(\text{nominal\_feature\_shap})$ 
12:   Compute upper_threshold =  $\max(\text{nominal\_feature\_shap})$ 
13:   Store thresholds shap_reference_values[feature] = {lower_threshold,
upper_threshold}
14: end for
15: }
16:
17: Explanation Phase {
18: for each anomalous_gesture do
19:   Calculate global_shap_value for each feature
20:   Ask user for input about real gesture class
21:   if any( $\text{global\_shap\_value} \geq \text{lower\_threshold}$  or  $\leq \text{upper\_threshold}$ ) for each
feature then
22:     global_shap_value is Nominal
23:   else
24:     global_shap_value is Anominal
25:   end if
26:   anomalous_slopes = [ $\text{diff}(\text{anomalous\_shap\_value})$ ]
27:   if any( $\text{anomalous\_slopes} > \text{slopes}$ ) for each feature then
28:     Feature importance increased from nominal to anomalous gestures
29:   else if any( $\text{anomalous\_slopes} < \text{slopes}$ ) for each feature then
30:     Feature importance decreased from nominal to anomalous gestures
31:   end if
32: end for
  }

```

---

## References

1. Wan, Q., Li, Y., Li, C., Pal, R.: Gesture recognition for smart home applications using portable radar sensors. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014. 2014, 6414–7 (2014). <https://doi.org/10.1109/EMBC.2014.6945096>

2. Wang, W., He, M., Wang, X., Ma, J., Song, H.: Medical gesture recognition method based on improved lightweight network. *Appl. Sci.* **12**, 6414 (2022). <https://doi.org/10.3390/app12136414>
3. Kabisha, M.S., Rahim, K.A., Khaliluzzaman, M., Khan, S.I.: Face and hand gesture recognition based person identification system using convolutional neural network. *Int. J. Intell. Syst. Appl. Eng.* **10**, 105–115 (2022). <https://doi.org/10.18201/ijisae.2022.273>
4. Cui, P., Athey, S.: Stable learning establishes some common ground between causal inference and machine learning. *Nat. Mach. Intell.* **4**, 110–115 (2022). <https://doi.org/10.1038/s42256-022-00445-z>
5. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. (2017)
6. Castelvocchi, D.: Can we open the black box of AI? *Nature News.* **538**, 20 (2016). <https://doi.org/10.1038/538020a>
7. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies – ScienceDirect. <https://www.sciencedirect.com/science/article/pii/S0004370221000102>
8. Petch, J., Di, S., Nelson, W.: Opening the black box: the promise and limitations of explainable machine learning in cardiology. *Can. J. Cardiol.* **38**, 204–213 (2022). <https://doi.org/10.1016/j.cjca.2021.09.004>
9. Weber, P., Carl, K.V., Hinz, O.: Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature. *Manag Rev Q.* (2023). <https://doi.org/10.1007/s11301-023-00320-0>
10. Arrieta, A.B., et al.: Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI (2019). <http://arxiv.org/abs/1910.10045>
11. Mueller, S.T., Hoffman, R.R., Clancey, W., Emrey, A.: Explanation in human-AI systems: a literature meta-review synopsis of key ideas and publications and bibliography for explainable AI
12. Krause, J., Perer, A., Ng, K.: Interacting with predictions: visual inspection of black-box machine learning models. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose California USA, pp. 5686–5697. ACM (2016)
13. Wang, Q., Huang, K., Chandak, P., Zitnik, M., Gehlenborg, N.: Extending the nested model for user-centric XAI: a design study on GNN-based drug repurposing. *IEEE Trans. Visual Comput. Graphics* **29**, 1266–1276 (2023). <https://doi.org/10.1109/TVCG.2022.3209435>
14. Wang, D., Yang, Q., Abdul, A., Lim, B.Y.: Designing theory-driven user-centric explainable AI. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow Scotland UK, pp. 1–15. ACM (2019)
15. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, New York NY USA, pp. 180–186. ACM (2020)
16. Baia, A.E., Poggioni, V., Cavallaro, A.: Black-box attacks on image activity prediction and its natural language explanations. In: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Paris, France, pp. 3688–3697. IEEE (2023)
17. Malinin, A., et al.: Shifts 2.0: Extending The Dataset of Real Distributional Shifts (2022). <http://arxiv.org/abs/2206.15407>
18. Dolopikos, C., Pritchard, M., Bird, J.J., Faria, D.R.: Electromyography signal-based gesture recognition for human-machine interaction in real-time through model calibration. In: Arai, K. (ed.) *Advances in Information and Communication*, pp. 898–914. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-73103-8\\_65](https://doi.org/10.1007/978-3-030-73103-8_65)
19. Chen, X., Wang, S., Fu, B., Long, M., Wang, J.: Catastrophic forgetting meets negative transfer: batch spectral shrinkage for safe transfer learning. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. (2019)

20. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G.: Experience replay for continual learning. In: *Advances in Neural Information Processing Systems*. Curran Associates, Inc. (2019)
21. Zhang, B.-B., Zhang, D., Li, Y., Hu, Y., Chen, Y.: Unsupervised domain adaptation for device-free gesture recognition (2021). <http://arxiv.org/abs/2111.10602>
22. Liu, H., et al.: mTransSee: enabling environment-independent mmWave sensing based gesture recognition via transfer learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **6**, 23:1–23:28 (2022). <https://doi.org/10.1145/3517231>
23. Strobel, M., Schoenfeldt, S., Daugalas, J.: Gesture Recognition for FMCW Radar on the Edge (2023). <http://arxiv.org/abs/2310.08876>
24. Shapley, L.S.: 17. A Value for n-person games. In: *17. A Value for n-Person Games*, pp. 307–318. Princeton University Press (2016)
25. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: *Proceedings of the 34th International Conference on Machine Learning*. pp. 3319–3328. PMLR (2017)



# Toward Understanding the Disagreement Problem in Neural Network Feature Attribution

Niklas Koenen<sup>1,2</sup>  and Marvin N. Wright<sup>1,2,3</sup>  

<sup>1</sup> Leibniz Institute for Prevention Research and Epidemiology – BIPS, Bremen, Germany

{koenen,wright}@leibniz-bips.de

<sup>2</sup> University of Bremen, Bremen, Germany

<sup>3</sup> University of Copenhagen, Copenhagen, Denmark

**Abstract.** In recent years, neural networks have demonstrated their remarkable ability to discern intricate patterns and relationships from raw data. However, understanding the inner workings of these black box models remains challenging, yet crucial for high-stake decisions. Among the prominent approaches for explaining these black boxes are feature attribution methods, which assign relevance or contribution scores to each input variable for a model prediction. Despite the plethora of proposed techniques, ranging from gradient-based to backpropagation-based methods, a significant debate persists about which method to use. Various evaluation metrics have been proposed to assess the trustworthiness or robustness of their results. However, current research highlights disagreement among state-of-the-art methods in their explanations. Our work addresses this confusion by investigating the explanations' fundamental and distributional behavior. Additionally, through a comprehensive simulation study, we illustrate the impact of common scaling and encoding techniques on the explanation quality, assess their efficacy across different effect sizes, and demonstrate the origin of inconsistency in rank-based evaluation metrics.

**Keywords:** XAI · Feature Attribution · Neural Networks · Disagreement Problem · Tabular Data · Simulation Study

## 1 Introduction

Over the past decade, one specific class of machine learning models has been rapidly integrated into our daily lives: neural networks. Thanks to increasing computational power and resources, it has become possible to control these exceptionally flexible and highly parameter-rich models. Their remarkable success spans from image recognition to financial forecasts and disease detection [12, 35]. Nevertheless, this black box and its prediction-making process are challenging or even impossible for humans to fully understand.

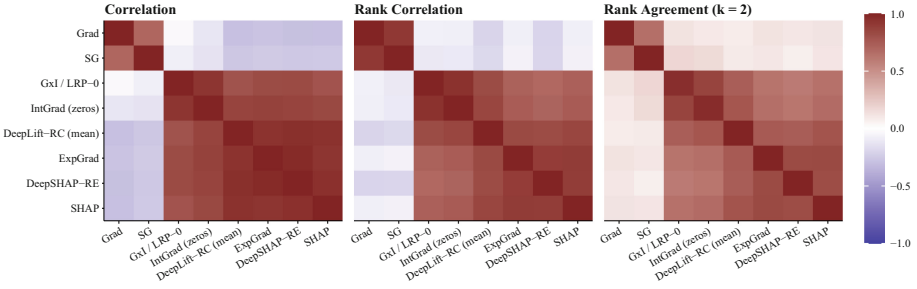
Building upon this question of explaining a model’s prediction, the term *explainable artificial intelligence* (XAI) has emerged, and a fast-growing research area has been established. Many methods have been proposed to explain black box models, ranging from intrinsic and self-explaining neural networks [4, 6] to concept-recognizing [11, 29] and perturbation-based approaches [27, 43]. However, the most well-known and commonly used group for post-hoc explanations consists of *feature attribution methods*. For a trained model, they assign relevance or contribution scores to the input variables, thus indicating the features or components on which the model bases its prediction. They have become known primarily for their type of visualization as heatmaps or saliency maps for image data. Nevertheless, most methods can also be applied to other data types, such as tabular data, due to the feature-individual attributions.

Driven by an increasingly diverse collection of methods [9, 47–50, 52, 54], feature attribution research has recently shifted away from method development and toward the question of which method is the ‘best’. However, this question can be answered differently depending on which aspect is being asked. In this sense, numerous partly heuristic evaluation metrics have been proposed measuring, e.g., an explanation’s trustworthiness/fidelity [4, 5, 17, 44, 58], robustness [34, 58], complexity [13, 42], or monotonicity [8]. Additionally, feature attribution methods have been compared in benchmarks using these metrics, mostly concluding that the choice is either model or dataset-dependent [2, 3, 36, 57]. Bhatt et al. [13] even propose a meta-metric based on aggregating various metrics.

Most of these evaluation metrics are based on a (simulated) information elimination of highly attributed features and the resulting observable change in the model’s prediction [5, 44], and do not consider the explanation’s distributional behavior. Apart from prediction-grounded evaluations, the magnitude of relevances is also used for pairwise comparisons of the method’s local explanations. In this context, Krishna et al. [34] postulated the so-called *disagreement problem*, as many state-of-the-art methods differ significantly in their assignment of important features. The authors defined rank-based metrics on the basis of Neely et al. [41], e.g., the rank agreement or rank correlation, and showed the disagreement using real-world datasets. We reproduce their comparison on the COMPAS dataset [10] and also observe this disagreement (see Fig. 1 middle and right). However, we claim this disagreement is not mainly caused by different explanation qualities but rather by how the effects are measured for many state-of-the-art methods. Moreover, we demonstrate that the magnitude of the relevance is strongly affected by an implicitly or explicitly chosen baseline value causing the method to answer different questions. For example, we can see the high correlation of the feature-wise distributions for non-plain gradient-based methods in the left heatmap in Fig. 1.

Our work aims to fundamentally understand the most prominent methods for feature attribution in neural networks while uncovering their limitations and resulting misinterpretations caused by visualizations and the choice of the baseline value. For example, a relevant feature can become less or even irrelevant for another baseline. To achieve this, we categorize these methods into four groups based on their underlying explanation target and analyze their distributional





**Fig. 1.** Pairwise comparison of state-of-the-art feature attribution methods (see Sect. 2 for the method descriptions) on the COMPAS dataset using (left) the mean feature-correlation, (middle) mean instance-wise Kendall rank correlation, and mean rank agreement of the two most important features (details can be found in Appendix A.1).

behavior. We demonstrate situations where certain methods fail to deliver adequate attributions through simulation studies employing a known data generation process for tabular data. Simultaneously, we show that the method’s feature-wise explanations are often strongly correlated even while failing. Furthermore, we investigate the impacts of various common data preprocessing techniques for continuous and categorical features on the explanation methods. Using the simulations, we clarify whether a feature attribution method can correctly attribute relevance to the prediction on a local level and whether rank-based aggregated relevance can be used as a global feature importance measure.

## 2 Background and Related Work

**Preliminaries.** To contextualize the topic of our work within the multitude of interpretation methods for machine learning models, we rely on the taxonomic classifications proposed by Doshi-Valez and Kim [20] and, specifically for neural networks, the framework provided by Zhang et al. [61]. Accordingly, feature attribution methods are considered (semi-)local post-hoc techniques, as they require a trained model (i.e., all parameters and internal structures remain unchanged) and explain the prediction of a single instance, e.g., an image or a patient. The term *feature attribution* originates from the fact that these methods assign relevance or contribution scores on a prediction to each feature. This means that for a model  $f$  and an instance  $\mathbf{x} \in \mathbb{R}^p$ , a feature attribution method results in a vector  $\mathbf{r} = (r_1, \dots, r_p)^T \in \mathbb{R}^p$  of feature-wise relevances. Ideally, they should provide a decomposition of the prediction (or a proportional objective) into individual feature-wise effects, also referred to in the literature as local accuracy [37], completeness [40], or summation-to-delta property [47], i.e., with  $r_0 \in \mathbb{R}$

$$f(\mathbf{x}) = r_0 + \sum_{i=1}^p r_i. \tag{1}$$

Even if not all methods fulfill this property exactly, a reasonable feature attribution method should comply with this fundamental principle or strive to approximate it. In the case of a linear model – which, in principle, is a neural network with only one dense layer and linear activation – the relevances for an ideal method should be proportional to the product of the regression coefficients and the feature values, i.e.,  $\beta_i x_i$ . In this notation, Shapley values provide proportional explanations, as they estimate the local effect against the constant marginal effect  $\beta_i x_i - \beta_i \mathbb{E}[X_i]$ .

**Methods.** This paper focuses on feature attribution methods specifically designed for neural networks, i.e., model-specific methods, which allow an application to image and tabular data. The pioneering work in this field is the gradient method (Grad) introduced by Simonyan et al. [49], which computes the feature-wise partial derivatives from the output to the respective input features. Originally applied to image data, the sum or maximum of absolute values across the color channels are calculated and became famous as saliency maps (Saliency). Subsequently, further methods emerged specifically for convolutional neural networks and ReLU networks [45, 51, 59]. These variations primarily differ in their computation of gradients within activations or their incorporation of the gradients to bias terms [52]. However, Smilkov et al. [50] critique the visually noisy nature of saliency maps, leading to the development of SmoothGrad (SG). This approach estimates the average gradient of Gaussian-disturbed inputs, resulting in a sharper appearance of the saliency map. Alternatively, Adeboye et al. [1] proposed computing the variance instead of the average. Notably, these methods predominantly rely on the plain gradients, which, from a mathematical perspective, do not provide a direct decomposition but rather highlight the features’ output sensitivity.

The first approach toward approximating the decomposition of the prediction  $f(\mathbf{x})$  was introduced through the backpropagation-based method layer-wise relevance propagation (LRP) by Bach et al. [9]. LRP starts its process from the output prediction, systematically redistributing relevances layer by layer to the lower layers using predefined rules until reaching the input layer. Shrikumar et al. [47] further advanced this concept with their deep learning important features (DeepLIFT) method, incorporating a reference value  $\tilde{\mathbf{x}}$  (also called the baseline value) to achieve a decomposition of  $f(\mathbf{x}) - f(\tilde{\mathbf{x}})$ . Integrated gradient (IntGrad) [54], sharing the same objective of decomposition as DeepLIFT, integrates the gradients along a path from  $\mathbf{x}$  to the reference value  $\tilde{\mathbf{x}}$ . More recent techniques such as DeepSHAP and expected gradient (ExpGrad) [15, 37] – also known as GradSHAP – employ multiple reference values for an explanation, bridging to Shapley values [46] by aiming for the decomposition the prediction regarding the expected prediction  $f(\mathbf{x}) - \mathbb{E}[f(X)]$ .

**Evaluation Metrics.** Evaluation metrics for feature attribution methods in current XAI research are subject to a broad debate. Doshi-Velez and Kim [20] categorize these metrics into human-grounded and function-grounded.

The former group assesses the explanation quality based on human judgments, measuring the overall comprehensibility of non-experts. On the other hand, function-grounded metrics consist of mathematically defined criteria that can be measured without human interactions. Within this group of evaluation metrics, a prominent subgroup focuses on verifying faithfulness [4, 58]. These metrics measure the extent to which highly relevant features based on the explanation method also crucially influence the model’s predictive power. They usually measure the loss change [40] or other correlations between the prediction drop and the explanations [5, 58] when the most or least relevant features are removed. Usually, the most or least relevant features are determined based on the explanations’ magnitude ignoring the sign, which is consistent with the rankings of the absolute values. In this context, ‘removing a feature’ mostly means simulating its absence, e.g., by setting it to zero or another baseline, conditional sampling [22], or retraining the entire model without this feature [26]. Although all these evaluation metrics justify the explanation method’s ability to detect highly decisive features, researchers have found that they are inconsistent and seem to measure different aspects [23, 56]. This inconsistency has recently become known as the disagreement problem [34, 41].

In addition to axiomatic approaches, which verify whether methods adhere to properties [28, 54], e.g., local accuracy from Eq. 1, our work deals with ground-truth evaluations. These evaluation methods assess the explanation’s quality on synthetic datasets or injected ground-truth elements, i.e., semi-natural datasets. However, current literature primarily focuses on so-called pointing games in image data [60], such as synthetic datasets like CLEVER-XAI [7] or overlaid images of prediction-relevant and irrelevant parts [30, 55, 57, 62].

There are comparatively fewer ground-truth analyses of model-specific feature attribution methods for tabular data. In Chen et al. [16], four data-generating processes with continuous Gaussian features are created, where not all features contribute to the prediction. They then compare the median rank of informative features regarding different feature attribution methods. Similarly, in Agarwal et al. [3], known methods are compared on synthetic and real-world data on a rank level with ground-truth values. However, there are also analyses of model-agnostic feature attribution methods that are more similar to our approach. For example, Guidotti et al. [24] create random data-generating processes (DGP) from simple transformations or feature distributions and assess the similarity of methods with the analytical gradients. However, we argue that gradients are not suitable ground-truth values for methods targeting an output decomposition. Further, Liu et al. [36] use additive models from simple feature transformations as DGPs but use Shapley values as ground truth instead. Carmichael et al. [14] propose a framework for comparing the explanations of model-agnostic feature attribution methods with the actual effects of additive structured DGPs.

### 3 Understanding the Explanation’s Distribution

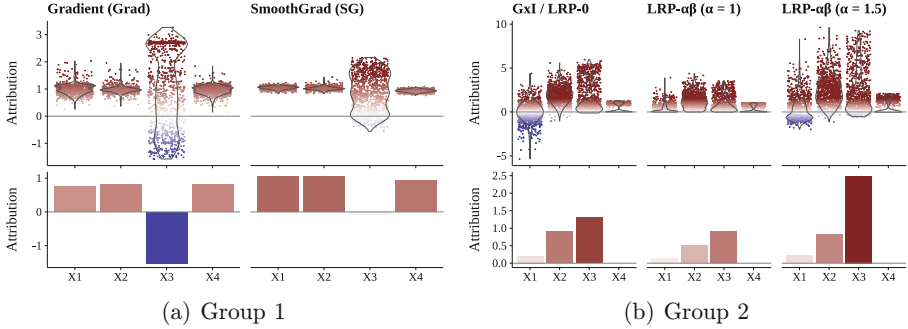
To adequately compare feature attribution methods, it is essential to understand their underlying principles before applying them in experiments or benchmark studies. A crucial distinction among these methods lies in how they quantify the effects or influence of features, which can significantly impact visual representations, such as heatmaps or bar plots, and feature rankings, potentially leading to misinterpretations. To illustrate the nuances and a statistical perspective of the state-of-the-art techniques, we consider the following data-generating process (DGP) describing a regression problem:

$$Y = X_1 + X_2 + X_3^2 + X_4 + \varepsilon \quad (2)$$

where  $X_1 \sim \mathcal{N}(0, 1)$ ,  $X_2 \sim \mathcal{N}(2, 1)$ ,  $X_3 \sim \mathcal{U}(-1, 2)$ ,  $X_4 \sim \text{Bern}(0.4)$ , including Gaussian noise  $\varepsilon \sim \mathcal{N}(0, 1)$ . In this setting, we generally expect a feature attribution method to generate normally distributed relevances for  $X_1$  and  $X_2$ . Similarly, for  $X_3$ , we expect mostly low with progressively fewer larger values, and for  $X_4$ , a strongly bimodal distribution. In the following, we group the most well-known feature attribution methods according to their similarities and analyze their distributional and individual behavior using a neural network with ReLU activation trained on  $n = 2,000$  instances.

**Prediction-Sensitive Methods (Group 1).** The first group consists of methods relying on plain gradients, such as the gradient (Grad) [49] method and its variant SmoothGrad (SG) [50]. Due to their mathematical definition, both methods calculate the output sensitivity of the features, causing them to be unsuitable as local attribution methods for individual effects. For instance, in Fig. 2a, it can be seen that both methods in our regression example consistently assign a relevance of closely one for the linear effects of  $X_1$ ,  $X_2$ , and  $X_4$ , and almost uniformly distributed relevances for the quadratic effect of  $X_3$ . Although SmoothGrad visibly reduces the variance of the Grad method, both methods fail to provide appropriate values for the local effects on the prediction, instead indicating the model’s sensitivity to changes in the variables. Despite not being further examined in this paper, other plain gradient-based methods like VarGrad [1], FullGrad [52], and GuidedBackprop [51] fall in this group.

**Fixed-Reference Methods (Group 2).** In the second group, we encompass methods that strive to approximate a decomposition of the prediction  $f(\mathbf{x})$  into feature-wise effects. These techniques mainly rely on a first-order Taylor approximation, where the reference point  $\tilde{\mathbf{x}}$  is implicitly fixed. Consequently, the quality of the approximation can significantly depend on the proximity between  $\mathbf{x}$  and  $\tilde{\mathbf{x}}$ . The most straightforward variant, Gradient×Input (GxI) [48], directly extends the Grad method by calculating the Hadamard product of the gradient and the corresponding input features. In a similar way as SG extends the Grad method, GxI can also be extended to SmoothGrad×Input (SGxI), which

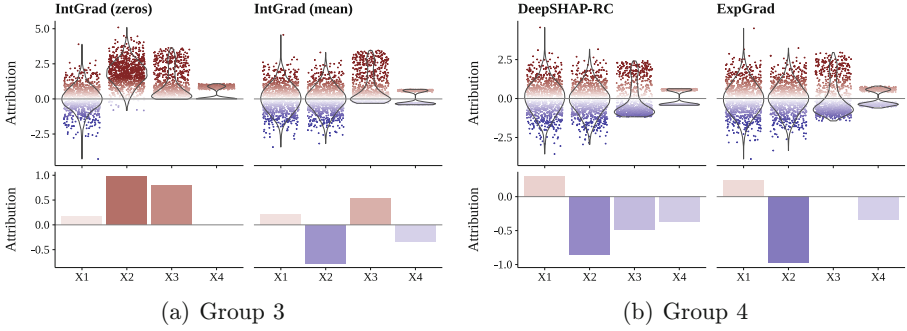


**Fig. 2.** Resulting attribution values of (a) prediction-sensitive and (b) fixed-reference methods of 1,000 test instances based on the DGP in Eq. 2. The distribution is shown as a violin plot at the top and a bar plot of the same single instance at the bottom.

inherits its smoothing effects. In contrast, the backpropagation-based method layer-wise relevance propagation (LRP) [9] represents a sequence of first-order Taylor approximations applied in each individual layer [40]. This involves a layer-wise redistribution of relevances from the upper to lower layers of the model until the input features are reached. Various rules have been proposed for this relevance redistribution, which can also be selected based on the layer type [33]. In addition to the initial LRP-0 rule, there is the LRP- $\varepsilon$  rule, which incorporates a relevance-absorbing stabilizer  $\varepsilon > 0$ , and the LRP- $\alpha\beta$  rule, which assigns different weights to positive and negative relevances using  $\alpha, \beta \in \mathbb{R}$  with  $1 = \alpha + \beta$ . Setting the  $\alpha$  parameter to one (i.e., retaining only positive relevances) and considering solely positive input values makes LRP- $\alpha\beta$  equivalent to the deep Taylor decomposition (DTD) method [40]. For a mathematical description and a more comprehensive overview, we refer to the work by Montavon et al. [39]. Despite their differing calculation approaches, GxI and LRP-0 implicitly use a reference value of  $\mathbf{0}$  for the Taylor approximations. Moreover, Ancona et al. [5] demonstrated that LRP-0 and GxI produce identical results for ReLU networks. Furthermore, the LRP- $\alpha\beta$  method and DTD can be interpreted as employing a root point as the reference value, i.e.,  $f(\tilde{x}) = 0$  [40]. Returning to our regression problem, we observe that GxI precisely yields the expected distribution of effects (see Fig. 2b) regarding a zero baseline. Since we exclusively utilized ReLU activations for the model, this method is equivalent to LRP-0. Conversely, with the LRP- $\alpha\beta$  method, one can observe varying weights’ influence on positive and negative relevances. When propagating purely positive relevances, all negative relevance values represented in the GxI are truncated to zero. On the other hand, one can see the stretching of positive relevance and compression of negative relevance with a positive-favored propagation in LRP- $\alpha\beta$ . At this point, it becomes apparent how different the whole explanation’s distribution and the visualizations for individual instances can be, as exemplified by features  $X_1$  and  $X_3$  in Fig. 2a compared to Fig. 2b.

**Reference-Based Methods (Group 3).** The main distinction between the previously discussed methods and our third group is how the local effects are measured. Whereas the former assesses effects with respect to zero or another implicitly defined baseline, the latter group attributes the relative effect of features  $\mathbf{x}$  in relation to an arbitrarily chosen reference value  $\tilde{\mathbf{x}}$  acting as a hyperparameter. For instance, feature  $X_2$  theoretically returns an effect of 2 on average in the GxI method assuming a perfect model fit, which is a valid explanation based on our data-generating process in Eq. 2. However, the third group attributes the relative effect of  $\mathbf{x}$  with respect to the baseline value  $\tilde{\mathbf{x}}$ , e.g., the average feature value. They aim to decompose the output differences  $f(\mathbf{x}) - f(\tilde{\mathbf{x}})$  and consequently answer the question of how significant the feature’s contribution is compared to the contribution of the reference value. The most prominent methods following this underlying characteristic are integrated gradient (IntGrad) [54] and deep learning important features (DeepLIFT) [47]. The former integrates the gradients along a path from  $\mathbf{x}$  to the reference value  $\tilde{\mathbf{x}}$ . While this integral is discretized and approximated in applications, an exact decomposition of the difference is asymptotically guaranteed for models that are differentiable almost everywhere. The DeepLIFT method achieves this decompositional target by incorporating the respective layer’s intermediate reference value in the layer-wise backpropagation scheme, akin to LRP-0. Additionally, the authors propose two rules for propagating through activation functions: the rescale (-RE) and reveal-cancel (-RC) rules. Ancona et al. [5] also demonstrated that DeepLIFT-RE using a zero-baseline is equivalent to the computationally efficient GxI method in neural networks using only ReLU activations and with zero bias vectors. Furthermore, empirical evidence from the authors and other researchers indicates a remarkably similar behavior of IntGrad and DeepLIFT-RE in practice [5, 47]. In our regression example, we similarly observe this phenomenon and, hence, present only the results of the IntGrad method for the reference value of zeros (left) and of the feature-wise empirical means (right) in Fig. 3a. With a zero baseline, the similarity to the GxI method in Fig. 2b is clearly visible. However, when the reference value is set to the mean feature values, changes in features  $X_2$ ,  $X_3$ , and  $X_4$  become apparent due to the negative shift. Particularly evident in the second variable is the influence of the reference value, addressing different questions in our regression setting. With  $\tilde{\mathbf{x}} = 0$ , the contribution of the variable to the prediction is determined, whereas with  $\tilde{\mathbf{x}}$  set to the average feature value, the effect is attributed relative to the effect of  $\tilde{\mathbf{x}}$  (see Fig. 3 bottom variable  $X_2$ ). This shows, in particular, that features with a high assigned magnitude can generally be less relevant or even irrelevant for another baseline.

**Shapley-based Methods (Group 4).** The final group consists of methods based on the game-theoretic Shapley values [46] adapted for feature attribution in machine learning. They aim to quantify the contribution of each feature to the change in the model prediction concerning the average prediction. Shapley values are computationally expensive to calculate due to the consideration of all possible feature combinations, particularly with high-dimensional data. Therefore,



**Fig. 3.** Resulting attribution values of (a) reference-based and (b) Shapley-based methods of 1,000 test instances based on the DGP in Eq. 2. The distribution is shown as a violin plot at the top and a bar plot of the same single instance at the bottom.

DeepSHAP and expected gradient (ExpGrad) [15, 37], also known as GradSHAP, have been developed for neural networks to approximate these values efficiently. These methods build upon previously explained techniques like DeepLIFT and IntGrad, providing a feature-wise decomposition of predictions compared to the average model prediction. Thus, they represent the feature effect compared to the estimated marginal effect of the feature, i.e., they incorporate the whole feature distribution. These methods are computed by averaging the DeepLIFT or IntGrad results across various reference values. The reference values should originate from the same distribution as the training data, thereby addressing the out-of-distribution problem associated with dataset-independent choices in the plain methods. For example, the mean feature values don't necessarily follow the data distribution. However, this approach entails higher computational costs due to the evaluation of the representative sample of baseline values. Especially for ExpGrad, which involves both aggregating over samples and approximating integrals, Erion et al. [21] proposed a purely sample-based estimation approach. In Fig. 3b of our running example, only minor differences between the two methods are apparent. However, compared to all other methods, it is clearly evident that the explanation distributions are centered around zero, thus consistently attributing the effect relative to the marginal effect.

### 4 Do Feature Attribution Methods Attribute?

In the preceding section, we demonstrated the varying behavior of the state-of-the-art feature attribution methods, particularly in scenarios where quadratic effects are present in the data-generating process (DGP) or where the feature distribution is not mean-centered. However, we observed that, at least in simple cases, the explanations' distributions are often proportional. This implies that, while the mean and scale of the distribution may differ due to different baselines, the relative distances of explanations should remain consistent across

methods. To assess this fundamental behavior of the methods for different types and strengths of effects, we employ an additive data-generating process in the following simulations with only numerical or categorical inputs separately:

$$Y = \beta_0 + g(X_1)\beta_1 + \dots + g(X_p)\beta_p + \varepsilon. \quad (3)$$

In this DGP, the function  $g : \mathbb{R} \rightarrow \mathbb{R}$  determines the type of effect (e.g.,  $g(x) = x$  for linear or  $g(x) = x^2$  for quadratic effects), and the coefficients  $\beta_1, \dots, \beta_p \in \mathbb{R}$  control the feature-individual (global) strength of the effect. Additionally, we add a standard normal distributed error term  $\varepsilon \sim \mathcal{N}(0, 1)$ . Numerical data is sampled from a normal distribution with uniformly sampled mean and variance and then transformed with a linear, piece-wise linear, and non-continuous function  $g$ . We use equidistant effects from  $-1$  to  $1$  for the levels of categorical variables. After the data generation, neural networks are trained on this data, and the corresponding feature attribution methods are applied to the test data. See Appendix A.2 for more simulation details. The efficacy of a method is evaluated by computing the Pearson correlation between the feature-wise effects  $g(x_j^{(1)})\beta_j, \dots, g(x_j^{(n)})\beta_j$  and the corresponding generated explanations across the test dataset of 1,000 samples. Consequently, we measure the explanation’s distributional fidelity to the shape of the ground-truth effects, still allowing potential linear transformations within the distributions.

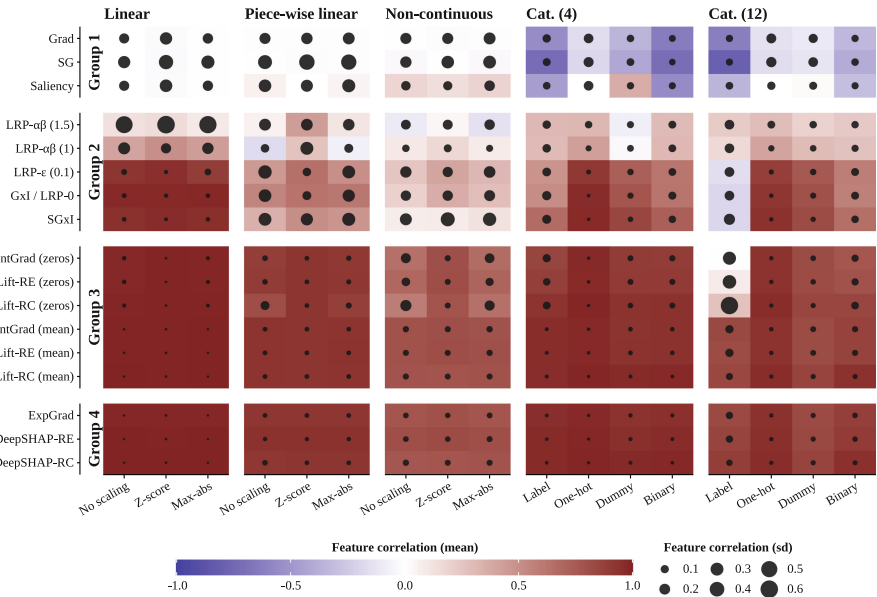
#### 4.1 Impact of Data Preprocessing

For image data, it is already known that not all feature attribution methods are invariant to constant shifts in the input data [31]. In particular, for reference-based methods, the question of a suitable baseline has arisen in recent years [25]. However, how the data is preprocessed is closely linked to this question. As seen in Sect. 3, for mean-centered variables, many methods coincide, or shifted variables can be transformed back with the appropriate baselines. Furthermore, to our knowledge, no one has yet analyzed the influence of different encoding techniques for categorical variables on the quality of explanation.

In our simulation setup for continuous variables, we consider three common scaling methods: none, z-score, and max-abs scaling. Without scaling, the variables are passed unprocessed to the neural network. In contrast, with the z-score method, the empirical mean is subtracted and then divided by the empirical standard deviation for each variable, resulting in zero-centered and unit-variance data. The min-max preprocessing is particularly well-known for image data, as pixel values are naturally bounded. Nevertheless, this scaling technique is also used for other continuous input data, restricting the data to values between  $-1$  and  $1$  by scaling with the feature-wise maximum absolute values. In particular, all these scaling methods are invariant to the correlation metric as they describe linear transformations. The scaling parameters are calculated based on the training data and then also used for the test data. For categorical variables, we employ the following encoding techniques: label, one-hot, dummy, and binary encoding. With label encoding, the levels are naively mapped to integer numbers, inducing



a particular non-existent order of values. One-hot encoding creates a zero-or-one column for each of the  $C$  categories, while dummy encoding transforms a category into  $C - 1$  columns to avoid redundancy. On the other hand, binary encoding represents each category as a binary number, and then each digit is transformed into a column. This results in a relatively low number of columns, especially for a high number of levels, i.e., only  $\lfloor \frac{\log(C)}{\log(2)} + 1 \rfloor$  instead of  $C$ . For our DGP from Eq. 3, we use  $p = 12$  variables with identical effect sizes for the two cases of all continuous and all categorical variables, allowing correlations to be unbiasedly aggregated across all variables. For the categorical variables, we consider both a small number of levels (4) and a high number of levels (12). Each setting is repeated 200 times, and then the mean correlation across the repetitions and features is calculated and summarized in Fig. 4. In addition, we calculate the standard deviation, which is shown next to the aggregated correlation as a size-varying dot.



**Fig. 4.** Results of the preprocess simulations for state-of-the-art feature attribution methods (y-axis) showing the averaged correlation with the ground-truth effects across 200 repetitions and  $p = 12$  features with equal effect strengths. The individual columns represent different types of effects, and the x-axis shows various preprocessing functions. The size-varying dot describes the standard deviation of the aggregation.

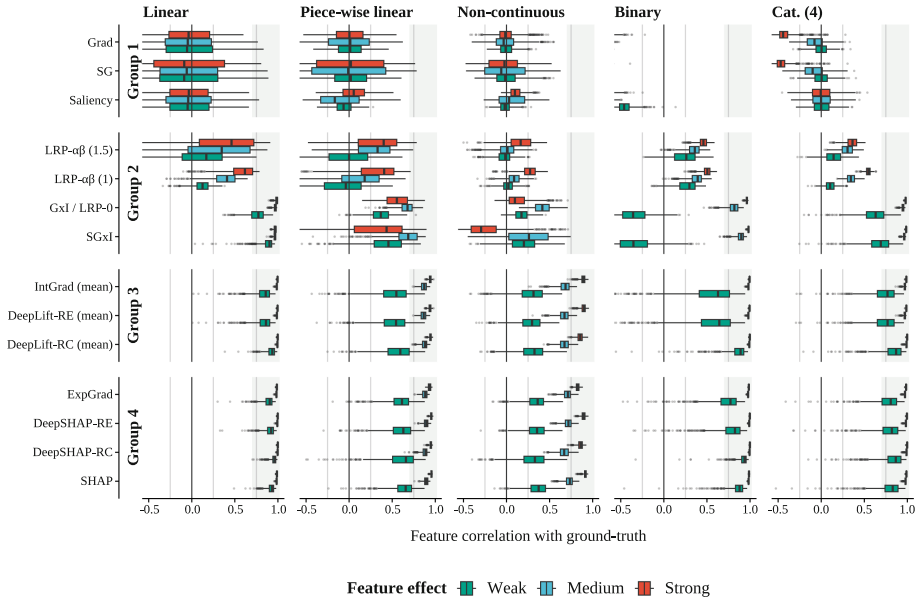
Generally, the results from Fig. 4 need to be interpreted column-wise since each column describes either a different DGP or a different model performance caused by another preprocessing function. However, the models performed notably similarly for continuous variables in the individual effect groups and

within a small error range for the categorical variables (see Table 1 and 2 in the Appendix). As expected, the prediction-sensitive methods (Grad, SG, and Saliency) are unable to correctly assign proportional local effects for any of the considered effect types or even result in misinterpretations due to negative correlations for categorical variables (see rows of Group 1 methods in Fig. 4). In addition, the larger dot describing the standard deviation across the simulation's repetitions and features shows an inconsistent attribution and, thus, a strong model dependency. Similarly, the uneven weighting of positive and negative relevances in LRP- $\alpha\beta$  proves to be counterproductive or even resembles guessing for non-linear effects or dummy-encoded variables. The remaining Group 2 methods (LRP-0/GxI, LRP- $\varepsilon$ , and SGxI) show only slight deviations from Group 3 and 4 methods concerning linear effects and categorical variables. However, their performance significantly worsens and destabilizes when confronted with non-linear effects. For the zero-baseline methods, no scaling and max-abs scaling increasingly degrade the quality and stability with the complexity of the effect types. This phenomenon is also the case for label-encoded categorical variables. This could be mainly due to the fact that zero is no longer a distribution-neutral baseline value for max-abs scaling or label encoding and thus reduces the quality of the Taylor approximation. However, this can be adjusted by using the feature-wise mean value as reference (see lower rows for Group 3 methods in Fig. 4). Otherwise, there is hardly any difference between the methods in Group 3 with empirical means as reference values, and Shapley-based methods (Group 4), which consistently show high correlations with the ground truth. Nevertheless, the overall results show that, with the exception of the prediction-sensitive methods and LRP- $\alpha\beta$ , the methods mostly agree using the zero-centered scaling and common encoding techniques.

## 4.2 Faithfulness of Effects

Which method to choose is strongly debated in the literature, with disagreement regarding which method provides adequate explanations and which does not. Commonly, the most influential features are removed, and the impact on prediction is evaluated. However, as seen in Sect. 3, the strength measured as the absolute magnitude of a variable varies and is baseline-dependent. Nevertheless, by adjusting  $\beta_1, \dots, \beta_p$ , we want to investigate how the methods behave with different effect sizes. Since weak effects are also more difficult for the model to capture due to a small signal-to-noise ratio, we expect that a good method also detects them less accurately. Similarly to the setting in Sect. 4.1, we simulate  $p = 12$  normally distributed, binary, and categorical variables to evaluate the performance of feature attribution methods concerning different effect sizes. In this scenario of the DGP from Eq. 3, grouped variables are considered, where  $\beta_1, \dots, \beta_4 = 0.1$  for weak,  $\beta_5, \dots, \beta_8 = 0.4$  for medium, and  $\beta_9, \dots, \beta_{12} = 1$  for strong effects. Furthermore, we employ z-score scaling for continuous variables, label encoding for binary, and one-hot encoding for categorical variables with four levels. The results of this simulation, conducted over 200 trained neural

networks and  $n = 2,000$ , are illustrated in Fig. 5. For a comparison with a state-of-the-art model-agnostic method, we also execute the well-established SHAP method [37] on the test data, which approximates Shapley values.



**Fig. 5.** Results of the faithfulness simulations for state-of-the-art feature attribution methods (y-axis) showing the correlation with the ground-truth effects across 200 repetitions and  $p = 12$  features with grouped (weak, medium, strong) effect strengths as box plots. The individual columns represent different types of effects, and the x-axis shows the correlation.

The results for the prediction-sensitive methods are similar to those for the preprocessing simulation: the continuous variables tend to have a strongly varying correlation around zero for all effect strengths and types, which represents a guessing of the local contributions to the prediction. The strong negative correlation for binary variables is probably due to unusual inter- or extrapolations of the model around the values 0 and 1. For the Group 2 methods,  $LRP-\alpha/\beta$  moderately captures the ground-truth attributions for linear data but increasingly fails to provide reliable explanations for more complex relationships. The method GxI (and thus LRP-0) appears to provide relevances quite similar to ground truth for (partially) linear effects. However, it only yields moderate correlations for non-linear effects and struggles to recognize the association with strong effect sizes. This inconsistency likely stems from the method’s reliance on the first-order Taylor approximation, causing it to falter in non-linear relationships. Interestingly, the GxI and SGxI methods inherit the negative correlation from the gradients for binary variables with weak effects, but this is corrected

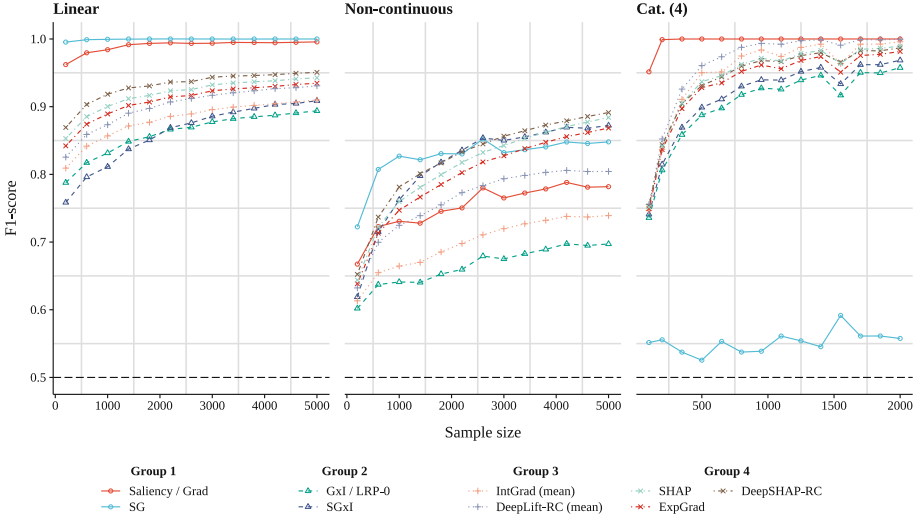
for one-hot encoded variables. This could presumably be because the Taylor approximation collapses when zero is used as both the reference and explained value. As for reference-based and Shapley-based methods, it becomes evident that these methods accurately identify the actual effects across all considered settings. However, Shapley-based methods tend to attribute weak or moderate effects better than all other methods. Furthermore, DeepLIFT-RC and its extension DeepSHAP-RC demonstrate an outstandingly strong correlation with the ground-truth effects, especially for binary and categorical variables. The comparison with the Shapley values from the model-agnostic method SHAP further indicates that the Group 3 and 4 methods align with this popular XAI method and provide a fast and accurate approximation of Shapley values for neural networks.

### 4.3 Beyond Feature Attribution Toward Importance

The previous simulations have shown to what extent feature attribution methods are able to attribute the exact proportions of a variable’s contribution to the prediction, i.e., how adequate the explanation’s distribution aligns with the ground-truth effects. As shown in Sect. 3, it can now occur that a feature reflecting the mean value correctly, receives a relevance score close to zero if the effect is measured to a zero baseline, even though it is a very important feature from a global perspective. This is comparable to a Gaussian-distributed variable in a linear model with a high regression coefficient: although the global importance of this variable is very high, the attributed relevance for feature values close to zero can vanish. From a local perspective, this means that instance-wise feature rankings based on the magnitude of relevance are not suitable for answering whether one feature is more important than another. Instead, the feature attribution methods do what they are developed for and give the local effect to the prediction relative to a method-dependent baseline. However, all methods should follow the property of *consistency* and assign little to no relevance to unimportant features, while tending to assign higher relevances to crucial features across the dataset. Thus, ranked relevances are more appropriate to answer whether or not a feature is important.

To evaluate the effectiveness of the methods in identifying important and unimportant features, we simulate  $p = 20$  normally distributed and categorical variables with 4 levels using the DGP from Eq. 2, where only the even feature indices affect the regression outcome. For each instance from the test data, the ranking of absolute explanations is converted into the binary decision of whether the feature belongs to the top 10 most important ones. Since the ground-truth importance values are 1 for even and 0 for odd indices, we compute the F1-score for each instance and then average the scores over the 1,000 test instances. Additionally, we vary the sample size  $n$  of the training data to assess the question of detecting important/unimportant features at different model qualities.

Although this group of methods performed poorly in the previous simulations, Fig. 6 demonstrates the strength of prediction-sensitive methods. They



**Fig. 6.** The figure shows the F1-score on the y-axis depending on the sample size for various feature attribution methods (colors) and effect types (columns) averaged over 500 repetitions.

are capable of nearly perfect discrimination between important and unimportant features across all model qualities for linear effects and categorical variables. Except for SG for categorical variables, which probably smooths the gradients too much due to the high variance in perturbations. Even with limited training data, Grad and especially SG perform exceptionally well for non-linear effects. Aside from Group 1 methods, it is apparent that Shapley-based methods, particularly DeepSHAP-RC, are the most reliable in distinguishing important and unimportant features. They even outperform Grad and SG when dealing with non-linear effects and large sample sizes. Despite the Group 3 methods with the feature-mean reference performing nearly identically to the Group 4 methods in the previous simulations (see Sect. 4.2), the single-reference methods show significant weaknesses in discriminating important and unimportant features. This is either due to disrupted rankings resulting from shifted relevances (as observed in Sect. 3) or because Shapley-based methods attribute low relevances to unimportant features more accurately due to the incorporation of multiple baselines. Furthermore, it is again evident that the methods GxI and LRP-0 encounter significant issues with non-linear effects and generally perform worse. Moreover, the Shapley-based model-specific feature attribution methods are able to outperform the established model-agnostic method SHAP.

## 5 Discussion

We have demonstrated through simulations how differently state-of-the-art feature attribution methods measure local effects on a prediction and how visual-

izations could be manipulated in favor of chosen features using baseline values (see Sect. 3), which can distort the quality of rank-based measures for determining important features. This confirms previous research that has already demonstrated this influence on heatmaps for reference-based methods in applications [19, 38, 53]. Closely connected to the choice of baseline values, we – to the best of our knowledge – for the first time analyzed the impact of common data preprocessing steps on the quality of attribution for continuous and categorical variables. We found that, except for prediction-sensitive methods, standard normalization techniques such as z-score scaling for continuous and one-hot encoding for categorical variables provide the most accurate attributions of local effects. However, fixed-reference methods such as LRP-0, GxI, and LRP- $\alpha\beta$  struggle noticeably with non-linear effects. Nevertheless, our simulations have shown that an appropriate baseline can adjust for non-zero-centered scaling techniques in reference-based methods.

Furthermore, in Sect. 4.2, we observed that the distributions of explanations provided by reference-based and Shapley-based methods exhibit remarkable similarities for standard preprocessing techniques. These explanations correlate strongly with ground-truth effects as the effect size increases, with Shapley-based methods generally tending to be more accurate. These simulations demonstrate that reference-based, Shapley-based, and fixed-reference-based methods (except for linear transformations) yield proportional explanations’ distributions, contradicting the disagreement problem [34, 41]. Hence, the disagreement is mainly caused by different baselines. However, our findings underscore that feature attribution methods, notably influenced by the choice of baseline, pursue distinct objectives in attribution, consequently resulting in varying local magnitudes of relevances. This fact became particularly evident in Sect. 4.3, where the methods exhibited noticeable differences in discriminating between important and unimportant features based on ranked magnitudes of the explanations. While prediction-sensitive methods may falter in correctly attributing relevances, they performed well in determining whether or not a feature is important. Nevertheless, Shapley-based methods, especially DeepLIFT-RC, appear to consistently excel in addressing this binary classification problem, as also observed by other researchers [62].

While our simulations are based on simple synthetic data without correlated features and interaction effects and we only trained dense neural networks, they represent the first independent comparison of feature attribution methods on tabular data considering the attributions’ correlation. Additionally, dense layers serve as a fundamental building block for many modern deep neural networks, such as convolutional neural networks or attention modules, providing insights into their behavior. Furthermore, the restriction to regression problems is negligible, as most feature attribution methods ignore the activation of the final layer, which computes class probabilities, and instead apply the method to the preactivation values [9, 47, 49]. Investigating the methods for interaction effects remains an attractive direction for future work, building upon the theoretical groundwork already explored by Deng et al. [18].

## 6 Conclusion

Our study provides a fundamental understanding of state-of-the-art feature attribution methods for neural networks through simulation studies. Initially, we demonstrated the variability in the explanation’s distribution and individual explanations across different methods. Particularly, we highlighted how crucial the implicitly or explicitly set reference value can influence the magnitude of a feature’s relevance and, thus, the ranking regarding the importance of a feature on a local level. Furthermore, we illustrated how preprocessing techniques of training data can affect and destabilize the quality of attribution and can only be corrected in reference-based or Shapley-based methods through appropriate baseline values.

Nevertheless, we have shown that most state-of-the-art methods, when utilizing z-score scaling for continuous variables and one-hot encoding for categorical variables, deliver relevances that closely correlate with the ground-truth values as the effect strength increases. However, this comparison does not consider linear transformations of the distributions, leading to the methods’ disagreement when transitioning to rank-aggregated values. Additionally, we have demonstrated that plain gradient methods, such as the gradient (Grad) and SmoothGrad (SG), are not suitable as attribution methods for effect decompositions while being highly capable of distinguishing important and unimportant features on a global scale.

**Acknowledgments.** This project was funded by the German Research Foundation (DFG), Emmy Noether Grant 437611051.

**Disclosure of Interests.** The authors have no conflict of interests to declare.

## A Appendix

All figures and simulation results presented in this work are reproducible using the code hosted on our GitHub repository, available at [https://github.com/bips-hb/Toward\\_Understanding\\_Disagreement\\_Problem](https://github.com/bips-hb/Toward_Understanding_Disagreement_Problem).

### A.1 COMPAS Dataset

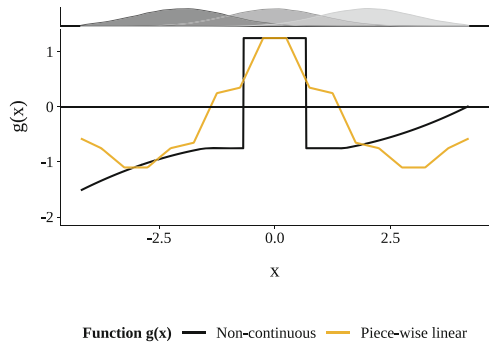
For this example, we load the COMPAS dataset from the R package `mlr3fairness`<sup>1</sup> and train it with respect to the variable `two_year_recid`. We use a neural network model with four layers: 256, 128, and 64 neurons in the hidden layers, along with ReLU activations. Additionally, a dropout layer is added after each hidden layer for regularization. Continuous variables are preprocessed using z-scores, and categorical variables are one-hot encoded. Using an 80/20 train-test split, we achieve an F1-score of 74.36%. The training procedure is the same as in the simulations described below.

<sup>1</sup> <https://mlr3fairness.ml-org.com/reference/compas.html>.

## A.2 Simulation Details

**Data Generation.** An additive model with independent variables is used for all simulations, following the data-generating process from Eq. 3. The continuous variable  $X_i$  is sampled from a normal distribution  $\mathcal{N}(\mu_i, \sigma_i)$ , where  $\mu_i$  and  $\sigma_i$  are uniformly distributed within the range  $[-2, 2]$  for the mean and  $[0.9, 1.1]$  for the variance. This approach allows us to simulate variations in the mean and scale of the Gaussian distributions. For a categorical variable  $X_i$  with  $c \in \mathbb{N}$  levels  $A_1, \dots, A_c$  with equal level probabilities, equidistant effects ranging from  $-1$  to  $1$  are assigned, i.e.,  $g(X_i = A_k) = -1 + \frac{2(k-1)}{c-1}$ . This zero-centered distribution of effects across categories ensures that only the coefficient  $\beta_i$  controls the effect strength so that the level-specific effects do not disturb it.

In order to simulate different types of effects for continuous variables also covering the mean shifts, we consider the linear function  $g(x) = x$  and the transformations  $g : \mathbb{R} \rightarrow \mathbb{R}$  described in Fig. 7. Unless otherwise stated, 4,000 training instances are generated for continuous and 2,000 for categorical/binary variables, with one-third of each generated as evaluation data for the neural network training. The smaller number of samples for the categorical and binary variables is due to their simpler and discrete relationships with the regression outcome. Additionally, we use 1,000 instances as test data for the feature attribution methods.



**Fig. 7.** Transformations  $g$  used for non-linear relationships of the variables and the regression outcome.

**Neural Network Training.** The neural networks consist of three dense layers with 256, 128, and 64 neurons for continuous variables and, due to simpler relationships, 128, 64, and 32 neurons for categorical variables. ReLU is always used as the activation function, and a dropout layer with a dropout rate of 0.4 is added after the activation in the hidden layers. Each network is trained for a maximum of 300 epochs using the Adam optimizer, where the initial learning rate of 0.01 is multiplied by 0.2 every 50 epochs, and the training terminates after 50 unimproved epochs on the evaluation data.

**Hyperparameters of Feature Attribution Methods.** We apply the feature attribution methods using the R package `innsight` [32] on the generated test data and the trained model. No hyperparameters are needed for the methods,



Grad, Saliency, and GxI; only the corresponding rules and baselines are required for LRP, DeepLIFT, and DeepSHAP. For SmoothGrad, we take 50 samples with a noise level of 0.2. Similarly, we use 50 samples for IntGrad, ExpGrad, and SHAP methods. Since methods for encoding categorical variables, except for label encoding, generate new artificial variables, we subsequently summed them up, ensuring that each categorical feature is assigned only one relevance.

### A.3 Model Performance

Since methods for encoding categorical variables, except for label encoding, generate new artificial variables, we subsequently summed them up, ensuring that each categorical feature is assigned only one relevance value (see Table 1 and 2).

**Table 1.** Results of the neural network performance on test data consisting of continuous variables measured by the average  $R^2$  value ( $\pm$  standard deviation) over 200 repetitions. As a reference, the performance of a linear model is also included to show that the neural network learned the non-linear relationships.

	Scaling	Effect type		
		Linear	Piece-wise linear	Non-continuous
Section 4.1	No scaling	0.92 $\pm$ 0.01	0.74 $\pm$ 0.02	0.57 $\pm$ 0.03
	Z-Score	0.91 $\pm$ 0.01	0.75 $\pm$ 0.02	0.60 $\pm$ 0.03
	Max-Abs	0.92 $\pm$ 0.01	0.75 $\pm$ 0.02	0.57 $\pm$ 0.04
	(Linear model)	0.92 $\pm$ 0.00	0.47 $\pm$ 0.06	0.23 $\pm$ 0.04
Section 4.2	Z-Score	0.81 $\pm$ 0.01	0.60 $\pm$ 0.03	0.60 $\pm$ 0.03
	(Linear model)	0.82 $\pm$ 0.01	0.38 $\pm$ 0.08	0.20 $\pm$ 0.06

**Table 2.** Results of the neural network performance on test data consisting of categorical variables measured by the average  $R^2$  value ( $\pm$  standard deviation) over 200 repetitions.

	# Levels	Encoding			
		Label	One-hot	Dummy	Binary
Section 4.1	4	0.57 $\pm$ 0.02	0.60 $\pm$ 0.02	0.55 $\pm$ 0.02	0.56 $\pm$ 0.02
	12	0.37 $\pm$ 0.05	0.48 $\pm$ 0.03	0.37 $\pm$ 0.03	0.42 $\pm$ 0.03
Section 4.2	Binary	0.81 $\pm$ 0.01	–	–	–
	4	–	0.7 $\pm$ 0.02	–	–

## References

1. Adebayo, J., Gilmer, J., Goodfellow, I., Kim, B.: Local explanation methods for deep neural networks lack sensitivity to parameter values. In: ICLR 2018 Workshop Track (2018)
2. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. *Adv. Neural. Inf. Process. Syst.* **31**, 9525–9536 (2018)
3. Agarwal, C.: OpenXAI: towards a transparent evaluation of model explanations. *Adv. Neural. Inf. Process. Syst.* **35**, 15784–15799 (2022)
4. Alvarez Melis, D., Jaakkola, T.: Towards robust interpretability with self-explaining neural networks. *Adv. Neural. Inf. Process. Syst.* **31**, 7786–7795 (2018)
5. Ancona, M., Ceolini, E., Öztireli, C., Gross, M.: Towards better understanding of gradient-based attribution methods for deep neural networks. In: Proceedings of the 6th International Conference on Learning Representations (2018)
6. Arik, S.Ö., Pfister, T.: TabNet: attentive interpretable tabular learning. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence, vol. 35, pp. 6679–6687 (2021)
7. Arras, L., Osman, A., Samek, W.: CLEVR-XAI: a benchmark dataset for the ground truth evaluation of neural network explanations. *Inf. Fusion* **81**, 14–40 (2022)
8. Arya, V., et al.: One explanation does not fit all: A toolkit and taxonomy of AI explainability techniques (2019). arXiv preprint [arXiv:1909.03012](https://arxiv.org/abs/1909.03012)
9. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **10**(7), 1–46 (2015)
10. Bao, M., et al.: It’s COMPASlicated: The messy relationship between RAI datasets and algorithmic fairness benchmarks. In: Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2021)
11. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: quantifying interpretability of deep visual representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)
12. Bengio, Y., Lecun, Y., Hinton, G.: Deep learning for AI. *Commun. ACM* **64**(7), 58–65 (2021)
13. Bhatt, U., Weller, A., Moura, J.M.F.: Evaluating and aggregating feature-based model explanations. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence, pp. 3016–3022 (2020)
14. Carmichael, Z., Scheirer, W.: How well do feature-additive explainers explain feature-additive predictors? In: XAI in Action: Past, Present, and Future Applications (2023)
15. Chen, H., Lundberg, S., Lee, S.I.: Explaining models by propagating Shapley values of local components. *Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability*, pp. 261–270 (2021)
16. Chen, J., Song, L., Wainwright, M., Jordan, M.: Learning to explain: an information-theoretic perspective on model interpretation. In: Proceedings of the 35th International Conference on Machine Learning, vol. 80, pp. 883–892 (2018)
17. Dasgupta, S., Frost, N., Moshkovitz, M.: Framework for evaluating faithfulness of local explanations. In: Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 4794–4815 (2022)
18. Deng, H., et al.: Unifying fourteen post-hoc attribution methods with Taylor interactions. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**(7), 4625–4640 (2024). <https://ieeexplore.ieee.org/document/10414149>


19. Dolci, G., Cruciani, F., Galazzo, I.B., Calhoun, V.D., Menegaz, G.: Objective assessment of the bias introduced by baseline signals in XAI attribution methods. In: Proceedings of the IEEE International Conference on Metrology for eXtended Reality, Artificial Intelligence and Neural Engineering, pp. 266–271 (2023)
20. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning (2017). arXiv preprint [arXiv:1702.08608](https://arxiv.org/abs/1702.08608)
21. Erion, G., Janizek, J.D., Sturmfels, P., Lundberg, S.M., Lee, S.I.: Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nat. Mach. Intell.* **3**(7), 620–631 (2021)
22. Fong, R.C., Vedaldi, A.: Interpretable explanations of black boxes by meaningful perturbation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3449–3457 (2017)
23. Gevaert, A., Rousseau, A.J., Becker, T., Valkenburg, D., De Bie, T., Saeys, Y.: Evaluating feature attribution methods in the image domain (2022). arXiv preprint [arXiv:2202.12270](https://arxiv.org/abs/2202.12270)
24. Guidotti, R.: Evaluating local explanation methods on ground truth. *Artif. Intell.* **291**, 103428 (2021)
25. Haug, J., Zürn, S., El-Jiz, P., Kasneci, G.: On baselines for local feature attributions (2021). arXiv preprint [arXiv:2101.00905](https://arxiv.org/abs/2101.00905)
26. Hooker, S., Erhan, D., Kindermans, P.J., Kim, B.: A benchmark for interpretability methods in deep neural networks. *Adv. Neural Inf. Process. Syst.* **32** (2019). [https://proceedings.neurips.cc/paper\\_files/paper/2019/hash/fe4b8556000d0f0cae99daa5c5c5a410-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2019/hash/fe4b8556000d0f0cae99daa5c5c5a410-Abstract.html)
27. Ivanovs, M., Kadikis, R., Ozols, K.: Perturbation-based methods for explaining deep neural networks: a survey. *Pattern Recogn. Lett.* **150**, 228–234 (2021)
28. Khakzar, A., Khorsandi, P., Nobahari, R., Navab, N.: Do explanations explain? Model knows best. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10244–10253 (2022)
29. Kim, B., et al.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: Proceedings of the 35th International Conference on Machine Learning, vol. 80, pp. 2668–2677 (2018)
30. Kim, J.S., Plumb, G., Talwalkar, A.: Sanity simulations for saliency methods. In: Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 11173–11200 (2022)
31. Kindermans, P.J., et al.: The (un)reliability of saliency methods. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280 (2019)
32. Koenen, N., Wright, M.N.: Interpreting deep neural networks with the package innsight (2023). arXiv preprint [arXiv:2306.10822](https://arxiv.org/abs/2306.10822)
33. Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., Lapuschkin, S.: Towards best practice in explaining neural network decisions with LRP. In: Proceedings of the International Joint Conference on Neural Networks, pp. 1–7 (2020)
34. Krishna, S., et al.: The disagreement problem in explainable machine learning: A practitioner’s perspective (2022). arXiv preprint [arXiv:2202.01602](https://arxiv.org/abs/2202.01602)
35. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
36. Liu, Y., Khandagale, S., White, C., Neiswanger, W.: Synthetic benchmarks for scientific research in explainable machine learning. In: Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (2021)

37. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **30**, 4765–4774 (2017)
38. Mamalakis, A., Barnes, E.A., Ebert-Uphoff, I.: Carefully choose the baseline: lessons learned from applying XAI attribution methods for regression tasks in geoscience. *Artif. Intell. Earth Syst.* **2**(1), e220058 (2023)
39. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.R.: Layer-wise relevance propagation: An overview. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 193–209 (2019)
40. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recogn.* **65**, 211–222 (2017)
41. Neely, M., Schouten, S.F., Bleeker, M.J., Lucic, A.: Order in the court: explainable AI methods prone to disagreement. In: *ICML 2021 Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI* (2021)
42. Nguyen, A.p., Martínez, M.R.: On quantitative aspects of model interpretability (2020). arXiv preprint [arXiv:2007.07584](https://arxiv.org/abs/2007.07584)
43. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized input sampling for explanation of black-box models. In: *Proceedings of the British Machine Vision Conference* (2018)
44. Samek, W., Binder, A., Montavon, G., Lapuschkin, S., Müller, K.R.: Evaluating the visualization of what a deep neural network has learned. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(11), 2660–2673 (2017)
45. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE International Conference on Computer Vision* (2017)
46. Shapley, L.S.: A value for n-person games. *Contrib. Theor. Games* **II**, 307–318 (1953)
47. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3145–3153 (2017)
48. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: Learning important features through propagating activation differences (2016). arXiv preprint [arXiv:1605.01713](https://arxiv.org/abs/1605.01713)
49. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps (2013). arXiv preprint [arXiv:1312.6034](https://arxiv.org/abs/1312.6034)
50. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: SmoothGrad: removing noise by adding noise (2017). arXiv preprint [arXiv:1706.03825](https://arxiv.org/abs/1706.03825)
51. Springenberg, J., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: the all convolutional net. In: *3rd International Conference on Learning Representations, Workshop Track* (2015)
52. Srinivas, S., Fleuret, F.: Full-gradient representation for neural network visualization. *Adv. Neural Inf. Process. Syst.* **32** (2019). [https://papers.nips.cc/paper\\_files/paper/2019/hash/80537a945c7aaa788ccfcd1b99b5d8f-Abstract.html](https://papers.nips.cc/paper_files/paper/2019/hash/80537a945c7aaa788ccfcd1b99b5d8f-Abstract.html)
53. Sturmfels, P., Lundberg, S., Lee, S.I.: Visualizing the impact of feature attribution baselines. *Distill* (2020)
54. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3319–3328 (2017)
55. Tjoa, E., Guan, C.: Quantifying explainability of saliency methods in deep neural networks with a synthetic dataset. *IEEE Trans. Artif. Intell.* **4**(4), 858–870 (2023)

56. Tomsett, R., Harborne, D., Chakraborty, S., Gurram, P., Preece, A.: Sanity checks for saliency metrics. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 6021–6029 (2020)
57. Yang, M., Kim, B.: Benchmarking attribution methods with relative feature importance (2019). arXiv preprint [arXiv:1907.09701](https://arxiv.org/abs/1907.09701)
58. Yeh, C.K., Hsieh, C.Y., Suggala, A., Inouye, D.I., Ravikumar, P.K.: On the (in)fidelity and sensitivity of explanations. *Adv. Neural Inf. Process. Syst.* **32** (2019). [https://papers.nips.cc/paper\\_files/paper/2019/hash/a7471fdc77b3435276507cc8f2dc2569-Abstract.html](https://papers.nips.cc/paper_files/paper/2019/hash/a7471fdc77b3435276507cc8f2dc2569-Abstract.html)
59. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Proceedings of the 13th European Conference on Computer Vision, pp. 818–833 (2014)
60. Zhang, J., Bargal, S.A., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. *Int. J. Comput. Vision* **126**(10), 1084–1102 (2017)
61. Zhang, Y., Tiño, P., Leonardis, A., Tang, K.: A survey on neural network interpretability. *IEEE Trans. Emerg. Top. Comput. Intell.* **5**(5), 726–742 (2021)
62. Zhou, Y., Booth, S., Ribeiro, M.T., Shah, J.: Do feature attribution methods correctly attribute features? In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 9623–9633 (2022)



# ConformaSight: Conformal Prediction-Based Global and Model-Agnostic Explainability Framework

Fatima Rabia Yapicioglu<sup>(✉)</sup> , Alessandra Stramiglio , and Fabio Vitali 

DISI, Department of Computer Science and Engineering, University of Bologna,  
Via Zamboni, 33, 40126 Bologna, Italy

{fatima.yapicioglu2,a.stramiglio,fabio.vitali}@unibo.it  
<https://www.unibo.it/en>

**Abstract.** Conformal inference or prediction is a method in statistics to yield resilient uncertainty bounds for predictions from black-box models regardless of any presupposed data dissemination. It has emerged as a simple practice to establish intervals of uncertainty, especially in critical scenarios. By incorporating a user-defined probability threshold, conformal inference ensures that the resulting sets—such as the predicted range in regression tasks or the prediction set in classification scenarios—reliably encompass the actual value. For instance, by defining a threshold probability, we can compute price or quality tiers ranges for pre-owned cars, assuring that the actual values will fall within these intervals. While these models offer transparency in terms of uncertainty quantification, they often come up short in explainability when it comes to grasping comprehension of factors driving changes in conformal metrics such as set size, coverage, and thus formation of prediction set-type outputs. Our paper introduces a comprehensive global explainability framework based on conformal inference, addressing the void of accommodating prediction set-type outputs in various classifiers. This understanding not only enhances transparency but furthermore ensures verifiability in comprehending the factors driving changes in conformal metrics and the formation of prediction sets which are assured to have actual value with the help of counterfactual instances of calibration-sets. Moreover, ConformaSight’s capability to capture and rank significant features, boosting classifier coverage, enables it to effectively identify the minimal dataset required for optimal model performance. We also showcase the flexibility of employing user-defined thresholds and re-calibration techniques to generate robust and reliable global feature importance estimates on test sets with significantly diverse distributions, obtained by perturbing the original test sets.

**Keywords:** Generative Explainability · Conformal Prediction · Uncertainty Estimation · Model-Agnostic Explainability

## 1 Introduction

In the realm of machine learning (ML), classification tasks play a pivotal role across various domains, from medical diagnosis and fraud detection to image recognition and sequence models. As artificial intelligence (AI) models become more complex and impact significant decisions in critical scenarios, the ability to interpret model decisions and estimate the uncertainty associated with predictions becomes crucial. At this point, Explainable Artificial Intelligence (XAI) emerges to introduce understanding to AI models, which hide the underlying working mechanism and are exceedingly complex [1]. Throughout this paper, we will use the terms “explainability” and “interpretability” interchangeably. For global explainability, which aims to provide insights into the overall model behavior, popular methods of explanation include PDP (Partial Dependence Plots) [2], ALE (Accumulated Local Effects) [3], and Permutation Feature Importance [4–6]. On the other hand, SHAP (Shapley Additive Explanations) [7] and LIME (Locally Interpretable Model Agnostic Explanations) [8] have been widely applied for local explainability, focusing on the decision-making process underlying local instances.

However, it is also impossible to overestimate the significance of uncertainty estimation which involves assessing the range of potential outcomes and the confidence in a model’s outcomes and uncertainty quantification that measures to quantify the uncertainty in AI models, especially in applications where making the wrong choice could have catastrophic consequences. For instance, considering autonomous vehicles, they must navigate through uncertain traffic environments with noisy perceptions and approximated models, where erroneous decisions could potentially endanger human lives [9].

Furthermore, model and data uncertainty are indicated as main causes of uncertainty estimation. The former, uncertainty of model, arises due to noise present in the data, often stemming from insufficient data collection methods. On the other hand, the latter, data uncertainty, results from distributional imbalances in the training set [10, 11]. However, model complexity is another significant contributor when evaluating uncertainty of model [12]. The use of quantification of uncertainty methods are essential for mitigating the effects of uncertainty in the optimization and decision-making processes [13]. As a solution, quantification of uncertainty methods emerged to communicate with the users in which extent the output should be trusted. Henceforward, conformal prediction has proven to be a potent instrument for the rigorous quantification of uncertainty in vital applications in various research fields [14–17]. The main source of uncertainty reflected by conformal inference models is epistemic (model’s doubt about its own predictions). It provides model-agnostic, statistically assured uncertainty estimate with flexible sets, revealing outliers and offering explainability insights. Suppose you’ve developed a machine learning model to predict the selling price of pre-owned cars based on features like model, mileage, year, and condition. While your model provides a single predicted value for each car, you want to enhance the usefulness of your predictions by providing an interval within which the parameters lie actual selling price is likely to fall, with a specified level

of confidence (e.g., 90%). Imagine a pre-owned car is listed for sale, and your model predicts its price to be \$15,000. You know the distribution of errors (non-conformity scores) for the 90% level of confidence corresponds to a range of  $\pm\$2,000$  around the predicted prices. Therefore, you could report the prediction as: Predicted Price Range: \$13,000 to \$17,000 with 90% confidence.

While current techniques for estimating uncertainty exist, there remains a gap in effectively integrating these with explainable artificial intelligence methods [18]. Additionally, existing model-agnostic explainability approaches primarily target models with single prediction outputs, neglecting methods that provide statistically assured, flexible prediction sets as we experience in conformal inference models. Conformal inference distinguishes itself through its resilience to deviations from normality, as it has the capacity to accommodate diverse data characteristics without necessitating specific assumptions about error distributions as defined in literature [19].

In this paper, standing on the uncertainty assessment and XAI convergence, we frame the work of creating an explainability methodology which can be adapted to conformal inference models with user-specified error rate and flexible prediction-set outcomes. Proposed framework produces discrete feature importance table as an output, so that the non-experts can understand which factors lead to dramatic changes in conformal metrics such as coverage and set-size of prediction sets. In continuation of our previous example, which factor—whether it be mileage, age, brand, model, or condition-of a pre-owned car leads to significant variations in the price range or quality tiers, ensuring that the actual pre-owned price or quality tier aligns with the actual value?

We can characterize our primary contributions as follows:

1. ConformaSight, a framework that can explain any classifier in a robust way considering conformal prediction offerings based on coverage, i.e., how often the prediction set includes the true explanation, and the size of the prediction set. Specifically, we are providing flexibility of producing multiple explanations using pre-defined thresholds to include ground truth on the convergence of uncertainty quantification and XAI.
2. Another contribution is to provide detailed insights into the factors influencing coverage for each class in a classification task, thereby enabling targeted adjustments to enhance the prediction accuracy and coverage of specific classes.
3. Comprehensive evaluation of effectiveness, faithfulness, and resilience on different models and datasets. We also provide comparison on small and large real-world tabular data enabling non-experts to do reasoning with uncertainty-aware explainability methodology in high-risk setting environments.

We structure the paper as follows: Sects. 2 and 3 offer preliminaries and literature studies to prepare reader background, while Sect. 4 delves into the theoretical intricacies of proposed concept. Given the length of the theory, experiments and evaluations are presented separately in Sect. 5. Finally, Sect. 6 encompasses results, discussion, and plans for future research.



## 2 Preliminaries

### 2.1 Explainability

Explainable AI (XAI) emerged from the necessity for transparency and understanding due to the widespread adoption of machine learning models across various domains. Its goal is to develop interpretable models that maintain high performance while enabling human comprehension and trust [20]. However, to tackle the challenge of providing explanations, it's crucial to formalize the concept of *explainability*. We define *explainability* as the AI model's capacity to elucidate its decision-making process in a manner comprehensible to humans. This entails transparently revealing the factors and considerations influencing the model's outputs, fostering understanding, trust, and accountability [21]. In addressing what needs explanation - the *explanandum* - a combination of different XAI mechanisms is necessary to ensure a minimal understanding from the recipient of the explanation - the *explainee* - regarding the internal logic of a black-box AI [22].

In XAI methods, two main categories exist: *Intrinsic Methods* and *Post-hoc Methods*. Intrinsic Methods are integrated directly into the architecture or training process of AI models, ensuring interpretability by design. Examples include Decision Trees, Rule-based Systems, and Linear Models, which operate on explicit rules or clear calculations. Post-hoc Methods, on the other hand, offer interpretability after training opaque or complex AI models. Local Explainability focuses on explaining individual predictions, aiding in understanding specific outcomes, with methods like SHAP and LIME [7,8]. Global Explainability provides insights into overall model behavior across datasets or domains, revealing broader patterns and decision-making processes, as seen in algorithms like PDP [2] and ALE [3].

### 2.2 Uncertainty Estimation and Quantification

Estimating uncertainty in machine learning refers to the process of quantifying the confidence or reliability of the predictions made by a model. It involves assessing the degree of uncertainty associated with each prediction, recognising that models do not always produce deterministic results, but rather predictions accompanied by a certain level of uncertainty. This uncertainty can stem from various factors, including limited or noisy training data, model complexity, or inherent stochasticity in the underlying processes being modeled [23].

When estimating the uncertainty of a forecast, - *predictive uncertainty* -, the most common way of estimating it is based on the separate modelling of model-induced uncertainty (epistemic or model uncertainty) and data-induced uncertainty (random or data uncertainty). While the former can be addressed by improving the model, the latter cannot be reduced [24].

From data collection to the development of a neural network (NN), numerous phases introduce potential sources of uncertainty and error. These include

variability in real-world situations, measurement errors, architectural flaws in the NN, training errors, and uncertainties stemming from unknown data [24].

Referring to input data domain, as in [24], predictive uncertainty comes in three types: in-domain, domain-shift, and out-of-domain. In-domain uncertainty relates to data distribution inputs and can be reduced by enhancing training data quality. Domain-shift uncertainty occurs when inputs differ from the training distribution, posing challenges for neural networks to explain. Out-of-domain uncertainty arises from inputs beyond the known data distribution, where neural networks struggle to explain samples beyond their trained knowledge.

### 2.3 Conformal Prediction

Here, we follow the steps outlined by the tutorial elaborated in [25]. Conformal inference or prediction offers a refreshingly convenient and easy way to create sound confidence ranges for any arbitrary model outcomes. Unlike traditional methods that assume a specific dispersion in data, conformal inference does not assume any specific dispersion in data, making it applicable to a wide range of data types and modeling scenarios.

At the core of conformal inference is the notion of validity and efficiency. Validity ensures that the prediction bounds or sets constructed by the method have a certain coverage probability, while efficiency aims to minimize the size of these intervals or sets. The user-specified assurance threshold ( $\alpha$ ) in conformal inference sets the desired confidence level, while the coverage rate (*coverage*) quantifies the proportion of actual instances encompassed by estimation sets. It's essential to distinguish between these two concepts to avoid confusion regarding the interpretation of prediction intervals. The in-depth review of conformal inference is as follows.

To commence, we initiate with a trained predictive model, denoted as  $\hat{f}$  (for instance a neural network classifier). Subsequently, we embark on generating prediction sets, comprising potential labels, for this classifier by employing a limited quantity of supplementary calibration data-referred to as the calibration step on occasion.

In more scholarly terms, let us consider a scenario where serve input each corresponding to one of  $M$  classes. Initially, we possess a classifier that yields estimated probabilities (represented as scores of softmax in this example but it doesn't have to be) for each class:  $\hat{f}(x) \in [0, 1]^M$ . Subsequently, we allocate a reasonably smaller subset of newly acquired independent and identically distributed pairs of data, not observed in the training stage, denoted as  $(X_1, Y_1), \dots, (X_n, Y_n)$ , for utilization as calibration set  $(X^{(cal)}, y^{(cal)})$ . Leveraging  $\hat{f}$  and calibration subset, our objective is to formulate a prediction set of feasible labels  $C(X^{(test)}) = \{1, \dots, M\}$  that holds validity in the subsequent context:

$$1 - \alpha \leq P(Y^{(test)} \in C(X^{(test)})) \leq 1 - \alpha + \frac{1}{n + 1}, \quad (1)$$

where  $(X^{(\text{test})}, Y^{(\text{test})})$  be a new test point drawn from the same distribution, and let  $\varepsilon \in [0, 1]$  represent a user-chosen error rate as it has been officially stated in [26]. The calibration process for input  $x$  and output  $y$  can be summed up as follows in general.

1. Establish an indicator of uncertainty by leveraging the model you trained (a classifier in this case).
2. Introduce the score method  $s(x, y) \in \mathbb{R}$ , where greater scores signify greater disagreement among  $x$  and  $y$ .
3. Determine  $\hat{q}$  by using the following procedure

$$\frac{\lceil (n + 1) \cdot (1 - \alpha) \rceil}{n} \quad (2)$$

quantile of the calculated score values of calibration  $s_1 = s(X_1, Y_1), \dots, s_n = s(X_n, Y_n)$ .

4. Employ this quantile to delineate prediction compilation for new instances:

$$C(X^{(\text{test})}) = \left\{ y : s(X^{(\text{test})}, y) \leq \hat{q} \right\}. \quad (3)$$

As we indicated before, for every score method and data distribution, these sets meet the validity property in (1).

There are numerous conformal prediction variants available as of today. These variations consist of the concepts of risk control [27, 28] and also the covariate shift [29]. Moreover, distribution shift, or when the test set diverges from the distribution in the calibration data, is another important area of study. For instance, [30] introduces a conformal procedure robust to shifts of known  $f$ -divergence in the score approach. A weighted variant of conformal inference that offers methods for handling non-exchangeable data was developed by [31]. Furthermore, by perpetually re-estimating the conformal quantile, [32] creates estimation ranges in a data stream with an altering distribution over time.

### 3 Related Work

Based on our literature review, this section provides studies that shed light on the links between uncertainty estimation and XAI. For instance, [33] underlines the importance of explanations, particularly in scenarios of uncertain models. Meanwhile, [34] explores the utilization of conformal inference methodologies for enhancing interpretability and reliability in AI models, albeit without explicit mention of uncertainty. Furthermore, [35] delves into oracle coaching with conformal inference, aiming to create highly accurate and interpretable models tailored to specific test sets. Additionally, [36] proposes non-conformity measures designed to approximate explanations efficiently, demonstrating superiority over traditional methods. Meanwhile, [37] proposes a statistical framework leveraging conformal inference for node-level explanations, highlighting their impact on

neighboring nodes. Moreover, [38] provides insights into constructing uncertainty sets for optimal explanations, exploring the implications of explanations derived from the true data-generating distribution. In the domain of self-explaining networks, [39] establishes a framework that emphasizes uncertainty without relying on distributional presumptions, particularly in generating efficient and effective prediction sets. Also, [18] investigated a relationship to XAI and created a framework specifically for neural networks that has been provided with XAI-like uncertainty estimates. The research results showed, however, that estimates of uncertainty in the model are susceptible to variations in the data dissemination. Similarly, [40] presented a novel framework that enables the conversion of any random neural network explanation technique into a Bayesian neural network explanation method. Finally, [41] suggests using “Monte Carlo Dropout” and trust scores to tackle uncertainty in counterfactual explanations.

ConformaSight, in contrast, is a model-agnostic and global explainer that provides robust explanations for variations in data distribution, and can elucidate not only neural networks but also any type of classifier capable of generating probabilities thanks to conformal inference basement.

## 4 Methodology

This section introduces the basic principles, essential elements, practical information, and validity requirements of the ConformaSight framework. Following this, we provide a practical example scenario demonstrating how to split your data, set arguments, and call the explainer method.

### 4.1 ConformaSight Structure and Mechanism

Presented herein is ConformaSight, our proposed framework rooted in conformal prediction methodology. It is designed to furnish robust and insightful explanations independent of the data distribution, produce explanations for set-type outputs by the conformal predictors. Unlike traditional methods where explanations focus solely on feature importance, explanations in conformal prediction-based models consider how the calibration process influences the prediction outcomes.

For instance, if a particular feature in the calibration set has a strong influence on the coverage of prediction intervals, it indicates that this feature plays a crucial role in determining the model’s confidence in its predictions. Therefore, explaining the model’s predictions involves not only highlighting the importance of features but also elucidating how the calibration process impacts the prediction outcomes. This challenge is encapsulated by several different research questions outlined as below.

**Adapting Explainability to Conformal Prediction:** *How can we enhance the interpretability of conformal prediction results by investigating the factors influencing the formation of prediction sets?* In the realm of classification tasks, understanding the factors influencing prediction set formation within conformal prediction frameworks is crucial for enhancing model interpretability and

trust [42]. A meticulous examination of metrics such as weighted coverage and weighted set size provides profound insights into the uncertainty representation of the model [25]. Weighted coverage elucidates the proportion of instances correctly classified and enclosed within prediction sets, offering a measure of the model’s reliability in delineating uncertain regions with respect to class distribution [43]. Meanwhile, weighted set size furnishes critical information regarding the granularity of uncertainty representation, indicating the average number of instances grouped within prediction sets while considering class imbalance [25]. By scrutinizing these metrics, researchers can unravel the nuanced relationship between model predictions and input features.

Prediction sets  $P_i$  are obtained from the conformal prediction algorithm, while  $y$  represents the true class labels of the instances. To quantify the performance of these prediction sets, we define two key conformal metrics: *weighted coverage* and *weighted set size*.

**Definition 1. Weighted coverage:** *We define it as the proportion of correctly classified instances within prediction sets for each class, and it is calculated as the average proportion of correctly classified instances weighted by the class sizes:*

$$\frac{\sum_{i=1}^n (\text{Coverage}(P_i, y) \times c_i)}{\sum_{i=1}^n c_i} \tag{4}$$

where  $n$  is the total number of classes,  $\text{Coverage}(P_i, y)$  is the coverage for prediction set  $P_i$  given true labels  $y$ , and  $c_i$  is the count of instances in class  $i$ .

**Definition 2. Weighted set size:** *We define it as the average number of instances in prediction sets for each class, and it is computed as the average set size weighted by the class sizes:*

$$\frac{\sum_{i=1}^n (\text{Set Size}(P_i) \times c_i)}{\sum_{i=1}^n c_i} \tag{5}$$

where  $\text{Set Size}(P_i)$  is the average set size for prediction set  $P_i$ .

**Defining Imbalance-Free Baseline Conformal Metrics:** *How can we obtain baseline weighted coverage and set size metrics while addressing class imbalance in the dataset?* To ensure the reliability of baseline conformal metrics, it is imperative to adopt conformal classification techniques that maintain equal coverage across classes. By setting thresholds independently for each class, we can guarantee coverage across all classes, thus enhancing the robustness of the baseline metrics. Specifically, for each class, thresholds can be determined to encompass certain percent (that can be decided by user) of instances within that class [44]. This approach not only addresses class imbalance but also ensures the safety and accuracy of the baseline conformal metrics. The presented Algorithm 1 aims to establish thresholds for each class independently in a conformal classification model, ensuring equitable coverage across all classes. Given the error rate  $\alpha$ ,

calibration dataset  $X^{(cal)}$  with corresponding labels  $y^{(cal)}$ , and a conformal classification model  $\hat{f}$ , the algorithm computes thresholds such that certain percentage of instances within each class fall below the threshold. This is achieved by predicting probabilities for each class label in  $X^{(cal)}$  using the classifier, computing the s-scores (complement of probabilities), and determining the  $\hat{q}$ , the  $1 - \alpha$  quantile of s-scores for each class, adjusted for class size. Specifically,  $\hat{q}$  is computed as in Eq. (2), where  $n$  represents the size of each class.

---

**Algorithm 1: Get Individual Thresholds For Each Class**


---

**Input:** Classifier  $\hat{f}$ , Error Rate  $\alpha$ , List of All Classes  $classes\_list$ , Calibration Set  $(X^{(cal)}, y^{(cal)})$

**Output:** Thresholds  $thresholds$

```

thresholds = [ ];
foreach class in classes_list do
    // Extract probabilities for each instance of  $X^{(cal)}$  which has label
    //  $y^{(cal)} \in class$ .
    y_cal_probs = get_probs( $\hat{f}$ ,  $X^{(cal)}$ ,  $y^{(cal)}$ , class);
    // Perform the following next two steps as a reference to Eq. (2).
    s_scores,  $\hat{q}$  = get_qhat(y_cal_probs,  $\alpha$ );
    threshold = percentile(s_scores,  $\hat{q}$ );
    append threshold to thresholds;
end
return thresholds;
```

---



---

**Algorithm 2: Get Prediction Sets For The Test Set**


---

**Input:** Classifier  $\hat{f}$ , Test set  $X^{(test)}$ , Thresholds  $thresholds$ , Number of Classes  $n\_classes$

**Output:** Prediction sets  $prediction\_sets$

```

prediction_sets = [ ];
// Extract probabilities for the entire  $X^{(test)}$ .
predicted_probas =  $\hat{f}$ .predict_proba( $X^{(test)}$ );
// Extract all probabilities from 1 to get s-scores.
si_scores = 1 - predicted_probas;
// Apply class-conditional thresholding.
for i = 0 to n_classes do
    | prediction_sets[i] = si_scores[:, i]  $\leq$  thresholds[i];
end
return prediction_sets;
```

---

The algorithms delineated in Algorithm 1 and 2 epitomize a pivotal stride in the realm of class-conditional conformal prediction [25], a methodology poised to tackle the challenges inherent in imbalanced datasets within the context of multiclass classification tasks. By ingesting pivotal inputs including the trained

classifier, test dataset, thresholds, and the number of classes, this algorithm systematically refines classifier predictions. It achieves this feat by meticulously crafting prediction sets customized to each class. This granular approach becomes indispensable in scenarios characterized by imbalanced class distributions or varying classification priorities, where conventional methods falter. Through the iterative application of class-specific thresholds to predicted probabilities, the algorithm adeptly adapts to the nuanced landscape of each class, mitigating the adverse effects of data imbalance. Finally, a full algorithm to produce weighted coverage  $W_{coverage}$  (4), weighted set size  $W_{set\_size}$  (5) can be found in Algorithm 3. We use the same algorithm to set baseline conformal metrics.

**Providing Distribution Shift in the Calibration Set:** *By leveraging the conformal prediction framework to provide distribution shift in calibration set, how can we systematically provide variations?* Through the lens of the conformal prediction framework, addressing the question of how to systematically provide variations in calibration set-size to discern significant features through the change in weighted coverage and set-size emerges as a crucial endeavor in enhancing model interpretability and reliability. Advancing beyond a single calibration, the approach entails repeated generation of prediction sets with perturbed calibration sets, echoing insights into data drift and concept evolution. Analysis of variations across these sets illuminates nuanced shifts in model behavior, offering a pathway to discern significant features driving prediction set formation.

---

**Algorithm 3:** ConformaSight Calibration and Metrics Production

---

**Input:** Calibration Set  $(X^{(cal)}, y^{(cal)})$ , Test Set  $(X^{(test)}, y^{(test)})$ , Error Rate  $\alpha \in [0, 1]$ , Class Labels  $l$ , Classifier  $\hat{f}$

**Output:** Weighted Coverage  $W_{coverage}$ , Weighted Set-Size  $W_{set\_size}$

**Calibration and Metric Production:**

```
// Perform the following step as we described in Algorithm 1.
thresholds =
    get_individual_thresholds_for_each_class( $\hat{f}$ ,  $\alpha$ ,  $l$ ,  $X^{(cal)}$ ,  $y^{(cal)}$ );
// Perform the following step as we described in Algorithm 2.
C( $X^{(test)}$ ) =
    get_individually_thresholded_prediction_sets( $\hat{f}$ ,  $X^{(test)}$ , thresholds,  $l$ );
Acoverage = get_coverages_per_class(C( $X^{(test)}$ ),  $y^{(test)}$ );
Atotal_instances = get_total_instances_per_class( $y^{(test)}$ );
Aset_sizes = get_set_size_per_class(C( $X^{(test)}$ ));
Wcoverage = get_weighted_coverage(Acoverage, Atotal_instances);
Wset_sizes = get_weighted_set_size(Aset_sizes, Atotal_instances);
return Wcoverage, Wset_size;
```

---

Now we outline the key perturbation methods utilized to introduce variations in the calibration datasets. Specifically, we employ counterfactual perturbations to systematically alter the numerical attributes of the data and permutation-based perturbations for categorical attributes with a certain number of severity

that can be supplied by user. We present counterfactual perturbations using both Gaussian Noise [45] and Uniform Noise [46]. It is crucial to investigate the impact of different types of noise on the experimental outcomes. While Gaussian noise preserves the distribution shift of the original data, uniform noise introduces variability that can alter the distribution shift [47]. By systematically applying noise separately to individual columns (we perturb one column at a time, keep others fixed and measure conformal metrics:  $W_{coverage}$ ,  $W_{set\_size}$ ) and repeating the experiment, we can discern how these variations influence the conformal metrics.

This analysis provides valuable insights into the robustness and generalizability of our findings under different perturbation scenarios. We provide mathematical and statistical definitions for each perturbation method that we utilize in this paper as follows.

**Definition 3. Permutation-based Perturbations:** *Let  $X$  be a categorical column with  $n$  unique categories. The permutation-based perturbation function  $permute(X, k)$  produces a perturbed column  $X'$ , where the values of  $X'$  are randomly permuted  $k$  times. Mathematically, this can be expressed as:*

$$X' = permute(X, k) \quad (6)$$

**Definition 4. Gaussian Noise Perturbations:** *Let  $N$  be a numerical column with standard deviation  $\sigma$ . Given a severity parameter  $s$ , the counterfactual perturbation function adds noise to  $N$  according to:*

$$N' = N + \text{Noise}, \quad \text{where } \text{Noise} \sim \mathcal{N}(0, s \times \sigma). \quad (7)$$

**Definition 5. Uniform Noise Perturbations:** *Let  $N$  be a numerical column. Given a severity parameter  $s$ , the uniform noise perturbation function adds noise sampled from a uniform distribution to  $N$  according to:*

$$N' = N + \text{Noise}, \quad \text{where } \text{Noise} \sim \text{Uniform}(-s \times R, s \times R), \quad (8)$$

where  $R$  represents the range of values in  $N$ . The uniform noise perturbation introduces variability across the dataset, allowing for the exploration of different distributional shifts.

**Observing Relative Change in Conformal Metrics:** *How can we measure the relative deviation in weighted coverage and set size from the baseline reciprocals obtained through initial set calibration?* To evaluate the collective deviation in weighted coverage and set size across systematically perturbed calibration sets compared to the original calibration sets, we adopted two distinct functions. Initially, we quantified the weighted coverage and set size across all perturbed datasets, providing an aggregate perspective. Subsequently, to delve deeper into the perturbation effects and gain further insights, we introduced another function to compile averages in relative changes per class in by weighted coverage and set size. This finer-grained analysis enables us to discern the nuanced impacts



of perturbations on individual classes, contributing to a more comprehensive understanding of model behavior under varying conditions [48].

Let  $C_0$  denote the baseline weighted coverage obtained through initial set calibration, and  $S_0$  represent the baseline weighted set size. The Algorithm 4 computes the relative change in coverage and set size for each perturbed dataset compared to the baseline for each perturbed dataset. Finally, a full algorithm which outlines the steps one-by-one to execute explanations with ConformaSight can be found in Algorithm 5.

---

**Algorithm 4:** Calculate Relative Changes - Algorithm for calculating relative changes in weighted coverage and set size

---

**Input:** Baseline weighted coverage  $C_0$ , weighted set size  $S_0$ , and *perturbed\_datasets*

**Output:** List of Relative changes in coverage  $\Delta C_i$  and set size  $\Delta S_i$  for each perturbed dataset  $i$  [ $\Delta C_i, \Delta S_i$ ]

**foreach** *perturbed dataset*  $i$  **in** *perturbed\_datasets* **do**

// Perform the following step as we described in Algorithm 3.

Calculate weighted coverage  $C_i$  (4) and set size  $S_i$  (5);

Compute relative changes:

$$\Delta C_i = \left| \frac{C_i - C_0}{C_0} \right| \times 100$$

$$\Delta S_i = \left| \frac{S_i - S_0}{S_0} \right| \times 100$$

**end**

**return** [ $\Delta C_i, \Delta S_i$ ];

---

**Threats to Validity:** *What are the conditions that may pose threats to validity of conformal prediction models?* When assessing validity of conformal prediction procedures, two key aspects come into play: correctness and adaptivity. Correctness checks verify implementation accuracy, particularly focusing on coverage satisfaction, which demands a detailed examination of finite-sample variability. Meanwhile, adaptivity extends beyond average set size, requiring the procedure to produce small sets for straightforward inputs and larger ones for more complex ones, accurately mirroring model uncertainty.

## 4.2 ConformaSight in Practice: A Sample Scenario

In this section, we demonstrate a practical example which gives notions of algorithm usability overall and interpretation of output. We create a sample scenario with Breast Cancer [50] dataset to produce model agnostic and post-hoc explanations with ConformaSight, it is necessary to split dataset as follows.

The user should reserve a good amount of data also for calibration ( $X_{Cal}, y_{cal}$ ) unlike traditional train/test split. There is no rule to set exact number

of calibration data instances, one should adjust it depending on the coverage metric. However, setting it  $N=1000$  is sufficient for diverse purposes [25]. Subsequently, we train a basic XGB Classifier with default settings [53] to do binary classification. However, you can train any deterministic or probabilistic classifier at this point because the method works in model-agnostic nature.

---

**Algorithm 5:** ConformoSight Full Algorithm
 

---

**Input:** Classifier  $\hat{f}$ , Calibration Set  $(X^{(cal)}, y^{(cal)})$ , Test Set  $(X^{(test)}, y^{(test)})$ , Perturbation Range [min\_severity, max\_severity], Error Rate  $\alpha \in [0, 1]$ , Class Labels  $l$

**Output:** Feature Importance Table  $w$

**Baseline Calibration:**

// Perform the following step as we defined in Algorithm 3.

$[W_{cov}^1, W_{set}^1, C^1] \leftarrow \text{Calibrate}(X^{(cal)}, y^{(cal)}, X^{(test)}, y^{(test)}, \alpha, l, \hat{f});$

**Perturbation Generation:**

$perturbed\_datasets \leftarrow \text{GeneratePerturbations}(X_{Cal}, noise, [\min, \max], n);$

**Relative Changes:**

// Perform the following step as we defined in Algorithm 4.

$[\Delta C, \Delta S] \leftarrow$

$\text{CalculateRelativeChanges}([W_{cov}^1, W_{set}^1, C^1], perturbed\_datasets);$

**Call The ConformoSight Explainer:**

// User specifies explainer metric and noise arguments

$metric = type\_of\_metric \in \{“coverage”, “set-size”, “pred-set”\}$

$noise = noise\_type \in \{“Gaussian”, “Uniform”\}$

feature\_importance\_table =

$\text{plot\_conformal\_explainer}(\hat{f}, X^{(cal)}, y^{(cal)}, X^{(test)}, y^{(test)}, \alpha, l, metric, noise)$

**return**  $w$ ;

---

We developed a straightforward method called `plot_conformal_explainer`, which accepts several parameters. These parameters include  $\alpha$ , as defined in Eq. 1,  $X_{Cal}$ ,  $y_{cal}$ ,  $X_{test}$ , and  $y_{test}$ , as illustrated in Listing 1.1. Additionally, the method takes the `noise_type` parameter, which can be either “gaussian” or “uniform”, and the `type_of_metric` parameter, which can take on values such as “coverage”, “set\_size”, or “pred\_set”.

**Listing 1.1.** Splitting the data into training, testing, and calibration sets

```

1 # Features and labels extraction
2 y = df["target"]
3 X = df.drop(["target"], axis=1)
4 # Train, test, and calibration split
5 X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.5, random_state=42, stratify=y)

```

```

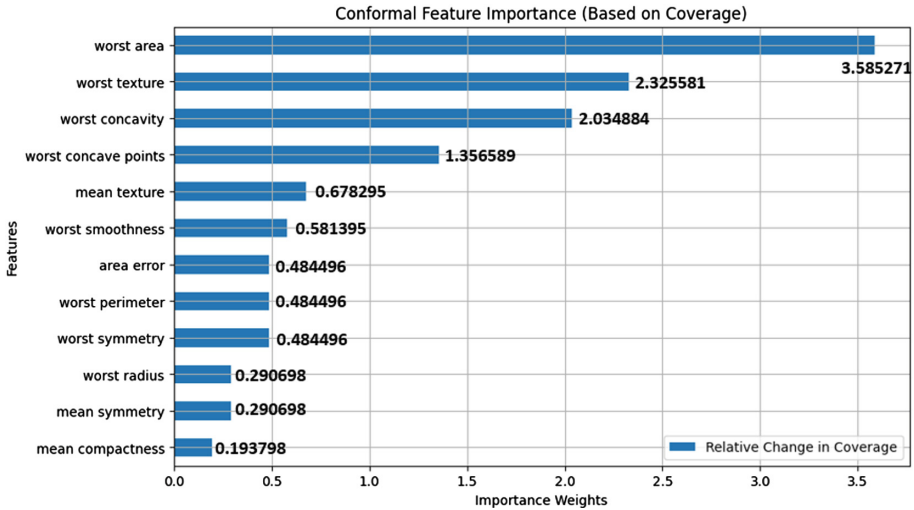
6 X_Cal, X_test, y_cal, y_test = train_test_split(X_test, y_test,
        test_size=0.5, random_state=42)
    
```

Moreover, as demonstrated in Listing 1.2, if the `type_of_metric` is specified as “coverage”, the method calculates relative changes based on the coverage metric, as shown in Fig. 1. Alternatively, if `type_of_metric` is set to “set.size”, the relative change calculations are performed according to the “set.size” metric. Lastly, when `type_of_metric` is specified as “pred.set”, the method generates class-specific changes in coverage, providing coverage calculations tailored to each class type. Further, with other functions one can set the range of counterfactual perturbations. Due to this flexibility, it provides wide range of experimentation options even to non-experts.

**Listing 1.2.** Calling the Plot Conformal Explainer Method

```

1 type_of_metric = "coverage" # Choose conformal metric
2 noise_type = "gaussian" # Choose noise type
3 alpha = 0.05 # Set the error rate
4 plot_conformal_explainer(classifier_base, X_Cal, y_cal, X_test,
        y_test, alpha, class_labels, type_of_metric, noise_type)
    
```



**Fig. 1.** Sample Output of Plot Conformal Explainer Method with Breast Cancer Dataset The values represent the mean relative change in coverage and non-zero values are filtered in the sample demonstration.

Moreover, for class-wise feature importance investigation, by setting the `type_of_metric` as “pred.set” we can get the output as illustrated in Table 1 with the exact values of change in relative class-wise coverage.

**Table 1. Mean Class-Wise Relative Change in Coverage** MCRC (Mean Class-Wise Relative Change) in Coverage: The first five rows for each class are filtered.

Group: Benign		Group: Malignant	
Feature	MCRC	Feature	MCRC
worst area	6.01	worst concavity	5.46
worst perimeter	1.85	worst texture	4.16
worst texture	1.23	worst radius	2.08
mean texture	1.23	mean concave points	1.82
mean concave points	0.92	worst symmetry	1.56

For instance, looking at the “worst area” feature, which demonstrates a MCRC in coverage of 6.01 for the Benign group, we can observe that this feature significantly affects the correct classifications in the benign class more than in the malignant class. This suggests that larger tumor areas might be more indicative of benign tumors, influencing classification outcomes accordingly. Similarly, examining the “worst concavity” feature with a MCRC of 5.46 for the Malignant group, we find that this feature has a substantial impact on the correct classifications of malignant tumors compared to benign tumors.

### 4.3 Computational Complexity of the ConformaSight

The computational complexity of the provided algorithm for perturbation generation can be described as  $O((n_{\text{cat}} + n_{\text{num}}) \times s)$ , where  $n_{\text{cat}}$  represents the number of categorical columns,  $n_{\text{num}}$  represents the number of numerical columns, and  $s$  denotes the maximum severity level. The computational complexity of Conformal Prediction varies depending on factors such as dataset size, model complexity. Generally, it involves training the models (with complexities varying from  $O(n^2 \cdot d)$  to  $O(n^3 \cdot d)$  for SVM [49]), calibrating conformity scores (ranging from  $O(n^2)$  to  $O(n \log n)$ ), and predicting new instances ( $n$ : data points,  $d$ : dimensions of each data point). Finally, the complexity for calculating relative changes in weighted coverage and set size for  $N$  perturbed datasets, each containing  $M$  instances, is  $O(N \cdot M)$ . The scripts and sample notebooks can be found at <https://github.com/rabia174/ConformaSight>.

## 5 Experiments and Evaluations

In this section, we begin by outlining the experimental settings, which encompass the datasets utilized in our evaluations, the configurations of the models and explainers employed, and the evaluation metrics employed for assessment. Subsequently, we delve into a comprehensive analysis of the qualitative and quantitative results derived from our evaluations. Experiments are implemented with a Python 3.8 Kernel on an Intel vPRO i5 processor equipped device.

### 5.1 Experimental Settings

**Datasets.** We assess the generalizability of our approach using three distinct publicly available datasets with varying sizes and dimensions. Consistent with our framework, we concentrate on classification problems characterized by multivariate datasets comprising both categorical and numerical columns. Table 2 presents a general overview of the datasets that have been used in our assessments, respectively, Breast Cancer [50], Glass Identification [51], and Cover Type [52].

**Table 2.** Summary of Datasets used to make assessments

Dataset	Type	Instances	Features	Target Column	Class
Breast Cancer	Real-world	569	30	Malignant or Benign	2
Glass Identification	Real-world	214	9	Type of glass	6
Cover Type	Real-world	581012	54	Forest Cover Type	7

The Breast Cancer dataset includes data on breast cancer patients’ characteristics, with 569 instances and 30 features, classified into 2 classes indicating tumor malignancy. The Glass Identification dataset provides information on glass samples’ attributes, with 214 instances and 9 features, categorized into 6 classes based on the type of glass. The Cover Type dataset contains forest cover type data derived from cartographic variables, with 581,012 instances and 54 features, categorized into 7 classes.

**Models.** We opted four distinct models that can be found in Table 3. By using diverse models, we aim to harness complementary advantages in modeling diverse patterns and quantifying uncertainty, thereby enhancing the comprehensibility of our analysis.

**Table 3.** Summary of Models and Parameters/Settings

Model Type	Library/Framework	Parameters/Settings
XGBoost	DMLC XGBoost [53]	Max depth: 3 Learning rate: 0.1 Number of estimators: 100 Objective: multi-softmax Booster: gbtree
Neural Network	TensorFlow [54]	Dense layers: (128, 64, 32) Activation: (relu, relu, softmax) Optimizer: Adam Loss: Sparse Cat. Cross-entropy Epochs: 10 Batch size: 32
SVM [49]	scikit-learn [55]	Kernel: Linear C: 1.0 Gamma: scale
Logistic Regression [56]	scikit-learn [55]	Penalty: l2 C: 1.0 Solver: lbfgs

**Explainers.** To demonstrate performance of our proposed approach ConformaSight, we compare its outputs with those generated by established explanation methods aimed at elucidating the general behavior of the model. Firstly, we employ a model like XGB interpretable by nature. We reference XGB’s “Gain” feature importance metric over “Cover” and “Weight” as it closely aligns with our approach [53]. Additionally, we utilize Permutation-Based feature importance, a widely adopted method that assesses the importance of input features by shuffling their values and measuring the impact on model performance [57]. Furthermore, we incorporate mean SHAP values computed by averaging local SHAP explanations, which offer insights into the contribution of each feature to individual predictions while maintaining a global perspective [58, 59]. By employing these explanation methods with their default settings, we aim to provide a comprehensive assessment of ConformaSight’s performance and elucidate its interpretability.

**Evaluation Setup.** In our evaluation, we employed three approaches: Effectiveness, Faithfulness, and Resilience. For this purpose, we initially trained a base model using all available features from the dataset to assess and quantify feature importance with diverse explainers. Subsequently, we compiled a feature-set comprising a subset of features and retrain the base models. This subset, referred to as the  $F_{golden\_set}$  was curated by selecting the top  $N$  most significant features identified through the explanation methods discussed in the previous section, along with our proposed method, ConformaSight. In case of models with XGB and LR which are intrinsically interpretable, the top  $N$  most significant features referred to as  $F_{gold}$  to avoid confusion when evaluating Faithfulness. Finally, when evaluating Resilience the explanations extracted from the test set without noise (original) referred to as  $F_{org\_exp}$  and the explanations extracted from noisy test set referred to as  $F_{noisy\_exp}$ . During all evaluations, we set ConformaSight perturbation severities ranging from 0.1 to 0.95 with a step size of 0.05 for numerical columns and from 1 to 10 with a step size of 1 for categorical columns.

**Effectiveness:** To measure effectiveness, we trained all models in Table 3 using the possible minimal golden-set obtained from each explainer and ConformaSight (comprising the same number of data instances but with less number of features, resulting in a significant reduction in dimensions). We then produced the following Weighted F1-Score on the same test set used for prediction:

$$\text{Weighted F1-Score} = \frac{2 \times \left( \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FP_i)} \right) \times \left( \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FN_i)} \right)}{\left( \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FP_i)} \right) + \left( \frac{\sum_{i=1}^N TP_i}{\sum_{i=1}^N (TP_i + FN_i)} \right)} \quad (9)$$

where True Positive (TP) represents correctly predicted positive instances, False Negative (FN) indicates incorrectly predicted negative instances, False Positive (FP) denotes incorrectly predicted positive instances, and True Negative (TN) signifies correctly predicted negative instances.

**Resilience:** We assess ConformaSight’s resilience by introducing Uniform noise that changes the distribution shift in the same test set which we produce explanations (we make sure that the distribution is significantly changed with Kolmogorov-Smirnov (KS) test [60,61]). Subsequently, we measure the extent to which the explanations of the noisy test set ( $F_{\text{noisy\_exp}}$ ) include those of the original test set ( $F_{\text{org\_exp}}$ ), as demonstrated in Eq. 10.

$$\text{Extent of Common Features} = \frac{|F_{\text{org\_exp}} \cap F_{\text{noisy\_exp}}|}{|F_{\text{org\_exp}}|} \quad (10)$$

**Faithfulness:** We measure the faithfulness of explanations with classifiers that can be interpretable easily by design, such as XGBoost (XGB) and Logistic Regression (LR). After training these classifiers using the entire feature set, we extract a subset of features (top  $N$ ) identified as most significant. This subset serves as our baseline gold standard ( $F_{\text{gold}}$ ), reflecting the features deemed important by an inherently explainable model. Subsequently, we generate explanations, extract top  $N$  from each ( $F_{\text{exp}}$ ) and measure the extent to which these explanations accurately capture the features in the gold standard set as demonstrated in Eq. 11. This metric assesses the overall faithfulness of our approach to diverse models.

$$\text{Fraction of common features} = \frac{|F_{\text{gold}} \cap F_{\text{exp}}|}{N} \quad (11)$$

## 6 Results and Discussion

In this section, we present our results belonging to evaluations of ConformaSight’s feature importance outputs, focusing on effectiveness, faithfulness, and resilience perspectives.

**Evaluating Effectiveness.** The performance of various predefined models in Table 3 on the Breast Cancer and Glass Identification golden-sets is presented in Table 4. ConformaSight metrics often outperform other global explainability techniques across both datasets, irrespective of the model used.

The experiments reported in Table 4 highlight the superiority of ConformaSight metrics compared to other feature selection techniques, across various models, in accurately classifying both the Breast Cancer and Glass Identification datasets.

Moreover, to demonstrate the effectiveness of our model also on models like neural networks (NN) and large-scale datasets, we conducted experiments on another real-world dataset, Cover Type. The results, summarized in Table 5, showcase the weighted F1-Score rates obtained for the Cover Type dataset using different models. Notably, ConformaSight often outperformed other techniques also in a large-scale and real-world dataset. Conformal prediction tends to perform better with larger datasets with wider classes due to its ability to provide calibrated confidence estimates for predictions.

**Table 4. Comparison of Average F1-Score of Small-Scale Real-World Datasets.** The table displays average F1-Score values from diverse models for Breast Cancer and Glass Identification datasets. ConformaSight metrics utilize Gaussian noise perturbation. From all explainers, we picked the top 7 features for the Breast Cancer and top 6 for Glass Identification, aiming for optimal performance with the possible minimal set of features. All model details can be found in Table 3.

	Breast Cancer			Glass Identif.		
	XGB	SVM	LR	XGB	SVM	LR
Random	0.88	0.88	0.88	0.68	0.56	0.58
Permutation	0.92	0.91	0.92	0.70	0.61	0.62
SHAP	0.92	0.92	0.92	0.63	0.59	0.62
XGB (Gain)	0.90	–	–	0.71	–	–
ConformaSight Coverage-Based	<b>0.95</b>	<b>0.94</b>	<b>0.93</b>	<b>0.78</b>	<b>0.64</b>	<b>0.64</b>
ConformaSight Set Size-Based	<b>0.94</b>	<b>0.93</b>	<b>0.93</b>	<b>0.73</b>	<b>0.64</b>	<b>0.62</b>

**Table 5. Comparison of Average F1-Scores of Large-Scale Real-World Dataset.** The table displays average F1-Score values from diverse models for Cover Type dataset. ConformaSight metrics utilize Gaussian noise perturbation. For all explainers, we picked the top 10 features. All model details can be found in Table 3.

	Cover Type			
	XGB	NN	SVM	LR
Random	0.39	0.53	0.37	0.29
Permutation	0.85	0.70	0.70	0.65
SHAP	0.85	0.70	0.70	0.69
XGB (Gain)	0.69	–	–	–
ConformaSight Coverage-Based	<b>0.87</b>	<b>0.84</b>	<b>0.75</b>	<b>0.72</b>
ConformaSight Set Size-Based	<b>0.87</b>	<b>0.81</b>	<b>0.73</b>	<b>0.70</b>

**Evaluating Resilience.** We present in Table 6 the measure to which extent noisy test set (distribution shift is verified with Kolmogorov-Smirnov as  $p < 0.05$ ) explanations cover original test set explanations. Specifically, the percentage of common features between the explanations derived from the original and noisy test sets remains consistently high, ranging from 84.6% to 100%.

Overall, the fraction of common features between the original and noisy test sets was found to be suggesting a great substantial overlap in the features analyzed across both datasets. Furthermore, the observed resilience highlights the potential utility of the models in practical applications where data may be subject to inherent variability or noise. The ability of producing consistent and resilient explanations in dramatically noisy test sets underscores

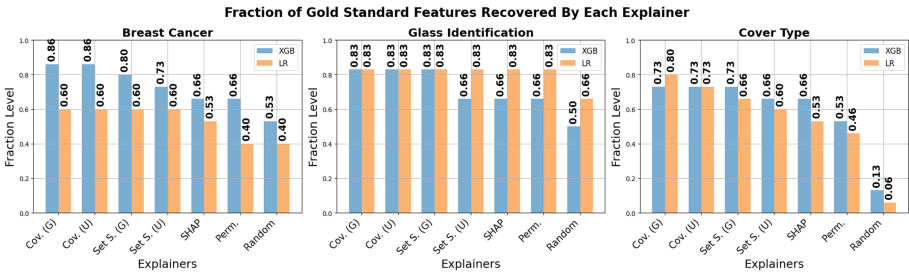


ConformaSight’s adaptability to highly uncertain environments, thus suggesting promising prospects for their application in real-world scenarios.

**Table 6. Resilience Assessment Table** Calibration set perturbation were set to Gaussian. Glass Identification is excluded in this experiment due to low dimensions. All features in the explanation set were taken into consideration.

Type	Breast Cancer			Cover Type		
	XGB	SVM	LR	XGB	SVM	LR
Coverage-Based	84.6%	85.7%	<b>100%</b>	94.1%	96.7%	<b>100%</b>
Set Size-Based	91.6%	93.3%	<b>100%</b>	94.4%	<b>100%</b>	<b>100%</b>

**Evaluating Faithfulness.** In assessing faithfulness, we evaluate the extent to which explanation methods accurately represent a model’s decision-making process. Our findings, illustrated in Fig. 2, reveal that ConformaSight’s Coverage-Based and Set Size-Based explanations demonstrate robust faithfulness, with fractions ranging from approximately 73% to 86% across all datasets and machine learning models we utilized. This underscores the reliability of ConformaSight’s explanation techniques in providing accurate insights into model predictions, thereby fostering transparency and trust in AI systems.



**Fig. 2. Fraction of Gold Standard Features Recovered By Each Explainer.** G (Gaussian Noise), U (Uniform Noise), Perm. (Permutation), Cov. (Coverage-Based), Set S. (Set Size-Based). For both datasets top 15 features from all explanations ( $F_{exp}$ ) including intrinsic explanations( $F_{gold}$ ) were picked.

## 7 Conclusion, Limitations and Future Work

In this paper, we propose a novel method for generating explanations in conformal prediction classifiers by incorporating model uncertainty estimates into the explanation generation process. Our approach addresses class-imbalance and distribution shift issues by creating counterfactual instances through realistic noise

introduction to calibration sets, followed by model re-calibration. We measure mean relative changes in coverage and set-size metrics within the conformal prediction context through systematic perturbation on each feature in the calibration set. Our method allows users to specify noise generation intervals, types, and coverage rates, offering flexibility in producing diverse explanations while ensuring ground truth containment within prediction sets. Explanations can be generated in three ways: mean relative change in coverage, mean relative change in set-size, and mean class-wise relative change in coverage.

Evaluations on three real-world datasets using diverse models (XGBoost, SVM, LR, NN) assess effectiveness, faithfulness and resilience of our explanations. ConformaSight often outperforms mean SHAP, Permutation, and even intrinsic explanations with minimal feature sets on the same classifier. This strength of ConformaSight lies in its ability to effectively identify the minimal dataset required to maximize model performance. Leveraging its capability to capture and rank features that enhance coverage in classifiers, ConformaSight identifies and prioritizes features that contribute most significantly to model performance, thereby optimizing classifier performance with minimal input data. This highlights the superior effectiveness of ConformaSight in providing comprehensive and accurate explanations compared to traditional and intrinsic explanation methods. Notably, our experiments have demonstrated that ConformaSight consistently produces resilient explanation sets, even when confronted with significantly perturbed versions of the same test sets. This highlights the robustness of ConformaSight in providing stable and reliable explanations across varying conditions thanks to the nature of conformal-prediction.

However, conformal prediction-based models face challenges in computational expense, scalability with large datasets or complex models, and conservative predictions leading to wider intervals. Model choice and feature representation quality also affect effectiveness, robustness, and generalization.

For future improvements, we propose offering variations of conformal procedures for generating explanations, exploring adaptive prediction sets and group-balanced conformal prediction techniques. Enabling users to provide custom uncertainty estimates and noise generation functions would enhance flexibility and robustness. Extending our approach to regression and time-series data, as well as producing explanations for data characteristics like bias and outliers, are additional areas for enhancement.

**Disclosure of Interests.** The authors declare no relevant competing interests associated with the content of this article.

## References

1. Weber, L., Lapuschkin, S., Binder, A., Samek, W.: Beyond explaining: opportunities and challenges of XAI-based model improvement. *Inf. Fus.* **92**, 154–176 (2023)
2. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 1189–1232 (2001)

3. Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **82**(4), 1059–1086 (2020)
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
5. Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable’s importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.* **20**(177), 1–81 (2019)
6. Wei, P., Lu, Z., Song, J.: Variable importance analysis: a comprehensive review. *Reliab. Eng. Syst. Saf.* **142**, 399–432 (2015)
7. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
8. Ribeiro, M. T., Singh, S., Guestrin, C.: “Why should i trust you?” Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016)
9. Loquercio, A., Segu, M., Scaramuzza, D.: A general framework for uncertainty estimation in deep learning. *IEEE Robot. Autom. Lett.* **5**(2), 3153–3160 (2020)
10. Malinin, A., Gales, M.: Predictive uncertainty estimation via prior networks. In: *Advances in Neural Information Processing Systems*, vol. 31 (2018)
11. Nandy, J., Hsu, W., Lee, M.L.: Towards maximizing the representation gap between in-domain & out-of-distribution examples. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 9239–9250 (2020)
12. Snowling, S.D., Kramer, J.R.: Evaluating modelling uncertainty for model selection. *Ecol. Model.* **138**(1–3), 17–30 (2001)
13. Abdar, M., et al.: A review of uncertainty quantification in deep learning: techniques, applications and challenges. *Inf. Fus.* **76**, 243–297 (2021)
14. van Kuijk, K., Dirksen, M., Seiler, C.: Conformal regression in calorie prediction for team Jumbo-Visma. *arXiv preprint [arXiv:2304.03778](https://arxiv.org/abs/2304.03778)* (2023)
15. Wilm, A., et al.: Skin Doctor CP: conformal prediction of the skin sensitization potential of small organic molecules. *Chem. Res. Toxicol.* **34**(2), 330–344 (2020)
16. Zhan, X., Wang, Z., Yang, M., Luo, Z., Wang, Y., Li, G.: An electronic nose-based assistive diagnostic prototype for lung cancer detection with conformal prediction. *Measurement* **158**, 107588 (2020)
17. Henne, M., Schwaiger, A., Weiss, G.: Managing uncertainty of AI-based perception for autonomous systems. In: *AISafety@ IJCAI*, pp. 11–12 (2019)
18. Thuy, A., Benoit, D.F.: Explainability through uncertainty: trustworthy decision-making with neural networks. *Eur. J. Oper. Res.* (2023)
19. Balasubramanian, V., Ho, S.S., Vovk, V.: Conformal prediction for reliable machine learning: theory, adaptations and applications. *Newnes* (2014)
20. Arrieta, A.B., et al.: Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fus.* **58**, 82–115 (2020)
21. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* **6**, 52138–52160 (2018). <https://doi.org/10.1109/ACCESS.2018.2870052>
22. Sovrano, F.: How to explain: from theory to practice, [Dissertation thesis], Alma Mater Studiorum Università di Bologna. Dottorato di ricerca in Data science and computation, 34 Ciclo. (2023). <https://doi.org/10.48676/unibo/amsdottorato/10943>
23. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Cham (2006). <https://doi.org/10.1007/978-0-387-45528-0>

24. Gawlikowski, J., et al.: A Survey of Uncertainty in Deep Neural Networks (2021). <http://arxiv.org/abs/2107.03342>
25. Angelopoulos, A.N., Bates, S.: A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint [arXiv:2107.07511](https://arxiv.org/abs/2107.07511) (2021)
26. Vovk, V., Gammerman, A., Saunders, C.: Machine-learning applications of algorithmic randomness. In: International Conference on Machine Learning, pp. 444–453 (1999)
27. Angelopoulos, A.N., Bates, S., Malik, J., Jordan, M.I.: Uncertainty sets for image classifiers using conformal prediction. In: International Conference on Learning Representations (2021)
28. Angelopoulos, A.N., Bates, S., Candès, E.J., Jordan, M.I., Lei, L.: Learn then test: calibrating predictive algorithms to achieve risk control. [arXiv:2110.01052](https://arxiv.org/abs/2110.01052) (2021)
29. Tibshirani, R.J., Foygel Barber, R., Candès, E., Ramdas, A.: Conformal prediction under covariate shift. In: Advances in Neural Information Processing Systems, vol. 32, pp. 2530–2540 (2019)
30. Cauchois, M., Gupta, S., Ali, A., Duchi, J.C.: Robust validation: confident predictions even when distributions shift. [arXiv:2008.04267](https://arxiv.org/abs/2008.04267) (2020)
31. Barber, R.F., Candès, E.J., Ramdas, A., Tibshirani, R.J.: Conformal prediction beyond exchangeability. [arXiv:2202.13415](https://arxiv.org/abs/2202.13415) (2022)
32. Gibbs, I., Candès, E.: Adaptive conformal inference under distribution shift. [arXiv:2106.00170](https://arxiv.org/abs/2106.00170) (2021)
33. Artelt, A., Visser, R., Hammer, B.: I do not know! but why? - Local model-agnostic example-based explanations of reject. *Neurocomputing* **558**, 126722 (2023). ISSN 0925-2312. <https://doi.org/10.1016/j.neucom.2023.126722>
34. Carlevaro, A.N., Narteni, S., Dabbene, F., Muselli, M., Mongelli, M.: CONFIDERA: a novel CONFormal Interpretable-by-Design score function for Explainable and Reliable Artificial Intelligence (2023)
35. Johansson, U., Löfström, T., Boström, H., Sönströd, C.: Interpretable and specialized conformal predictors. In: Proceedings of the Eighth Symposium on Conformal and Probabilistic Prediction and Applications, PMLR, vol. 105, pp. 3–22 (2019)
36. Alkhatib, A., Bostrom, H., Ennadir, S., Johansson, U.: Approximating score-based explanation techniques using conformal regression. In: Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications, PMLR, vol. 204, pp. 450–469 (2023)
37. Park, H.: Providing post-hoc explanation for node representation learning models through inductive conformal predictions. *IEEE Access* **11**, 1202–1212 (2023). <https://doi.org/10.1109/ACCESS.2022.3233036>
38. Marx, C., Park, Y., Hasson, H., Wang, Y., Ermon, S., Huan, L.: But are you sure? An uncertainty-aware perspective on explainable AI. In: Proceedings of The 26th International Conference on Artificial Intelligence and Statistics, PMLR, vol. 206, pp. 7375–7391
39. Qian, W., Zhao, C., Li, Y., Ma, F., Zhang, C., Huai, M.: Towards Modeling Uncertainties of Self-explaining Neural Networks via Conformal Prediction, 2401.01549, cs.LG (2024)
40. Bykov, K., et al.: How much can i trust you?—quantifying uncertainties in explaining neural networks. arXiv preprint [arXiv:2006.09000](https://arxiv.org/abs/2006.09000) (2020)
41. Delaney, E., Greene, D., Keane, M.T.: Uncertainty estimation and out-of-distribution detection for counterfactual explanations: pitfalls and solutions. arXiv preprint [arXiv:2107.09734](https://arxiv.org/abs/2107.09734) (2021)

42. Vovk, V., Gammerman, A., Shafer, G.: *Algorithmic Learning in a Random World*. Springer, New York (2005). <https://doi.org/10.1007/b106715>
43. Shafer, G., Vovk, V.: A tutorial on conformal prediction. *J. Mach. Learn. Res.* **9**, 371–421 (2008)
44. Zadrozny, B., Elkan, C.: Learning and making decisions when costs and probabilities are both unknown. In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2001)
45. Box, G.E.P., Muller, M.E., Tiao, G.C.: Robustness in the strategy of scientific model building. In: *Robustness in Statistics*, pp. 201–236. Academic Press (1978)
46. Devroye, L.: *Non-uniform Random Variate Generation*. Springer, New York (1986). <https://doi.org/10.1007/978-1-4613-8643-8>
47. Franceschi, J.Y., Fawzi, A., Fawzi, O.: Robustness of classifiers to uniform  $\ell_p$  and Gaussian noise. In: *International Conference on Artificial Intelligence and Statistics*, pp. 1280–1288. PMLR (2018)
48. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: *International Conference on Machine Learning* (2017)
49. Vapnik, V., Cortes, C.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
50. Zwitter, M., Soklic, M.: Breast Cancer. UCI Machine Learning Repository (1988). <https://doi.org/10.24432/C51P4M>
51. German, B.: Glass Identification. UCI Machine Learning Repository (1987). <https://doi.org/10.24432/C5WW2P>
52. Blackard, J.: Coverttype. UCI Machine Learning Repository (1998). <https://doi.org/10.24432/C50K5N>
53. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794 (2016)
54. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems. Software available from tensorflow.org (2015)
55. Pedregosa, F., et al.: *JMLR* **12**, pp. 2825–2830 (2011)
56. Cox, D.R.: The application of the logistic function to experimental data. *Biometrics* **14**(1), 59–67 (1958)
57. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
58. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
59. Lundberg, S.M., et al.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 56–67 (2020)
60. Fasano, G., Franceschini, A.: A multidimensional version of the Kolmogorov-Smirnov test. *Mon. Not. R. Astron. Soc.* **225**(1), 155–170 (1987)
61. Virtanen, P., et al.: SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Meth.* **17**(3), 261–272 (2020)



# Differential Privacy for Anomaly Detection: Analyzing the Trade-Off Between Privacy and Explainability

Fatima Ezzeddine<sup>1,2(✉)</sup>, Mirna Saad<sup>3</sup>, Omran Ayoub<sup>1</sup>, Davide Andreoletti<sup>1</sup>, Martin Gjoreski<sup>2</sup>, Ihab Sbeity<sup>3</sup>, Marc Langheinrich<sup>2</sup>, and Silvia Giordano<sup>1</sup>

<sup>1</sup> University of Applied Sciences and Arts of Southern Switzerland, Manno, Switzerland

<sup>2</sup> Università della Svizzera italiana, Lugano, Switzerland  
fatima.ezzeddine@usi.ch

<sup>3</sup> Lebanese University, Beirut, Lebanon

**Abstract.** Anomaly detection (AD), also referred to as outlier detection, is a statistical process aimed at identifying observations within a dataset that significantly deviate from the expected pattern of the majority of the data. Such a process finds wide application in various fields, such as finance and healthcare. While the primary objective of AD is to yield high detection accuracy, the requirements of explainability and privacy are also paramount. The first ensures the transparency of the AD process, while the second guarantees that no sensitive information is leaked to untrusted parties. In this work, we exploit the trade-off of applying Explainable AI (XAI) through SHapley Additive exPlanations (SHAP) and differential privacy (DP). We perform AD with different models and on various datasets, and we thoroughly evaluate the cost of privacy in terms of decreased accuracy and explainability. Our results show that the enforcement of privacy through DP has a significant impact on detection accuracy and explainability, which depends on both the dataset and the considered AD model.

**Keywords:** Explainable AI · Differential Privacy · Anomaly Detection

## 1 Introduction

Within the realm of data-driven decision-making, anomalies, which are data points exhibiting statistically significant deviations from expected patterns, have multifaceted impact. Anomalies are indicative of errors or inconsistencies within the data, but they also hold the potential to reveal novel or critical situations. Deviations are subsequently flagged for further investigation as they can be highly informative, often signaling underlying issues or emerging trends. For instance, in the context of cyber-security, an anomaly might indicate a security breach or an attempted attack. In healthcare, it could highlight rare illnesses. The timely identification of anomalies is crucial for maintaining security,

F. Ezzeddine and M. Saad—Co-first author.

efficiency, and safety in various fields, such as cyber-security, healthcare, network monitoring, and transportation [2, 25]. Therefore, developing highly effective Anomaly Detection (AD) systems is of paramount importance, as they represent a critical tool to address the challenge of detecting these anomalies [10]. By leveraging diverse statistical and machine learning (ML) techniques, AD establishes a statistical baseline for normal behavior within a dataset.

Recently, a main requirement of AD systems emerged, in addition to that of high predictive performance, which is the need of providing stakeholders with relevant information on why and how a specific data point is considered an anomaly, such as to enhance the transparency of AD systems, with the final aim of fostering trust in these systems [5, 19, 47]. In addition to that, privacy guarantees are also essential for AD in scenarios as data owners may delegate the AD task to a third party in possession of the technical expertise needed for effective AD. This third-party data access introduces privacy concerns, especially when dealing with sensitive data such as healthcare information that includes confidential patient medical histories [14, 35, 52]. In this context, we are confronted with the challenge of ensuring both transparency and privacy in AD systems. To ensure the privacy of data and the transparency of decisions, differential privacy (DP) [16] and explainable artificial intelligence (XAI) have been the main methods attracting the attention of the research community, respectively.

A conflict however arises between ensuring transparency (through XAI) and privacy (through DP) due to their opposing goals. XAI aims to provide insights into model behavior for transparency, while privacy-preserving solutions obscure data to prevent data leakage. Specifically, XAI techniques are proposed to demystify the inner workings of complex ML models and AD through different types of explanations such as, e.g., feature importance, which scores the contribution and impact of each feature on the model’s output, permitting data owners to identify which features in the data were most influential in identifying an anomaly. DP, on the contrary, works by injecting calibrated noise into the data before it is released, introducing a quantifiable privacy guarantee, and allowing control of the level of information revealed about individual data points.

The intersection of XAI with privacy-preserving techniques presents complex and nuanced challenges. As privacy concerns arise, the implementation of privacy-preserving mechanisms often involves obfuscating sensitive information, which may not only impact the performance and utility of AD models but also their explainability. Therefore, there exists a critical need to precisely quantify how privacy-preserving techniques such as DP affect the explainability of any ML or AD systems [28]. In this paper, to investigate the complex relationship among DP, AD, and XAI. As a XAI framework, we rely on SHapley Additive exPlanations (SHAP) [30]. We focus on SHAP since it is widely applied for feature importance in tabular data. To investigate this interplay in the context of AD, we formulate two research questions (RQs) and address them as follows:

1. *RQ.1 To what extent does increasing DP noise affect the fidelity and stability of SHAP values in AD?* We explore XAI in DP-AD, specifically, we investigate SHAP with a focus on understanding how the SHAP values of AD

models are affected by varying levels of DP noise on AD. By extensively analyzing quantitatively and qualitatively the output of SHAP, we shed light on the potential trade-offs between privacy protection and model explainability.

2. *RQ2. To what extent does the application of DP impact the performance and explainability of AD algorithms designed for specific anomaly types (local vs. global)?* Leveraging the inherent specialization of AD algorithms towards distinct anomaly types (local vs. global), this research delves into the potential for DP and its impact on their performance and explainability under varying privacy constraints.

The paper is organized as follows. Section 2 discusses related work. Section 3 presents background on AD and DP. Section 4 details the problem and objectives. Section 5 presents the experimental setup and evaluation settings. Section 6 showcases the obtained results, and analyzes their implications, offering valuable insights for practitioners navigating the trade-off between privacy and explainability in AD models. Section 7 concludes the paper.

## 2 Related Work

In this section, we first discuss studies that have applied either privacy-preserving or explainability techniques to AD and then studies that have investigated the impact of privacy on the model’s explainability. The related works highlight the growing tension between privacy and explainability. While research has explored privacy-preserving techniques and explainable models in many contexts, their intersection with AD remains largely unaddressed. This is particularly crucial because AD often deals with sensitive data and requires different considerations compared to traditional classification and regression tasks. In this paper, we analyze this intersection to reveal insights into the trade-offs involved, guiding the development of future robust, explainable, and privacy-preserving AD methods that empower informed decision-making.

### 2.1 Privacy-Preserving Anomaly Detection

AD methods play a crucial role in diverse sectors such as finance, healthcare, transportation, and smart grids [3, 14, 22, 25]. Most existing methods rely primarily on unsupervised ML techniques (e.g., [7, 10, 27, 38, 56]), with some supervised approaches as well (e.g., [24]). Numerous AD algorithms have been proposed in the literature, and each of them comes with strengths and weaknesses and is suitable for specific contexts and data types. For example, Isolation Forest (iForest) and Local Outlier Factor (LOF) are well suited for tabular data, Deep Learning-based AD with auto-encoders is suitable for images and time series data [10, 11, 29]. Several recent studies on AD have proposed employing privacy-preserving techniques to address the critical challenge of balancing effective AD with individual data protection. These techniques mitigate public concern over data sharing, reduce the risk of re-identification from



anonymized data, facilitate compliance with data privacy regulations like the General Data Protection Regulation (GDPR), and foster collaboration between organizations, leading to more comprehensive and effective AD across various sectors [14, 32]. To achieve AD while safeguarding data privacy, several methods have been employed such as DP [6, 15, 18], homomorphic encryption and training on encrypted data [4, 36, 48], Secure Multi-Party Computation [54], and even novel, custom-designed approaches [31], such as generating synthetic data [35], or approaches for specific cases such as body movement and power systems [6, 26].

Specifically, among the various techniques, several studies have explored the integration of AD algorithms and DP, e.g., [15, 42]. Notably, Du et al. [15] show how DP can improve the utility of AD and novelty detection, with a focus on detecting poisoning samples in backdoor attacks. Jiang et. al [25] propose a privacy-preserving social network model that utilizes restricted local DP to sanitize user information collection. Moreover, Chukkapalli et al. in [12] propose an approach for privacy-preserving AD in smart farming by adding noise to individual farm data. Giraldo et. al [18] examine how AD can be combined with DP to provide robust privacy and security for individuals. In addition to Degue et al. [14] DP is employed with AD in correlated data to analyze the trade-off between privacy level and detection accuracy of multivariate Gaussian signals.

Other studies have utilized homomorphic encryption and performed training of AD models with encrypted data [4, 20, 54]. For instance, Alabdulatif et al. [4] focus on cloud-based models and propose an AD model that preserves data privacy with reliance on ciphertext, while, Mehnaz et al. [36] present a framework for efficient AD on real-time-series encrypted data. Guo et al. [20] propose an AD scheme for encrypted video bitstreams with format-compliant encryption. Zhang et al. [54] propose a semi-centralized privacy-preserving secure multiparty computation protocol for the PCA-based AD. Other approaches have been proposed based on synthesizing data and generating samples, such as [35]. Mayer et al. [35] analyze many approaches for creating synthetic data and the utility of the created datasets for AD in supervised, semi-supervised, and unsupervised settings. Prioritizing privacy has been the main focus of these recent AD studies. However, explainability in AD is also emerging as a crucial area of research, which we explore in the following subsection.

## 2.2 Explainable Anomaly Detection

Yuan et al. [53] discuss crucial challenges of AD such as trustworthiness, explainability, and robustness. In this context, several studies explore methods for extracting valuable insights for anomalies explanations [5] and as detailed in [44]. For instance, Panjei et al. [44] categorize various types of explanations and analyze existing techniques for interpreting anomalies, paving the way for more meaningful AD analysis. Roshan et al. [46] demonstrate the use of XAI to explain the results of the autoencoder AD model, and in [47] leverages the Kernel SHAP method to explain network anomalies. Moreover, Ravi et al. [45] explore the feasibility and compare the performance of several state-of-the-art

XAI frameworks on Convolutional Autoencoders for building AD systems in the visual domain. Finally, Tritscher et al. [49] highlight the growing interest in categorizing and analyzing these explainable methods based on their access to training data and the specific AD model. These works have demonstrably investigated interpretability and leveraged XAI techniques for AD, but without considering privacy concerns.

### 2.3 Impact of Privacy on Explainability

Recent years have seen a surge in studies investigating the interplay of privacy with XAI [8, 17, 40] as both interpretability and privacy represent a requirement for deploying ML models and datasets. Some studies investigate the inherent trade-off between these concepts, while others propose novel methods to generate privacy-preserving explanations [23, 37, 50]. Bozorgpanah et al. [8] investigate the impact of various privacy-preserving techniques such as masking, and DP noise addition on the effectiveness of regression-based explainability methods utilizing Shapley values and show different behaviors in Shapley for different models by computing correlation metrics. Also, the authors in [40] show the impact of DP on the interpretability of Deep Neural Networks particularly in medical imaging application classification, and show significant visual differences in explanation with DP. Another study [13] investigates the use of example-based explainability models for retinal image analysis. The authors propose leveraging Generative Adversarial Networks (GANs) to generate synthetic examples that provide explanations for model predictions while preserving the privacy of the original retinal images. Nori et al. [41], a method for adding DP to Explainable Boosting Machines enables the training of interpretable classification and regression models with state-of-the-art accuracy while preserving privacy. Harder et al. [21] addresses the challenge of balancing interpretability and privacy in ML models by proposing a novel approach using simple models with locally linear maps to approximate complex models. This method achieves high classification accuracy while providing differentially private explanations for the classifications. Montenegro et al. [37] propose privacy-preserving GANs for privatizing case-based explanations in classification tasks. This GAN incorporates a counterfactual module, enabling the generation of both factual and counterfactual explanations while safeguarding privacy. Moreover, Jetchev et al. [23] introduces a novel, privacy-preserving algorithm for calculating Shapley values on decision tree ensembles within a secure multi-party computation framework, ensuring data privacy. Veugen et al. [50] propose the generation of privacy-friendly explanations by leveraging local foil trees.

## 3 Background

This section introduces the concept of AD and the theoretical background relative to the AD algorithms considered in this work, namely LOF and iForest.

### 3.1 Anomaly Detection Algorithms

The term anomalies refers to data points that deviate significantly from the expected patterns or behaviors observed within a dataset. This deviation can indicate various underlying factors, ranging from irregularities to errors in data collection or processing. AD refers to the process of identifying these anomalies [10]. In many applications, practitioners are particularly interested in finding data points that deviate from their immediate neighbors locally (local anomalies) or globally, referring to data points that deviate significantly from the overall distribution of the data set rather than just its immediate neighbors (global anomalies). To address both local and global anomalies, our study incorporates LOF [9] (for local AD), and iForest [29] (for global AD), which are two unsupervised models that proved scalable and efficient [39].

**Isolation Forest.** The core principle behind iForest [29] lies in constructing decision trees using randomly sampled data to isolate outliers. Given the intrinsic sparsity of anomalies relative to normal data points, their isolation pathways within the iForest exhibit demonstrably shorter lengths, therefore, anomalies are isolated faster within the constructed trees. In other words, the fewer branches that need to be traversed in the tree to isolate a data point (indicating a shorter path), the higher the likelihood that it is an anomaly.

To build a tree, it starts by randomly selecting a feature and a random split value between its minimum and maximum. It then partitions the data into two sets based on the chosen split value. This process is recursively repeated, building out the decision tree structure. A pre-defined maximum tree depth can be set to ensure consistency and avoid excessively deep trees. After that, an anomaly score is computed as the average path length of a data point across all the trees in the forest. Lower scores indicate a higher likelihood of being an anomaly, since deeper paths, indicating more effort to isolate, suggest higher normality, while shallower paths, signifying easier isolation, point toward potential anomalies. Since iForest focuses on random partitioning, faster isolation of anomalies, and independence from local density distribution, it is a strong choice for detecting global anomalies that deviate significantly from the overall data distribution.

**Local Outlier Factor.** LOF [9] algorithm identifies anomalies by comparing the local density of a data point with the density of its  $k$ -nearest neighbors. Local density comparison assesses how much an individual point deviates from its surrounding environment. LOF first identifies the  $k$ -nearest neighbors for each data point. Then, it computes a local reachability density (LRD) that estimates how dense the area surrounding a particular data point is compared to its neighbors. Then, LOF calculates a score for each data point by comparing its LRD to the average LRD of its  $k$ -nearest neighbors. Points with a significantly lower LRD than their neighbors are considered potential outliers, and higher LOF scores indicate higher local density and normality. In contrast, lower scores suggest potential anomalies, deviating significantly from their surrounding data points.

LOF focuses on identifying local anomalies and excels at this task by considering local density measures. This approach makes LOF ideal for identifying local anomalies that deviate from the norm within their specific local area.

### 3.2 Differential Privacy

DP offers a mathematical framework to ensure individual privacy in data analysis. It achieves this by injecting calibrated noise into various stages of the process, including the input data itself [16], the output of ML models, or even the model weights or internal parameters [1]. DP allows for extracting valuable statistical insights from datasets while demonstrably protecting the privacy of any single record within them. In essence, DP ensures that the overall statistical properties of the dataset remain preserved irrespective of the presence or absence of any specific individual data point in the training set. A mechanism is defined as any mathematical computation that applies to and interacts with the data. Therefore, if the likelihood of any given result is nearly equal for two datasets that differ by just one record, then the mechanism guarantees DP. The degree of privacy is governed by a parameter known as  $\epsilon$ , which dictates how closely the outputs of a DP mechanism resemble each other when applied to two neighboring databases (i.e., datasets that are identical except for the presence or absence of a single individual's data). A smaller  $\epsilon$  offers stronger privacy protection. In Def. 1, we detail the DP inequality.

**Definition 1 (Differential Privacy).** *A randomized algorithm  $M$  with domain  $\mathbb{N}^2$  is  $(\epsilon, \delta)$ -differentially private if for all  $S \subseteq \text{Range}(M)$  and for all  $x, y \in \mathbb{N}^2$  such that  $\|x - y\|_1 \leq 1$ :*

$$\Pr[M(x) = S] \leq e^\epsilon \cdot \Pr[M(y) \in S] + \delta,$$

For any subset  $S$  and neighboring datasets  $x, y$ , the probabilities of  $M$  on  $x$  are less than  $e^\epsilon$  times the probabilities on  $y$ , plus  $\delta$ . This indicates that a randomized algorithm  $M$  is  $(\epsilon, \delta)$ -differentially private. One common method for achieving DP is by adding Laplace noise to query responses [55]. Let  $f$  be a function representing the AD algorithm, and  $\epsilon$  be the privacy parameter. The Laplace mechanism adds noise according to the formula:

$$\hat{f}(D) = f(D) + \text{Lap}\left(\frac{\Delta f}{\epsilon}\right) \quad (1)$$

where  $\hat{f}(D)$  is the DP query result on dataset  $D$ ,  $f(D)$  is the true query result,  $\Delta f$  is the sensitivity of the function  $f$ , and  $\text{Lap}(\lambda)$  represents Laplace noise with scale parameter  $\lambda$ . An alternative approach is to add Gaussian noise to query responses, which adds Gaussian noise instead of Laplacian noise.

## 4 Problem Formulation and Approach

We aim to investigate the effectiveness of AD techniques, specifically iForest and LOF, in identifying two types of anomalies within a dataset: local and global outliers (*RQ2*). This investigation is conducted within a privacy-preserving setup, where we employ DP techniques. Crucially, we also aim to analyze how the SHAP explanations of the AD models change as a result of employing DP. Estimating these explainability changes will allow us to assess the impact of the privacy-preserving mechanisms on the reasoning behind the identified anomalies. To achieve our goals, we will employ Privacy-Preserving AD with DP on the training data level. This means that we will introduce calibrated noise directly to the training data itself before applying the AD algorithms. Following the noise injection into the data, we employ SHAP to analyze how feature contributes to anomaly scores change under different  $\epsilon$ -DP guarantees (*RQ1*). Figure 1 summarizes the steps followed for the conducted analysis. Let  $D$  be a dataset  $\mathcal{D}: \mathcal{X} \in \mathbb{R}^d \in \mathbb{R}$ . The AD algorithm is trained on  $D$  in an unsupervised manner to estimate the anomaly score  $y_i$  for each datapoint  $x_i \in \mathcal{X}$ . To ensure a robust level of privacy protection measured by  $\epsilon$ -differential privacy ( $\epsilon$ -DP), we aim for small values of  $\epsilon$  to minimize the influence of individual records on the overall performance. We introduce noise into the data using either a Laplacian or Gaussian distribution, with varying  $\epsilon$  values (0.01, 0.1, 1, and 5). Subsequently, we retrain the AD models on the data augmented with noise. This process enables us to assess the impact of DP on both the performance and explainability of the models. Specifically, we compare the AD performance of models trained on noisy and original data using diverse metrics, focusing on understanding how DP affects the features driving AD with SHAP. Moreover, we explore the trade-off between privacy and explainability by employing SHAP.

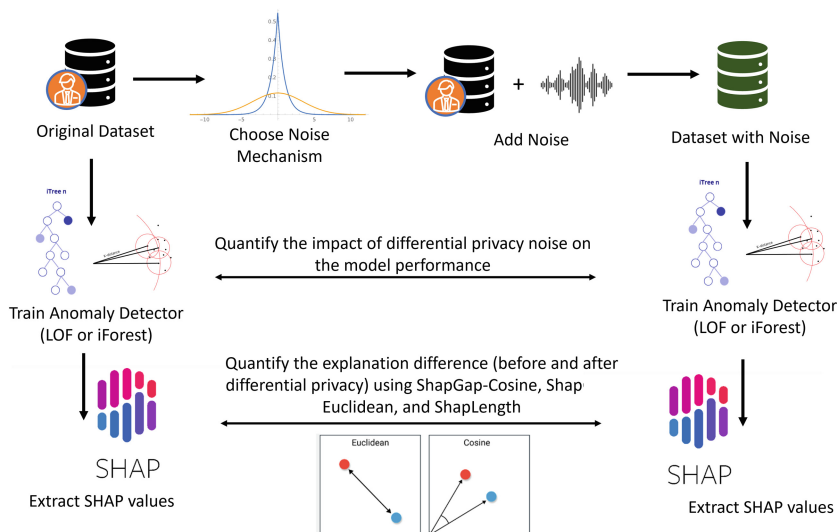


Fig. 1. Overall scheme of the experimental setup

## 5 Experimental Setting

This section describes the settings of our experiments, including the datasets, the procedures to train and evaluate the AD models, and the metrics used to quantify the change in explainability.

### 5.1 Datasets Description

We consider three datasets that are widely used for AD:

1. Mammography [51]: The mammography dataset corresponds to radiological scans to diagnose breast cancer. It consists of 6 features and 11183 records, of which 10923 are non-anomalous and 260 are anomalous.
2. Thyroid dataset [43]: The thyroid dataset corresponds to thyroid diseases. It comprises 21 features and contains 7,200 records, of which 6,666 are non-anomalous and 534 are anomalous.
3. Campaign (bank) dataset [43] includes banking information about individuals. It has 62 features and contains 41188 records, of which 36548 are non-anomalous and 4640 are anomalous.

### 5.2 Anomaly Detection Model Training

We perform hyperparameter tuning of the AD models using a Grid search with cross-validation (for a separate subset of the data). The hyperparameters that undergo an optimal selection are `aremax_features`, `n_estimators` for iForest, and `n_neighbors` for LOF. It is important to emphasize that all the datasets are labeled. However, as iForest and LOF are unsupervised AD algorithms, we use the labels only for evaluating their performance, as illustrated in Fig. 1.

### 5.3 Evaluation Metrics

**AD Performance Metrics.** We evaluate the performance of the AD model in terms of predictive performance and output consistency. As for the former, we use precision, because it focuses on the ratio of true anomalies identified among all flagged data points, and Area under the ROC curve (AUC) because it is suitable for imbalanced datasets as metrics. Precision measures the proportion of true positives among all predicted positives, while AUC evaluates the model's ability to distinguish between positive and negative classes. As for the latter, we compute the fidelity score, which measures the degree of agreement between the predictions of two different models on the same set of inputs (i.e., how closely the outputs of one model mirror those of another). In our case, the fidelity score quantifies the agreement between the AD model's outputs before and after applying DP (as shown in Fig. 1).

**Quantitative Analysis of SHAP Difference Metrics.** To quantify the SHAP value changes in scenarios with and without DP, we use three recently-proposed metrics, ShapGAP-Euclidean and ShapGAP-Cosine distances [34], and ShapLength [33]. While initially proposed for comparing SHAP values between black-box models and their corresponding surrogate white-box models, we leverage these metrics for a different purpose. Specifically, we compute the *ShapGAP* and *ShapLength* between SHAP values of data points before and after adding DP noise, with the final aim of quantifying the impact to assess how DP alters the explainability of AD models. More details about these metrics are provided in the following:

1. **ShapGAP-Euclidean Distance** [34] (**Eq. 2**): ShapGAP-Euclidean provides a magnitude-based measure of the difference between the SHAP values generated for the same data point from two models  $S$  (without and with DP) across  $n$  data points  $x_i$  of a dataset  $D$ . It is useful for understanding the overall magnitude of changes.

$$\text{ShapGAP}_{L_2}(D) = \frac{1}{n} \sum_i^n \|S_{\text{without DP}}(x_i) - S_{\text{with DP}}(x_i)\|_2 \quad (2)$$

2. **ShapGAP-Cosine Distance** [34] (**Eq. 3**): ShapGAP-Cosine computes the magnitude and directional relationship of the difference between two SHAP values of the same point from two models  $S$  (without and with DP) across  $n$  data points  $x_i$  of a dataset  $D$ . The ShapGAP-Cosine can range between 0 and 2, where higher values indicate higher dissimilarity. It is useful for capturing how similar the directions are, even if the magnitudes differ.

$$\text{ShapGAP}_{Cos}(D) = \frac{1}{n} \sum_i^n \left(1 - \frac{S_{\text{without DP}}(x_i) \cdot S_{\text{with DP}}(x_i)}{\|S_{\text{without DP}}(x_i)\|_2 \|S_{\text{with DP}}(x_i)\|_2}\right) \quad (3)$$

3. **ShapLength** [33]: ShapLength is a model-agnostic and computationally efficient metric for assessing how human-understandable a model is. It builds upon the  $p\%$ -complete explanation property, which finds the smallest set of features whose SHAP values sum exceed a defined threshold. Shap Length represents the number of features included in this  $p\%$ -complete explanation. A higher ShapLength indicates a model that relies on a larger number of features or complex interactions, making it harder to explain and interpret.

## 6 Experimental Results

In this section, we quantitatively evaluate the impact of DP on the effectiveness (Subsect. 6.1) and explainability of the AD models. Explainability is assessed through quantitative and qualitative evaluations, in Subsects. 6.2 and 6.4, respectively.

**Table 1.** AUC and precision of iForest across the mammography, thyroid, and bank datasets, without DP and with varying DP- $\epsilon$  values, considering the two noise-adding mechanisms: Gaussian and Laplace.

Dataset	Metric	Without Privacy	Laplace				Gaussian			
			$\epsilon$				$\epsilon$			
			5	1	0.1	0.01	5	1	0.1	0.01
mammography	AUC	74	73	72	66	53	76	72	69	54
	Precision	90	91	90	88	84	92	90	89	85
thyroid	AUC	89	54	56	51	50	54	53	53	48
	Precision	90	58	60	56	55	58	58	58	53
bank	AUC	64	58	57	52	52	57	56	51	49
	Precision	68	64	63	58	58	63	62	58	55

**Table 2.** AUC and precision of LOF across the mammography, thyroid, and bank datasets, without DP and with varying DP- $\epsilon$  values, considering the two noise adding mechanism: Gaussian and Laplace.

Dataset	Metric	Without Privacy	Laplace				Gaussian			
			Epsilon				Epsilon			
			5	1	0.1	0.01	5	1	0.1	0.01
Mammography	AUC	74	74	74	66	74	74	73	70	
	Precision	91	91	91	88	91	91	91	89	
Thyroid	AUC	56	56	53	51	56	57	53	51	
	Precision	60	60	58	56	60	61	58	56	
Bank	AUC	59	59	59	59	59	59	59	59	
	Precision	65	65	64	65	65	64	64	64	

### 6.1 Impact of Differential Privacy on Anomaly Detection Models

We start by analyzing the impact of employing DP on the performance of iForest and LOF. Table 1 reports the AUC and precision of iForest across the three considered datasets (mammography, thyroid, and bank), and across the two noise-adding mechanisms (Gaussian and Laplace) for varying values of  $\epsilon$ . When DP is not employed, iForest achieves an AUC of 74%, 89%, and 64% for mammography, thyroid, and bank datasets, respectively. A precision of 90% for both the mammography and thyroid datasets, and 68% for the bank dataset. As DP is introduced, iForest generally exhibits decreased performance compared to the non-DP models. AUC and precision decrease already for high values of  $\epsilon$  (i.e., less privacy). As expected, the difference is higher for smaller values of  $\epsilon$  (i.e., more privacy). By decreasing  $\epsilon$  in the Laplace case, the AUC decreases from 73% to 53% for the mammography dataset, from 54% to 50% for the thyroid dataset, and from 58% to 52% for the bank dataset, and a similar decrease



happens when decreasing  $\varepsilon$  with Gaussian noise. This is except for one case, that is for the mammography dataset for  $\varepsilon = 5$  with Gaussian.

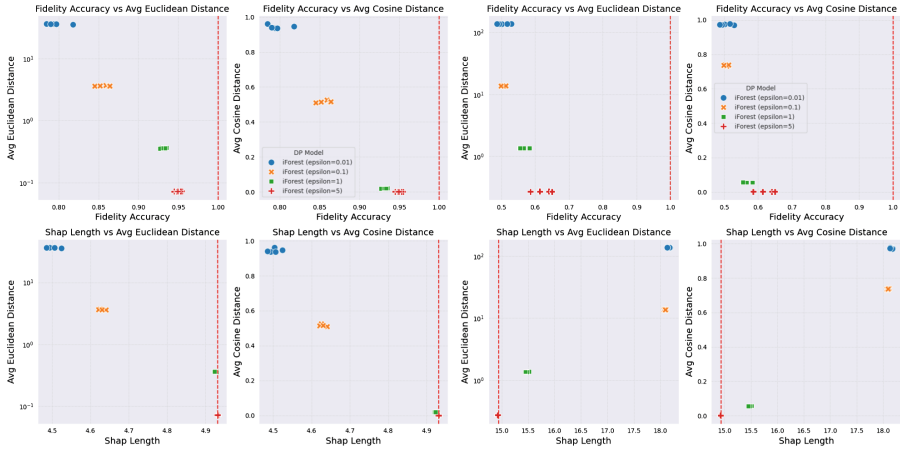
Table 2 reports the AUC and precision achieved by LOF across three considered datasets. Results show that in the case without applying DP, LOF achieves an AUC of 74%, 56%, and 59% for mammography, thyroid, and bank datasets, respectively, and a precision of 91%, 60%, and 65%. As DP is introduced, unlike with iForest, we observe a similar value for both the AUC and the precision across all datasets and all values of  $\varepsilon$  except for  $\varepsilon = 0.01$ .

This suggests that while iForest initially outperformed LOF without DP, LOF proved to be more robust and resilient to DP, maintaining its effectiveness under such constraints better than iForest, potentially due to its focus on k-nearest neighbors and local data density. This investigation suggests a trade-off between local and global outlier detection capabilities under DP. We explore this trade-off further with respect to SHAP explanations.

## 6.2 Impact of Differential Privacy on SHAP Explanations

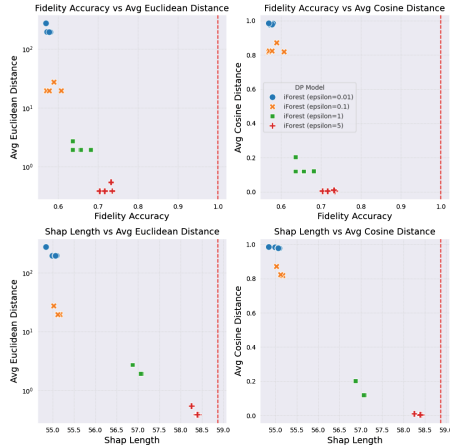
Figures 2 and 3 report the results of ShapGap-Cosine and ShapGap-Euclidean distances along with the fidelity accuracy and ShapLength metrics of iForest and LOF, across the three datasets and for the various values of  $\varepsilon$  considered. Each row of the plot focuses on a specific metric on the x-axis. The y-axis consistently displays both ShapGap-Cosine and ShapGap-Euclidean distances across 5 runs for each  $\varepsilon$  experiment. Each point on the plots corresponds to the average of the corresponding ShapGap distance across all data points of each dataset for one single run. In an ideal scenario, the AD models with and without DP should agree to 100%, producing identical outputs for all data points.

**iForest:** Figure 2, presents the results for iForest across the three datasets. Results show that ShapGAP-Euclidean and ShapGAP-Cosine distances across all datasets increase as the privacy guarantee increases (i.e.,  $\varepsilon$  decreases). This difference means that the vectors of the features in the explanations extracted using SHAP before and after the application of DP change in both magnitude (captured by the Euclidean) and direction (captured by the Cosine). These findings reveal that as the  $\varepsilon$  decreases, the magnitude of SHAP values tends to deviate more from those obtained in the absence of privacy constraints, for instance, a ShapGap-Cosine value close to 1 indicates high dissimilarity in the magnitude and direction of the SHAP value vectors before and after applying DP with  $\varepsilon$  of 0.01. Conversely, ShapGap-Euclidean has no upper bound, with higher values signifying greater dissimilarity. Specifically, the ShapGap-Euclidean metric for the mammography dataset ranges between 0 and 10, while for the thyroid and bank dataset, it ranges between 0 and 100. In contrast, the ShapGap-Cosine metric (Fig. 2 a,b,c bottom and up right) scored between 0 to 1 for both datasets depending on  $\varepsilon$ . The results also show a consistent trend between the value of  $\varepsilon$  and the fidelity accuracy (Fig. 2 a,b,c up right), as ShapGap-Cosine with smaller  $\varepsilon$  values correspond to lower fidelity accuracy, progressively decreasing



(a) Mammography dataset

(b) Thyroid dataset



(c) Bank dataset

**Fig. 2.** Fidelity Accuracy of iForest and average ShapGap-Euclidean distance, ShapGap-Cosine distance and ShapLength computed across the explanations extracted using SHAP for the various iForest models and the various values of epsilon, across (a) Mammography, (b) Thyroid and (c) Bank datasets. The vertical dashed line represents the without DP metric presented at the x-axis.

from 100% relative to the ideal scenario when DP is not applied. This trend is evident for both distances (Fig. 2 a,b,c up) and across all the datasets.

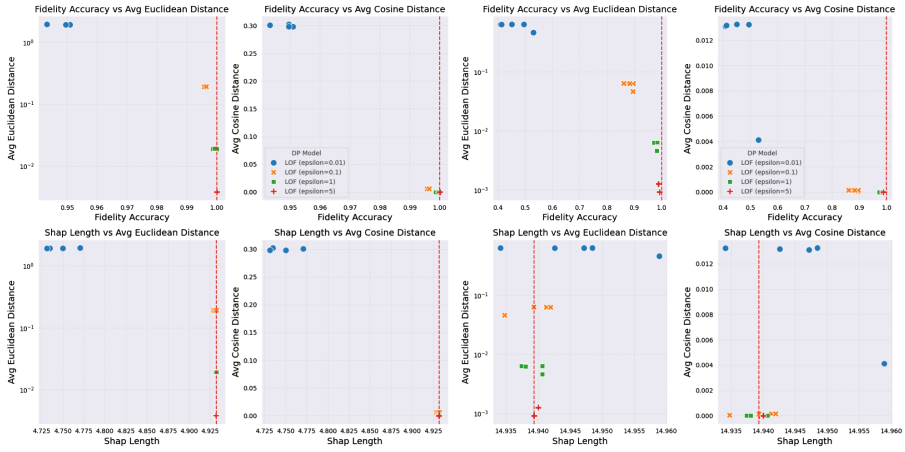
Another notable trend is the negative correlation between fidelity scores and distance values (i.e., higher distances imply lower fidelity scores, and vice versa). In other words, when the model’s reasoning aligns more closely with the original non-DP model (higher fidelity scores), the distances between SHAP values are minimized, indicating a stronger agreement in the influence of features on

predictions. This trend is observable for both ShapGap-Euclidean and cosine distances, showing that both the magnitude and direction of the SHAP value vectors are closely linked to the model’s reasoning and fidelity. *This result suggests a correlation between the difference in explanations and the model’s ability to faithfully represent the original predictions, as indicated by the fidelity scores.*

We now discuss the complexity captured by the value of ShapLength. The findings vary based on the privacy budget ( $\epsilon$ ) value for DP and depending on the dataset. For the mammography dataset (Fig. 2a bottom), observations at  $\epsilon$  values of 1 and 5 indicate stability and consistency in ShapLength, maintaining an average value of 4.95, identical to that of the model without DP. *This implies that the implementation of DP, while ensuring a moderate level of privacy protection, does not impact the ShapLength, preserving the explainability complexity of the model as in the non-DP setting.* However, at lower  $\epsilon$  values, specifically 0.1 and 1, there is a noticeable reduction in ShapLength to 4.5, indicating a slight deviation in model complexity compared to the model without DP. This indicates that DP with stricter privacy has reduced the model’s complexity. For the thyroid dataset (Fig. 2b bottom), we observe a different outcome: the ShapLength is highly affected by the application of DP, as with the decrease of  $\epsilon$ , the ShapLength is increasing. When observing it with the SHAP Gap distances, smaller ShapLength aligns with larger Euclidean and Cosine distances. The bank dataset (Fig. 2c bottom) exhibits a similar outcome in ShapLength compared to the mammography, as we observe a decrease in SHAP Length with smaller  $\epsilon$  values even with  $\epsilon = 5$  where ShapLength has also decreased.

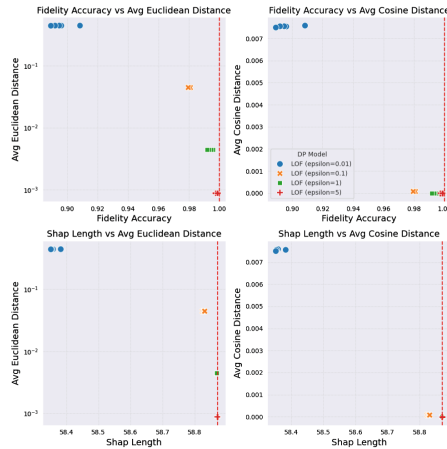
Our investigation reveals a privacy-explainability trade-off in applying DP to iForest models with SHAP explanations. Stricter privacy leads to increased divergence in SHAP values (magnitude and direction), decreased fidelity to the original model, and simpler models with less detailed explanations (reduced ShapLength) and are dependent. Conversely, relaxed privacy settings show better-preserved interpretability with explanations closer to the non-DP model and stable model complexity. *This negative correlation between explanation divergence and fidelity scores suggests a link between interpretability and the model’s ability to faithfully represent the original model’s predictions and explanations.*

**LOF:** Figure 3 shows the ShapGap metrics for the performance of the LOF model, highlighting the association between model fidelity and privacy levels achieved through DP (Fig. 3 a,b,c up). Across the mammography, thyroid, and bank datasets, model fidelity remains impressively high, ranging from 90% to 100% for  $\epsilon$  values of 0.1, 1, and 5, nearly mirroring the fidelity seen in the AD model without DP. However, a notable fidelity reduction occurs at  $\epsilon = 0.01$ , with drops by approximately 5%, 40%, and 10% for the mammography, thyroid, and bank datasets, respectively. Despite this high fidelity, we observe shifts and increases in Euclidean distances (Fig. 3 a,b,c up left), indicating alterations in the scale of the underlying data that fidelity scores do not capture. Conversely, the ShapGap-Cosine distances (Fig. 3 a,b,c up right), remain largely stable across 0.1, 1, and 5,  $\epsilon$  values, suggesting that the directionality of the



(a) Mammography dataset

(b) Thyroid dataset



(c) Bank dataset

**Fig. 3.** Fidelity Accuracy of LOF and average ShapGap-Euclidean distance, ShapGap-Cosine distance, and ShapLength computed across the explanations extracted using SHAP for the various iForest models and the various values of  $\epsilon$ , across (a) Mammography, (b) Thyroid and (c) Bank datasets. The vertical dashed line represents the without DP metric presented at the x-axis.

SHAP value vectors stays consistent despite the application of DP except for  $\epsilon$  of 0.01. Specifically, for the mammography dataset, it increases to around 0.3. For the thyroid, and bank dataset, the ShapGap-Cosine also increases but stays within a very low cosine measure of around 0.07 and 0.012 respectively, indicating a low magnitude and direction of change within SHAP values. *This phenomenon suggests that while the overall magnitude of model explanations is*

*influenced by DP, the vector direction reflecting feature contributions towards predictions remains minimally unaffected with DP for LOF.*

Analyzing model complexity through the ShapLength (Fig. 3 a,b,c bottom), we find that in both the mammography and bank datasets, ShapLength remains relatively unchanged for  $\varepsilon$  values of 0.1, 1, and 5, suggesting minimal variations in model complexity. However, at a  $\varepsilon$  of 0.01, we notice a decline in ShapLength, which decreases from 4.925 to 4.725 in the mammography dataset and from 58.8 to 58.4 in the bank dataset. This indicates that the complexity of the model decreases slightly under more strict privacy conditions. In contrast, the thyroid dataset demonstrates a different pattern: ShapLength stays closely aligned with the small variance in complexity without DP, except at a  $\varepsilon$  of 0.01, where we note a relatively small increase in the variance of complexity as the privacy level increases across the different runs (from 14.935 to 14.96).

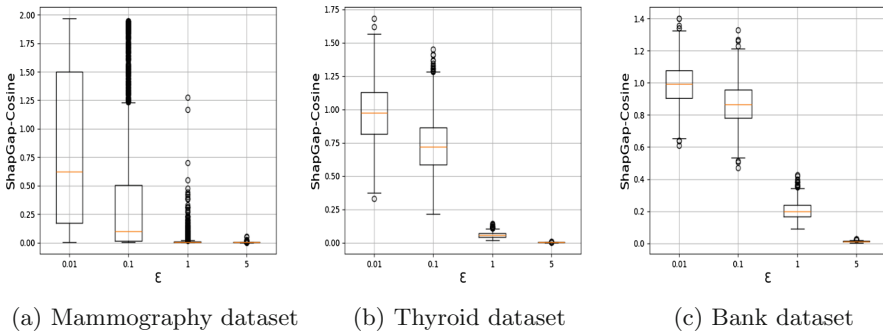
The observed inconsistencies across datasets in fidelity and ShapGap-Euclidean and Cosine distances likely stem from the inherent randomness introduced by DP and the chosen AD algorithms. While DP guarantees privacy, its added noise can alter various data characteristics, therefore the noise level and privacy level should be carefully chosen in a way that the overall distribution of the data is not largely affected. Therefore, this means that fidelity scores can remain elevated despite variations in Euclidean distance within higher  $\varepsilon$ . This also suggests that the data distribution undergoes modifications caused by the DP noise, but these modifications are limited such that the overall statistical properties remain largely preserved, thus maintaining high fidelity and low ShapGap-Cosine.

As anticipated, these findings indicate that iForest is more sensitive to data points that significantly deviate from the overall distribution of the data, while LOF more easily detects deviations within specific data regions. Since DP focuses on preserving overall data statistics, it can alter the distribution of the data, impacting how anomalies appear in the global picture. This, in turn, affects the SHAP values in iForest, as features contributing to isolation might be masked by the DP noise. LOF focuses on Local outliers and analyzes how different a data point is from a local perspective. DP's impact on local neighborhoods might be less significant compared to its effect on the entire data distribution. Additionally, LOF's SHAP values might focus on features relevant to the local anomaly score, which might be less sensitive to global distribution changes caused by DP.

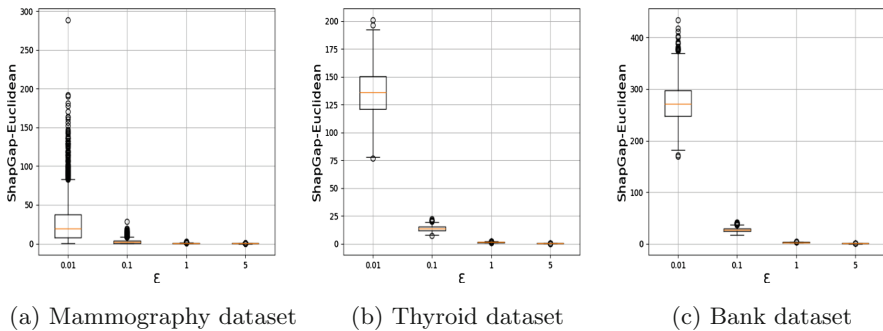
### 6.3 ShapGap Distribution Analysis

To further explore the distribution of SHAP divergence across data points, we report the findings using box plots for ShapGap-Euclidean and Cosine. In Fig. 4, we visualize the distribution of ShapGap-Cosine for iForest for all data points across the three datasets. We observe that for  $\varepsilon$  values of 1 and 5 across the three datasets, the ShapGap-Cosine has a lower spread in comparison to smaller  $\varepsilon$  of 0.1 and 0.01. This low spread indicates that most of the data points have a small cosine distance between 0 and 0.25, which means closer to the non-DP scenario. However, for smaller  $\varepsilon$  values (0.1 and 0.01), we see a wider spread

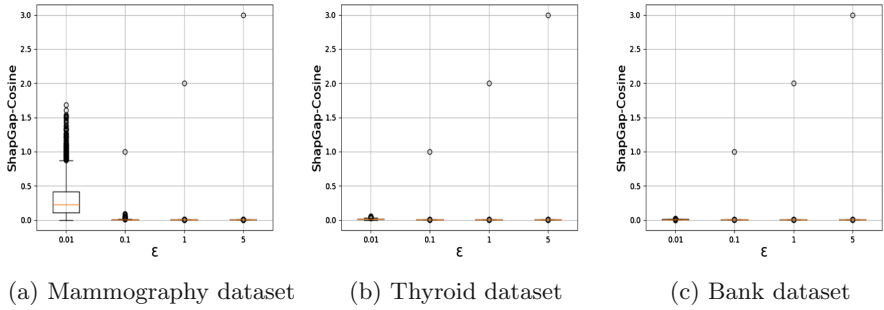
of the ShapGap-Cosine distribution. The values range between 0 and 2 for  $\varepsilon = 0.01$  with the mammography dataset, between 0.25 and 1.75 for thyroid, and between 0.6 and 1.4 for bank. This wider distribution suggests a significant portion of data points exhibiting high ShapGap, deviating from the behavior observed without DP. Regarding Euclidean distances, Fig. 5 we visualize the distribution of ShapGap-Euclidean for iForest across all data points for each dataset. We observe that for  $\varepsilon$  values of 0.1, 1, and 5 across the three datasets, the ShapGap Euclidean distance has a small distribution compared to  $\varepsilon$  of 0.01. This indicates that the data points with small Euclidean distances (ranging between 0 and 20 for mammography, from 0 to 25 for thyroid, and from 0 to 50 for the bank), are closer to the non-DP scenario. However, for smaller  $\varepsilon$  values (0.01), we note a larger distribution of ShapGap Euclidean measures reaching up to 500 in the bank dataset, which is extremely large. Figure 6 shows the distribution of ShapGap Cosine across all data points with LOF. For  $\varepsilon$  values of 0.1, 1, and 5, the distribution of ShapGap Cosine distance is relatively small (less than 0.2 for all the datasets), indicating that the SHAP values with and without



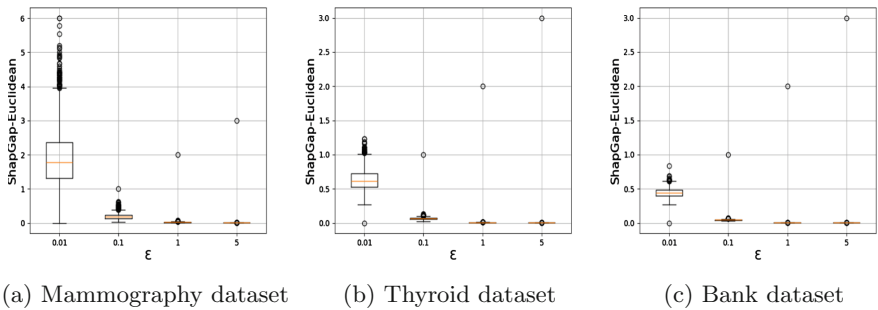
**Fig. 4.** Distribution of iForest ShapGap-Cosine distances across a) Mammography, b) Thyroid, and c) Bank Datasets for the various  $\varepsilon$  values



**Fig. 5.** Distribution of iForest ShapGap-Euclidean distances across a) Mammography, b) Thyroid, and c) Bank Datasets for the various  $\varepsilon$  values



**Fig. 6.** Distribution of LOF ShapGap-Cosine distances across a) Mammography, b) Thyroid, and c) Bank Datasets for the various  $\epsilon$  values

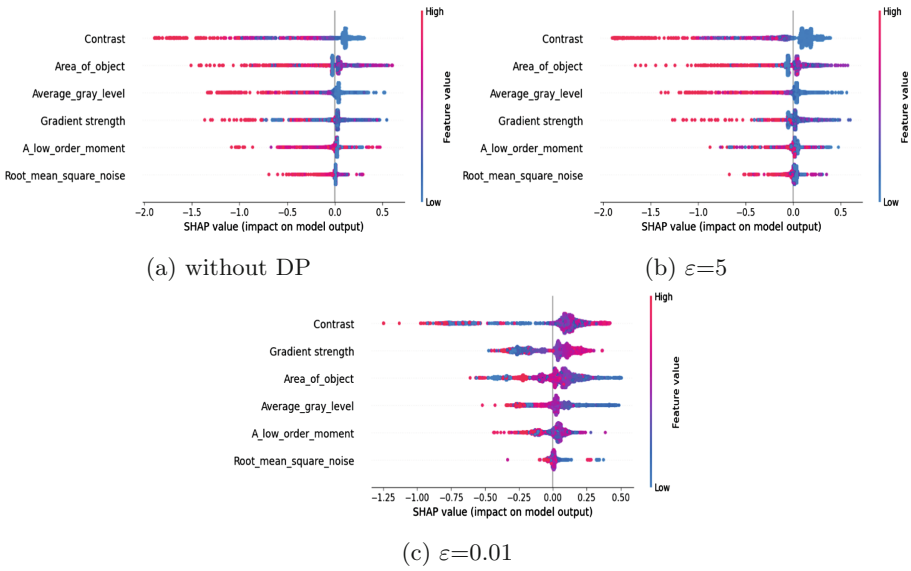


**Fig. 7.** Distribution of LOF ShapGap-Euclidean distances across a) Mammography, b) Thyroid, and c) Bank Datasets for the various  $\epsilon$  values

DP are similar for most of the data points, suggesting minimal impact on the data due to DP at these privacy levels. For  $\epsilon$  equal to 0.01, the distribution of ShapGap Cosine distance is larger (ranging from 0 to 1.75), indicating greater differences between SHAP values with and without DP. Regarding Euclidean distances, Fig. 7 visualizes the distribution of ShapGap Euclidean for LOF across all data points. We observe that for  $\epsilon$  of 0.1, 1, and 5, for the 3 datasets, the ShapGap Euclidean concentrates around at most between 0 and 1, and there is no diverged distribution. While only for the very small  $\epsilon$  we observe the distribution of ShapGap Euclidean diverges covering a range between 0 and 6 as maximum, which is still relatively very small compared to the magnitude change scale of iForest. These findings suggest that moderate privacy levels have minimal impact on SHAP with LOF. However, iForest appears to be more sensitive to the DP noise, evidenced by the large distribution of ShapGap distances and reflected by the distance distribution across the data points.

### 6.4 Impact of Differential Privacy on SHAP Summary Plots

After having quantitatively analyzed the impact of DP on SHAP values, we now examine the visual interpretation and analysis of SHAP summary plots to illustrate how DP noise influences the interpretability of SHAP explanations visually. Figure 8 and Fig. 9 present the summary plots relative to the mammography dataset for the iForest and LOF models respectively<sup>1</sup>.



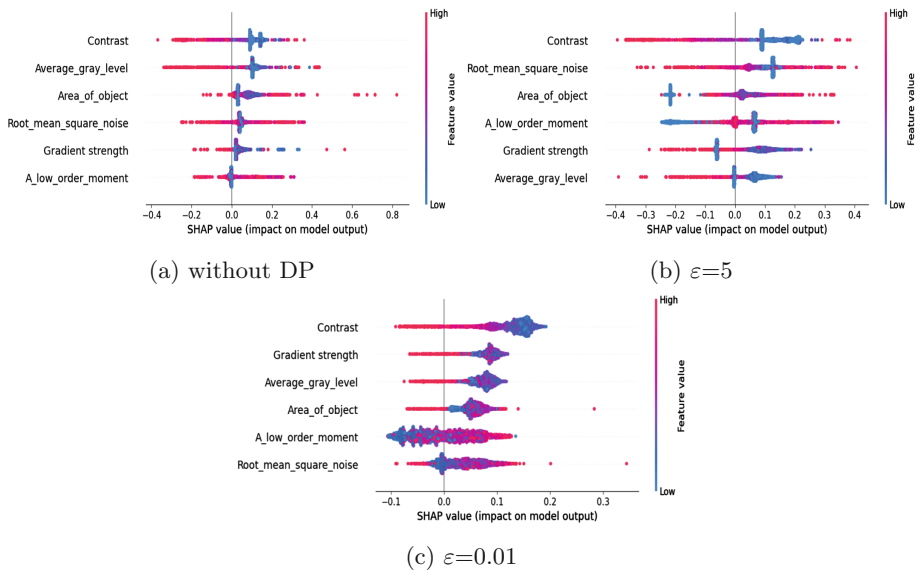
**Fig. 8.** Summary plot for iForest for the mammography dataset with various  $\epsilon$

Figure 8 shows that the feature *Contrast* emerges as the most critical feature for the model’s prediction, while a *low\_order\_moment* and *Root\_mean\_square\_noise* remain the two least important regardless of the level of privacy  $\epsilon$  introduced. This fact indicates that, for some features, the level of importance remains consistent even after applying DP, regardless of the value of  $\epsilon$ . Instead, concerning other features, such as *Gradient\_Strength*, *Area\_of\_object*, and *average\_gray\_level*, their importance according to SHAP varies with  $\epsilon$ . For example, for a low level of privacy protection (e.g.,  $\epsilon = 5$ ), the consistency concerning the scenario without DP is maintained considering both feature importance and feature contribution, as observable by the similar color distributions

<sup>1</sup> The summary plots display SHAP values for each feature and data point, indicating their impact on classifying normal or abnormal. On the x-axis, SHAP values show a feature’s influence on predictions, with positive or negative values indicating a tendency towards an abnormal or normal prediction, respectively. The y-axis ranks features by importance, and point colors signify feature values—red for high and blue for low.



in the various scenarios. If, instead, stronger privacy guarantees are set in place (e.g.,  $\epsilon = 0.01$ ), both the order and contribution of the three features significantly change. Indeed, the clear distinction between blue and red fades, indicating that the significant level of noise obscures the clarity of SHAP values, rendering the interpretation of output trends more difficult<sup>2</sup> Figure 9 shows the summary plots obtained for the LOF model. We observe that the *contrast* feature is consistently the most influential, irrespective of the value  $\epsilon$ -DP. Specifically, we note that higher values of the contrast feature continue to be highlighted in blue and have a positive impact on the output, indicating their significance, while lower values are consistently highlighted in red and have negative SHAP values. Instead, for  $\epsilon = 0.01$ , the SHAP values are no longer distinguishable by color, signifying a less clear impact of feature values on the output of the model. Concerning the other features, there is a variation in their order of influence between the various  $\epsilon$ , yet the underlying rationale behind their values remains consistent across most  $\epsilon$  values. However, for a stricter privacy budget of  $\epsilon = 0.01$ , there is a notable departure from this consistency, as the distinction between blue and red becomes less clear, thus reducing considerably the interpretability of the model through this plot. These findings align with the ShapGap and ShapLength measures, demonstrating that iForest exhibits a greater sensitivity to DP perturbations compared to LOF in terms of SHAP interpretability. This is evidenced by the



**Fig. 9.** Summary plot for LOF for the mammography dataset with various  $\epsilon$  (Color figure online)

<sup>2</sup> As similar trends in the SHAP summary plots are observed in the two other datasets, we omit to show them and the relative discussion. To illustrate the visual changes, we show summary plots for only two  $\epsilon$  values (0.01 and 5).

more pronounced shift observed in the SHAP summary plots for iForest with decreasing privacy budgets. While both models experience a decline in interpretability for stricter privacy constraints, LOF retains a relatively clearer distinction between influential and non-influential features even at lower  $\varepsilon$  values, specifically at  $\varepsilon$  equal to 0.01, where iForest’s interpretability obscures significantly. Another finding is that while applying DP safeguards individual data privacy through noise injection, this mechanism hindered the interpretability of SHAP summary plots. DP’s impact manifests in different ways such as distorted features importance and misleading interpretations. Firstly, the introduced randomness leads to fluctuations in SHAP feature attributions, making it difficult to accurately detect their true impact on model predictions. Secondly, the noise obscured data patterns, diminishing the overall precision of the AD and SHAP values and affecting the extraction of meaningful insights.

## 7 Conclusion

In this paper, we investigate the impact of differential privacy (DP) on the performance and explainability of anomaly detection (AD) models. We compare the performance of Isolation Forest (iForest) and Local Outlier Factor (LOF) under various DP noise conditions and across multiple datasets. The results show that while iForest initially outperforms LOF without DP, LOF exhibits greater robustness to DP. Furthermore, we analyze the impact of DP on explainability by comparing them across different distance metrics, both with and without DP applied. For explainability, we use SHapley Additive exPlanations (SHAP). We observe a correlation between the DP parameter ( $\varepsilon$ ) and the magnitude and direction of changes in SHAP values across the metrics. Notably, the impact of DP on SHAP values manifested diversely across datasets and with the different AD techniques. This implies that distinctive data characteristics might affect the sensitivity of SHAP values to DP noise. These findings underscore the trade-off between privacy and explainability when employing DP alongside SHAP values in AD. For future work, we aim to explore techniques to mitigate the effect of DP on SHAP values while upholding adequate privacy guarantees. Additionally, we aim to evaluate the effects of DP on other explainability techniques utilized with deep learning-based AD methodologies.

**Acknowledgements.** F. Ezzeddine was supported by the Swiss Government Excellence Scholarship. Dr. M. Gjoreski’s work was funded by SNSF through the project XAI-PAC (grant number PZ00P2\_216405).

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Abadi, M., et al.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 308–318 (2016)
2. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. *J. Netw. Comput. Appl.* **60**, 19–31 (2016)
3. Ahmed, M., Mahmood, A.N., Islam, Md.R.: A survey of anomaly detection techniques in financial domain. *Future Gener. Comput. Syst.* **55**, 278–288 (2016)
4. Alabdulatif, A., Khalil, I., Kumarage, H., Zomaya, A.Y., Yi, X.: Privacy-preserving anomaly detection in the cloud for quality assured decision-making in smart cities. *J. Parallel Distrib. Comput.* **127**, 209–223 (2019)
5. Alharbi, B., Liang, Z., Aljindan, J.M., Agnia, A.K., Zhang, X.: Explainable and interpretable anomaly detection models for production data. *SPE J.* **27**(01), 349–363 (2022)
6. Angelini, F., Yan, J., Naqvi, S.M.: Privacy-preserving online human behaviour anomaly detection based on body movements and objects positions. In: ICASSP 2019 – 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8444–8448. IEEE (2019)
7. Bergmann, P., Fauser, M., Sattlegger, D., Steger, C.: MVTEC AD—a comprehensive real-world dataset for unsupervised anomaly detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9592–9600 (2019)
8. Bozorgpanah, A., Torra, V., Aliahmadipour, L.: Privacy and explainability: the effects of data protection on Shapley values. *Technologies* **10**(6), 125 (2022)
9. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LoF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 93–104 (2000)
10. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: a survey. *ACM Comput. Surv. (CSUR)* **41**(3), 1–58 (2009)
11. Chen, Z., Yeo, C.K., Lee, B.S., Lau, C.T.: Autoencoder-based network anomaly detection. In: 2018 Wireless Telecommunications Symposium (WTS), pp. 1–5. IEEE (2018)
12. Chukkapalli, S.S.L., Ranade, P., Mittal, S., Joshi, A.: A privacy preserving anomaly detection framework for cooperative smart farming ecosystem. In: 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), pp. 340–347. IEEE (2021)
13. de Araújo, F.M.N.: XAIPrivacy-XAI with Differential Privacy. Ph.D. thesis, Universidade do Porto (Portugal) (2023)
14. Degue, K.H., Gopalakrishnan, K., Li, M.Z., Balakrishnan, H.: Differentially private outlier detection in correlated data. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 2735–2742. IEEE (2021)
15. Du, M., Jia, R., Song, D.: Robust anomaly detection and backdoor attack detection via differential privacy. *arXiv preprint [arXiv:1911.07116](https://arxiv.org/abs/1911.07116)* (2019)
16. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). [https://doi.org/10.1007/11787006\\_1](https://doi.org/10.1007/11787006_1)
17. Ezzeddine, F., Ayoub, O., Andreoletti, D., Tornatore, M., Giordano, S.: Vertical split learning-based identification and explainable deep learning-based localization of failures in multi-domain NFV systems. In: 2023 IEEE Conference on Network

- Function Virtualization and Software Defined Networks (NFV-SDN), pp. 46–52. IEEE (2023)
18. Giraldo, J., Cardenas, A., Kantarcioglu, M., Katz, J.: Adversarial classification under differential privacy. In: Network and Distributed Systems Security (NDSS) Symposium 2020 (2020)
  19. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM Comput. Surv. (CSUR)* **51**(5), 1–42 (2018)
  20. Guo, J., Zheng, P., Huang, J.: Efficient privacy-preserving anomaly detection and localization in bitstream video. *IEEE Trans. Circuits Syst. Video Technol.* **30**(9), 3268–3281 (2019)
  21. Harder, F., Bauer, M., Park, M.: Interpretable and differentially private predictions. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 4083–4090 (2020)
  22. Ul Hassan, M., Rehmani, M.H., Chen, J.: Differential privacy in blockchain technology: a futuristic approach. *J. Parallel Distrib. Comput.* **145**, 50–74 (2020)
  23. Jetchev, D., Vuille, M.: Xorshap: privacy-preserving explainable AI for decision tree models. *Cryptology ePrint Archive* (2023)
  24. Jia, W., Shukla, R.M., Sengupta, S.: Anomaly detection using supervised learning and multiple statistical methods. In: 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1291–1297. IEEE (2019)
  25. Jiang, H., Pei, J., Dongxiao, Yu., Jiguo, Yu., Gong, B., Cheng, X.: Applications of differential privacy in social network analysis: a survey. *IEEE Trans. Knowl. Data Eng.* **35**(1), 108–127 (2021)
  26. Keshk, M., Sitnikova, E., Moustafa, N., Jiankun, H., Khalil, I.: An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. *IEEE Trans. Sustain. Comput.* **6**(1), 66–79 (2019)
  27. Leung, K., Leckie, C.: Unsupervised anomaly detection in network intrusion detection using clusters. In: Proceedings of the Twenty-Eighth Australasian Conference on Computer Science, vol. 38, pp. 333–342 (2005)
  28. Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: a review of machine learning interpretability methods. *Entropy* **23**(1), 18 (2020)
  29. Liu, F.T., Ting, K.M., Zhou, Z.-H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE (2008)
  30. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
  31. Lyu, L., Law, Y.W., Erfani, S.M., Leckie, C., Palaniswami, M.: An improved scheme for privacy-preserving collaborative anomaly detection. In: 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pp. 1–6. IEEE (2016)
  32. Ma, S., et al.: Privacy-preserving anomaly detection in cloud manufacturing via federated transformer. *IEEE Trans. Ind. Inform.* **18**(12), 8977–8987 (2022)
  33. Mariotti, E., Alonso-Moral, J.M., Gatt, A.: Measuring model understandability by means of Shapley additive explanations. In: 2022 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–8. IEEE (2022)
  34. Mariotti, E., Sivaprasad, A., Moral, J.M.A.: Beyond prediction similarity: Shapgap for evaluating faithful surrogate models in XAI. In: Longo, L. (ed.) xAI 2023. CCIS, vol. 1901, pp. 160–173. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-44064-9\\_10](https://doi.org/10.1007/978-3-031-44064-9_10)

35. Mayer, R., Hittmeir, M., Ekelhart, A.: Privacy-preserving anomaly detection using synthetic data. In: Singhal, A., Vaidya, J. (eds.) DBSec 2020. LNCS, vol. 12122, pp. 195–207. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-49669-2\\_11](https://doi.org/10.1007/978-3-030-49669-2_11)
36. Mehnaz, S., Bertino, E.: Privacy-preserving real-time anomaly detection using edge computing. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE), pp. 469–480. IEEE (2020)
37. Montenegro, H., Silva, W., Cardoso, J.S.: Privacy-preserving generative adversarial network for case-based explainability in medical image analysis. *IEEE Access* **9**, 148037–148047 (2021)
38. Munir, M., Siddiqui, S.A., Dengel, A., Ahmed, S.: Deepant: a deep learning approach for unsupervised anomaly detection in time series. *IEEE Access* **7**, 1991–2005 (2018)
39. Muruti, G., Rahim, F.A., bin Ibrahim, Z.-A., A survey on anomalies detection techniques and measurement methods. In: 2018 IEEE Conference on Application, Information and Network Security (AINS), pp. 81–86. IEEE (2018)
40. Naidu, R., Priyanshu, A., Kumar, A., Kotti, S., Wang, H., Mireshghallah, F.: When differential privacy meets interpretability: a case study. *arXiv preprint arXiv:2106.13203* (2021)
41. Nori, H., Caruana, R., Bu, Z., Shen, J.H., Kulkarni, J.: Accuracy, interpretability, and differential privacy via explainable boosting. In: International Conference on Machine Learning, pp. 8227–8237. PMLR (2021)
42. Okada, R., Fukuchi, K., Sakuma, J.: Differentially private analysis of outliers. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (eds.) ECML PKDD 2015. LNCS (LNAI), vol. 9285, pp. 458–473. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-23525-7\\_28](https://doi.org/10.1007/978-3-319-23525-7_28)
43. Pang, G., Shen, C., van den Hengel, A.: Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and data mining, pp. 353–362 (2019)
44. Panjei, E., Gruenwald, L., Leal, E., Nguyen, C., Silvia, S.: A survey on outlier explanations. *VLDB J.* **31**(5), 977–1008 (2022)
45. Ravi, A., Yu, X., Santelices, I., Karray, F., Fidan, B.: General frameworks for anomaly detection explainability: comparative study. In: 2021 IEEE International Conference on Autonomous Systems (ICAS), pp. 1–5. IEEE (2021)
46. Roshan, K., Zafar, A.: Utilizing xAI technique to improve autoencoder based model for computer network anomaly detection with Shapley additive explanation (Shap). *arXiv preprint arXiv:2112.08442* (2021)
47. Roshan, K., Zafar, A.: Using kernel Shap xAI method to optimize the network anomaly detection model. In: 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 74–80. IEEE (2022)
48. Sridharan, R., Maiti, R.R., Tippenhauer, N.O.: Wadac: privacy-preserving anomaly detection and attack classification on wireless traffic. In: Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks, pp. 51–62 (2018)
49. Tritscher, J., Krause, A., Hotho, A.: Feature relevance xAI in anomaly detection: Reviewing approaches and challenges. *Front. Artif. Intell.* **6**, 1099521 (2023)
50. Veugen, T., Kamphorst, B., Marcus, M.: Privacy-preserving contrastive explanations with local foil trees. *Cryptography* **6**(4), 54 (2022)
51. Woods, K.S., Doss, C.C., Bowyer, K.W., Solka, J.L., Priebe, C.E., Kegelmeyer Jr., W.P.: Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *Int. J. Pattern Recogn. Artif. Intell.* **7**(06), 1417–1436 (1993)

52. Yang, M., Song, L., Xu, J., Li, C., Tan, G.: The tradeoff between privacy and accuracy in anomaly detection using federated XGBoost. arXiv preprint [arXiv:1907.07157](https://arxiv.org/abs/1907.07157) (2019)
53. Yuan, S., Wu, X.: Trustworthy anomaly detection: a survey. arXiv preprint [arXiv:2202.07787](https://arxiv.org/abs/2202.07787) (2022)
54. Zhang, P., Huang, X., Sun, X., Wang, H., Ma, Y.: Privacy-preserving anomaly detection across multi-domain networks. In: 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, pp. 1066–1070. IEEE (2012)
55. Zhang, Z., et al.: {PrivSyn}: differentially private data synthesis. In: 30th USENIX Security Symposium (USENIX Security 21), pp. 929–946 (2021)
56. Zong, B., et al.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection, In: International Conference on Learning Representations (2018)



# Blockchain for Ethical and Transparent Generative AI Utilization by Banking and Finance Lawyers

Swati Sachan<sup>1</sup> , Vinicius Dezem<sup>2</sup> , and Dale Fickett<sup>3</sup> 

<sup>1</sup> Financial Technology, Management School, University of Liverpool, Liverpool L69 7ZH, UK  
swati.sachan@liverpool.ac.uk

<sup>2</sup> Knowledge Engineering, Federal University of Santa Catarina, Florianopolis, Brazil  
vinicius.dezem@porgrad.ufsc.br

<sup>3</sup> Robins School of Business, University of Richmond, Virginia, VA 23173, USA  
dfickett@richmond.edu

**Abstract.** Generative AI tools powered by Large Language Models (LLMs) have attracted significant attention from the banking, finance, legal, and technology sectors due to their ability to generate and articulate coherent human-like text and images. Legal firms have raised ethical concerns regarding LLM's ability to emulate legal reasoning, accountability of erroneous outcomes, and security and privacy of confidential legal data. To address these challenges, this research paper proposes a blockchain-based monitoring framework that ensures the responsible and secure application of Generative AI in drafting legal decisions by utilizing the anonymized output from an existing eXplainable Artificial Intelligence (XAI) algorithm within a law firm, which assists in legal decision-making. The lawyers are expected to comprehend explainable algorithmic decisions expressed in terms of probabilities and feature importance instead of textual explanations. The immutability and decentralization of blockchain technology form the basis of a transparent and tamper-proof record-keeping system. It ensures consistent and tamper-resistant responses by generative AI, which has been used by lawyers in the past. A case study on data security and tort liability claims on banking data breaches is presented to demonstrate the practical application.

**Keywords:** Generative AI · Explainable AI · Blockchain · Banking · Law

## 1 Introduction

Legal decision-support systems driven by artificial intelligence (AI) algorithms to assist human lawyers in legal reasoning can be attributed to advances in eXplainable AI (XAI) techniques and accessibility of high-quality data [1]. The launch of Generative AI tools based on Large Language Models (LLMs), such as OpenAI's Generative Pre-trained Transformer (GPT) and Google AI's Google Bard, have attracted significant attention from the banking, finance, legal, and technology sectors due to its ability to generate and

articulate coherent human-like text and images [2]. Experimental studies have demonstrated that the automated legal reasoning of LLMs is getting closer to human legal practitioners. For Instance, GPT-3.5 achieved a C+ grade in actual law school examinations [3], and GPT-4 successfully passed the Uniform Bar Exam [4]. The experiment on the Uniform Bar Exam was conducted on multiple choice questions and open-ended questions in a tightly controlled environment under a restricted set of prompts and parameters.

Law preserves justice and societal equilibrium. Therefore, algorithms cannot be held accountable for making incorrect high-stakes legal decisions. In-house banking lawyers (stakeholders) and law firms are actively seeking potential use cases for these tools by gaining an in-depth understanding of the decision-making capabilities, data utilization, and security protocols of AI tools [5]. Generative AI usage has three main ethical concerns: (a) the challenge to replicate nuanced legal reasoning, (b) the accountability of erroneous outcomes, and (c) the mishandling or unauthorized access to confidential information. In response to these ethical and security concerns, firms across multiple domains in Europe, the USA, and Canada have restricted their employees' use of generative AI tools to prevent potential data breaches.

This research integrates blockchain technology to monitor and govern the usability of Generative AI tools and XAI decision-making models by human legal professionals. It aims to make the following contributions:

- (a) **Application of Generative AI in Legal Drafting:** It presents the ethical and safe application of Generative AI tools to support the drafting of legal decisions derived from a pre-existing XAI algorithm. Generative AI is not used for the direct processing of legal facts and pieces of evidence. Instead, it processes the raw but anonymized output (or detailed decision analysis) from an XAI model for efficient drafting of various correspondence for anticipated pre-litigation decisions.
- (b) **Blockchain for Responsible AI:** It proposes the responsible use of AI tools by transforming the anonymized input prompt and the AI-generated text into a hash value, a unique digital fingerprint of the data. The hash values are then permanently recorded on a blockchain network for a tamper-evident ledger for future auditing purposes.
- (c) **Case Study on Data Security and Tort Claims:** A case study on tort liability claims on banking data breaches is presented where the XAI Evidential Reasoning (ER) algorithm was used to give legal decisions, usability in terms of quality text generated by four LLM models was tested. Two blockchain networks, Ethereum and Hyperledger Fabric, were tested to audit the AI-generated content.

## 2 Legal Decisions by AI

The literature on the XAI framework for legal knowledge representation based on Abstract Dialectical Frameworks (ADF), Hybrid Rule-Based Expert Systems, and Explainable Deep Learning is summarized in Table 1. ADF is a framework that represents arguments or legal propositions and their relationships. It uses logical conditions to determine the status of each node. ADFs are flexible and can be adapted to model a range of legal arguments. However, as altercations grow in complexity, the computational resources needed can increase significantly. Hybrid Rule-Based Expert Systems combine a set of rules with case-based reasoning to simulate the decision-making abilities of human experts. Each technique provides unique perspectives on constructing



legal arguments, transparent decision-making for tort liability claims, summarization of legal texts, and representation of statutory laws through rules in expert systems. It is crucial to note that widely recognized AI interpretation techniques such as LIME (Local Interpretable Model-Agnostic Explanations) [6], Shapley additive explanations (SHAP) [7], LRP (Layer-wise Relevance Propagation) [8], and Taylor decomposition [9] are inapplicable in legal context because AI-driven legal decision-making is not conducted through opaque, ‘black-box’ models.

**Table 1.** XAI for Legal Knowledge Representation

Approach	Author and Year	Main Contribution
Abstract Dialectical Frameworks (ADF)	J. Collenette, et al., 2023 [10]	Comparative study on the primary approach of legal reasoning based on HYPO [11], CATO [12], and IBP [13] with ADF [14]. ADF is a directed graph that represents and reasons with complex legal argumentation structures
Hybrid Rule-Base: Legal Knowledge and Data-Driven	S. Sachan, et al., 2021 [15]	A hybrid rule-based method based on human expert knowledge and data-driven training for transparent decisions on tort liability claims
Explainable Deep Learning	M. Norkute, et al., 2021 [16]	Legal text summarization by highlighting the text based on attention score by an explainable deep-learning model
Rule-Base	M. Sergot, et al., 1986 [17] U. Schild, 1990 [18]	Both papers have utilized a rule-based approach to fit statutory law representation but are less apt lower legal layers, with the assumption that users will provide case law knowledge

Legal and technical challenges of Generative AI tools, AI algorithms, and Blockchain technology are demonstrated in Fig. 1. A singular technology is not good enough to boost responsible AI. For instance, blockchain provides immutable data storage, but it conflicts with the data protection mandate on the “Right to be Forgotten” because once personal information is recorded on the blockchain, it cannot be erased [19]. Therefore, multiple technology integration is required to overcome each other shortcomings, as proposed in this research.

Adopting AI and blockchain technologies in the legal sector requires a comprehensive architectural framework with ethical considerations such as compliance with data protection law, confidentiality and integrity, quality assurance, and security. The proposed approach complies with stringent data protection laws such as the General Data Protection Regulation (GDPR) and the California Privacy Rights Act (CPRA) to respect privacy rights, the “Right to be Forgotten” [20], and the “Right to Explanation [21].” It provides a dual-layered solution, hashing sensitive data for blockchain storage (on-chain) and offloading larger files to secure cloud services (off-chain).

The regular audits and checks ensure that the automated content generated by AI tools utilized by humans (lawyers) aligns with legal principles and precedents. It verifies the authority and responsibility of humans to promote accountability. Robustness and security are crucial for trustworthy AI and blockchain systems. The proposed multi-technology approach can withstand the tampering of prompts containing anonymized legal decisions derived from XAI and AI-generated text by malicious actors through the use of blockchain’s immutable record-keeping. Commitment to ethical standards promotes transparency and control over personal data and accountability of human users, which is particularly important in sensitive fields such as legal services, finance, and banking.

	Legal	Technical
<b>Generative AI</b>	<ul style="list-style-type: none"> <li>- Data confidentiality risks</li> <li>- Intellectual property Infringement</li> <li>- Ambiguity in accountability and liability for AI caused damages</li> <li>- Cybercrimes powered by Generative AI</li> </ul>	<ul style="list-style-type: none"> <li>- Fabrication of facts</li> <li>- Biased responses</li> <li>- High computational cost (GPT: Cost per 1K prompt tokens, Google Bard: Free)</li> <li>- Control or influence on human judgement</li> </ul>
<b>AI Algorithms</b>	<ul style="list-style-type: none"> <li>- GDPR Article 22 - Need for decision transparency</li> <li>- Indeterministic responsibility of damages caused by AI</li> </ul>	<ul style="list-style-type: none"> <li>- Explanations incomprehensible to domain experts</li> <li>- Bias and discrimination in decision-making</li> </ul>
<b>Blockchain Technology</b>	<ul style="list-style-type: none"> <li>- GDPR Article 17 - 'Right to be forgotten'</li> <li>- Mandates deletion of personal data upon request</li> <li>- Regulatory uncertainty: relatively new and constantly evolving law</li> <li>- Data privacy and protection</li> </ul>	<ul style="list-style-type: none"> <li>- High on-chain computational Cost</li> <li>- High data storage expenses</li> <li>- Scalability issue due to slower transactions by large number of users</li> </ul>

**Fig. 1.** Legal and Technical Challenges

### 3 Blockchain for Accountability in AI

The decentralized solution by blockchain technology eliminates the need for trust in a central authority by utilizing an immutable and distributed ledger consisting of time-stamped transaction blocks [22]. These blocks are linked through the hash of information stored in the previous block to ensure the integrity of transactions. Any attempt to modify a transaction in one block would require the alteration of subsequent blocks, which is computationally expensive for malicious actors. Therefore, the data recorded on the blockchain remains unaltered and secure, which provides a robust solution to preserve accountability and trustworthiness in AI decision-support systems [23].

Literature on AI-powered auditing systems has acknowledged the potential of combining Blockchain and AI technologies for developing advanced auditing systems [24]. However, there is a need for more research on the productive integration of blockchain as a security layer in AI-based systems. A study proposes the utilization of the InterPlanetary File System (IPFS) as a potential solution for accountability of explainable artificial intelligence (XAI) decisions [25]. It points out the inherent storage limitations associated with the Ethereum blockchain and suggests IPFS's decentralized storage capabilities to manage large datasets effectively. However, the proposed solution lacks experimental results on robustness by performance metrics such as throughput and latency. Another research utilized a combination of IPFS files, blockchain, and cloud storage to link Generative AI and XAI metadata, such as XAI decisions and AI system parameters, to safeguard the system against malicious attacks and manage the usability of AI tools by humans [26]. It stores the Merkle tree (hash tree of SHA256) of previous blocks, which increases computational overhead for auditing. A framework proposed decentralized consensus of several XAI predictors by a smart contract to estimate a final decision [27]. Furthermore, a blockchain-integrated approach with Explainable Deep Neural Networks (x-DNN) has suggested the security and interoperability of sensitive data for medical indemnity insurance [28]. Blockchain has also been employed to amalgamate the expertise of multiple experts for reliable lending decisions to promote financial inclusivity for the underserved community [29].

## 4 Methodology

### 4.1 Explainable Legal Decisions by ER

ER algorithm can combine multiple pieces of independent and highly conflicting evidence by considering the weights and reliability of each piece of evidence to add a more comprehensive explanation for decisions [30, 31]. It is inherently explainable and does not rely on model-agnostic methods such as SHAP, LIME, LRP, or Taylor decomposition to elucidate the reasoning behind decisions made by non-linear models.

The weight of evidence points to the importance of evidence, and reliability points to the quality of the information supporting the evidence. It is an extension of the Dempster-Shafer theory, which extends the probability theory or a generalization of the Bayesian inference [32].

Let, a dataset has  $N$  number of cases, there are  $q$  attributes to evaluate the legal liability, each with  $v$  referential values ( $q \in \{1, \dots, Q\}$ ,  $v \in \{1, \dots, V_q\}$ ). The target attribute ( $\theta$ ) has a possible decision defined in the frame of discernment  $\Theta = \{\theta_1, \dots, \theta_z, \dots, \theta_Z, z \in \{1, \dots, Z\}\}$ . These decisions are mutually exclusive and collectively exhaustive. A piece of evidence for a legal case is denoted by  $e$ ; the  $v^{\text{th}}$  evidence in  $q^{\text{th}}$  attribute as  $e_{v,q}$ . The decision by ER is provided over the power set of  $\Theta$ :

$$P(\Theta) = \{\emptyset, \{\theta_1\}, \dots, \{\theta_Z\}, \dots, \{\theta_1, \dots, \theta_{Z-1}\}, \Theta\} \quad (1)$$

Uncertainty is quantified by the number of samples supporting a class, represents the belief for an outcome. Evidence  $e_{v,q}$  in attribute  $q$  is profiled over a belief distribution:

$$e_{v,q} = \{(e_{\theta,v,q}, \hat{m}_{\theta,v,q}), \forall \theta \in P(\Theta)\} \quad (2)$$

With the belief distribution summing up to 1:

$$\sum_{\theta \in P(\Theta)} \hat{m}_{\theta,v,q} = 1 \quad (3)$$

The normalized probability mass of set of evidence in all attributes for a given legal claim is:

$$\hat{m}_{\theta,v,q} = \begin{cases} 0 & \theta = \emptyset \\ \frac{m_{\theta,v,q}}{(1+w_{\theta,v,q}-r_{\theta,v,q})} & \theta \subseteq \Theta, \theta \neq \emptyset \\ \frac{(1-r_{\theta,v,q})}{(1+w_{\theta,v,q}-r_{\theta,v,q})} & \theta = P(\Theta) \end{cases} \quad (4)$$

Here,  $m_{\theta,v,q}$  is the basic probability mass of evidence  $e_{v,q}$  for a decision  $\theta$ ,  $\hat{m}_{\theta,v,q}$  is the probability mass normalized by weight ( $w_{\theta,v,q}$ ) and reliability ( $r_{\theta,v,q}$ ) of evidence.

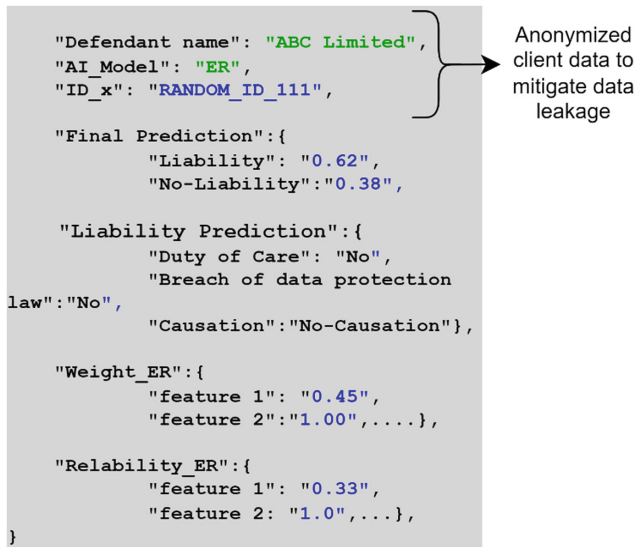
The weight and reliability of evidence could be a subjective judgment of lawyers. Lawyers first assess the initial beliefs for singleton evidence and joint pieces of evidence to incorporate their judgments into the system. The process of human-AI collaboration for convergence to an ultimate-true decision by mitigating noisy decisions can be achieved by Evidential-Reasoning eXplainer (ER-X), which is based on maximum-likelihood evidential reasoning [33]. The set of evidence is initially mapped with the training data. However, in real applications, the comprehensive data coverage for the entire discourse universe is not always available for training purposes. The parameters of evidence supported by data are refined by data-driven optimization. The objective function to optimize weight and reliability is:

$$\begin{aligned} \text{Minimize} : f(w_{\theta,v,q}) &= \frac{1}{2N} \sum_{i=1}^N \sum_{\theta \in P(\Theta)} (m^o - \hat{m}(w_{\theta,v,q}, r_{\theta,v,q}))^2 \\ \text{constraints} : & 0 \leq w_{\theta,v,q} \leq 1, 0 \leq r_{\theta,v,q} \leq 1 \end{aligned} \quad (5)$$

The legal decision (output) from the ER algorithm is not just a result but a carefully crafted presentation designed with the user in mind. It visually demonstrates the algorithm's final decision, along with the belief, weight, and reliability of multiple pieces of evidence representing the circumstances of a legal case. The user interface design and color scheme are thoughtfully chosen to support all kinds of users. For instance, utilizing colorblind-friendly palettes. Frontend developers conduct rigorous usability tests to

fine-tune the features of an interface to make it intuitive and user-friendly for everyone within the organization.

The legal case decisions rendered by the ER algorithm are securely stored in the cloud as text-based JSON files. Each case's explanation is anonymized to strip out client information and specific features of the AI model to mitigate the risk of sensitive information disclosure through prompts designed for the Generative AI, as shown in Fig. 2.

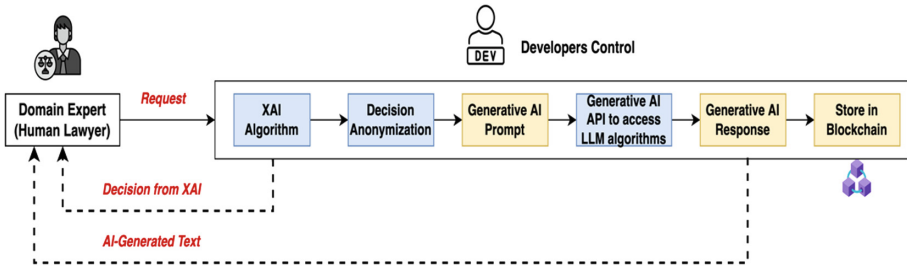


**Fig. 2.** Format of the Explanation Generated by the ER Algorithm for Use as a Prompt in Legal Document Drafting and Response Formulation

## 4.2 Blockchain to Monitor the Usage-Generative AI

The anonymized JSON format of the explanation is concatenated with a fixed prompt to generate a response. Python scripts are used to interact with the LLM models provided by the OpenAI and Google Bard APIs. It automates the submission of prompts in a loop for a large batch of legal cases. For confidentiality and integrity, the work computers of lawyers are strictly restricted from direct access or copy-pasting the information into Generative AI platforms.

The immutability of a blockchain ledger enhances trust within a legal firm, as it ensures that any text generated by an LLM of Generative AI tool can be reliably referenced in the future. The auditing mechanism by blockchain tracks the usage or exclusion of AI-generated text by the lawyer while drafting correspondence. The lawyers leverage AI-generated content to get drafting assistance; they are ultimately responsible for the final draft based on their professional judgment and understanding. Figure 3 demonstrates how human lawyers can collaborate with AI and blockchain technology in a developer-controlled environment.



**Fig. 3.** Human Lawyers in a Developer Controlled Environment to Interact with AI and Blockchain

In the proposed system, only the hash values of the AI-generated response for an  $x^{th}$  individual is stored in the blockchain, along with the case ID ( $ID_x$ ) and *date*. A hash functions as a fingerprint of the information and cannot be reverted to the original data, ensuring non-disclosure of the original information. The SHA256 algorithm is employed for hashing, offering a robust and reliable method for preserving the integrity of the text [34]. SHA256 create a hash value of  $2^{64} - 1$  bits. A malicious actor to tamper with data in the blockchain requires  $2^{128}$  input to create a collision pair with 50% probability; it requires 175,000 CPU-years to find a collision pair [34].

The blockchain-based monitoring mechanism is illustrated in Fig. 4. The raw text-based explanation files are stored off-chain (outside the blockchain) in a secured cloud environment, and a smart contract pushes the hash values of generated files inside the blockchain (on-chain storage). The off-chain data and on-chain data are denoted by  $D_\alpha^x$  and  $D_\beta^x$ , respectively. The prompt and response for an  $x^{th}$  individual is denoted by  $P_x$  and  $R_x$ , respectively. The hash value of  $R_x$  generated by the SHA256 algorithm is denoted by  $H_R^x$ .

The non-tampering or consistency of a Generative-AI response used by a lawyer in the past is verified by an auditor or developers by utilizing the  $GET()$  function within a smart contract (or chain code). The  $GET()$  initiates a query to the blockchain to locate a specific dataset known as a triple  $(ID_x, H_R^x, date)$ , demonstrated in Fig. 5. A full node was managed by Infura blockchain service provider, which allows the storage of hash addresses. The non-existence of this triple indicates tampering with the original off-chain Generative AI response, as this data does not match the on-chain data.

## 5 Case Study on Bank Data Breach

### 5.1 Banking Data Breach Claims

A banking data breach can lead to financial loss, identity theft, and emotional distress. This case study is conducted in a law firm. The law firm acted as an intermediary, serving as a legal advisor and partner to both the bank involved in the data breach and its associated insurance company. According to the UK’s Information Commissioner’s Office (ICO) guidelines, organizations must report any breaches within 72 h of discovery to be eligible for compensation. The ICO shares banking data breaches with regulators, law enforcement, and cybercrime agencies such as the Financial Conduct Authority (FCA)

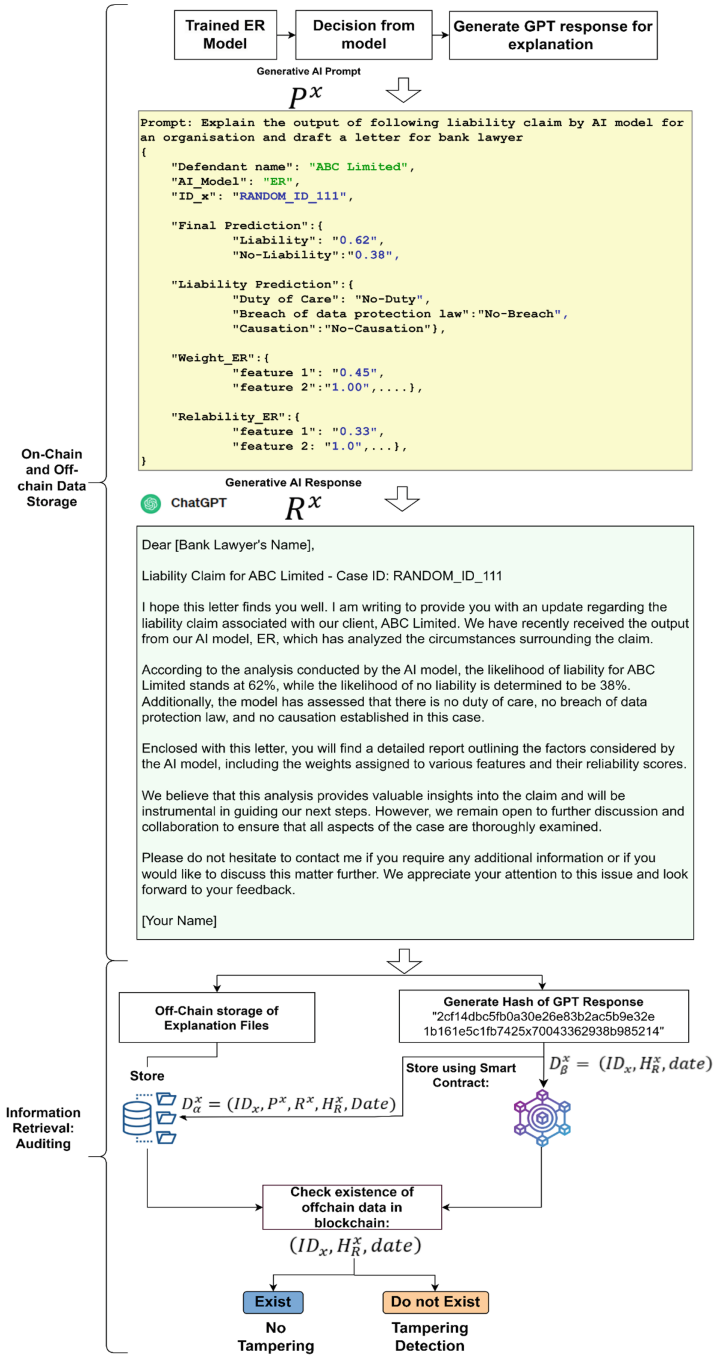


Fig. 4. Blockchain-Based Mechanism to Monitor Generative AI

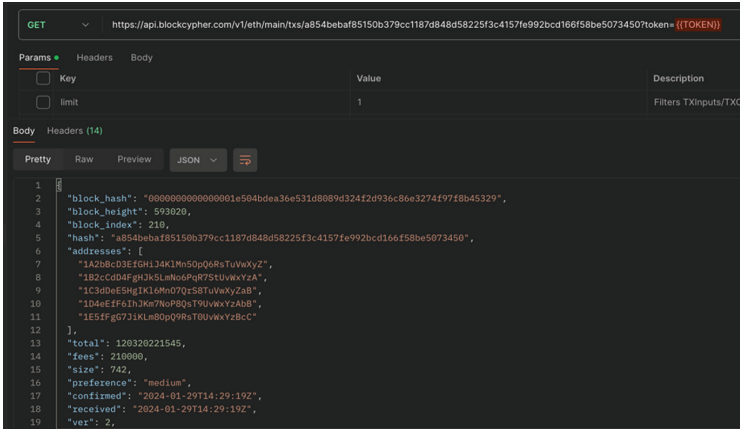


Fig. 5. GET Request in API to Find a Given Case Data in Blockchain. The statement on the top is a GET Request, and below is the Response Body (Data).

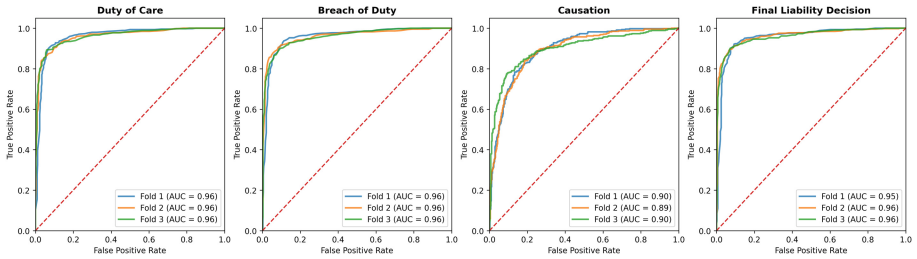
and the National Cyber Security Centre (NCSC) to investigate and identify possible bank liabilities, which further helps insurers pay compensation to the claimant. In this study, the bank liable for the data breach is the defendant, and the client could be an individual, business client, or external partner such as FinTech.

The banking data leakage is a common law ‘tort’ (a civil wrong that causes someone to suffer loss or harm). Tort has three fundamental components. First, “Duty of Care,” also known as the “Quincecare duty,” points to a bank’s obligation to safeguard the confidential financial information of their customers and clients. Second, a “Breach of Data Protection Law” occurs when a bank fails to adhere strictly to data protection regulations such as the General Data Protection Regulation (GDPR) or the Revised Payment Services Directive (PSD2) due to negligence such as cyber theft or bank’s employee misconduct. Third, “Causation” is established when a data breach by a bank infringes an individual’s rights and freedoms or causes financial loss and emotional distress.

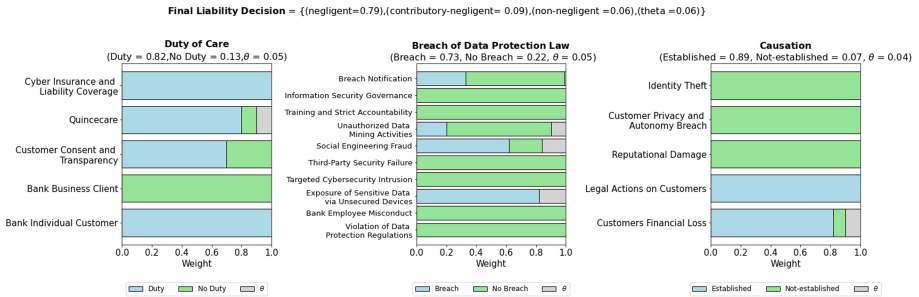
The ER algorithm was trained using a dataset containing 2712 bank data breach cases. The dataset had 8.31%, 21.17%, and 70.52% negligent, contributory-negligent, and non-negligent cases. Figure 6 illustrates the AUC score (model accuracy) for duty of care, breach of duty, causation, and the final liability decision. ER is an explainable AI algorithm.

Figure 7 demonstrates the weight of evidence from all variables points towards a strong case for bank liability due to failure of the bank’s duty of care towards a given customer, data breach from inadequately secured devices, and the establishment of causation due to legal action against the customer by an external organization and financial loss. The end-users, such as legal practitioners, can utilize the reasoning behind the decisions derived from the XAI model to make informed decisions and uphold their professional accountability.





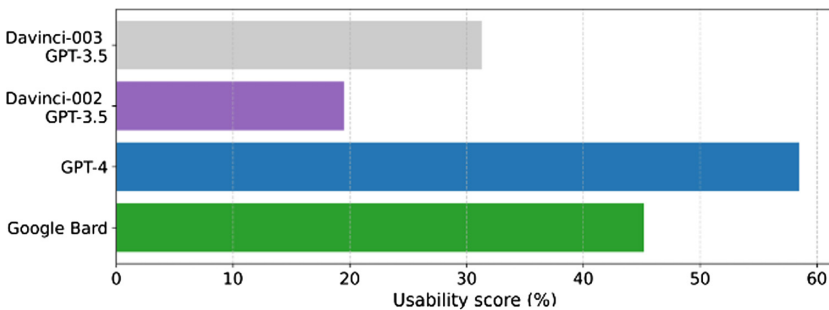
**Fig. 6.** ROC Curve demonstrating AUC Score for Bank Liability Decision by ER



**Fig. 7.** Explainable Decision for a True Positive Liability Case

## 5.2 Usability Test of LLM Models by Generative AI APIs

Multiple versions of LLM models, such as GPT-3.5 (text-davinci-003 and text-davinci-002), gpt-4 (GPT-4), and Google Bard, were accessed through their APIs via a Python script to generate text for the decision rendered in a JSON format for each  $x^{th}$  defendant. The prompt structure for each input transmitted through the loop by the Python script was a string datatype.



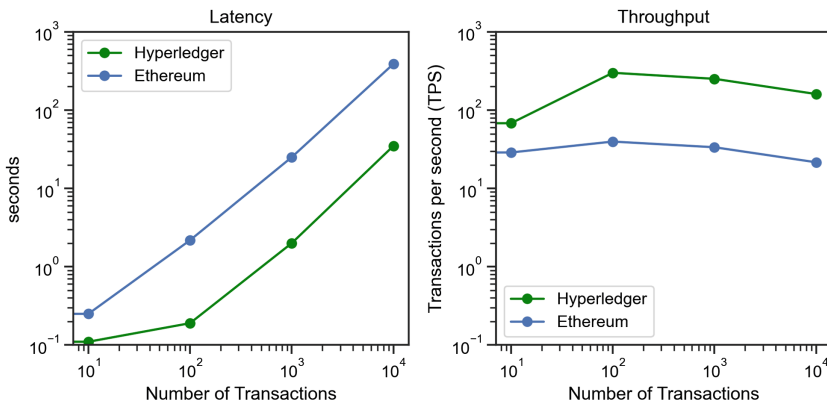
**Fig. 8.** Usability Testing of Generative AI Tools

An experimental analysis was conducted to assess the usability of text generated by three OpenAI GPT models and Google Bard, specifically in the context of banking cyber-attack cases. A panel of 25 legal professionals evaluated the applicability of content generated by LLM models in crafting legal narratives on bank negligence in protecting financial data resulting in customer harm. Each lawyer was provided with a single response from each LLM algorithm; they were restricted from using multiple versions obtained from multiple prompts.

Turnitin software was utilized to measure the percentage of text similarity between the draft written by lawyers and OpenAI’s and Google algorithms. Figure 8 indicates a strong preference for the GPT-4 model among legal professionals; 60.5% of its output was utilized to draft legal documents. Google Bard’s contributions were found to be 15.3% less utilized than GPT-4. The legal experts preferred to use only 33.3% and 19.5% of the content generated by the text-davinci-003 and text-davinci-002, respectively.

### 5.3 Blockchain Auditing Performance

The performance of blockchain networks in auditing the past usage of AI-generated content by human lawyers is assessed by throughput and latency. Throughput is the number of valid transactions committed by the blockchain per time unit (usually time in seconds). It is also called the transaction per second (TPS) rate [35, 36]. Latency is the time taken for data transfer.



**Fig. 9.** Latency and Throughput in Blockchain Networks to Record Audit Data

AI-generated content utilized by professionals such as banking lawyers must be recorded quickly in the blockchain network for future auditing and monitoring. Therefore, an optimal blockchain network should feature high throughput and low latency. Figure 9 demonstrates the experimental results for the performance of blockchain implementations on Ethereum (Public Network) and Hyperledger Fabric (Private Network) based on variations in the number of transactions from 1 to 10,000. The 10,000 transactions were executed by repeatedly dispatching the hash data of 2,712 legal cases for performance evaluation. The result indicates that Hyperledger Fabric outperforms

Ethereum; it exhibits higher throughput and reduced latency. Each case file archived on the blockchain averaged 0.10 Megabytes in size. Post-storage, the average time required to audit on Hyperledger Fabric was 90 Secs, and Ethereum was 120 Secs.

## 6 Discussion, Limitations, and Future Research

The pilot study was successfully conducted with the assistance of a banking and finance lawyer. Lawyers participating in the study provided positive feedback on the effectiveness of using LLMs to mitigate concerns about data leakage and potential accountability issues among their peers. Tech firms such as Microsoft Azure provide OpenAI services. They augment prompts with data retrieved from client data sources without exposing it to the OpenAI database. However, the firms still encounter the challenge of confidential information leakage to LLMs. Additionally, the current systems do not provide immutable records of AI-generated content utilized by lawyers, which is crucial for tracking accountability in legal practices.

In this study, each lawyer received a single response from each LLM algorithm and was prohibited from utilizing multiple versions obtained from multiple prompts. The protocol ensured that lawyers could not directly copy and paste into the Generative AI tools, as their respective firms monitored and controlled their computer usage. Instead, they submitted their requests through an API gateway that anonymized the prompts before forwarding them to a Generative AI API, such as GPT-4. However, in a real application, a lawyer might demand multiple versions of different prompt statements.

A significant limitation of this approach is its reliance on a fixed prompt; the study did not explore prompt engineering to optimize the input for generating high-quality prompt engineering practices that could eliminate the need for generations of multiple versions and the subsequent storage of multiple version hashes in the blockchain.

On the positive side, this framework is not intended to make legal decisions by LLM but rather to assist in generating decision texts for legal reasoning derived from an XAI algorithm such as ER. This approach minimizes the risk of legal facts hallucinations and reduces the oversight by human lawyers if they utilize the content generated from LLM algorithms. However, further investigation is needed to assess whether lawyers might overlook errors in the AI-generated text when preparing legal documents.

## 7 Conclusion

This research presents the technique for the responsible and secure application of Generative AI tools by legal practitioners in supporting the drafting of legal decisions derived from an XAI algorithm by leveraging the immutability feature of blockchain technology. The proposed framework has high confidentiality and integrity, as it uses anonymized inputs for AI prompt requests to avoid data leakage and stores hash values of legal documents and AI-generated texts in blockchain for auditability. A blockchain-based auditing process ensures the consistency and non-tampering of Generative AI responses used by lawyers in the past. The immutability and decentralization of the blockchain ledger provide a secure and transparent record-keeping system.

This research concludes the findings from a case study conducted within a law firm on tort liability cases against banks for data breaches. It shows the application and results of explainable legal decisions by the ER algorithm and the subsequent use of its raw textual decisions for drafting legal documents. Comparative analysis of LLM models indicated that GPT-4 produces high-quality legal drafting content. The auditing performance of two blockchain networks, Ethereum and Hyperledger Fabrics, shows that Hyperledger Fabric has better throughput and latency.

**Acknowledgments.** This work is fully funded by the University of Liverpool, the ULMS FinTech impact research grant. We are grateful to four anonymous reviewers for their valuable feedback.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Padovan, P.H., Martins, C.M., Reed, C.: Black is the new orange: how to determine AI liability. *Artificial Intell. Law* **31**(1), 133–167 (2023)
2. Dwivedi, Y., Yogesh, K., Kshetri, N., Hughes, L., et al.: So what if ChatGPT wrote it? Multidisciplinary perspectives on opportunities, challenges and implications of generative conversational AI for research, practice and policy. *Int. J. Inf. Manage.* **71**, 102642 (2023)
3. Choi, J.H., Hickman, K.E., Monahan, A.B., Schwarcz, D.: ChatGPT goes to law school. *J. Leg. Educ.* **71**, 387 (2021)
4. Katz, D.M., Bommarito, M.J., Gao, S., Arredondo, P.: Gpt-4 passes the bar exam. *Phil. Trans. R. Soc. A* **382**(2270), 20230254 (2024)
5. Sleiman, J.P.: Generative artificial intelligence and large language models for digital banking: First outlook and perspectives. *J. Dig. Bank.* **8**(2), 102–117 (2023)
6. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144 (2016)
7. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
8. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, K.R.: Layer-wise relevance propagation: an overview. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 193–209 (2019)
9. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recogn.* **65**, 211–222 (2017)
10. Collenette, J., Atkinson, K., Bench-Capon, T.: Explainable AI tools for legal reasoning about cases: a study on the European court of human rights. *Artif. Intell.* 103861 (2023)
11. Rissland, E.L., Ashley, K. D.: A case-based system for trade secrets law. In: *Proceedings of the 1st International Conference on Artificial Intelligence and Law*, pp. 60–66 (1987)
12. Alevan, V.A.: *Teaching Case-Based Argumentation Through a Model and Examples*. University of Pittsburgh, Pittsburgh (1997)
13. Bruninghaus, S., Ashley, K.D.: Predicting outcomes of case based legal arguments. In: *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, pp. 233–242 (2003)
14. Al-Abdulkarim, L., Atkinson, K., Bench-Capon, T.: Abstract dialectical frameworks for legal reasoning. In: *Legal Knowledge and Information Systems*, pp. 61–70 (2014)

15. Sachan, S., et al.: Augmented intelligence for transparent decision making in insurance claims. In: 31st European Conference on Operational Research (2021)
16. Norkute, M., Herger, N., Michalak, L., Mulder, A., Gao, S.: Towards explainable AI: assessing the usefulness and impact of added explainability features in legal document summarisation. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–7 (2021)
17. Sergot, M.J., Sadri, F., Kowalski, R.A., Kriwaczek, F., Hammond, P., Cory, H.T.: The British Nationality Act as a logic program. *Commun. ACM* **29**(5), 370–386 (1986)
18. Schild, U.J.: Open-textured law, expert systems and logic programming (1990)
19. Bayle, A., Koscina, M., Manset, D., Perez-Kempner, O.: When blockchain meets the right to be forgotten: technology versus law in the healthcare industry. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 788–792 (2018)
20. Solove, D.J.: The limitations of privacy rights. *Notre Dame Law Rev.* **98**, 975 (2022)
21. Rosen, J.: The right to be forgotten. *Stanford Law Rev. Online* **64**, 88 (2011)
22. Austin, T.H., Di Troia, F.: A blockchain-based tamper-resistant logging framework. In: Silicon Valley Cybersecurity Conference, pp. 90–104 (2022)
23. Kayikci, S., Khoshgoftaar, T.M.: A general survey on combining ML and Blockchain: blockchain meets machine learning: a survey. *J. Big Data* **11**, 9 (2024)
24. Salah, K., Rehman, M.H.U., Nizamuddin, N., Al-Fuqaha, A.: Blockchain for AI: review and open research challenges. *IEEE Access* **7**, 10127–10149 (2019)
25. Malhotra, D., Srivastava, S., Saini, P., Singh, A.K.: Blockchain-based audit trailing of XAI decisions: storing on IPFS and ethereum blockchain. In: International Conference on Communication Systems & Networks (COMSNETS), pp. 1–5 (2021)
26. Sachan, S., Liu, X.: Blockchain-based auditing of legal decisions supported by explainable AI and generative AI tools. *Eng. Appl. Artif. Intell.* **129**, 107666 (2024)
27. Nassar, M., Salah, K., Ur Rehman, M.H., Svetinovic, D.: Blockchain for explainable and trustworthy artificial intelligence. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **10**(1), 1340 (2020)
28. Sachan, S., Muwanga, J.: Integration of Explainable Deep Neural Network with Blockchain Technology: Medical Indemnity Insurance (2023)
29. Sachan, S., Fickett, D.S., Kyaw, N.E.E., Purkayastha, R.S., Renimol, S.: A blockchain framework in compliance with data protection law to manage and integrate human knowledge by fuzzy cognitive maps: small business loans. In: IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 1–4 (2023)
30. Sachan, S., Almaghrabi, F., Yang, J.-B., Xu, D.-L.: Evidential reasoning for preprocessing uncertain categorical data for trustworthy decisions: an application on healthcare and Finance. *Expert Syst. Appl.* **185**, 115597 (2021)
31. Sachan, S., Yang, J.-B., Xu, D.-L., Benavides, D.E., Li, Y.: An explainable AI decision-support-system to automate loan underwriting. *Expert Syst. Appl.* **144**, 113100 (2020)
32. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. In: *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pp. 57–72 (2008)
33. Sachan, S., Almaghrabi, F., Yang, J.-B., Xu, D.-L.: Human-AI collaboration to mitigate decision noise in financial underwriting: a study on FinTech innovation in a lending firm. *Int. Rev. Financ. Anal.* **93**, 103149 (2024)
34. Bouam, M., Bouillaguet, C., Delaplace, C., Noûs, C.: Computational records with aging hardware: Controlling half the output of SHA-256. *Parallel Comput.* **106**, 102804 (2021)
35. Tran, T.H., Pham, H.L., Nakashima, Y.: A high-performance multimem SHA-256 accelerator for society 5.0. *IEEE Access* **9**, 39182–39192 (2021)
36. Kuzlu, M., Pipattanasomporn, M., Gurses, L., Rahman, S.: Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability. In: IEEE International Conference on Blockchain (Blockchain), pp. 536–540 (2019)



# Multi-modal Machine Learning Model for Interpretable Malware Classification

Fahmida Tasnim Lisa<sup>(✉)</sup>, Sheikh Rabiul Islam<sup>(✉)</sup>, and Neha Mohan Kumar

Department of Computer Science, Rutgers University - Camden,  
Camden, NJ 08102, USA

{fahmidatasnim.lisa,sheikh.islam,neha.m}@rutgers.edu

**Abstract.** As mobile devices have become universal and are now integral to every facet of our everyday lives, the alarming rise in mobile malware poses a significant threat to the security of sensitive and private information stored or transferred to/from our mobile devices (e.g., smartphones, and tablets). This paper introduces an innovative method for mobile malware detection using a multimodal deep learning approach on two different modalities of datasets: grayscale images of android malware and tabular data. We leverage Explainable AI (XAI) to enhance the interpretation of classification results for both unimodal and multimodal approaches. Furthermore, we create an explainable malware classifier using Knowledge Graph to compare its performance with multimodal learning. The classifiers provide improved explainability with minimal to no compromise in accuracy when classifying malware samples.

**Keywords:** Mobile Malware · Multimodal Learning · Knowledge Graph

## 1 Introduction

The surge in mobile malware presents a growing concern for consumer devices. With its extensive user base, the Android operating system, being one of the most widely embraced mobile platforms, becomes a focal point for malicious actors aiming to exploit vulnerabilities. Android's popularity, while advantageous, introduces vulnerabilities due to its open nature. The system's permission for users to download applications from third-party sources serves as a breeding ground for malware. This accessibility, though providing flexibility, elevates the risk of inadvertent exposure to harmful software and creates a thriving environment for various mobile malware types such as viruses, worms, mobile bots, phishing attacks, ransomware, and spyware. This vulnerability exposes users to a spectrum of threats, including the theft of sensitive data, deceptive fraud schemes, and compromise of device functionality. Phishing attacks, in particular, deceive users into divulging confidential information, while ransomware can lock users out of their devices, demanding payment for restoration. The range and

severity of these threats underscore the need for a proactive approach to secure mobile devices.

As mobile devices increasingly integrate into our personal and professional lives, safeguarding against these evolving threats becomes paramount. Heightened awareness, education on potential risks, and adherence to secure practices, especially in the realm of third-party app installations, are essential for ensuring the security of Android devices.

Studies on Android malware detection using machine learning approaches can be broadly categorized into two methodologies: static analysis and dynamic analysis. In static analysis, researchers extract detailed information from the Android Package Kit (APK) installation file. This includes insights into the app's manifest, permissions, API calls, intents, and more. On the contrary, dynamic analysis focuses on monitoring an application's behavior during execution, examining aspects such as shared memory usage, system calls, and process activity within a controlled environment or sandbox. A number of deep learning and ensemble learning-based techniques for Android malware detection have been proposed in earlier research [2, 15]. Malware data can be represented using a variety of modalities, including text, code, images, and tabular data. This multifaceted nature of malware data needs an approach that can handle and process various modalities, efficiently. Numerous studies have used multimodal learning techniques on multiple data modalities. For example, [6] employs multimodal deep learning for problem report classification while working with text, graphics, and code. Similarly, multimodal deep learning and ensemble learning are employed by [15]. This paper focuses on the data collected from CIC-AndMal-2020 [5, 13] of dynamic analysis and image data on Android samples [4]. We propose a multimodal machine learning model that combines image (i.e., grayscale image) and tabular numerical data (i.e., features acquired from app memory, APIs, and other sources) for Android malware detection. The combination of modalities enables us to capture both the functional components of an app and its grayscale visual representation. Our proposed method achieves 98.54% prediction accuracy for mobile malware detection.

The contributions of this study can be summarized as follows:

- We integrate malware data from multiple sources, including tabular data from CIC-AndMal-2020 [5, 13] and grayscale image data from [4], to develop multimodal deep learning model that can perform with good accuracy.
- We evaluate the effectiveness of our multimodal model by comparing it with the unimodal image model and unimodal tabular data model. We also compare the multimodal model with and without including Principal Component Analysis (PCA). Furthermore, we use Knowledge graph embeddings in the tabular dataset to compare the performance of the multimodal model. We also compare the concatenation-based multimodal fusion to elementwise-multiplication-based multimodal fusion. Additionally, we also compare the multimodal model trained on knowledge graph embeddings on the tabular dataset.

- Through explainable AI, we demystify the classification process, empowering users, and security researchers to understand the underlying threats. We used multiple explainable artificial intelligence techniques to explain our proposed model.

The remainder of the paper is organized as follows. Section 2 introduces the related works. Section 3 proposes our approach. Section 4 describes the experimental results and discussion of our results. Section 5 concludes the paper.

## 2 Related Works

The authors in [4] proposed a novel approach for Android malware detection and family identification by utilizing image-based representations of mobile applications as input for an explainable deep learning model. Their method demonstrated high effectiveness with an average accuracy ranging from 0.96 to 0.97 across 8,446 Android samples, encompassing six malware families and one trusted sample family, while also providing interpretability for model predictions. There's also an extensive review of machine learning-based malware detection techniques for Android platforms [8]. It focuses on the advantages and disadvantages of certain methods, such as static, dynamic, and hybrid analysis. Static analysis of an application's permissions, code, and API calls has been used by researchers to identify malware. Support vector machines (SVM) are used in the work of [9] to handle sensitive data, classify Android applications according to their functions, and analyze relevant subjects and data flows. There has been a study [7] that used machine learning on statically generated app attributes in a different study and obtained promising detection accuracy. This strategy is effective and simple to use.

Several studies have been conducted using multimodal learning with deep learning to detect malware. Researchers [2] present a taxonomy that divides deep malware detection and classification methods that are resistant to zero-day exploits into four groups: adversarial resistant, few-shot, unsupervised, and semi-supervised. In a separate work [16] presents SusTriage, a technique for triaging bug reports that combines ensemble learning and multi-modal deep learning to achieve excellent prediction performance while also preserving the long-term viability of open source communities. One research [6] introduces a novel multimodal model for issue report classification, leveraging text, images, and code information. Experimental results demonstrate a substantial improvement (5.07% to 14.12% higher F1-score) compared to traditional text-based models. This approach highlights the efficacy of utilizing heterogeneous data for enhanced issue classification. The significance of multimodal deep learning to enhance information processing by combining several modalities-image, video, text, audio, body motions, face expressions, and physiological signals-was also investigated [15]. Using an OCR-generated noisy text as input, researchers [1] integrated word and image embeddings using a multimodal neural network. Their proposed method greatly increased the Tobacco3482 and RVL-CDIP datasets'



classification accuracy. The work of [4] offered a useful approach in a different study for creating interpretable models related to mobile malware detection. This paper [12] worked with prior knowledge represented as Cybersecurity Knowledge Graphs (CKGs), to guide the exploration of a Representation Learning algorithm to detect malware. Our work highlights the significance of interpretable deep learning and machine learning models in malware detection by combining data from many sources to classify mobile malware. Explainability allows us to better understand the characteristics and actions of malware across a variety of modalities, which improves our ability to identify its ever-evolving avenues.

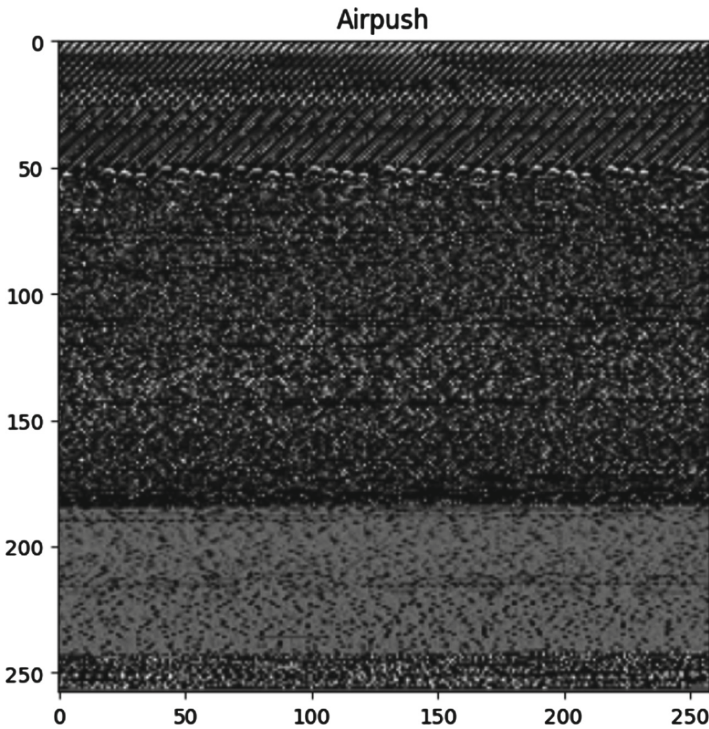


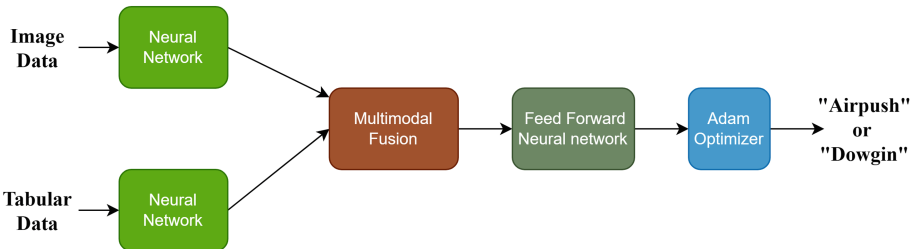
Fig. 1. Sample Android Malware Image

### 3 Proposed Methodology

#### 3.1 Overview

Our proposed approach uses multimodal deep learning on image and tabular data. Additionally, we implement explainable AI techniques to help understand decisions depicted by the most contributing features. Figure 2 depicts a high level overview our proposed approach of multimodal mobile malware classification.

We evaluate the unimodal tabular data on eight distinct machine-learning algorithms namely, K-Nearest Neighbors (KNN), Decision Trees, Logistic Regression, Naive Bayes, Random Forest, XGBoost, Support Vector Machine(SVM) with a linear kernel, and SVM with a Radial Basis Function (RBF) kernel. Each algorithm contributes to the exploration and comprehensive understanding of the dataset's characteristics. Similarly, we evaluate the image data on a Feed-forward Neural Network architecture. For multimodal fusion of both modalities of data, we use separate neural networks on the image and tabular data and then use the output layer of both neural network channels to do multimodal concatenation of the features. Afterward, we evaluated the concatenated features on a neural network model with fully-connected layers. We also do a multimodal feature fusion using element-wise multiplication of features to compare the performance of the concatenation-based feature fusion of the multimodal approach. Furthermore, we also use Knowledge graph node embeddings on the tabular dataset and feed it into the multimodal model to calculate different performance metrics. For the interpretability of the unimodal tabular model, we use a heatmap to investigate feature correlation and importance. We also implemented GradCAM [14] to understand how the unimodal image model classifies grayscale malware images. Explanations from multiple perspectives ultimately help in a better understanding of the decision.



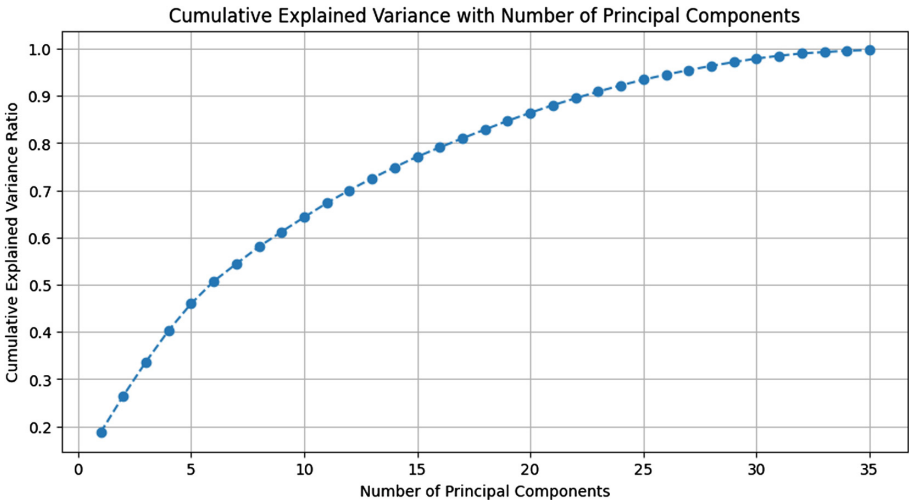
**Fig. 2.** Proposed Methodology

### 3.2 Dataset

For our study, we used two different types of datasets CCCS-CIC-AndMal-2020 [5, 13] and Android Image dataset collected from the works of this paper [4].

The CCCS-CIC-AndMal-2020 dataset, publicly released in 2020, is a collaborative effort of the Canadian Centre for Cyber Security and the Canadian Institute for Cybersecurity. This extensive dataset comprises a total of 400,000 Android applications, with half being regular benign apps and the remaining half categorized as malicious apps. For the dynamic analysis, the dataset encompasses 141 features, categorized into Memory (23 features), API (105 features), battery (2 features), network (4 features), logcat (6 features), and process (1 feature). This dataset comprises over 50,000 samples, having 14 distinct malware

categories, including Riskware, Adware, No\_Category, Zero\_Day, Trojan, Ransomware, Trojan\_Spy, Trojan\_SMS, Trojan\_Dropper, Potentially Unwanted Apps (PUA), Backdoor, Scareware, FileInfector, and Trojan\_Banker. To maintain consistency with the Android image dataset, which focuses on three specific classes (Airpush, Dowgin, and Fusob), we selected the Adware and Ransomware categories from the CIC dataset. Unfortunately, the CIC dataset's Ransomware category lacked Fusob samples, contrary to our expectations. There are a total of 2,927 samples which are split as: 2017 samples in the Airpush class and 910 samples in the Dowgin class. We excluded the 'Hash' and 'Category' columns from the datasets due to their lack of relevance to the model. To ensure compatibility with the Image dataset [4], we explicitly use features taken from the Airpush and Dowgin families in the tabular dataset. Furthermore, we eliminated the use of Fusob samples from the image dataset. We collected approximately 1,200 samples of grayscale android malware images. From this dataset, 749 samples of Airpush, and 489 samples of Dowgin. Figure 1 is a visualization of a random sample's grayscale image.



**Fig. 3.** Variance distribution with the number of principal components.

### 3.3 Dataset Preprocessing and Feature Extraction

In the data preprocessing phase, two distinct datasets were utilized - the CIC-AndMal-2020 dataset [5, 13] and the Android Image Dataset [4].

From the CIC-AndMal-2020 dataset, only specific classes, namely Airpush and Dowgin, were selected, resulting in a collection of 2,927 samples. There were 2017 samples in the Airpush class and 910 samples in the Dowgin class.

Using similar feature values in machine learning algorithms leads to faster and more effective training compared to a data set with dissimilar values in data points. Dissimilar points or feature values might lead to slower understanding and lower accuracy. We used Scikit-Learn’s Standard Scaler to standardize the data. To prepare data for feature extraction, we dropped two redundant columns. Note that there are no missing values in this dataset. Finally, we apply a One-HotEncoder on our label as a standard data pre-processing task. This encoding process assigns a unique numeric label to each distinct family category, facilitating the integration of categorical data into machine learning models. Recursive Feature Elimination (RFE) [3] along with GridSearch was performed with four classifiers-Random Forest (RF), Decision Tree (DT), Logistic Regression (LR), and Adaboost. GridSearch was applied during this process. The outcome of RFE was a list of 41 common features out of 141 features. To further reduce dimensionality, Principal Component Analysis (PCA) [10] was employed with a grid search, resulting in the selection of 35 components. The goal of PCA is to transform data from a high-dimensional space to a low-dimensional one while preserving its essential features. Dealing with high-dimensional data can be challenging due to computational complexities and sparsity caused by the curse of dimensionality. Smaller datasets are easier to analyze and visualize, making it simpler for machine learning algorithms to process information quickly. Our analysis revealed that utilizing 35 principal components could effectively capture over 99% of the dataset’s variance. For a visual representation, refer to Fig. 3 illustrating the distribution of variance concerning the number of principal components.

Furthermore, for the Android Image Dataset, we collected approximately 1,200 samples of grayscale android malware images. See Fig. 1 for visualization of the image dataset. The images are of two categories: Airpush, and Dowgin which have similarities with the same malware categories as the CIC-AndMal-2020 dataset. Since the image samples were of different sizes, we resized the entire dataset. After that, we normalized the images and then put the images through a CNN-based channel. The images are then inputs of size  $(258 \times 258)$ . The model also maintains three convolution layers with a kernel size of  $(5, 5)$  followed by the ReLU operation. The model also applies a max-pooling operation twice with a kernel size of  $(5, 5)$ . After the final max-pooling operation, the results go through a fully connected layer. Finally, it extracts the image features in a feature size of  $(1 \times 1 \times 256)$ .

### 3.4 Models

#### Unimodal Model

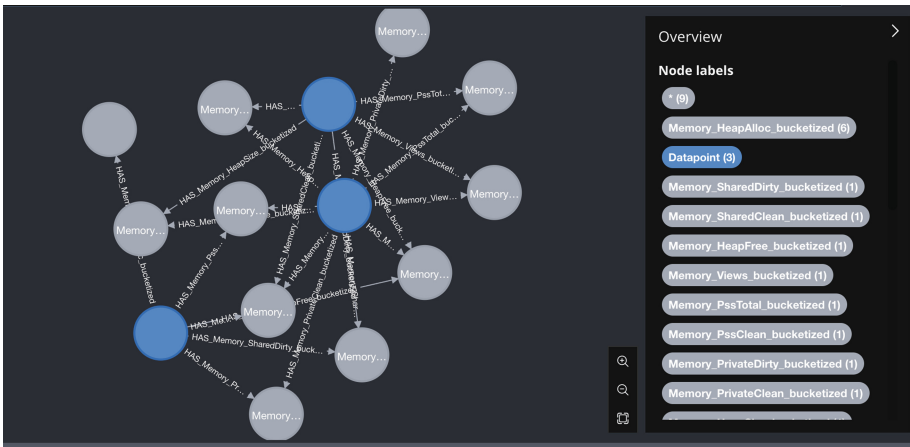
- Tabular Data

We use 2,927 samples of tabular data from CIC-AndMal-2020 [5, 13]. Of which, 2017 samples are in the Airpush class and 910 samples in the Dowgin class. Here we apply multiple machine learning classifiers. Following feature

extraction, we use Decision Tree (DT), Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM) with Linear Kernel, Support Vector Machine with RBF kernel, K-Nearest Neighbors (KNN), and XGBoost classifiers. Our decision to include these algorithms was motivated by their prevalence in the literature and their established effectiveness across various domains. The results are presented in Table 3 in terms of accuracy, recall, precision, and f1 score.

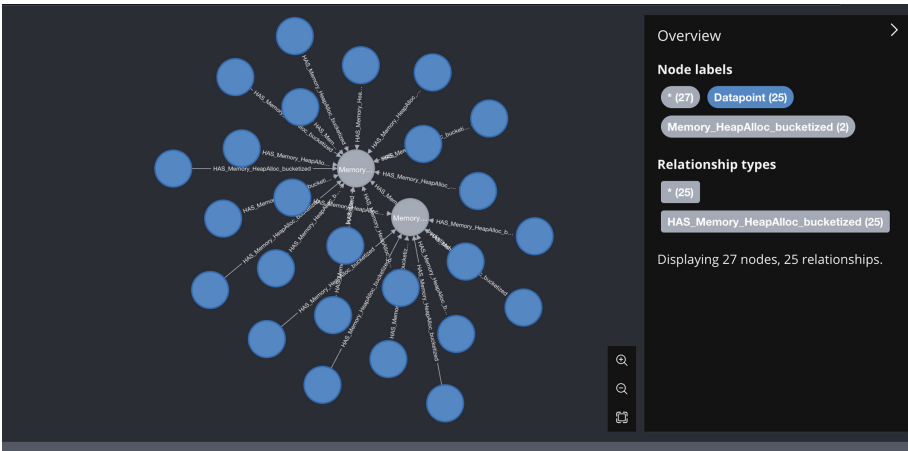
– Image Data

There are about 1,200 samples of grayscale android malware images we used from the Android Image Dataset [4]. We then apply a Convolution Neural Network (CNN) model on the preprocessed image data [4]. This choice stems from the inherent suitability of CNNs for handling image data due to their ability to capture spatial hierarchies and local patterns effectively. Moreover, CNNs have demonstrated remarkable performance in numerous image-related tasks, making them a natural choice for our analysis. The employed CNN consists of three convolutional layers with a kernel size of (5, 5) followed by the ReLU activation function. The model also applies a max-pooling operation twice with a kernel size of (5, 5). After the final max-pooling layer, the output is passed to a fully connected layer. Finally, it extracts the image features in a feature size consistent with the feature size of the tabular data. We also use a sigmoid activation function for the output layer. As a result, we achieve 91.35% prediction accuracy. Table 1 displays the final results in terms of accuracy, recall, precision, and f1 score.



**Fig. 4.** Multiple bucketized column nodes connected to data points with respective relationships

**Multimodal Model.** For preprocessing, we sort both the image and tabular data based on the labels to ensure consistency. Due to the presence of a larger amount of image data compared to the tabular data, we take a random subset of the training and test data so we have an equal amount for our model. The tabular data is processed through RFE as discussed earlier and the images are resized to a standard size of  $128 \times 128$ . The multimodal model has two neural network branches. The tabular input branch is made up of a fully connected layer with 32 neurons and a ReLU activation function. The image input branch has a convolutional layer with 16 filters with a kernel size of  $(3 \times 3)$  and a ReLU activation function. The output of this layer is then flattened. The output of these layers is then concatenated using the concatenate function from *Keras*. This concatenated data is a unified representation that captures both the image and tabular information. After concatenation, the data is passed through a fully connected layer with 64 neurons and a ReLU activation function. The output layer is a fully connected layer with a single neuron and a sigmoid activation function. The model is compiled with the Adam optimizer, binary cross entropy loss, and accuracy as the metric. It is then trained for 20 epochs with a batch size of 32. Furthermore, instead of using the concatenate function, we run the multimodal model on a simple elementwise multiplication as an entry-level multimodal approach, which has a similar model architecture and is run for 10 epochs with a batch size of 32.



**Fig. 5.** Memory HeapAlloc column’s bucketized nodes connected to numerous data points

Our approach leverages knowledge graph node embeddings applied to the tabular dataset to enhance the dataset’s interpretability. We categorized several columns into five distinct bins each. We then created new columns where the original values were replaced with their corresponding bin names. For example,

if the original column was named ‘abc’, the bins were labeled as ‘abc-1’ through ‘abc-5’. Subsequently, we established a graph-based representation of the data. Each data point in the dataset was assigned a node, and additional nodes were created to represent each bin as shown in Figs. 4 and 5. We connected the data point nodes to the bin nodes based on the original dataset’s bin assignments. Using the graph2vec [11] algorithm, we generated embeddings for each data point node, capturing the local structural information within the graph. These embeddings were combined with the original feature set, enriching the model’s representation. Finally, we trained the multimodal classifier on the augmented feature set, allowing the model to leverage both the original features and the graph-derived representations to improve predictive performance.

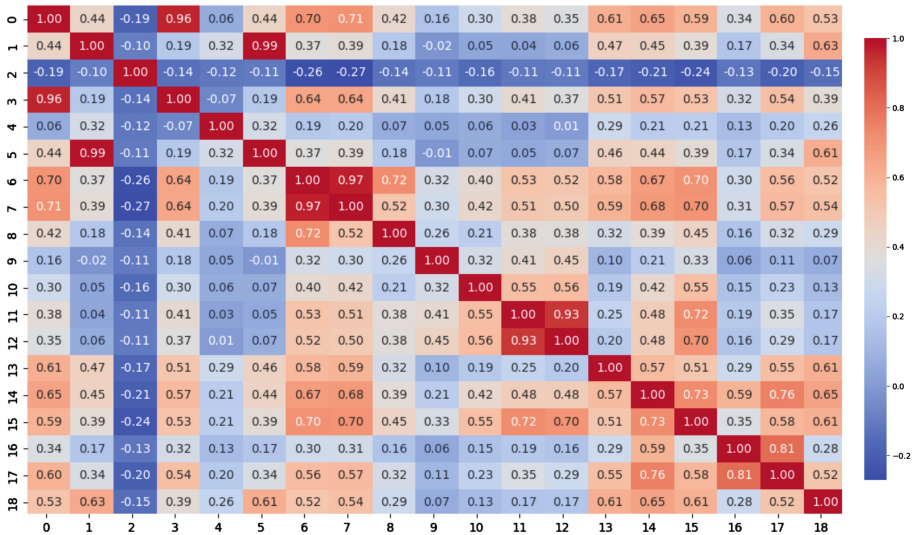


Fig. 6. Heatmap of feature correlation of CIC-AndMal-2020 (first 18 features out of 141)

## 4 Results and Discussion

We evaluate our multimodal deep learning model by comparing it with the unimodal model for images and the unimodal model for tabular datasets. For the unimodal image model, we applied CNN architecture and achieved 91.35% accuracy lower than our multimodal model, where it achieved 98.71% accuracy. We are able to get diverse perspectives and explainable predictions from the use of multiple modalities and explainability techniques (Heatmap and GradCAM). We expect that our combined model can be translated to malware detection

**Table 1.** Comparative Analysis between methods

Modality	Accuracy	Precision	Recall	F-1 Score
Unimodal model for Image Data	91.35%	90.38%	87.63%	88.98%
Multimodal model (Concatenation) w/ PCA	98.71%	98.33%	97.67%	98.1%
Multimodal model (Concatenation) without PCA	97.66%	96.33%	96.07%	50.19%
Multimodal model (Fusion)	96.2%	95.28%	94.58%	66.78%
Multimodal model with KG node embeddings (Concatenation)	95.39%	95.54%	88.16%	92.16%

techniques from both tabular and image datasets. Our model is able to detect android malware of “Airpush” and “Dowgin” class of mobile malware.

During data pre-processing of CIC-AndMal-2020 tabular data, we use a heatmap to check the feature correlation. The heatmap is visualized in Fig. 6. From the heatmap, we conclude that the features had varied correlations with each other, some high and some low. The heatmap gave us a quick idea about the feature importance although represents only linear correlations. Additionally, we perform Recursive Feature Elimination (RFE). RFE is essential in selecting more informative features and achieving high accuracy. After performing RFE, we used PCA. Upon employing Principal Component Analysis (PCA) to reduce dimensionality, we observed a notable enhancement in the performance of our multimodal model, achieving an accuracy of 98.71%. Conversely, when running the multimodal model without PCA, the accuracy experienced a discernible drop to 97.66%. Alternatively, running the machine learning models with PCA we see a drop in accuracies, precision, recall, and F-1 score among the machine learning classifiers. Among the models trained without PCA, RF and XGBoost classifiers performed the best with 96% and 95% prediction accuracies, respectively. SVM with linear and RBF kernel achieve 94% and 96% accuracies while KNN and LR both achieve 94% prediction accuracy. DT classifier achieves 92% prediction accuracy while Naive Bayes achieves only 80% prediction accuracy. Despite the notable improvements observed in the performance of the multimodal model with PCA, it’s imperative to note the impact on individual modalities, particularly the tabular dataset. Notably, when employing PCA, the Random Forest (RF) and XGBoost classifiers exhibited accuracies of 92% and 94%, respectively, which were marginally lower compared to their counterparts without PCA. Table 3 provides a comprehensive overview of the machine learning models’ metrics trained on the tabular dataset with PCA, while Table 2 delineates the results without PCA. The classifiers’ overall accuracy aligns with their precision, recall, and F1 scores, providing a comprehensive evaluation of their classification capabilities. Evidently, there’s a discernible drop in accuracies across all machine learning models in the absence of PCA. However, despite the reduction in performance for tabular data alone, our decision to utilize PCA stems from its overall enhance-



**Table 2.** Classification Matrix for Machine Learning classifiers on CIC-AndMal-2020 tabular dataset without PCA. Here Class 0 is Airpush and Class 1 is Dowgin

Classifier	Class	Precision	Recall	F1 Score	Accuracy
XGBoost	Class 0	99%	99%	99%	98%
	Class 1	98%	97%	97%	
Decision Tree	Class 0	97%	97%	97%	95%
	Class 1	93%	92%	92%	
Logistic Regression	Class 0	95%	96%	94%	94%
	Class 1	90%	89%	90%	
KNN	Class 0	95%	99%	97%	96%
	Class 1	97%	87%	92%	
SVM with RBF Kernel	Class 0	95%	97%	96%	95%
	Class 1	92%	92%	92%	
SVM with Linear Kernel	Class 0	96%	96%	96%	94%
	Class 1	90%	90%	90%	
Random Forest	Class 0	98%	99%	99%	98%
	Class 1	97%	96%	97%	

ment of the multimodal model’s performance. By incorporating PCA, we achieve an overarching improvement in multimodal accuracy, albeit at the expense of some efficacy in tabular data processing. This strategic trade-off underscores the pivotal role of dimensionality reduction techniques in optimizing multimodal fusion and bolstering overall predictive capabilities.

Applying GradCAM [14] on our image model enables us to observe how our model is learning its classification process. Figure 7 shows a convolutional layer of an image labeled “Airpush” as the heatmap and GradCAM image. This helps us to deduce that our model is learning in an interpretable manner as it highlights all the necessary characteristics of the Airpush grayscale image. Thus, CNNs have proved to be highly effective for our image dataset. The high accuracy of the CNN model on image data reflects its capability to learn hierarchical features from grayscale malware images, providing a strong foundation for the multimodal fusion process.

In comparing concatenation-based multimodal fusion with elementwise multimodal fusion, our analysis reveals a clear performance advantage for the concatenation based approach. Specifically, the concatenation based multimodal fusion model demonstrates superior accuracy, outperforming its element-wise counterpart with an accuracy of 96.3%. Although the element-wise multimodal fusion model achieves accuracy levels close to those of the concatenation-based model, exhibiting comparable precision and recall values, it notably suffers from a lower F1 score. This discrepancy suggests that while the element-wise approach may effectively identify true positives and negatives, it encounters challenges in accurately identifying positive instances, leading to a reduced F1 score. In contrast,

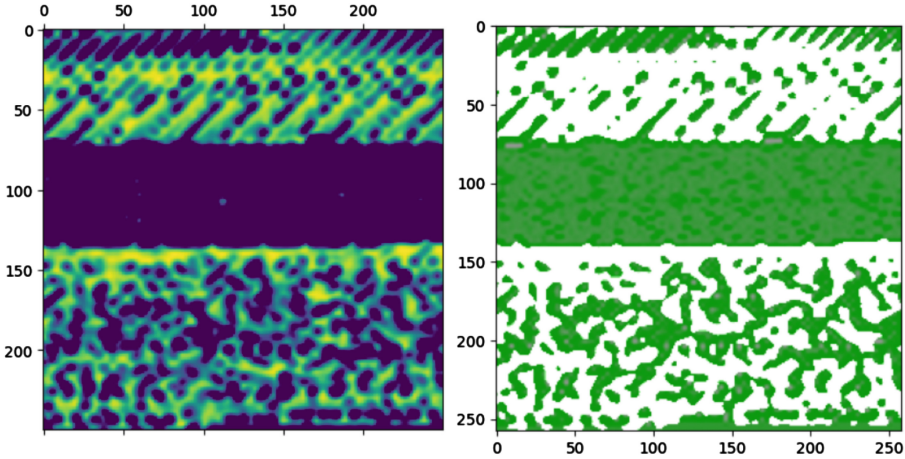


Fig. 7. GradCAM [14] on convolutional layer of Image data model. Left: Heatmap. Right: GradCAM image

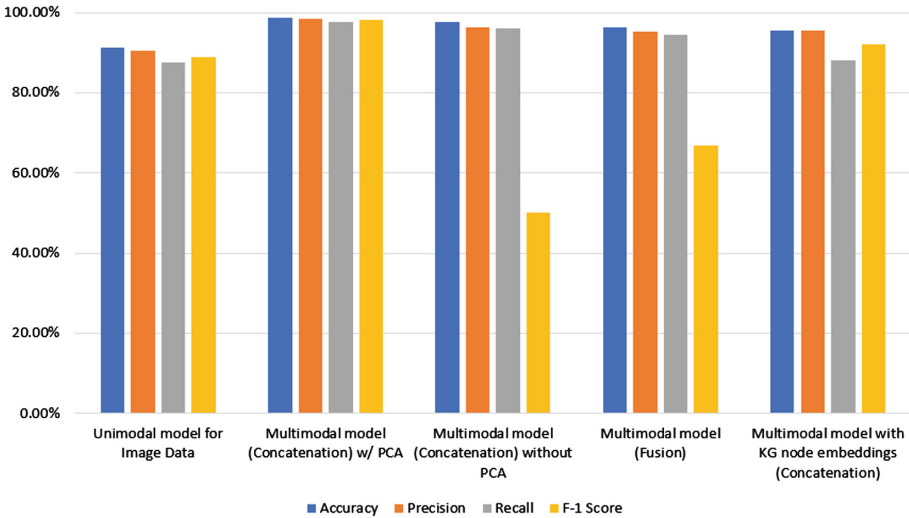


Fig. 8. Visualization of the metrics of different modalities

the concatenation-based multimodal fusion model leverages the comprehensive information integration enabled by concatenation, resulting in a more robust performance across all evaluation metrics with an accuracy of 98.71%, precision, recall, and F-1 score of 98.33%, 97.67%, and 98.1% respectively. Thus, our findings highlight the effectiveness of concatenation-based multimodal fusion for enhancing classification accuracy and overall model efficacy.

**Table 3.** Classification Matrix for Machine Learning classifiers on CIC-AndMal-2020 tabular dataset with PCA. Here Class 0 is Airpush and Class 1 is Dowgin

Classifier	Class	Precision	Recall	F1 Score	Accuracy
XGBoost	Class 0	98%	98%	97%	94%
	Class 1	98%	97%	97%	
Decision Tree	Class 0	91%	92%	92%	88%
	Class 1	81%	79%	80%	
Logistic Regression	Class 0	95%	96%	95%	94%
	Class 1	90%	89%	90%	
KNN	Class 0	93%	97%	95%	93%
	Class 1	91%	84%	88%	
SVM with RBF Kernel	Class 0	95%	97%	96%	95%
	Class 1	92%	92%	92%	
SVM with Linear Kernel	Class 0	95%	95%	95%	94%
	Class 1	89%	89%	89%	
Random Forest	Class 0	91%	98%	95%	92%
	Class 1	95%	79%	86%	

While the incorporation of knowledge graph node embeddings in the multimodal model does enhance its performance compared to unimodal models, it falls short of achieving the superior performance demonstrated by the proposed approach. This discrepancy can be attributed to the inherent nature of the dataset, which primarily consists of numerical features. In such datasets, discerning meaningful relationships between node embeddings derived from knowledge graphs proves challenging. Consequently, despite the enrichment provided by knowledge graph embeddings, the multimodal model's performance remains moderate.

## 5 Conclusion

Our multimodal deep learning approach emerges as a promising avenue for mobile malware classification, harnessing the synergistic capabilities of both image and tabular data. By using neural networks, machine learning classifiers, explainability techniques, knowledge graph embeddings, and meticulous data pre-processing, the deployed models demonstrate acceptable performance, interpretability, and robustness in detecting mobile malware. By embracing a multifaceted approach, researchers can pave the way for the development of more effective and resilient models, thereby fortifying mobile security measures and safeguarding users against evolving cyber threats.

However, there are avenues for further exploration. Given the inherent nature of our dataset, primarily comprising numerical features, future research could

extend beyond this scope by considering alternative datasets. Exploring datasets with diverse characteristics, such as textual or temporal data, could provide valuable insights into the generalizability and adaptability of our approach across different domains. Additionally, conducting comparative studies on varied datasets can offer deeper insights into the performance and scalability of our model, facilitating its broader applicability in real-world scenarios.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Audebert, N., Herold, C., Slimani, K., Vidal, C.: Multimodal deep networks for text and image-based document classification. In: Cellier, P., Driessens, K. (eds.) ECML PKDD 2019. CCIS, vol. 1167, pp. 427–443. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-43823-4\\_35](https://doi.org/10.1007/978-3-030-43823-4_35)
2. Deldar, F., Abadi, M.: Deep learning for zero-day malware detection and classification: a survey. *ACM Comput. Surv.* (2023)
3. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Mach. Learn.* **46**(1), 389–422 (2002). <https://doi.org/10.1023/A:1012487302797>
4. Iadarola, G., Martinelli, F., Mercaldo, F., Santone, A.: Towards an interpretable deep learning model for mobile malware detection and family identification. *Comput. Secur.* **105**, 102198 (2021)
5. Keyes, D.S., Li, B., Kaur, G., Lashkari, A.H., Gagnon, F., Massicotte, F.: Entropylyzer: android malware classification and characterization using entropy analysis of dynamic characteristics. In: 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), pp. 1–12. IEEE (2021)
6. Kwak, C., Jung, P., Lee, S.: A multimodal deep learning model using text, image, and code data for improving issue classification tasks. *Appl. Sci.* **13**(16), 9456 (2023)
7. Li, L., et al.: Static analysis of android apps: a systematic literature review. *Inf. Softw. Technol.* **88**, 67–95 (2017)
8. Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., Liu, H.: A review of android malware detection approaches based on machine learning. *IEEE Access* **8**, 124579–124607 (2020)
9. Lou, S., Cheng, S., Huang, J., Jiang, F.: Tfdroid: android malware detection by topics and sensitive data flows using machine learning techniques. In: 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), pp. 30–36. IEEE (2019)
10. Maćkiewicz, A., Ratajczak, W.: Principal components analysis (PCA). *Comput. Geosci.* **19**(3), 303–342 (1993). [https://doi.org/10.1016/0098-3004\(93\)90090-R](https://doi.org/10.1016/0098-3004(93)90090-R), <https://www.sciencedirect.com/science/article/pii/009830049390090R>
11. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005* (2017)
12. Piplai, A., Ranade, P., Kotal, A., Mittal, S., Narayanan, S.N., Joshi, A.: Using knowledge graphs and reinforcement learning for malware analysis. In: 2020 IEEE International Conference on Big Data (Big Data), pp. 2626–2633. IEEE (2020)

13. Rahali, A., Lashkari, A.H., Kaur, G., Taheri, L., Gagnon, F., Massicotte, F.: Didroid: android malware classification and characterization using deep image learning. In: 2020 The 10th International Conference on Communication and Network Security, pp. 70–82 (2020)
14. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
15. Summaira, J., Li, X., Shoib, A.M., Li, S., Abdul, J.: Recent advances and trends in multimodal deep learning: a review. arXiv preprint [arXiv:2105.11087](https://arxiv.org/abs/2105.11087) (2021)
16. Zhang, W., Zhao, J., Wang, S.: Sustriage: sustainable bug triage with multi-modal ensemble learning. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 441–448 (2021)



# Explainable Fraud Detection with Deep Symbolic Classification

Samantha Visbeek<sup>1</sup>, Erman Acar<sup>2(✉)</sup>, and Floris den Hengst<sup>3(✉)</sup>

<sup>1</sup> RiskQuest, Amsterdam, The Netherlands

<sup>2</sup> ILLC and IvI, University of Amsterdam, Amsterdam, The Netherlands

Erman.Acar@uva.nl

<sup>3</sup> Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

F.den.Hengst@vu.nl

**Abstract.** There is a growing demand for explainable, transparent, and data-driven models within the domain of fraud detection. Decisions made by the fraud detection model need to be explainable in the event of a customer dispute. Additionally, the decision-making process in the model must be transparent to win the trust of regulators, analysts, and business stakeholders. At the same time, fraud detection solutions can benefit from data due to the noisy and dynamic nature of fraud detection and the availability of large historical data sets. Finally, fraud detection is notorious for its class imbalance: there are typically several orders of magnitude more legitimate transactions than fraudulent ones. In this paper, we present Deep Symbolic Classification (DSC), an extension of the Deep Symbolic Regression framework to classification problems. DSC casts classification as a search problem in the space of all analytic functions composed of a vocabulary of variables, constants, and operations and optimizes for an arbitrary evaluation metric directly. The search is guided by a deep neural network trained with reinforcement learning. Because the functions are mathematical expressions that are in closed-form and concise, the model is inherently explainable both at the level of a single classification decision and at the model's decision process level. Furthermore, the class imbalance problem is successfully addressed by optimizing for metrics that are robust to class imbalance such as the F1 score. This eliminates the need for problematic oversampling and undersampling techniques that plague traditional approaches. Finally, the model allows to explicitly balance between the prediction accuracy and the explainability. An evaluation on the PaySim data set demonstrates competitive predictive performance with state-of-the-art models, while surpassing them in terms of explainability. This establishes DSC as a promising model for fraud detection systems.

**Keywords:** Fraud Detection · Classification · Deep Symbolic Regression · Deep Reinforcement Learning

---

E. Acar and F. den Hengst—Equal contribution.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 350–373, 2024.

[https://doi.org/10.1007/978-3-031-63800-8\\_18](https://doi.org/10.1007/978-3-031-63800-8_18)

## 1 Introduction

Deep learning has shown remarkable results in many application domains, including fraud detection [1, 13, 24]. However, a major drawback of these models is their lack of transparency. Their black-box nature makes it difficult to justify a single decision, let alone explain their overall decision-making processes. Understanding these is necessary because (i) frauds need not only be detected, but the opportunity for fraud needs to be mitigated with, e.g., more stringent security measures, and (ii) the nature of fraud detection changes daily: new types of fraud are developed, whereas existing types fall out of favor or become impossible due to novel security measures. Furthermore, according to the European Union’s 2018 General Data Protection Regulation [9], financial institutions are required to justify their decisions to legal authorities and customers. These requirements highlight the need for inherently transparent and explainable models.

Explainable AI has been gaining attention in recent years, with one area of research being Symbolic Regression (SR). SR aims to find analytical (concise, closed-form) expressions that describe functional dependencies in a data set. Since an expression can be understood simply by inspection, SR can be used to create a model that is transparent and explainable. Recently, [23] proposed Deep Symbolic Regression (DSR), an approach to SR based on deep reinforcement learning. In DSR, a recurrent neural network (RNN) is trained with deep reinforcement learning on a task-specific reward function to generate expressions with high predictive power and low complexity. DSR effectively leverages gradient-based deep learning to capture complex relationships in large data sets that are nevertheless easily described with an analytical function.

In this paper, we propose extensions to the DSR framework for the supervised fraud detection problem. The resulting Deep Symbolic Classification (DSC) approach extends DSR with: the addition of a sigmoid layer to the output of the expressions to turn regression into binary classification, the incorporation of a threshold-based decision mechanism, and a reward function based on an accuracy metric for class-imbalanced problems.

Our approach results in a novel framework for fraud detection, characterized by the following strengths:

- (1) robust predictive performance from large-scale, high-dimensional data with the use of deep reinforcement learning,
- (2) analytical expressions that can be transformed into a concise set of rules and with explanations at both the decision- and the model levels,
- (3) intrinsically robust to highly imbalanced data without the need for problematic techniques like over- or undersampling,
- (4) explicitly trading off predictive power and explainability,
- (5) ability to capture non-monotonic, and non-linear relationships without an excessive number of polynomial terms or complex neural network architectures.

We train and evaluate DSC on the PaySim data set [19] which is highly imbalanced with only  $\sim 0.13\%$  being the fraud cases. We compare our approach to de

facto industry standards, including XGBoost [10] and show comparable predictive performance. Additionally, we evaluate the explainability of the expression obtained along two axes. Firstly, we assess whether the expression aligns with domain knowledge with an expert from the field, and find that the expression can be understood successfully. Secondly, we investigate the trade-off between explainability and predictive performance by constructing a Pareto front of performance and complexity of obtained expressions. We find that more complex expressions do not necessarily yield better predictive performance. Finally, in an investigation of the relation between expression complexity and overfitting, we find that the approach does not suffer from overfitting for simple and more complex expressions.

## 2 Related Work

We discuss various related works on explainable models, including (deep) symbolic regression, and other fraud detection methods based on supervised learning. These can be complemented by unsupervised ones e.g., the one-class SVM [30] which we do not detail further since our work is on supervised setting.

### 2.1 Explainable Models

Explainable AI has gained increasing attention in recent years, particularly in fields with high societal stakes such as finance and medicine [16]. While models focusing solely on predictive performance keep surpassing the state of the art, their black-box nature prevents adoption. This is especially seen in application domains where accountability is a prerequisite and decision-making based on black-box models can have harmful consequences [31, 33]. To address this issue, the field of explainable AI has emerged [18, 20, 26, 32]. This field focuses on approaches that involve approximating a secondary model to explain the predictions made by the original black-box model. However, these approaches may be insufficiently reliable, robust, or hard to interpret themselves, which has motivated the study of inherently explainable methods [2, 8, 15, 25].

Furthermore, the implementation of the European Union’s General Data Protection Regulation (GDPR) in 2018 has given citizens the ‘right to explanation’ of automated decision-making models that can significantly affect them [9]. Banks often make these decisions as part of their fraud prevention efforts. One example is the use of FDSs to block suspicious payment transactions in order to reduce losses and ensure the satisfaction of law-abiding customers. According to [22], bank institutions must justify their actions to customers, anti-money laundering authorities, and legal organizations. Since fraud detection algorithms have legal, operational, strategic and ethical constraints, banks must balance explainability with predictive performance [22]. Our approach allows for explicitly selecting a solution that is Pareto-optimal w.r.t. explainability and performance.



## 2.2 Symbolic Regression

The field of study known as Symbolic Regression (SR) aims to obtain analytical expressions that describe functional dependencies of a data set [7]. Formally, given a set of characteristics  $X$  and target values  $\mathbf{y}$ , with  $X_i \in \mathbb{R}^n$  and  $y_i \in \mathbb{R}$ , SR aims to find a function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  that best describes the data set, where  $f$  is a closed-form analytical expression such that  $f(X) = \mathbf{y} + \epsilon$ . Essentially, the SR problem is a discrete combinatorial search for the optimal function  $f^*$  that minimizes the distance metric  $D(f(X), \mathbf{y})$ :

$$f^* = \underset{f}{\operatorname{argmin}} D(f(X), \mathbf{y}). \tag{1}$$

The function  $f$  is an analytical expression that can be interpreted by inspection. As a result, SR is frequently used to produce models that are transparent and explainable [16].

Conventionally, SR has emerged within the field of Genetic Programming (GP) [16]. First introduced by [14], the combinatorial search problem is addressed with an algorithm inspired by the Darwinian principle of natural selection and genetic recombination. This process begins with the evaluation of a population of candidate solutions. Each candidate consists of a syntax tree where the leaf nodes represent features, and the internal nodes represent operators. A syntax tree represents an analytical expression and is evaluated on the training data set. In this evaluation, a predictive performance metric such as accuracy corresponds to the notion of fitness in Darwinian terminology. The fittest candidates are selected for reproduction, where their subtrees are recombined, with the goal of creating even more fit candidates. Additionally, each node has a probability of randomly mutating, i.e. changing the node’s operator. The process can be seen as a version of combinatorial discrete search for an accurate expression by training on a sufficient yet tractable number of generations.

Quite recently, [27] showed that GP-based SR often outperforms the top machine learning methods in classification tasks, highlighting its potential for achieving high performance. However, there are also concerns regarding the use of GP for SR, one of which is its sensitivity to hyperparameter configurations, which can result in suboptimal performance. Additionally, GP has been found to be computationally demanding and may not scale to large high-dimensional data sets with complex relationships [23].

## 2.3 Deep Symbolic Regression

To address the limitations of SR with GP, a recent work by [23] has proposed a gradient-based approach to SR based on RNN and reinforcement learning, known as Deep Symbolic Regression. During training, an RNN produces analytical expressions that are then evaluated on how well they describe the data set, a measure called “fitness”. This fitness is linked to a reward that is used to train the RNN through a risk-seeking policy gradient algorithm. The RNN adjusts the probabilities of sampling an expression according to its corresponding reward.

This results in expressions that describe the data set relatively well. The authors demonstrate that DSR outperforms the included baselines on a set of benchmark problems, including Eureqa, which is considered the gold standard for symbolic regression [23].

Building on the foundation laid by DSR, [21] have proposed a hybrid approach that combines GP and gradient-based methods. In this enhanced framework, the RNN initially generates a set of expressions (or candidate solutions). Subsequently, GP is employed as an inner optimization loop to facilitate selection, recombination, and mutation on these candidates. This enables the exploration of a more diverse solution space and enables the algorithm to escape local optima more effectively. After this GP step, the fitness of the resulting expressions is reassessed, and the RNN is then used again to generate a fresh batch of candidate solutions. This process continues iteratively, with the GP component serving as the inner optimization loop, while the neural-guided gradient-based approach operates as the outer loop.

The hybrid approach in this study combines the strengths of both methods to enhance their respective performance. First, the integration of the RNN trained with reinforcement learning allows for improved restarts in the GP process, overcoming the limitations of random restarts that are normally associated with GP. Second, the inclusion of GP enables a more diverse exploration of the solution space, reducing the risk of being trapped in local optima. The authors demonstrate the superior performance of the hybrid DSR approach compared to vanilla DSR in various benchmark problems [21].

In this study, we present a novel framework called Deep Symbolic Classification, which is a modified version of hybrid DSR, tailored specifically for the binary classification task of fraud detection. Our proposed approach incorporates a sigmoid layer into the prediction mechanism, enabling it to produce a probabilistic output suitable for classification. Additionally, we utilize the F1 score as the reward function to address the prevalent issue of high class imbalance often encountered in fraud detection scenarios.

## 2.4 Uninterpretable Fraud Detection

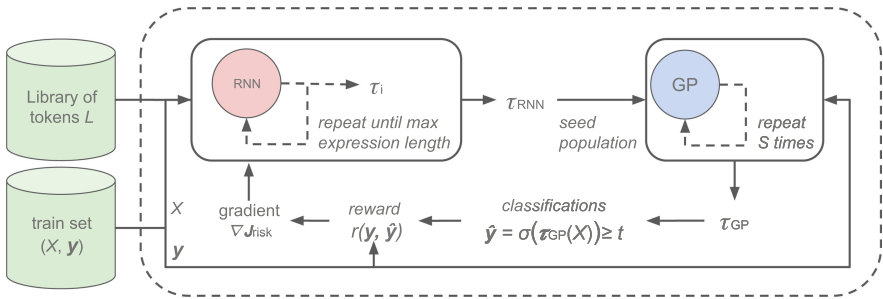
In 2022, [10] proposed an XGBoost-based framework that was empirically shown to achieve state of the art (SOTA) performance on the PaySim data set. XGBoost, short for Extreme Gradient Boosting, is a decision tree ensemble method that combines multiple ML models to produce superior performance relative to that of a singular model [5]. It is based on the principle of sequentially adding weaker models to correct for the errors made by previous models, utilizing gradient descent to optimize the model's performance. However, one of its primary limitations is its lack of explainability, which makes it unsuitable as an FDS in practical scenarios.

In the same study, several supervised models with varying accuracy and levels of explainability were used as baselines for comparative analysis. These include (in decreasing order of explainability [22])  $k$ -Nearest Neighbors ( $k$ -NN), Random Forest (RF) and Support Vector Machines (SVM). Their findings indicated that  $k$ -NN and SVM are ineffective for fraud detection due to their inability to address the class imbalance. Conversely, RF performed exceptionally well on the data set, yet it is still considered a black-box model because of the high number of deep decision trees generated within it and due to the lack of controlling the rule complexity. Although some strategies have been proposed to offer the required explainability for understanding the RF model [3], it is advisable to employ models that are inherently explainable and rely on compact closed-form rules instead, as previously detailed in Sect. 2.1.

In this work, we conduct a comparative analysis between DSC and the aforementioned models, assessing their respective predictive performances. This enables us to illustrate how DSC measures up against models with varying degrees of explainability, as well as its performance against the state of the art on the test set. Note that the pre-processing of our data set differs from the procedure adopted in the work of [10]. Therefore, we reimplement these models according to the hyperparameters specified in the original paper to maintain consistency in our analysis.

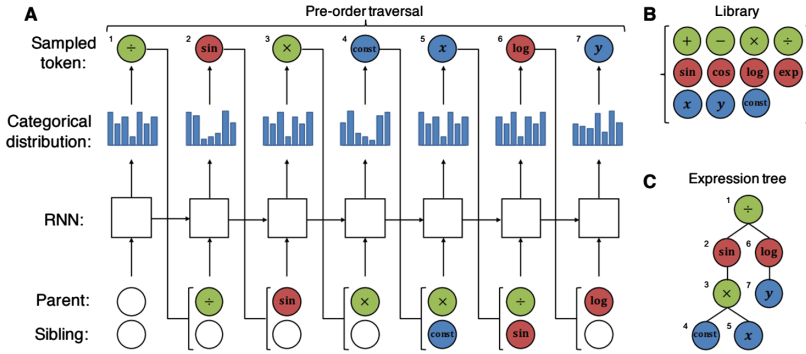
### 3 Method

#### 3.1 Model Implementation



**Fig. 1.** Train loop for deep symbolic classification. An RNN generates an expression by sampling tokens from a predefined library of tokens. The resulting expression is used to seed the population of a genetic programming process for further optimizing the expression. After optimization, classifications are generated using a sigmoid  $\sigma$  and a threshold  $t$ . The classifications are scored with a reward function, e.g. F1 score, which is used to train the RNN in a risk-based gradient estimate.

We implement hybrid DSR as described in [21], and adjust it to make it suitable for classification problems. In this subsection, we provide a comprehensive



**Fig. 2.** An example of the sampling process of (Hybrid) Deep Symbolic Regression (taken from [23]). **(A)** Elements are sampled from a categorical distribution emitted by the RNN on the library  $L$  of elements, which is given in **(B)**. The parent and sibling nodes of the next element are used as the next input to the RNN. The sampling process ends when all branches reach the leaf nodes. The resulting list  $[\div, \sin, \times, \text{constant}, x, \log, y]$  is the preorder traversal of the syntax tree that represents the equation  $\sin(cx)/\log(y)$ . **(C)** The syntax tree that can be reconstructed from the preorder traversal list from **(A)**.

description of our methodology for generating analytical expressions using a RNN, converting these expressions into classification models, and training the RNN using reinforcement learning techniques.<sup>1</sup> Fig. 1 details the steps described below.

**Generating Expressions.** Following [21], here we explain how expressions are generated with (deep) symbolic regression, reformulating previous works to suit the purposes of our work. Each expression is represented by a binary syntax tree, following the approach proposed by [14]. Each node in this tree is labeled with a token from a library of tokens. The library contains tokens that represent input features, constants, and mathematical operators. All leaf nodes of the syntax tree are labeled with tokens representing features or constants, and internal nodes represent mathematical operators. These operators can be unary (logarithm, sign, square root, etc.) or binary (summation, difference, multiplication, etc.). Within the algorithm, a syntax tree is represented as a list corresponding to a pre-order traversal of the tree, see Fig. 2 (taken from [23]). Since the arity of each mathematical operator is known upfront, a list of tokens represents a single, unique tree. The list representation brings the benefit of compatibility with existing neural network architectures that accept sequential input, including RNNs, LSTM-based models, and transformers [17].

The lists are generated from left to right, where each element is sampled from a probability distribution over all available features and mathematical operators.

<sup>1</sup> Source code available at [https://github.com/samanthav24/DSC\\_Fraud\\_Detection](https://github.com/samanthav24/DSC_Fraud_Detection).

The probability distribution for sampling element  $i$  is conditioned on the previously sampled elements  $i-1, i-2, \dots, 0$ . This conditional dependence is generated by an RNN, the outputs of which are passed through a softmax layer to obtain probabilities for each of the operators and features. Rather than using the current list as input, the RNN is only given the parent and sibling nodes of the previously sampled element as input. This is because the list does not capture the hierarchical structure of its syntax tree, as it was formed by preorder traversal.

In order to generate plausible expressions that make sense in the context of describing the data set, the syntax tree is subject to certain constraints: (1) *the length of the sampled expression* is bound by predefined minimum and maximum values; (2) it is enforced that *each pair of leaf nodes that descend from the same parent node*, should represent at least one feature: this prevents forming expressions of constants; (3) the tree has some constraints to ensure expressions that make sense mathematically: *unary operators* cannot have children that are the inverse of the operator, and *trigonometric operators* cannot have children that are trigonometric operators themselves. The process of generating expressions with RNN is visualized in Fig. 2.

**Inner Optimization Loop.** Entering the inner GP loop of our process, the expressions  $\tau_{RNN}$  generated by the RNN serve as the initial population  $\tau_{GP}^{(0)}$  for the GP algorithm. With each iteration  $i$ , a new population of expressions  $\tau_{GP}^{(i+1)}$  is systematically constructed through processes of selection, recombination, and mutation. This iterative procedure continues until the specified number of generations,  $S$ , is reached. The highest-performing expressions of the final population  $\tau_{GP}^{(S)}$ , are selected and subsequently used for gradient update.

This hybrid methodology introduced by [21] effectively combines the advantages of an inner GP-based optimization loop with those of an outer gradient-based optimization loop. The internal loop is similar to the standard GP with random restarts, with the distinctive addition of the RNN that offers progressively better starting populations ( $\tau_{RNN} = \tau_{GP}^{(0)}$ ) for each successive iteration of the outer loop. In the context of the outer loop, the GP element ensures a more diverse range of expression populations. This diversity helps avoid being confined to local optima, thus facilitating a more efficient learning process.

**Evaluating Expression.** Each expression  $f$  is passed through a sigmoid function  $\sigma$  to produce probabilities that are suitable for use in this binary classification problem. This allows the expression to be used to predict the likelihood that a transaction is fraudulent (1) or legitimate (0). The class prediction  $\hat{y}$  of a transaction with corresponding features  $\mathbf{x}$  is determined according to the following:

$$\hat{y} = \begin{cases} 1 & \text{if } \sigma(f(\mathbf{x})) \geq t \\ 0 & \text{if } \sigma(f(\mathbf{x})) < t \end{cases} \quad (2)$$

where  $t$  is a given threshold. A reward is assigned to the expression, which corresponds to its performance on the data set. It is conventional in classification

problems to minimize the cross-entropy loss  $CE$  to increase the performance of the model. Therefore, we define a reward function  $r_{CE}$ , which is the normalized inverse cross-entropy loss with respect to the ground truth classes  $\mathbf{y}$ . Specifically,

$$r_{CE} = \frac{1}{1 + CE(\mathbf{y}, \hat{\mathbf{y}})} \quad (3)$$

where we add 1 to the denominator to normalize the reward. However, a major problem in the domain of fraud detection is the significant class imbalance, which often leads to low precision (i.e., too many legitimate transactions are incorrectly classified as fraudulent). To address this, a different reward function  $r_{F1}$  is defined. This function is based on the binary F1 score, which directly optimizes the model's performance with respect to precision and recall:

$$r_{F1} = \frac{2pr}{p + r} \quad (4)$$

where  $p$  and  $r$  are respectively the precision and recall score of the expression on the data set. We conduct a comparison between the reward functions  $r_{CE}$  and  $r_{F1}$  based on the performance scores of the expressions generated using these functions, for different threshold values  $t \in [0.5, 0.9]$ . In standard logistic regression, the threshold value is typically set to 0.5. However, due to the issue of low precision arising from the class imbalance, we increase the threshold values to improve precision.

The RNN adjusts the probability distribution for sampling elements with this reward function (a more detailed description of this process is given in Sect. 3.1). In the context of reinforcement learning, the constituent elements of the environment, actions, episode, policy, and reward are represented by the parent and sibling nodes, the sampling of elements, the generation of an expression, the probability distribution, and the reward function, respectively. The algorithm terminates after a given number of iterations.

**RNN Training.** Since the reward function is based on the predictive power of the generated equation, rather than the parameters  $\theta$  of the RNN, it is not differentiable with respect to  $\theta$ . Therefore, reinforcement learning is used to train the RNN to generate better expressions. Here, actions correspond to sampled tokens, observations correspond to the current state of the expression tree (i.e., parent and sibling input), and sequences that form expressions correspond to episodes.

A naive approach to optimize the policy  $p(\tau|\theta)$  (which is represented by the distribution of samples  $\tau$ ) would be to use the standard policy gradient objective that aims to maximize the expected value of the reward. It is important to note that by maximizing an expectation, the focus is on optimizing the average performance of the generated expressions. However, in the context of fraud detection, the ultimate objective is to achieve the best possible performance of a single expression that is found during training, as it will be used as the final

model. Therefore, the standard policy gradient objective may not be suitable for this purpose.

Instead, a risk seeking policy gradient objective  $J_{risk}(\boldsymbol{\theta}; \varepsilon)$ , is used to maximize the performance of the highest fraction of samples  $\varepsilon$ , at the expense of sacrificing the performance of the other generated expressions. The risk-seeking policy gradient objective [23] is defined as

$$J_{risk}(\boldsymbol{\theta}; \varepsilon) \equiv \mathbb{E}_{\tau \sim p(\tau|\boldsymbol{\theta})}[R(\tau)|R(\tau) \geq R_\varepsilon(\boldsymbol{\theta})] \quad (5)$$

which aims to increase the reward  $R_\varepsilon$  of the top  $\varepsilon$  fraction of samples from the distribution, while disregarding the samples that fall below this threshold. The gradient of  $J_{risk}(\boldsymbol{\theta}; \varepsilon)$  is then given by

$$\nabla J_{risk}(\boldsymbol{\theta}; \varepsilon) = \mathbb{E}_{\tau \sim p(\tau|\boldsymbol{\theta})}[(R(\tau) - R_\varepsilon(\boldsymbol{\theta})) \cdot \nabla_{\boldsymbol{\theta}} \log p(\tau|\boldsymbol{\theta}) | R(\tau) \geq R_\varepsilon(\boldsymbol{\theta})] \quad (6)$$

To compute this, we can use the standard REINFORCE Monte Carlo estimate [35] with two adjustments. First, instead of using the expected return of all samples as a baseline, we substituted it with  $R_\varepsilon$ . Second, we only include the top  $\varepsilon$  fraction of samples from each batch in the gradient computation. The resulting Monte Carlo estimate can be expressed as

$$\nabla J_{risk}(\boldsymbol{\theta}; \varepsilon) \approx \frac{1}{\varepsilon N} \sum_{i=1}^N [R(\tau^{(i)}) - \tilde{R}_\varepsilon(\boldsymbol{\theta})] \cdot \mathbf{1}_{R(\tau^{(i)}) \geq \tilde{R}_\varepsilon(\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \log p(\tau^{(i)}|\boldsymbol{\theta}) \quad (7)$$

where  $\tilde{R}_\varepsilon$  is the empirical  $(1 - \varepsilon)$ -quantile of the batch of rewards, and  $\mathbf{1}_s$  takes the value 1 if the statement  $s$  is true, and 0 otherwise. In this implementation, the value of  $\varepsilon$  is set to 0.05, which is consistent with the approach taken in [23].

### 3.2 Data

**The PaySim Data Set.** We use the popular PaySim data set [19]. PaySim is a data set of simulated transactions based on proprietary real transactions [19]. Obtaining a real data set of payment transactions can be difficult due to privacy concerns. To address this issue, PaySim was developed to provide researchers with a simulated data set that exhibits statistical properties similar to a real payment transaction data set, while preserving the confidentiality of the underlying customer data. The data set contains  $\sim 6.3$  million transactions over a period of a month and with a fraudulent transaction rate of  $\sim 0.13\%$ . Columns represent various attributes associated with transactions:

- step - unit of time; one step corresponds to one hour of time,
- type - a categorical feature with values: cash-in, cash-out, debit, payment, or transfer,
- amount - amount of the transaction in local currency,
- nameOrig - name of the customer,
- oldbalanceOrg - customer's balance before the transaction,

- *newbalanceOrig* - customer’s balance after the transaction,
- *nameDest* - name of the recipient,
- *oldbalanceDest* - recipient’s balance before the transaction,
- *newbalanceDest* - recipient’s balance after the transaction,
- *isFlaggedFraud* - an indicator of whether the transaction has been flagged as fraudulent in the simulation,
- *isFraud* - an indicator of the transaction being legitimate or fraudulent,

where the column *isFraud* represents the target variable, while the remaining columns are used as features in the DSC model.

**Generating Additional Features.** Some of these features require the inclusion of pre-processing in an analytical expression targeted in this work. The *type* variable was represented with one-hot encoding. All other categorical features (*nameOrig* and *nameDest*) were discarded to ensure the explainability of the model. Because information about individual customers and recipients can be essential for identifying fraud [4,34], aggregation features were added to model the behavior of account holders. These features include the *mean* and *maximum* transaction amounts of the last 3 and 7 transactions of the recipient account. Further analysis of the dataset shows that only 0.15% of the accounts participated in more than one transaction, compared to 83% of the recipient accounts. Therefore, the aggregation of the previous 3 and 7 transaction amounts is limited to the recipient account.

The PaySim data set contains many transactions with nonzero transaction amounts and before and after balances of zero. These transactions model accounts at counterparty banks, whose balances are not known and were imputed with zero. To mitigate this, the data set is enhanced with two features, *externalOrig* and *externalDest*, which indicate whether both balances before and after the transaction are zero, respectively, and thus are considered to belong to an external account. Furthermore, zero balances are imputed so that *oldbalanceOrig* is set equal to *amount* if the customer’s account is external and *newbalanceDest* is set equal to *amount* if the recipient’s account is external. This method ensures that the balances are proportional to the transaction amount. The indicator features *externalOrig* and *externalDest* identify such instances and differentiate between true zeros and zeros due to missing values. Furthermore, the inclusion of *externalOrig* and *externalDest* ensures that a possible relationship between fraudulent transactions and the involvement of external banks is properly considered.

The imputation of balances also mitigates a form of data leakage. A known limitation of the PaySim data set is that a model that predicts fraud if the transaction amount is equal to *oldbalanceOrig* achieves exceptionally high accuracy. This has raised concerns that the data set might have been generated according to this rule. However, the authors refute this possibility and assert that transactions recognized as fraud (as determined by the bank of the original data set from which these transactions are simulated) are likely to be cancelled<sup>2</sup>, resulting

<sup>2</sup> See <https://www.kaggle.com/datasets/ealaxi/paysim1/discussion/99799>.



in *oldbalanceOrig* being set to zero. Therefore, the aforementioned imputation of *oldbalanceOrig* helps circumvent this issue of data leakage.

A second form of data leakage was also mitigated in our preprocessing. According to the description of the data set, the *isFlaggedFraud* feature should be True if the transaction amount exceeds 200,000. However, when analyzing the data, it becomes apparent that this condition is not met and the actual meaning of this variable remains unclear. Nevertheless, it is worth noting that almost all (99.87%) of the transactions where *isFlaggedFraud* is True, are indeed fraudulent. Due to the ambiguity surrounding the interpretation of *isFlaggedFraud*, this feature is ultimately discarded.

**Final Preprocessing Steps.** Some of the baseline models we use for comparative analysis require balanced training data. To accommodate this requirement, an additional balanced training set is generated by randomly undersampling the training data. Details on preprocessing and all features used can be found in Appendix B.

### 3.3 Evaluation

We conduct a comparative analysis to assess the predictive performance of DSC in relation to the SOTA XGBoost-based method and the baseline models and hyperparameters proposed by [10]. In order to assess the trade-off between expression complexity and predictive performance, we create the set of generated expressions where no other generated expression is superior in both complexity and performance [28]. Such a set is typically known as the Pareto front. We define the complexity  $C$  of an expression  $f$  of length  $T$  with sampled tokens  $\tau_i$  as

$$C(f) = \sum_{i=0}^T c(\tau_i) \quad (8)$$

where  $c(\tau_i)$  is the complexity of a token  $\tau_i$ . The complexity of different types of tokens is taken from [23] and reproduced in Table 1.

The optimal expression can then be determined via the *elbow method*, i.e. by selecting the point at which adding more complexity to the expression does not result in a sufficient increase in the F1 score. This prevents overfitting and ensures that the expressions are not overly complex, thereby preserving explainability in its interpretation and aligning with *fruitfulness*, *exactness* and *similarity* in the framework by Sovrano and Vitali [29].

## 4 Results and Discussion

Table 2 lists the performance in the test set averaged over 5 runs: the baseline classification models with and without Random Undersampling (RUS) (see Sect. 2) is compared to the best DSC configuration, i.e. with  $r_{F1}$  and a threshold value of 0.8 and the best expression obtained by DSC (Eq. 11).

**Table 1.** Complexity of tokens

token $\tau$	complexity $c$
+, -, ×, feature, constant	1
÷, square	2
sin, cos	3
exp, log, square-root	4

**Table 2.** Average F1 scores on the test set. Std between parentheses if  $\leq 0.00$ , column-wise best in bold.

method	accuracy	precision	recall	F1 score
RF + RUS	0.93	0.02	0.93	0.03
XGBoost + RUS	0.95	0.02	<b>0.94</b>	0.05
$k$ -NN + RUS	0.94	0.02	0.83	0.03
SVM + RUS	0.95	0.02	0.70	0.03
RF	<b>0.99</b>	<b>0.99</b>	0.67	0.81
XGBoost	<b>0.99</b>	0.98	0.70	<b>0.82</b>
DSC (average)	<b>0.99</b>	0.95 (.01)	0.67	0.78
DSC (best expression)	<b>0.99</b>	0.95	0.67	0.78

Table 2 indicates that undersampling (RUS) does not improve results. While models trained with RUS demonstrate high recall rates, their precision values are notably low, leading to low F1 scores. This observation suggests that an excessive amount of information is lost and that models overfit to the small number of examples in the train set when using undersampling. We note that  $k$ -NN and SVM demonstrate effective training only when applying RUS to the train set. Otherwise, the training time for these models took more than 50 h and was aborted, highlighting the complexities and resource requirements associated with highly imbalanced data sets.

In terms of F1 score, DSC demonstrates comparable performance to RF and the SOTA XGBoost model without RUS. The relatively small drop in performance compared to these baselines stems from a drop in precision and not accuracy. The precision for DSC can be considered acceptable and signifies that a relatively low number of legitimate transactions is incorrectly classified as fraudulent. The DSC model provides inherent explainability at only a limited drop in predictive performance, making it an attractive choice for the fraud detection problem.

### 4.1 Derivation of the Decision Rule

The expression that resulted in the highest performance was:

$$f = \sqrt{\text{externalDest} + \text{type\_cash-out}} \cdot (\text{amount} - \text{maxDest7} + \text{type\_transfer}), \tag{9}$$

where we have three Boolean features that either have value 0 or 1: *externalDest*, *type\_cash-out* and *type\_transfer*. The other features *amount* and *maxDest7* are numerical and positive. The decision rule is defined as:

$$\hat{y} = 1(\text{fraud}), \text{ if } \sigma(f) > 0.7,$$

as this expression was found by training DSC on a threshold  $t = 0.7$ . Rewriting the sigmoid  $\sigma(f) = (1 + e^{-f})^{-1}$  gives us:

$$\hat{y} = 1(\text{fraud}), \text{ if } f > 0.85.$$

So for each transaction,  $f$  is calculated with the feature values of that transaction, and if  $f > 0.85$ , we classify that transaction as fraudulent. It should be noted that

$$\text{amount} - \text{maxDest7} \leq 0, \tag{10}$$

since *maxDest7* is the maximum of the last 7 transaction amounts (**including** the current amount) associated with a particular recipient.

Furthermore, we know that if *type\_transfer* = 0, then *type\_cash-out* = 1, and vice versa, since the feature *type* was one-hot encoded. There are essentially two scenarios:

**1. *type\_transfer* = 0 and *type\_cash-out* = 1.** For explainability, let us divide expression 9 in two parts, such that  $f = A \cdot B$ , where

$$A = \sqrt{\text{externalDest} + \text{type\_cash-out}}$$

$$B = (\text{amount} - \text{maxDest7} + \text{type\_transfer}).$$

Given the inequality 10, we know that  $B$  must be smaller than or equal to 0. Since *externalDest* is a Boolean, which makes  $A$  positive, it follows that  $f \leq 0$ . Therefore, in the scenario where the transaction is of type *cashout*, the model will never classify the transaction as fraudulent.

**2. *type\_transfer* = 1 and *type\_cash-out* = 0.** For a transaction to be classified as fraudulent within this scenario, it is imperative that the value of *externalDest* is equal to 1. Otherwise,  $A$  would evaluate to 0, resulting in the overall value of  $f$  being 0 due to multiplication with 0.

Now, let us assume that *externalDest* = 1, this would reduce Eq. 9 to:  $f = \text{amount} - \text{maxDest7} + 1$ . Given that a transaction is considered fraudulent only if  $f$  is greater than 0.85, it follows that  $\text{amount} - \text{maxDest7}$  must be greater than -0.15.

We can now summarize our findings according to the following rules:

**classify a transaction as fraudulent, if**

- type = transfer, and
- externalDest = True, and
- amount - maxDest\_7 > -0.15

**classify a transaction as legitimate, otherwise**

## 4.2 Explainability

The best average performance was obtained with  $r_{F1}$  and threshold  $t = 0.8$  based on grid-based hyperparameter optimization, see 3. However, the best individual run was trained at  $t = 0.7$ . This expression has comparable predictive performance and lower complexity. Figure 3 shows the F1 scores of the Pareto front of the best run. We refer to the expression  $f$  with complexity level  $x$ , by  $f_{C=x}$ . The figure indicates that the best expression, based on the F1 score, is either  $f_{C=9}$  or  $f_{C=13}$ . When analyzing the overfitting, one might be inclined to favor  $f_{C=9}$  over  $f_{C=13}$  as the performance is similar at lower complexity. However, when evaluating the expressions on the test set, it becomes apparent that  $f_{C=9}$  yields a score of 0.76, while  $f_{C=13}$  yields a score of 0.78. Given that  $f_{C=13}$  produces a higher F1 score on the test set, this suggests that there is no overfitting for this expression.

The key consideration now is whether the observed increase in performance justifies the corresponding increase in complexity, potentially affecting the explainability of the model. The expression that demonstrated the highest performance is given by:

$$f_{C=13} = \sqrt{\text{externalDest} + \text{type.cash-out}} \cdot (\text{amount} - \text{maxDest7} + \text{type.transfer}), \quad (11)$$

where *maxDest7* denotes the maximum amount among the last 7 transactions (including the current amount), associated with a particular recipient. This expression can be readily transformed into a straightforward decision rule suitable for deployment as a detection model, see Sect. 4.1.

Compared to the best expression  $f_{C=13}$  with  $f_{C=9}$ , the key difference is the absence of the square root operator and the substitution of *maxDest7* with *maxDest3*. However, the decision rule derived from the best expression eliminates the square-root operator, making both expressions equally explainable. The only remaining disparity lies in the utilization of either the *maxDest7* or *maxDest3* feature. Therefore, we consider  $f_{C=13}$  as the optimal expression.

The absence of weighted features and the lack of periodic relationships in this expression are somewhat unexpected. One plausible explanation for this finding is that the PaySim data set covers a single month, while fraudulent behavior in general exhibits seasonality over a longer period of time [6, 11]. Hence, the complex periodicity of real-world fraud may not be present in the data set. Furthermore, we observe that the features *type.cash-out* and *type.transfer* are present

in the optimal expression. Exploratory Data Analysis confirms that all fraudulent transactions in the data set indeed fall under these two types. However, when examining the subsequent decision model derived from the expression, it becomes apparent that the model does not detect fraudulent transactions of the type “cash-out”. Although the model acknowledges the significance of this type, it is likely that further training is necessary to effectively capture this specific relationship.

### 4.3 Expert Interpretation

We participated in a discussion with a senior expert in fraud detection employed at a large international bank based in the Netherlands. Our discussion focused on whether our findings align with expert knowledge and the potential applicability of our approach within the bank the expert is currently employed, considering both its performance and explainability. The expression’s simplicity and ease of interpretation make it more manageable than the complex set of rules and large-scale random forests that are typically in place. Moreover, the selected features and their relations within the expression are logically coherent. For example, the inclusion of the feature “*type=transfer*” aligns with criminal behavior. Transfers are popular for executing fraud, in contrast to other types of transactions such as payments. Similarly, the feature “*externalDest = True*” is informative. Specifically, in the event that a transaction is classified as fraudulent by an FDS, the bank may need to retrieve funds. The process of retrieving funds becomes more challenging if the transaction involves an external bank, compared to internal transfers. Fraudsters are well aware of this distinction and can exploit vulnerabilities in the system by diverting money to external institutions.

Furthermore, the requirement that the transaction amount must exceed the highest value among the previous six transactions, or differ by no more than 0.15, exemplifies an adaptation by criminals to evade detection. This adaptive behavior arises from the fact that earlier detection models successfully captured and flagged transactions that adhered to this particular behavior.<sup>3</sup> In response, fraudsters devised a new method known as “smurfing”, in which multiple transactions with small amounts are used to avoid detection by the system<sup>4</sup>.

Finally, in a hypothetical scenario where DSC demonstrates comparable performance on the expert’s bank’s internal data set, it would be regarded as a valuable addition to the FDS. The hypothetically adequate performance of DSC and its simplicity justify its consideration for use as a component in an FDS. One could also imagine that DSC could play a role in devising mitigations for new types of fraud.

---

<sup>3</sup> It is important to acknowledge that the insights are derived from the PaySim data set do not necessarily reflect current fraudulent behavior.

<sup>4</sup> <https://www.abnamro.com/nl/nieuws/meer-over-financiele-criminaliteit>.

**Table 3.** Mean (std) scores of DSC with different reward functions and thresholds over 5 runs, column-wise best in bold.

reward	$t$	accuracy	precision	recall	F1 score
$r_{CE}$	0.5	<b>0.99</b> (.00)	0.98 (.01)	0.52 (.05)	0.68 (.04)
	0.6	<b>0.99</b> (.00)	<b>0.98</b> (.00)	0.50 (.00)	0.66 (.00)
	0.7	<b>0.99</b> (.00)	0.98 (.01)	0.53 (.07)	0.69 (.05)
	0.8	<b>0.99</b> (.00)	0.98 (.01)	0.55 (.07)	0.70 (.05)
	0.9	<b>0.99</b> (.00)	0.95 (.03)	0.59 (.08)	0.72 (.05)
$r_{F1}$	0.5	<b>0.99</b> (.00)	<b>0.98</b> (.00)	0.50 (.00)	0.66 (.00)
	0.6	<b>0.99</b> (.00)	<b>0.98</b> (.00)	0.50 (.00)	0.66 (.00)
	0.7	<b>0.99</b> (.00)	0.97 (.01)	0.56 (.05)	0.71 (.03)
	0.8	<b>0.99</b> (.00)	0.95 (.01)	<b>0.67</b> (.00)	<b>0.78</b> (.00)
	0.9	<b>0.99</b> (.00)	0.94 (.03)	0.66 (.01)	0.78 (.01)

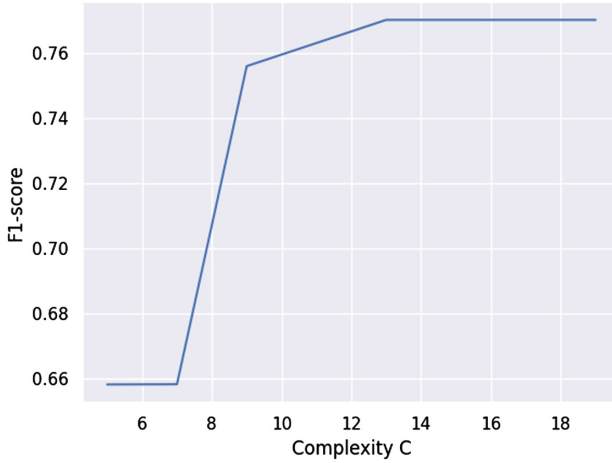
#### 4.4 Impact of Hyperparameters

Table 3 lists the predictive performance of DSC with different reward functions  $r_{CE}$  and  $r_{F1}$ , for various thresholds, averaged over 5 runs. It is important to note that the models were trained on the imbalanced data set and used the same threshold for both training and testing.

When using  $r_{CE}$  as the reward function, there appears to be a slight increase in the F1 score for higher thresholds. This increase is mostly explained by higher recall. However, in general, the recall score is relatively low: only a limited number of fraudulent transactions are detected. In contrast, the positive relationship between the threshold and the F1 score becomes more pronounced for  $r_{F1}$ . Although precision slightly decreases, reward increases significantly, leading to an increase in the F1 score. This trend continues until a threshold of  $t = 0.8$ . Using a threshold of 0.8 yields a recall rate of 0.67, i.e. two thirds of the fraudulent transactions are detected.

The difference in performance when using  $r_{CE}$  or  $r_{F1}$  can be explained as follows: the fraudulent minority class carries less weight in the calculation of the normalized inverse cross-entropy loss, resulting in minimal improvements. On the other hand, by directly optimizing the F1 score, the model ensures that the minority class is not neglected, as both precision and recall have equal importance in the calculation of the reward.

The training and testing phases use the same threshold and one might therefore assume that the threshold should not have a significant impact as feature weights can be adjusted accordingly. However, the optimization of constants includes an inner optimization loop. This loop forms a computational bottleneck. This is mitigated by faster convergence in the case of higher decision thresholds. We believe that longer run times may have resulted in comparable scores for lower decision thresholds. A higher threshold thus serves as a practical approach to reducing computational resources without sacrificing predictive



**Fig. 3.** Pareto front of predictive performance and complexity by the best DSC run ( $t = 0.7, r_{F1}$ ).

performance or model simplicity. Furthermore, we note that recall increases with higher thresholds, while the precision remains stable or even decreases for  $r_{F1}$ . A high decision threshold is a common strategy to favor precision when traditional machine learning models are used on imbalanced data. However, in this particular case, the class imbalance is substantial, with only 0.13% of the transactions being fraudulent. As a result, the model may exhibit overconfidence in the legitimate class, causing the sigmoid function to output probabilities that are lower than they should be. High decision thresholds force the model to predict a larger proportion of fraudulent transactions and increase the recall rate.

### 4.5 Limitations

Despite the promise shown by our approach, several limitations require further discussion. First, our data pre-processing and prevention of data leakage introduced noise into the aggregated features. This approach may not accurately reflect genuine behavioral patterns, and thus a larger data set could potentially improve performance. However, the computational expense of DSC raises practical considerations. Second, the representativeness of our data is a concern. The PaySim data set, simulated from real mobile money transaction may not be representative of transaction patterns globally. This, however, is a common issue in fraud detection research as the availability of realistic data is limited due to considerations on privacy, competitiveness and systemic risk. Additionally, while our model should adapt well to evolving tactics of fraudsters we did not evaluate our approach for this due to data set limitations. Moreover, we worked with only a single fraud expert. Carrying out the same study with multiple fraud experts can help to address and provide more perspectives. Third, in our pre-processing

of balance data, we favoured mitigating leakage and the integrity of the transaction amounts at the cost of accuracy with respect to balance amount. Fourth, due to the substantial runtime of experiments, we did not perform full cross-validation. We did perform multiple runs with varying random seeds to ensure robustness of results, however. Future studies should consider cross-validation to potentially enhance the robustness of the results even further. Fifth, DSC demonstrated a higher variance in performance relative to benchmark models such as RF and XGBoost. As our proposed framework incorporates probabilistic components, some runs may escape local optima more quickly than others. This suggests that our approach can benefit from existing approaches to escaping local optima that have already been adopted by established techniques, including those included in the benchmark. Due to the relatively high computational expense of experimentation, we leave these improvements as future work. Finally, since we are dealing with binary classification problem, we used binary F1 score; however, for multi-class classification problems, other metrics can be considered e.g., weighted F1 or Matthews Correlation Coefficient.

## 5 Conclusion

In this work, we introduced Deep Symbolic Classification, a novel framework for explainable fraud detection in financial transactions. Our approach involves training a deep symbolic regression algorithm to generate analytical expressions with a classification-based reward function. We incorporate a sigmoid layer and a tunable decision threshold to turn regression into classification. By using the F1 score as the reward function instead and by setting a decision threshold of 0.8, we have effectively mitigated the challenges associated with high class imbalance, a key issue in the fraud detection domain. By taking the class imbalance problem head on, DSC eliminates the need for problematic techniques such as oversampling or undersampling. The models generated by DSC are transparent and allow for straightforward inspection of features. In particular, our analysis has revealed that certain key features align with expert knowledge about fraudulent transactions. We observed that transaction type, intra-bank transactions, and the amount of the transaction relative to the last six transactions of the recipient were significant factors in determining fraudulent activities.

Our framework facilitates the creation of models with varying complexity and predictive performance and the creation of a Pareto front. Analysts and other stakeholders can select the model that best aligns with their desired trade-off between explainability and predictive performance from this set of optimal solutions. In our case study, we found an optimal solution that could be transformed into a concise decision rule based on only three Boolean variables.

Elaborating further on the aspect of predictive performance, DSC exhibits slightly lower performance compared to SOTA models on the PaySim data set. However, DSC achieves precision and recall scores of 0.95 and 0.67, respectively, indicating a minimal occurrence of misclassified legitimate transactions and a notable ability to detect approximately two thirds of fraudulent transactions.



It is important to note that the SOTA models lack explainability and exhibit only marginally better performance, thus positioning DSC as a promising model for fraud detection. However, additional research needs to be done on different data sets and different operators to provide a definitive conclusion regarding its practical implementation in industry.

Regarding future directions, several areas can be explored. Firstly, incorporating relational operators (e.g.,  $\geq$ ,  $<$ ,  $\neq$ ) or aggregational operators (e.g., mean, standard deviation, percentiles) in the library of tokens can help eliminate the need for manual feature engineering.

Additionally, exploring alternatives to the sigmoid function for mapping expression values to probability spaces could be fruitful. Multilayer Discriminant Classification presents an interesting option, wherein two expressions are created, one for each class, and the argmax of their weighted values determine the classification [27]. The weights of the features in both expressions directly optimize the likelihood of each class.

Moreover, the recurrent expression generation process of DSC, trained via reinforcement learning, lacks parallelization, resulting in relatively high computation times. A potential solution is to investigate transformer-based symbolic regression, as introduced by [12], to address this limitation.

**Acknowledgements.** We kindly thank Wim Tip for sharing his expertise on fraud detection and the anonymous reviewers for their useful suggestions to improve on this work. Floris den Hengst is generously funded by NWO Hybrid Intelligence Project (024.004.022).

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## A Baseline Model Configuration

The training set was randomly undersampled to achieve a balanced training set. Both the balanced training set and the original training set were used to train the baseline models. Subsequently, these models were tested on an unbalanced test set. The parameters of the baseline models are displayed in Table 4.

## B Preprocessing the PaySim Dataset

The following steps were taken into account to preprocess the data set:

- Certain transactions in the data set exhibited non-zero amounts, but had corresponding old and new balances of zero. To address this scenario, we introduced the features *externalOrig* and *externalDest* for the customer and recipient accounts, respectively (please refer to Table 5 for further details). Following this, we performed imputation of the balances according to the following relationships:

$$\begin{aligned} newbalanceDest &= oldbalanceDest + amount \\ oldbalanceOrig &= newbalanceOrig + amount \end{aligned}$$

**Table 4.** Parameters of the baseline models

Model	Library	Parameters
$k$ -NN	scikit-learn	$k = 2$
SVM	scikit-learn	complexity parameter $C = 1$ , kernel function = polynomial, $\gamma = 0.01$
RF	scikit-learn	number of trees = 200
XGBoost	XGBoost	booster = gbtree, $\eta = 0.3$ , $\gamma = 0$ , maximum depth of a tree = 3, sampling method = uniform, $\lambda = 1$ , $\alpha = 0$

- Additional features were obtained through aggregation techniques in the data set. Descriptions of these features are given in Table 5.
- The features *nameOrig*, *nameDest* and *isFlaggedFraud* were discarded.
- The feature *type* was one-hot encoded.
- The data was randomly split into a training, validation, and test set which encompassed 75%, 10% and 15% of the data, respectively.
- A standard scaler was fitted on the numerical columns of the training set. Subsequently, the numerical columns of the training, validation, and the test set were scaled using this fitted standard scaler.
- For some of the baseline models, an additional balanced training set was generated by randomly undersampling the training data. Specifically, all fraudulent transactions were retained and an equal number of legitimate transactions was randomly selected to match the count of fraudulent instances.

We here briefly describe and motivate some modeling decisions made in the experiments. In all experiments we aim to incorporate aggregation features that encompass *all* previous transactions of both the customer and the recipient, providing insight into their overall behavior patterns. The PaySim data set represents 30 d of transactions, which results in a major fraction of the account holders to participate in a low number of transactions. As a consequence, aggregation features may not accurately describe the individual’s overall behavior. To address this issue, we assume that subsequent transactions are independent from the current transaction: they primarily reflect the individual’s general behavior and exhibit similar distributions as those observed in previous (yet unseen) months. Therefore, we include future transactions as well in certain aggregation features. Thus, for each transaction, we add characteristics that show the *mean* and *maximum* transaction amount over the entire data set of both the customer and recipient. This approach has a risk of data leakage, as earlier transactions may contain information from subsequent time steps through the balance features. However, we argue that future transaction information primarily reflects general user behavior and therefore does not constitute a form of data leakage. To reflect that these features model overall customer behavior and reduce the risk of data leakage even further, we add a Gaussian noise to aggregation features that contain future information.

**Table 5.** Descriptions of the additional features that were added to the data set

Feature	Description
externalOrig	Boolean variable that indicates whether the customer account is likely associated with an external institution, an account is considered external if both <i>oldbalanceOrig</i> and <i>oldbalanceOrig</i> equal zero
externalDest	Boolean variable that indicates whether the recipient account is likely associated with an external institution, an account is considered external if both <i>oldbalanceDest</i> and <i>oldbalanceDest</i> equal zero
meanOrig	mean value of all transaction amounts (excluding the current amount) associated with a particular customer, with added Gaussian noise with $\mu = 0$ and $\sigma = 0.01 * (q - m)$ where $q$ denotes the 0.75 quantile of the customer’s transaction amounts and $m$ represents the minimum transaction amount
meanDest	mean value of all transaction amounts (excluding the current amount) associated with a particular recipient, with added Gaussian noise with $\mu = 0$ and $\sigma = 0.01 * (q - m)$ where $q$ denotes the 0.75 quantile of the recipient’s transaction amounts and $m$ represents the minimum transaction amount
maxOrig	maximum value of all transaction amounts (excluding the current amount) associated with a particular customer, with added Gaussian noise with $\mu = 0$ and $\sigma = 0.01 * (q - m)$ where $q$ denotes the 0.75 quantile of the customer’s transaction amounts and $m$ represents the minimum transaction amount
maxDest	maximum value of all transaction amounts (excluding the current amount) associated with a particular recipient, with added Gaussian noise with $\mu = 0$ and $\sigma = 0.01 * (q - m)$ where $q$ denotes the 0.75 quantile of the recipient’s transaction amounts and $m$ represents the minimum transaction amount
meanDest3	mean of the last 3 transaction amounts (including the current amount) associated with a particular recipient
meanDest7	mean of the last 7 transaction amounts (including the current amount) associated with a particular recipient
maxDest3	maximum of the last 3 transaction amounts (including the current amount) associated with a particular recipient
maxDest7	maximum of the last 7 transaction amounts (including the current amount) associated with a particular recipient
numTransOrig	total number of transactions associated with a particular customer
numTransDest	total number of transactions associated with a particular recipient

## References

1. Alarfaj, F.K., Malik, I., Khan, H.U., Almusallam, N., Ramzan, M., Ahmed, M.: Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms. *IEEE Access* **10**, 39700–39715 (2022)
2. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods (June 2018)
3. Aria, M., Cuccurullo, C., Gnasso, A.: A comparison among interpretative proposals for random forests. *Mach. Learn. Appl.* **6**, 100094 (2021)
4. Bahnsen, A.C., Aouada, D., Stojanovic, A., Ottersten, B.: Feature engineering strategies for credit card fraud detection. *Expert Syst. Appl.* **51**, 134–142 (2016)
5. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 785–794. ACM, New York, NY (2016)
6. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection and concept-drift adaptation with delayed supervised information. In: *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, iscataway, New Jersey (2015)
7. Diveev, A., Shmalko, E.: *Machine Learning Control by Symbolic Regression*. Springer, Cham (2021). <https://doi.org/10.1007/978-3-030-83213-1>
8. Garreau, D., Luxburg, U.: Explaining the explainer: a first theoretical analysis of lime. In: *International Conference on Artificial Intelligence and Statistics*, pp. 1287–1296. Springer, Cham, Switzerland (2020)
9. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a “right to explanation”. *AI Mag.* **38**(3), 50–57 (2017)
10. Hajek, P., Abedin, M.Z., Sivaraajah, U.: Fraud detection in mobile payment systems using an XGBoost-based framework. *Inf. Syst. Front.* **162**, 1–19 (2022)
11. Junger, M., Wang, V., Schlömer, M.: Fraud against businesses both online and offline: crime scripts, business characteristics, efforts, and benefits. *Crime Sci.* **9**(1), 13 (2020)
12. Kamienny, P.A., d’Ascoli, S., Lample, G., Charton, F.: End-to-end symbolic regression with transformers. *Proc. NeurIPS* **35**, 10269–10281 (2022)
13. Kim, E., et al.: Champion-challenger analysis for credit card fraud detection: hybrid ensemble and deep learning. *Expert Syst. Appl.* **128**, 214–224 (2019)
14. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA (1992)
15. Kumar, I.E., Venkatasubramanian, S., Scheidegger, C., Friedler, S.: Problems with shapley-value-based explanations as feature importance measures. In: *International Conference on Machine Learning*, pp. 5491–5500. PMLR, Vienna, Austria (2020)
16. La Cava, W., Orzechowski, P., Burlacu, B., de Franca, F.O., Virgolin, M., Jin, Y., Kommenda, M., Moore, J.H.: Contemporary symbolic regression methods and their relative performance. In: *Thirty-fifth Conference on Neural Information Processing Systems*. PMLR, online (2021)
17. Landajuela, M., et al.: A unified framework for deep symbolic regression. *Proc. NeurIPS* **35**, 33985–33998 (2022)
18. Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., Kochenderfer, M.J., et al.: Algorithms for verifying deep neural networks. *Found. Trends® in Optimization* **4**(3–4), 244–404 (2021)
19. Lopez-Rojas, E., Elmir, A., Axelsson, S.: PaySim: a financial mobile money simulator for fraud detection. In: *28th European Modeling and Simulation Symposium, EMSS, Larnaca*, pp. 249–255 (2016)

20. Mainali, P., Psychoula, I., Petitcolas, F.A.: ExMo: Explainable AI Model using inverse frequency decision rules. In: International Conference on Human-Computer Interaction, vol. 13336, pp. 179–198. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-05643-7\\_12](https://doi.org/10.1007/978-3-031-05643-7_12)
21. Mundhenk, T.N., Landajuela, M., Glatt, R., Santiago, C.P., Faissol, D.M., Petersen, B.K.: Symbolic regression via neural-guided genetic programming population seeding (2021)
22. Nesvijevskaia, A., Ouillade, S., Guilmin, P., Zucker, J.D.: The accuracy versus interpretability trade-off in fraud detection model. *Data Policy* **3**, e12 (2021)
23. Petersen, B.K., Larma, M.L., Mundhenk, T.N., Santiago, C.P., Kim, S.K., Kim, J.T.: Deep symbolic regression: recovering mathematical expressions from data via risk-seeking policy gradients. In: Proceedings of ICLR (2021)
24. Raghavan, P., El Gayar, N.: Fraud detection using machine learning and deep learning. In: 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), pp. 334–339. IEEE (2019)
25. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Mach. Intell.* **1**(5), 206–215 (2019)
26. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.): Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-28954-6>
27. Sipper, M.: Binary and multinomial classification through evolutionary symbolic regression. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 300–303 (2022)
28. Smits, G.F., Kotanchek, M.: Pareto-front exploitation in symbolic regression. *Genetic programming theory and practice II*, pp. 283–299. Springer, Cham (2005). [https://doi.org/10.1007/0-387-23254-0\\_17](https://doi.org/10.1007/0-387-23254-0_17)
29. Sovrano, F., Vitali, F.: An objective metric for explainable AI: how and why to estimate the degree of explainability. *Knowl.-Based Syst.* **278**, 110866 (2023)
30. Sundarkumar, G.G., Ravi, V., Siddeshwar, V.: One-class support vector machine based undersampling: application to churn prediction and insurance fraud detection. In: 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICIC), pp. 1–7. IEEE (2015)
31. Varshney, K.R., Alemzadeh, H.: On the safety of machine learning: cyber-physical systems, decision sciences, and data products. *Big data* **5**(3), 246–255 (2017)
32. Vilone, G., Longo, L.: Explainable artificial intelligence: a systematic review (2020)
33. Wexler, R.: When a computer program keeps you in jail. *NY Times* **13** (2017)
34. Whitrow, C., Hand, D.J., Juszczak, P., Weston, D., Adams, N.M.: Transaction aggregation as a strategy for credit card fraud detection. *Data Min. Knowl. Disc.* **18**, 30–55 (2009)
35. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**, 5–32 (1992). <https://doi.org/10.1007/BF00992696>



# Better Luck Next Time: About Robust Recourse in Binary Allocation Problems

Meirav Segal<sup>1</sup>(✉) , Anne-Marie George<sup>1</sup> , Ingrid Chieh Yu<sup>1</sup> ,  
and Christos Dimitrakakis<sup>2</sup> 

<sup>1</sup> University of Oslo, Oslo, Norway

{meiravs, annemage, ingridcy}@ifi.uio.no

<sup>2</sup> University of Neuchatel, Neuchatel, Switzerland

christos.dimitrakakis@unine.ch

**Abstract.** In this work, we present the problem of algorithmic recourse for the setting of binary allocation problems. In this setting, the optimal allocation does not depend only on the prediction model and the individual’s features, but also on the current available resources, utility function used by the decision maker and other individuals currently applying for the resource. We provide a method for generating counterfactual explanations under separable utilities that are monotonically increasing with prediction scores. Here, we assume that we can translate probabilities of “success” together with some other parameters into utility, such that the problem can be phrased as a knapsack problem and solved by known allocation policies: optimal 0–1 knapsack and greedy. We use the two policies respectively in the use cases of loans and college admissions. Moreover, we address the problem of recourse invalidation due to changes in allocation variables, under an unchanged prediction model, by presenting a method for robust recourse under variables’ distributions. Finally, we empirically compare our method with perturbation-robust recourse and show that our method can provide higher validity at a lower cost.

**Keywords:** Counterfactual explanations · Algorithmic recourse · Allocation problems

## 1 Introduction

Automated decision-making systems are currently employed in many high-risk applications such as granting loans [49] or admitting students to higher-education programs [45]. As these applications have a great impact on people’s lives and future trajectories, it is important to provide individuals with explanations regarding such decisions and algorithmic recourse—actions the individual can take in order to obtain the desired outcome. A widely used approach is counterfactual explanations (CE). With this approach, an explanation model outputs a feature vector that would have obtained the desired outcome, and requires minimal changes to the original feature vector of the individual. For example,

an individual who has been denied a loan might be told that they must increase their salary by 1000\$ in order to qualify for a loan. As such, CE and algorithmic recourse are of the same format: the description of individual features that would yield the desired outcome. While CEs interpret this as an explanation of what the current individual is lacking, in the view of recourse the action recommendation is to obtain the described features.

The CE and recourse literature is mainly focused on binary classification settings. In these problems, a model  $M$  predicts the probability of success (e.g., of repaying the loan or graduating) for each individual and then a function  $f : [0, 1] \rightarrow \{0, 1\}$  outputs a decision by setting a threshold over those scores [14, 41]. All individuals with success probability above this threshold are assigned with the desired label ('loan granted' or 'admitted') and the full classifier is  $h = f \circ M$ . We extend this line of work to allocation problems where budget constraints do not necessarily allow for a threshold policy. Here individuals with lower (prediction) value might be accepted because of their lower cost.

In allocation problems, a decision maker (DM) is allocating *limited* resources among a *population* in order to maximise some *objective* (such as profit to the bank), sometimes under additional constraints. The decision is determined according to the available resources, current population (or applicant pool) and the DM's utility function. Applications such as college admissions and loan granting, which are usually considered as classification problems [18, 32], are in fact dependant on resource constraints (and consequentially the whole population) and are thus better phrased as allocation problems. For example, for the lending use case, we can consider a bank making a decision every time step based on a batch of loan applications. The bank can offer a loan to a limited number of individuals depending on the bank's budget, these individuals are selected among the current set of loan applicants and the bank may have a utility function which takes into account different factors, for instance the current interest rate. Hence, it is insufficient to provide counterfactual explanations with respect to a prediction model (e.g., of the probability of repaying the loan), which usually only considers prediction accuracy. Instead, counterfactual explanations should be made with respect to the entire decision making process, i.e., the allocation problem and its variables.

All three allocation problem variables – resources, population and utility – may change over time. Following the lending example, possible changes of variables could include:

- *Resources*. The bank may have a different budget in the next time step, which would make it easier or harder to be granted a loan.
- *Applicants*. We do not expect to see the exact same population applying again for a loan.
- *Utility*. According to the current market, a bank may change their utility function to be more or less risk averse.

As the goal of acting upon a given recourse is to yield the desired outcome in the future, it is crucial that the recourse remains valid over time. That is, the

individual receives the desired outcome at a later time step following the implementation of the recommended recourse. Following the above lending example, a recourse that is based on the current resources, applicants and utility may not be valid at the next time-step.

Previous works that address recourse invalidity, due to changes of the prediction model or retraining following a distribution shift, consider robustness with respect to worst-case perturbations. As a natural result, many works also demonstrate the cost-robustness trade-off, meaning that the cost of a recourse increases with its robustness guarantees. In a work by Pawelczyk et al. [41] the users are granted control over this trade-off by setting a desired validity rate. However, these robust methods do not account for the probability of these perturbations or the likelihood of the current parameters. When taking the distribution of the problem variables into account and providing recourse that is valid with high probability, the cost of the recourse might in fact decrease. A user could be simply “unlucky” in a specific allocation, but could be assigned with a favorable decision given the same features with high probability, depending on the distribution over problem variables. This information is crucial for the user, which might otherwise invest significant efforts to implement an unnecessarily difficult recourse. To this end, we model changes in allocation variables by sampling them from a known distribution and propose a distribution-aware method for robust recourse.

In this paper, we focus on binary allocation problems with monotonic separable utilities (Sect. 3). We then present two use cases for such problems with different allocation policies: loans and college admissions with optimal 0–1 knapsack and greedy policies respectively (Sect. 4). For these problems, we provide a pipeline for generating CE under a black-box prediction model and an allocation policy (Sect. 5). Here we assume to have access to a CE-generator for classifiers and encapsulate this part in the pipeline. Next, we provide an algorithm for approximating a robust recourse given a distribution over the allocation variables and perform empirical analysis using budget distributions (Sect. 6).

Our contributions are as follows:

1. We propose allocation problems as a novel setting for considering robust algorithmic recourse.
2. In this setting, under a combination of a classification model and an allocation policy, we show through examples that the counterfactual explanations for allocations can be more reliable for static allocations compared to counterfactual explanations for the associated classification tasks.
3. For algorithmic recourse in repeated allocations, we empirically show that a distribution-aware robust recourse could reduce the cost in some cases while still providing high chances of achieving the desired outcome.

## 2 Related Work

*Recourse for Allocations.* A recent survey about algorithmic recourse [24] mentions that recourse should be extended to matching problems and allocation



problems. Yet, to the best of our knowledge, the problem of robust algorithmic recourse for allocation problems has not been addressed in the literature so far. The literature closest to this problem is from the field of scheduling and routing problems, where several contributions deal with explainability by answering “why-not” and “what-if” questions [9, 15, 29, 31]. Yet, most works address the end-user of the explanation as the scheduler (or employer), and do not consider the individual’s point of view (employee who was assigned to tasks). One line of work considers the perspective of the individual and generates CE using inverse optimization [28]. However, this work does not address the problem of algorithmic recourse and possible changes to the problem variables and constraints. Recently, a new work discussing the impact of recourse on the distribution of future population also included a resource constraint and addressed the competition between individuals currently applying for the resource [17]. Here, the recourse definition is limited to a threshold allocation policy, the paper does not regard the DM’s utility, and the need for robustness is not addressed.

*Recourse Invalidation.* The problem of algorithmic recourse invalidation and the need for robustness has already been recognised in recent years [35]. The majority of papers consider invalidation due to model retraining with different training data, usually following a distribution shift [5, 6, 14, 20, 27, 37, 38, 42, 46]. We propose that even with the same data distribution, the differences in sampled populations from one allocation to another may lead to recourse invalidation. Moreover, we also address possible invalidation due to change of resources or utility function. The latter was identified as an open problem in a recent survey of causal machine learning [23]. Other studied causes of invalidation are change of prediction model [40] and feature perturbation, which could be due to inaccurate implementation of the recourse [11, 41, 47] or privacy perturbation [36]. We do not address these kinds of invalidation and assume that the recourse is implemented in full.

*Robust Recourse.* Many works try to improve recourse robustness by considering the worst-case adversarial perturbation (e.g. of the data distribution) within a set of plausible changes, usually measured by distance up to a specific value [6, 11, 37, 38, 46, 47]. While these methods indeed improve the robustness of the recourse, they also present a trade-off between robustness and cost (e.g., distance of the counterfactual from the original feature vector) [40, 42, 46]. For deep networks, even if no explicit trade-off exists, the robust recourse is still presented as more costly [5]. Nonetheless, these methods do not take into account the probability of such worst cases or question whether the current variables should be used as a point of reference for increasing robustness. We present a robust recourse under the variables’ distribution, which could result in lower costs for the users, compared with the worst-case recourse with respect to current variables. When considering the distribution, we can also provide the user with more control over the robustness-cost trade-off. This was proposed in a recent paper [41] assuming a specific noise distribution over recourse implementation. A similar method was also suggested for generating counterfactual explanations under uncertainty of

the causal relations in the data [25]. We facilitate the same kind of control for allocation problems.

*Other Approaches.* Ferrario and Loi [16] suggest a different approach for handling recourse invalidation. They propose a method for retraining the prediction model such that counterfactual explanations generated in the past would still hold. A similar problem to recourse robustness is the uncertainty of counterfactual examples with respect to the data distribution [3, 10, 43]. We do not address this problem, and assume that the black-box explanation model provides a reasonable counterfactual explanation with respect to the data distribution.<sup>1</sup>

### 3 Binary Allocation Problems

A binary allocation problem is a triple  $\langle r, X, U \rangle$  where  $r$  represents the available quantity of the resource (such as budget),  $X$  is the given population of size  $n$  with  $x_i \in \mathbb{R}^l$  being the feature vector of individual  $i \in \{1, \dots, n\}$  which includes  $w_i$ , the resource amount requested by applicant  $i$ , and  $U$  is the utility function that the DM is trying to maximise. An allocation policy  $\pi$  outputs a valid allocation or assignment, represented by a binary vector  $Y = \{0, 1\}^n$ , where  $y_i = 1$  means that individual  $i$  is assigned with  $w_i$  of the resource, and  $y_i = 0$  means that they are assigned with none of the resource. A valid allocation is an allocation for which the allocated quantity of the resource does not exceed the available quantity, i.e.,  $\sum_i y_i w_i \leq r$ . In the following sections, we consider the CE, valid recourse and robust recourse to be with respect to the preferred assignment  $\hat{y}_i = 1$ .

*Separable Utility and Prediction Model.* In this paper, we focus on settings in which the DM's utility for allocation  $Y$  is separable over the population, meaning that it can be decomposed into a sum of individual utilities  $v_i$  for each person  $i$  to which a resource is allocated, i.e.,  $U(Y) = \sum_{i:y_i=1} v_i$ . The individual utility  $v$  is the output of an individual utility function  $u : \mathbb{R}^l \rightarrow \mathbb{R}$  which takes the individual's feature vector as input, i.e.  $u(x_i) = v_i$ . Moreover, we restrict the function  $u$  to be of a specific form – a composition of two functions  $u = S_\theta \circ M$ . The function  $M : \mathbb{R}^l \rightarrow [0, 1]$  is a prediction model, which maps a feature vector to a single value. This can, for example, represent the success probability of repaying a loan. The function  $S_\theta : [0, 1] \rightarrow \mathbb{R}$  is a monotonically increasing function parameterised by  $\theta$ . The parameter  $\theta$  could, for instance, represent the current interest rate. The individual utility can be interpreted as the predicted gain if we allocate the requested resource to the individual.

---

<sup>1</sup> We note that a problem which might be considered as related is the of use counterfactuals to explain classification uncertainty [30]. This is a different objective and in our work we do not account for prediction uncertainty.

## 4 Use Cases

We present two possible allocation policies for commonly used applications for high-risk decisions: lending and college admissions. Using these examples, we motivate the need for CE for binary allocation problems.

### 4.1 Lending

The lending use case is often seen as an example of a high-risk application of automated decision making systems [33]. In this problem, individuals apply for a loan by providing information such as requested credit, purpose of the loan, current salary and demographic information. Based on these features, the prediction model employed by the DM (in this case, the bank or lending institute) predicts the individual’s probability of repaying the loan. Previous papers consider this as a classification problem, and the allocation policy to be simply setting a constant threshold over these probabilities. We formulate this problem as an allocation problem and describe our concrete modelling choices in the following. This formalism is particularly relevant for student loans in the US, where the Federal Student Aid Programs operate under a limited budget and all applications for the next academic year are submitted up to a set deadline [2].<sup>2</sup>

*Utility Function.* Following the student loan use-case, we assume that the DM’s gain from each successful applicant is twofold: 1) the DM has a (monetary) profit—a constant fraction  $G_1 \in [0, 1]$  out of the requested credit,<sup>3</sup> and 2)  $G_2 \in \mathbb{R}$ , a value that represents the social value of granting a loan, e.g., by enabling an educational opportunity to an individual who could not have afforded this otherwise, and allowing them to increase future financial prospects. In case the individual was not able to repay the loan, the DM loses a fraction  $C \in [0, 1]$  of the loan. For simplicity, we assume that  $C$  is constant and has the same value for all applicants. Thus, the expected utility when granting a loan to individual  $i$  is

$$u(x_i) = M(x_i)(w_i G_1 + G_2) - (1 - M(x_i))Cw_i. \quad (1)$$

*Allocation Policy.* The DM is trying to maximise utility under budget constraints, where each applicant has individual utility and desired credit. This problem can be translated to the well known 0–1 knapsack problem [4]. Here, the weight capacity of the knapsack is the budget  $r$ , we have  $n$  items (individuals), each item  $i$  has value  $v_i = u(x_i)$  and a weight  $w_i$ . Items with negative utility can be removed since including them cannot increase the allocation utility. Considering weights and values to be non-negative, the problem is given

<sup>2</sup> Other examples of such allocation problems, with a limited budget and applicants requesting different quantities in batches, include funding agencies and grant applications.

<sup>3</sup> In practice, the utility function also depends on the time for which the loan is requested, but we ignore this component for simplicity.

by  $\max \sum_{i=1}^n v_i y_i$  s.t.  $\sum_{i=1}^n w_i y_i \leq r$ , i.e., filling the “knapsack” with the most value while respecting its capacity. This constrained optimisation problem is NP-complete, yet solvable in pseudo-polynomial time using dynamic programming. Note that we assume discretisation: the credit has a minimal step size (e.g. 100\$). We therefore assume that the DM’s allocation policy for this application is determined by the optimal solution.

**Table 1.** Motivating example: lending use case. Table of applicants with their predicted probability of repaying a loan according to model  $M$  and their features, the individual utility for the DM according to Eq. 1 and the credit each applicant is requesting (in thousands of dollars).

Applicant	$M(x_i)$	$u(x_i)$	<b>Credit</b> ( $w_i$ )
1	0.8	0.8	4
2	0.7	0.625	3
3	0.6	0.5	2
4	0.5	0.425	1

*Motivating Example.* Consider the applicants described in Table 1 under the utility function  $u(x_i) = M(x_i)(w_i(G_1 + C) + G_2) - Cw_i$  (Eq. 1 rearranged) using the parameters  $G_1 = 0.05, G_2 = 1, C = 0.2$  and budget of 6 (thousand dollars). The optimal allocation is  $Y = (0, 1, 1, 1)$ , meaning approving the loan for applicants 2, 3 and 4 with utility of 1.55 for the DM. Note that applicant 1 was not selected, even though their probability of repaying the loan is higher than that of the other applicants, as well as their individual utility for the DM. Thus, it would be difficult to explain the decision when only considering the prediction model, without the allocation mechanism, remaining population and budget constraint.

## 4.2 College Admission (Greedy)

College admission is a highly researched problem [22], and it concerns educational opportunities. As such, it is regarded as a high-risk application for which individuals are entitled to an explanation (according to the European AI Act [33]).

In the case of college admission, individuals apply by providing educational and demographic background information. Based on these features, the prediction model employed by the DM (i.e., the university) predicts the individual’s probability of graduation. For college admissions, the weight or requested quantity is identical for all candidates (one admission slot  $\forall i, w_i = 1$ ). Thus, the use of a threshold policy would be optimal, assuming that the utility of the allocation is the sum of individual utilities. Nonetheless, while previous papers set the threshold over the graduation probabilities, for allocation problems the threshold would be over the individual utilities, and applicants are greedily added as

long as the utility increases and the assignment does not exceed the budget [8]. This means that individuals with non-positive utility will not be included in the admitted set regardless of the budget constraint. Hence, the decision also depends on the resource and the utility function, which might lead to different results from those we get using only a predictor.

*Utility Function.* Let us assume that for every admitted student, the university pays a constant cost  $C \geq 0$ . In addition, for every admitted student who successfully graduates, the university receives a constant reward or gain  $G \geq 0$ . Thus, the expected (individual) utility for an admitted student  $i$  is

$$u(x_i) = M(x_i)(G - C) + (1 - M(x_i))(-C) = M(x_i)G - C. \quad (2)$$

Again, we denote  $u(x_i)$  as  $v_i$ .

**Table 2.** Motivating example: college admission. Table of applicants with their predicted probability of graduating according to model  $M$  and their features, and the individual utility for the DM.

Applicant	$M(x_i)$	$u(x_i)$
1	0.8	0.2
2	0.7	0.1
3	0.6	0.0
4	0.5	-0.1

*Motivating Example.* Let us consider the utility function from Eq. 2 with  $G = 1$  and  $C = 0.6$  as shown in Table 2. For the case of  $r = 2$  (two free study slots), the optimal allocation would be to select the top two applicants: 1 and 2 ( $Y = (1, 1, 0, 0)$ ). In this case, in order to be selected, applicant 3 should increase their utility such that it is higher than that of applicant 2. For the case of  $r = 3$ , the optimal allocation would also be  $Y = (1, 1, 0, 0)$ , since selecting applicant 3 would not increase the utility. Thus, in order to be selected, it is sufficient for applicant 3 to increase their utility by  $\epsilon > 0$ . This could be a significant difference of investment cost for the applicant, which could not be captured by considering the prediction model alone.

## 5 Counterfactual Explanations

We start off by giving a formal definition of counterfactual explanations that is based on the definition of counterfactual explanation for classification problems [19]. We then describe how to generate these in our specific setting.

**Definition 1 (Counterfactual Explanation for Binary Allocations).**

Given an allocation policy  $\pi$  that outputs the decision  $Y$  for population  $X$ , utility function  $U$  and given resource  $r$ , a counterfactual explanation for individual  $x_i \in X$  consists of an alternative vector of features  $x'$  for which the allocation  $Y' = \pi(r, X \cup \{x'\} \setminus \{x_i\}, U)$  is different from  $Y$  such that  $y'_i = 1$ . We define such a counterfactual explanation to be minimal if its cost to the individual  $d(x_i, x')$  is minimal under some metric  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$ .

Note that here, there could be another individual  $j \neq i$  for which  $y'_j \neq y_j$ , meaning that the CE might change the assignment for other individuals and not only the individual requesting the CE.

Assume we are given a prediction model  $M$ , an allocation policy  $\pi$ , an individual utility function  $u = S_\theta \circ M$  such that  $S_\theta$  is a monotonically increasing function, a population  $X$ , resources  $r$  and a metric  $d$  in the feature space. We propose to generate a CE according to the pipeline below.

1) *Computing the Minimal Utility-CE.* Given an allocation policy, we first produce a *minimal utility-CE*  $v'$ , i.e., the minimal utility that would have led to a preferred assignment. For the two allocation policies we focus on:

- **Optimal 0-1 knapsack policy.** Intuitively, the individual utility should increase by the difference between the current maximal allocation utility and the maximal allocation utility under the constraint of including individual  $i$ . We denote the optimal allocation for applicant set  $[n]$  and available resources  $r$  as  $Y^*([n], r)$ . We claim that the minimal utility-CE for individual  $i$  is  $v'_i = U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i))$ , where  $U$  is the utility of the allocation. A proof for this result and additional notes can be found in the appendix. We can thus use a dynamic programming algorithm<sup>4</sup> for 0-1 knapsack, see e.g., [34], to compute the minimal utility-CE. In practice, to avoid ties we set  $v'_i = U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i)) + \epsilon$  for some  $\epsilon > 0$ .
- **Greedy policy.** Following the example in Table 2, the utility-CE for individual  $i$  is either 1) larger than the utility of the selected applicant with the smallest utility, in case the budget was fully utilised or 2) larger than 0 in case there are still available vacancies. Formally, we propose for any  $\epsilon > 0$ :

$$v'_i = \begin{cases} (\min_{j:y_j=1} v_j) + \epsilon, & \text{if } \sum_j y_j = r \\ \epsilon, & \text{otherwise} \end{cases}$$

2) *Computing a Prediction-CE.* The minimal utility-CE is translated to a *prediction-CE*  $m'$ , i.e., the minimal success probability that would have led to a preferred assignment. Because  $S_\theta$  is monotonically increasing, it is also invertible. Then the prediction-CE is  $m' = S_\theta^{-1}(v')$ .

---

<sup>4</sup> Simply put, a table  $V$  of size  $n \times r$  is being filled. Each cell  $V[i, j]$  holds the value of the maximal utility that can be obtained given items  $1, \dots, i$  and maximal weight  $j$ .

3) *Computing a Minimal (Feature-Based) CE.* Using the prediction-CE, a minimal CE  $x'$  is generated by solving the following optimisation problem:

$$x' = \arg \min_z d(z, x) \quad \text{s.t.} \quad M(z) \geq m'. \quad (3)$$

For example, we can construct the function  $h_{m'}$  with  $h_{m'}(x) = 1$  if  $M(x) \geq m'$  and  $h_{m'}(x) = 0$  otherwise. Then, one of the many existing explanation models for classifiers [19, 39] (e.g., [48]) can be used on  $h_{m'}$  with metric  $d$ , which provides  $x'$ , a minimal CE with respect to the feature-based cost function  $d$ . Note that we consider  $w_i$  to be fixed.

At the end of this process,  $x'$  is minimal with respect to  $d$  and  $M(x') \geq m'$ . Hence,  $x'$  is a minimal CE for the allocation problem under the following assumptions: 1) the utility function is monotonic in the prediction scores, and 2) the allocation policy is monotonic in the utility, i.e., increasing the utility for an individual assigned with the resource could never change the allocation such that the individual is not assigned with the resource. Both policies (optimal 0–1 knapsack and greedy) satisfy these monotonicity assumptions.

To mitigate the effect of specific classification explanation choices in step 3), we can define the CE in terms of success probability or prediction score (prediction-CE). In the remainder of the paper, we assume the cost function is defined with respect to the predicted probability of success:  $d_M(M(x_i), m') = |M(x_i) - m'|$ .

Using our proposed method, we can see that for the example in Table 1 the optimal allocation under the constraint of including applicant 1 is  $Y' = (1, 0, 1, 0)$  with utility of 1.3. Hence, applicant 1 should increase their individual utility to be at least  $1.55 - 0.5 = 1.05$  which translates to increasing their probability of repaying the loan from 0.8 to 0.925.

## 6 Robust Recourse for Binary Allocations

Counterfactual explanations are used to explain the current decision, but for repeated settings, we wish to provide recommendations for the future, i.e. recourse. We assume that the available resources, population and utility function may change from one time step to the next, which may lead to invalidation of the recourse. We first define (robust) recourse for binary allocations under variable distributions, then describe how to generate approximate robust recourse and lastly evaluate this in our experiments.

**Definition 2 (Valid Recourse for Repeated Binary Allocations).** *At time  $t_1$ , given the allocation instance  $\langle r_{t_1}, X_{t_1}, U_{t_1} \rangle$ , a recourse for individual  $x_i \in X_{t_1}$  consists of an alternative vector of features  $x'$ . This recourse is valid at time  $t_2 > t_1$  if for the new allocation instance  $\langle r_{t_2}, X_{t_2} \in \mathbb{R}^{n-1, l}, U_{t_2} \rangle$ , the allocation policy outputs the allocation  $Y^{t_2} = \pi(r_{t_2}, X_{t_2} \cup \{x'\}, U_{t_2})$  such that  $y_i^{t_2} = 1$ . A recourse is said to be minimal (w.r.t.  $d$ ) if its cost  $d(x_i, x')$  is minimal under the cost metric  $d: \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$ .*

We assume that at each time step the available resources, applicants and utility function are sampled i.i.d. according to a joint distribution  $D$ . Using this distribution, we follow the approach of Pawelczyk et al. [41] and allow the user to control the robustness-cost trade-off by providing a validity probability  $\rho \in [0, 1]$ .

**Definition 3 ( $\rho$ -Robust Recourse for Binary Allocations).** *Let  $x'$  be a recourse generated at time  $t_1$  for individual  $i$  given an allocation problem. Given distribution  $D$  over resources, applicants and utility function,  $x'$  is  $\rho$ -robust if the expected validity at time  $t_2 > t_1$  is at least  $\rho$ , i.e.,*

$$\mathbb{E}_{r_{t_2}, X_{t_2}, U_{t_2} \sim D} [\mathbb{1}_{x' \text{ valid for } \langle r_{t_2}, X_{t_2}, U_{t_2} \rangle}] \geq \rho$$

where  $\mathbb{1}[\cdot]$  is an indicator function. Among all  $\rho$ -robust recourses, a recourse with minimal cost  $d(x_i, x')$  is denoted as a minimal  $\rho$ -robust recourse.

Note that we assume here that the allocation policy is constant, yet this could also be relaxed and added to the sampled variables.

Interestingly, under our definition, a robust recourse may be of cost 0, depending on the distribution and the initial allocation variables. For example, the recourse might have been generated under an extremely unlikely combination of variables, so that the individual was simply “unlucky”.

## 6.1 Approximated Robust Recourse

---

### Algorithm 1. Approximated $\rho$ -Robust Recourse

**Require:** sample size  $n > 0$ , prediction model  $M$ , feature-based explanation function  $E$ , allocation problem  $\langle r, X, u \rangle$ , allocation policy  $\pi$ , distribution  $D$  over resources, applicants and utility parameters  $\{(r_j, X_j, \theta_j)\}_{j \in [n]}$ , individual  $i$ , desired validation level  $\rho \in [0, 1]$ .

**for**  $j$  from 1 to  $n$  **do**

Get sampled variables  $(r_j, X_j, \theta_j) \sim D$

Get utility-CE  $v'_j$  with respect to  $\langle r_j, X_j, u_j = S_{\theta_j} \circ M \rangle$  and policy  $\pi$

Get prediction-CE  $m'_j = S_{\theta_j}^{-1}(v'_j)$

**end for**

Sort all prediction-CE: sorted  $\leftarrow \text{sort}([m'_j]_{j=1}^n)$

Get  $\rho$ -robust prediction-CE  $m_\rho \leftarrow \text{sorted}[\lceil \rho n \rceil]$

**return** feature-based CE  $x' = E(M, i, m_\rho)$

---

We approximate the  $\rho$ -robust recourse for binary allocations, a monotonic separable utility and a monotonic policy using a Monte-Carlo approximation (see Algorithm 1). Given a prediction model  $M$ , an allocation policy  $\pi$ , distribution  $D$  over resource  $r$ , applicants  $X$  and utility function parameter  $\theta$ , for each allocation problem  $\langle r, X, u = S_\theta \circ M \rangle$  such that  $(r, X, \theta) \sim D$ , we can generate a minimal prediction-CE for individual  $i$  as shown in Sect. 5. Given the minimal



prediction-CE for  $n$  sampled problems, we can find  $m_\rho$ , the prediction-CE that is valid for at least  $\rho$  of the sampled allocation problems. Such  $m_\rho$  exists as the allocation is monotonic with respect to the prediction score: for every allocation problem which requires individual  $i$  to have a prediction score of  $m$  in order to receive the resource, any larger prediction score  $q > m$  would also guarantee the resource being allocated to  $i$ . As we can estimate the distribution's quantiles using Monte Carlo approximation [12], this  $m_\rho$  approximates the validity over the entire distribution.<sup>5</sup> This  $\rho$ -robust prediction-CE can then be translated to features, as was proposed in step 3 in Sect. 5. The produced feature vector  $x'$  is then the minimiser of

$$\min_z d(z, x_i) \quad \text{s.t.} \quad \frac{1}{n} \sum_{j=1}^n \mathbb{1}_{M(z) > m'_j} \geq \rho. \quad (4)$$

Here,  $m'_j$  is the  $j$ -th prediction-CE. Note that it is sufficient to sort the thresholds, as is done in Algorithm 1. Hence,  $x'$  is the feature vector with the lowest cost w.r.t.  $d$  which provides individual  $i$  with the resource in approximately  $\rho$  of the allocations. We note that by using the intermediate step of prediction-CE, we reduce the problem to a one-dimensional monotonic recourse. Without this step, for each sampled allocation problem we would generate a different feature-based CE  $x'$ . We do not assume the prediction model  $M$  to be monotonic in the features, i.e., a specific value of feature  $j$  in  $x'$  does not guarantee that all feature vectors with a higher value for feature  $j$  would have a greater or equal prediction score.

## 6.2 Experiments

We empirically evaluate the performance of our robust recourse method in terms of cost and validity. We focus on the case of a changing budget, assuming that the utility of the DM is fixed and the recourse is generated with respect to the current population. In our experiments we produce prediction-CE or prediction-recourse, and measure the recourse cost with respect to the difference in prediction score. As there is no other method for generating a CE for allocation problems, we cannot compare our results with previous methods. Thus, the goal of the empirical results is twofold:

1. Evaluate the cost and validity of the robust-recourse compared to the static CE.
2. Compare our approach of distribution-aware robust-recourse to the previously proposed approach of perturbation-based robust recourse.

---

<sup>5</sup> The accuracy of the approximation depends on the sample size which we consider to be fixed. However, our method could be extended to include a parameter to control the required sample size.

## Datasets and Preprocessing

*Loans.* We use the German credit dataset [13] which is one of the most common benchmarks used for CE and algorithmic recourse (e.g. [5, 6, 14, 20]). The dataset consists of 1,000 samples with 20 features such as age, marital status, education, savings and requested credit. A binary label indicates whether the candidate repaid the loan. We scale numeric features to  $[0, 1]$  and encode categorical features as 1-hot vectors. In addition, the requested credit is divided by 100 and rounded. The data is split to train and test sets with the ratio of 70–30. Then, a random forest classifier with 200 trees is trained on the train set (achieves accuracy of 0.78 on the test set). We construct 20 allocation problems by uniformly sampling 20 individuals from the test set, set the utility function parameters to  $G_1 = 0.06, G_2 = 4, C = 0.5$  using the utility function in Eq. 1, and sample a budget from the budget distribution. The utility parameters are set such that the number of individuals with a positive utility is close to the number of individuals who were granted a loan based on the train set.

*College Admissions.* We sample applicant features and their success probability for 10,000 applicants according to the simulator described in [26], using data from the Norwegian Database for Statistics on Higher Education [1]. We construct 20 allocation problem by uniformly sampling 500 individuals from the simulated data, set the utility function parameters to  $G = 2, C = 1$  using the utility function in Eq. 2, and sample a budget from the budget distribution. The utility parameters are set such that an individual with success probability higher than 0.5 will have a positive utility.

*Budget Distributions.* For both datasets, we sample 50 batches of applicants (20 applicants for German credit and 500 for college admissions) and consider the sum of given credit or admitted students as the current budget or capacity. We then fit a normal distribution to it, and consider this as the budget distribution.

**Method and Baselines.** We test our  $\rho$ -robust recourse method, described in Algorithm 1, with  $\rho \in \{0.7, 0.9\}$ , with 200 budget samples, which we denote as the validation set. We compare our results to the static prediction-CE for allocations. In addition, we implement another recourse method we denote as *p-noisy*. This method is designed to be of a similar nature to perturbation robustness. According to this method, given an allocation problem with a specific budget  $r$ , and an individual  $i$ , we generate a validation set by sampling 200 values from a truncated normal distribution  $\nu_j \sim \mathbb{N}(0, \sigma^2)_{[a, b]}, j \in [200]$ . Then, we generate the minimal prediction-CE for all budgets  $\{r + \nu_j\}_{j \in [200]}$ . The *p-noisy* robust recourse is the maximal among them. The parameter  $p$  controls the range  $[a, b]$  such that  $p$  of the values lie according to distribution  $\mathbb{N}(0, \sigma^2)$  in the range  $[a, b]$ . We set  $\sigma$  to be the standard deviation of the underlying variable (budget) distribution. In our experiments we use  $p \in \{0.7, 0.9\}$ . Moreover, we define an optimistic baseline which is a  $\rho$ -robust recourse generated based on the test budget samples. For

this baseline we set  $\rho = 1$ , so that the generated recourse is valid for the entire test set.

Our assumption is that the test set would be more similar to the validation set used by the  $\rho$ -robust method (sampled from the same distribution), rather than the single sample of the original variable used by the static CE, or the validation set used by the noisy recourse method. Therefore, we expect to see that our method can achieve better results in terms of cost and validity compared to the baselines.

**Results.** For each allocation problem, we find the optimal allocation and provide recourse for all individuals not included in the allocation. The results are described in Table 3. The recourse validity of each individual is measured as the average validity over a test set of 200 samples from the budget distribution. The validity of the method is then the average validity across all individuals. The recourse cost for each method is the average prediction score difference. We normalise all costs by the cost of the optimistic baseline.

**Table 3.** Robust recourse under resource distribution.

Method	Loans		Admissions	
	Cost	Validity	Cost	Validity
Static CE	0.42	0.823	0.756	0.679
0.7-robust	<b>0.407</b>	0.84	<b>0.753</b>	0.771
0.9-robust	0.51	0.917	0.796	0.922
0.7-noisy	0.571	0.888	0.812	0.845
0.9-noisy	0.649	0.977	0.84	0.952
Optimistic	1	<b>1</b>	1	<b>1</b>

From the results in Table 3, we can observe that as expected, higher  $\rho$  or noise values achieve higher validity at a higher cost. We can also observe that the 0.7-robust method Parto-dominates the static-CE for both applications, as it achieves higher validity at a lower cost. This shows that the budgets of some of the allocation problems did not represent the test set and produced a higher-cost prediction-CE. Similarly, the 0.9-robust method Pareto-dominates the 0.7-noisy method in both applications. We can also observe that the  $\rho$ -robust methods are never Pareto-dominated by any other. This shows the advantage of our distribution-based robust-method.

When considering a single individual, by increasing the validity we also increase the cost of the recourse. This is due to our monotonicity assumption for the utility function and the allocation policy. However, when considering the average over the population and the test set, we can see it is possible for our method to achieve higher validity at a lower cost. This could be explained by the fact that the validation set is more likely to represent the test set. When the

original variable is more permissive, allowing resource allocation to more individuals, our method can provide a recourse that would be valid for more restricting samples of the distribution. Thus increasing the average validity and the average cost. When the original variable hinders resource allocation, our method would be able to find “unlucky” individuals that do not require a costly recourse (or recourse at all) to be allocated with the resource for many variable values. Thus, the average cost would be reduced and the validity would remain high.

Another observation we can make from the experimental results, is the difference between the validity on the test set and the requested validity. This gap can be explained by the fact that the validity is estimated based on the validation set and the final validity is computed based on the test set. Since the two sets are not identical, the recourse for which the estimated validity was  $\rho$  (the requested validity) may provide lower or higher validity on the test set. In addition, it is possible that the minimal recourse for the requested validity level already provides a higher validity. For example, let us assume a user is requesting 0.5 validity and the validation set produces the following minimal CE: (0.1, 0.2, 0.01, 0.2, 0.2, 0.25). If we wish to provide 0.5 validity, we must have a recourse of 0.2, but that recourse already provides us with a higher validity of 0.83. This could explain the fact that all methods provide test-set validity that is higher than the requested validity.

## 7 Generalisation and Open Problems

In this paper we only make the first step in solving this new setting of recourse for allocation problems. We address allocation problems with binary decisions and separable utilities. More complex problems within the scope of allocation problems could be addressed in the future. We propose here more general definitions for CE and recourse for a wider class of allocation problems and point out interesting aspects of these problems.

### 7.1 Definitions

We start by providing a general definition of an allocation problem.

**Definition 4 (Allocation Problem).** *An allocation problem is a triple  $\langle R, X, U \rangle$  where  $R = \{r_j\}_{j=1}^k$  represents the available resources, with  $r_j$  being the number of units available of resource  $j$ ,  $X$  is the given population of size  $n$  with  $x_i \in \mathbb{R}^l$  being the feature vector of individual  $i$ , and  $U$  is the utility function that the DM is trying to maximise.<sup>6</sup> An allocation policy  $\pi$  outputs a valid allocation or assignment, represented by a matrix  $A \in \mathbb{R}^{n,k}$  such that  $a_{i,j}$  is the number of units of resource  $j$  allocated to individual  $i$ . A valid allocation is an*

<sup>6</sup> The allocation problem could also be defined as  $\langle R, X, U, C \rangle$  where  $C$  represents additional constraints. The feature vector  $x_i$  could also include preferences over the resources  $[pref(i)]_{i=1}^k \in \mathbb{R}^k$ .

allocation that satisfies  $\forall j \in [k], \sum_{i=1}^n a_{i,j} \leq r_j$ , meaning that the sum of allocated resources is at most the set of available resources. The DM is then trying to find a policy which maximizes the utility function  $U : \mathbb{R}^{n,k} \rightarrow \mathbb{R}$ .

Next, we provide a definitions of counterfactual explanations and valid recourse for general allocation problems.

**Definition 5 (Counterfactual Explanation for Allocations).** Given an allocation policy  $\pi$  that outputs the decision  $A$  for population  $X$ , utility function  $U$  and given resources  $R$ , a counterfactual explanation for individual  $x_i \in X$  with respect to a preferred allocation or assignment for individual  $i$ :  $\hat{a}_i \in \mathbb{R}^k$  consists of an alternative vector of features  $x'$  for which the allocation  $A' = \pi(R, X \cup \{x'\} \setminus \{x_i\}, U)$  is different from  $A$  such that  $a'_i = \hat{a}_i$ . We define such a counterfactual explanation to be minimal if its cost  $d(x_i, x')$  is minimal under some metric  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$ .

**Definition 6 (Valid Recourse for Sequential Allocations).** At time  $t_1$ , given the allocation variables  $R_{t_1}, X_{t_1}, U_{t_1}$ , a recourse for individual  $x_i \in X_{t_1}$  with respect to a preferred allocation  $\hat{a}_i \in \mathbb{R}^k$  consists of an alternative vector of features  $x'$ . This recourse is valid at time  $t_2 > t_1$  if given the new set of allocation variables at  $t_2$ :  $R_{t_2}, X_{t_2} \in \mathbb{R}^{n-1,l}, U_{t_2}$ , the allocation policy outputs the allocation  $A^{t_2} = \pi(R_{t_2}, X_{t_2} \cup \{x'\}, U_{t_2})$  such that  $a_i^{t_2} = \hat{a}_i$ . A recourse is said to be minimal if its cost  $d(x_i, x')$  is minimal under some metric  $d : \mathbb{R}^l \times \mathbb{R}^l \rightarrow \mathbb{R}$ .

Note that if the preferred resources are not in  $R_{t_2}$ , a valid recourse for  $t_2$  does not exist.

## 7.2 Open Problems

*Recourse for Non-binary Decisions.* The problem of recourse for non-binary allocations is closely related to the problem of CE and recourse for regression [21, 44] (for a single resource) and multi-class predictions [7] (for multiple resources). Although some contributions have been made in that respect, these are still open problems. In our definitions, we wish to provide CE and recourse with respect to the individual's preferred outcome. When the allocation problem is not binary, it requires additional information regarding the individual's preferences, and possibly a cost function that takes these preferences as input.

*Non-separable Utility.* In this paper, we only address separable utility functions. Yet, as decisions are being made over batches and not over individuals, the use of non-separable utilities might be needed in some cases. For example, the probabilities of people repaying their loan might not be independent. They might, e.g., be influenced by sectoral or global crises. Thus, a decision maker might assign a higher utility to allocations with a sectoral balance, which cannot be represented by separable utilities. In these cases, it is even more important to consider the rest of the population when providing CE and recourse.

*Recourse Feedback and Multi-agent Recourse.* A mostly unexplored interesting facet of recourse for allocation problems is the fact that an implementation of a recourse by one individual might impact the allocation outcome for other individuals. This question was addressed via an empirical simulation study [17], in which the authors measure the effect of different parameters on recourse validity or “recourse reliability”. Yet, the provided recourse did not take into account the feedback effect of the recourse implementation.

## 8 Discussion

In this paper, we present the first attempt to define robust recourse for allocation problems. Under this setting, we show two examples of allocation policies for which methods for generating CE given a classifier would fail to explain the decision. For repeated allocations, we provide a distribution-aware method for generating robust recourse, as opposed to other methods which consider perturbations of the current problem variables. This approach allows for a recourse which might provide the user with high enough validity at the price of a lower cost. Moreover, our approach grants the user more control over the cost-robustness trade-off by choosing the requested validity probability.

*Assumptions and Limitations.* Our proposed method assumes full knowledge of the utility function structure and the allocation policy. These are reasonable assumptions when considering that the DM is the one providing the recourse. Moreover, we make no assumptions regarding the prediction model and address it as a black-box. In addition, we assume a specific structure of the individual utility function: composition of a parametric function  $S_\theta$  and a prediction model  $M$ , where  $S$  is monotonically increasing. As illustrated in Sect. 4, this structure is reasonable in some applications. However, it fails to capture other interesting applications in which the utility is affected directly by features. For example, for allocating research grants, the utility of a project may depend on the specific topic or planned collaborations, not only on the success probability of the proposed project. Our pipeline for generating CE and robust recourse does not provide a solution for these cases and an extension is left for future work.

We assume the allocation variables are sampled i.i.d. from a static distribution. When the true variable distribution is unknown, we can maintain a belief over the distribution and sample from the posterior to compute the robust recourse. Furthermore, this process can be adapted to consider changes in the underlying distribution over time. We also assume a constant population size, but that could be easily changed.

Our methods and definitions assume that the user’s requested resource remains unchanged. Yet, it could be reasonable for an individual to change their requested resource, for example in exchange for increasing their probability of receiving it. A CE which includes change of preferences is left for future work.

*Future Work.* Further research is required to explore the generalisation of our results to settings in which the assumptions mentioned above are relaxed. In addition, more complex problems within the scope of allocation problems, such as matching problems, could be addressed in the future, as well as the open problems mentioned in Sect. 7.

*Societal Impact.* Lastly, we note that the use of recourse is intended to provide users with more control over aspects in their lives controlled by automated decisions. Ideally, we would like the DMs to be held responsible for their provided recourse, such that the implementation of a recourse would guarantee access to the resource in the future. We choose to provide a probabilistic recourse in order to grant users more control over the robustness-cost trade-off. Yet, when the recourse takes a probabilistic nature, DMs might distance themselves from the responsibility for the robustness of the recourse.

**Acknowledgments.** This work was supported by the Research Council of Norway under project number 302203.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## Appendix

In Sect. 5 we claim the following:

**Lemma 1.** *The minimal utility-CE  $v'_i$  for individual  $i$  under an optimal 0-1 knapsack policy is*

$$v'_i = U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i)).$$

*Proof.* We prove this claim by contradiction. Suppose not, and let us assume that there exists  $\bar{v}_i < v'_i$  such that for  $\bar{v}_i$ , individual  $i$  is included in the optimal set. We assume that  $w_i \leq r$  (otherwise individual  $i$  could never be included in the allocation). As the order of the individuals does not change the optimal allocation, let us assume w.l.o.g. that individual  $i$  is the last individual inserted into table  $V$  ( $i = n$ ). Thus, when filling the cell  $V[n, r]$  we choose whether to include individual  $n$  or not:  $V[n, r] = \max(V[n - 1, W], V[n - 1, r - w[n]] + \bar{v}_n)$ . By assuming that the individual is included, we get that

$$\begin{aligned} V[n - 1, r] &\leq V[n - 1, r - w_n] + \bar{v}_n \\ \Leftrightarrow U(Y^*([n], r)) &\leq U(Y^*([n - 1], r - w_n)) + \bar{v}_n \\ \Leftrightarrow U(Y^*([n], r)) - U(Y^*([n - 1], r - w_n)) &\leq \bar{v}_n \\ \Leftrightarrow U(Y^*([n], r)) - U(Y^*([n] \setminus i, r - w_i)) &\leq \bar{v}_n \\ \Leftrightarrow v'_n &\leq \bar{v}_n \end{aligned}$$

Which contradicts our assumption of  $\bar{v}_i < v'_i$ . Note that  $U(Y^*([n], r)) = V[n - 1, r]$  as the individual was not originally included in the allocation. In practice, we add  $\epsilon > 0$  to the utility-CE in order to avoid ties.

Notes:

1. Another approach for generating CE for the 0–1 knapsack problem was previously proposed [28]. Yet, our approach allows efficient calculation of multiple CE for different budgets by filling the table  $V$  (both with and without individual  $i$ ) for a maximal budget  $r_{max}$ , which then provides all solutions for all  $r \in [r_{max}]$ .
2. In some cases, the required utility-CE would entail a prediction-CE that is greater than 1, which is impossible. Thus, in those cases, the applicant would learn that given the current allocation variables, there is nothing they could have changed in order to have received the requested loan. Nevertheless, we only consider the option to change user features excluding the requested credit, assuming the requested credit cannot be changed.

## References

1. Database for statistics on higher education (database for statistikk om høyere utdanning). <https://dbh.hkdir.no/>. Accessed 15 Apr 2024
2. Federal student aid in the U.S. department of education website. <https://www2.ed.gov/about/offices/list/rsa/index.html?exp=6>. Accessed 15 Apr 2024
3. Ali, G., Al-Obeidat, F., Tubaishat, A., Zia, T., Ilyas, M., Rocha, A.: Counterfactual explanation of Bayesian model uncertainty. *Neural Comput. Appl.* 1–8 (2021)
4. Assi, M., Haraty, R.A.: A survey of the knapsack problem. In: 2018 International Arab Conference on Information Technology (ACIT), pp. 1–6. IEEE (2018)
5. Black, E., Wang, Z., Fredrikson, M., Datta, A.: Consistent counterfactuals for deep models. arXiv preprint [arXiv:2110.03109](https://arxiv.org/abs/2110.03109) (2021)
6. Bui, N., Nguyen, D., Nguyen, V.A.: Counterfactual plans under distributional ambiguity. arXiv preprint [arXiv:2201.12487](https://arxiv.org/abs/2201.12487) (2022)
7. Carlevaro, A., Lenatti, M., Paglialonga, A., Mongelli, M.: Multi-class counterfactual explanations using support vector data description (2023)
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: *Introduction to Algorithms*. MIT Press, Cambridge (2022)
9. Čyras, K., Letsios, D., Misener, R., Toni, F.: Argumentation for explainable scheduling. In: Proceedings of the AAI Conference on Artificial Intelligence, vol. 33, pp. 2752–2759 (2019)
10. Delaney, E., Greene, D., Keane, M.T.: Uncertainty estimation and out-of-distribution detection for counterfactual explanations: pitfalls and solutions. arXiv preprint [arXiv:2107.09734](https://arxiv.org/abs/2107.09734) (2021)
11. Dominguez-Olmedo, R., Karimi, A.H., Schölkopf, B.: On the adversarial robustness of causal algorithmic recourse. In: International Conference on Machine Learning, pp. 5324–5342. PMLR (2022)
12. Dong, H., Nakayama, M.K.: A tutorial on quantile estimation via Monte Carlo. In: Tuffin, B., L'Ecuyer, P. (eds.) MCQMC 2018. SPMS, vol. 324, pp. 3–30. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-43465-6\\_1](https://doi.org/10.1007/978-3-030-43465-6_1)






13. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
14. Dutta, S., Long, J., Mishra, S., Tilli, C., Magazzeni, D.: Robust counterfactual explanations for tree-based ensembles. In: International Conference on Machine Learning, pp. 5742–5756. PMLR (2022)
15. Eifler, R., Frank, J., Hoffmann, J.: Explaining soft-goal conflicts through constraint relaxations. In: ICAPS 2022 Workshop on Explainable AI Planning (2022)
16. Ferrario, A., Loi, M.: The robustness of counterfactual explanations over time. *IEEE Access* (2022)
17. Fonseca, J., Bell, A., Abrate, C., Bonchi, F., Stoyanovich, J.: Setting the right expectations: algorithmic recourse over time. In: Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization, pp. 1–11 (2023)
18. Goyal, A., Kaur, R.: A survey on ensemble model for loan prediction. *Int. J. Eng. Trends Appl. (IJETA)* **3**(1), 32–37 (2016)
19. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. *Data Min. Knowl. Discov.* 1–55 (2022)
20. Guo, H., Jia, F., Chen, J., Squicciarini, A., Yadav, A.: Rocoursenet: distributionally robust training of a prediction aware recourse model. *arXiv preprint arXiv:2206.00700* (2022)
21. Hada, S.S., Carreira-Perpiñán, M.Á.: Exploring counterfactual explanations for classification and regression trees. In: Kamp, M., et al. (eds.) *ECML PKDD 2021. CCIS*, vol. 1524, pp. 489–504. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-93736-2\\_37](https://doi.org/10.1007/978-3-030-93736-2_37)
22. Hakimov, R., Kübler, D.: Experiments on centralized school choice and college admissions: a survey. *Exp. Econ.* **24**, 434–488 (2021)
23. Kaddour, J., Lynch, A., Liu, Q., Kusner, M.J., Silva, R.: Causal machine learning: a survey and open problems. *arXiv preprint arXiv:2206.15475* (2022)
24. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: contrastive explanations and consequential recommendations. *ACM Comput. Surv. (CSUR)* (2021)
25. Karimi, A.H., Von Kügelgen, J., Schölkopf, B., Valera, I.: Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 265–277 (2020)
26. Kleine Buening, T., Segal, M., Basu, D., George, A.M., Dimitrakakis, C.: On meritocracy in optimal set selection. In: *Equity and Access in Algorithms, Mechanisms, and Optimization*, pp. 1–14 (2022)
27. König, G., Freiesleben, T., Grosse-Wentrup, M.: A causal perspective on meaningful and robust algorithmic recourse. *arXiv preprint arXiv:2107.07853* (2021)
28. Korikov, A., Beck, J.C.: Counterfactual explanations via inverse constraint programming. In: *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2021)
29. Lerouge, M., Gicquel, C., Mousseau, V., Ouerdane, W.: Counterfactual explanations for workforce scheduling and routing problems. In: *12th International Conference on Operations Research and Enterprise Systems*, pp. 50–61. SCITEPRESS-Science and Technology Publications (2023)
30. Ley, D., Bhatt, U., Weller, A.: Diverse, global and amortised counterfactual explanations for uncertainty estimates. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 7390–7398 (2022)

31. Ludwig, J., Kalton, A., Stottler, R.: Explaining complex scheduling decisions. In: *IUI Workshops* (2018)
32. Lux, T., Pittman, R., Shende, M., Shende, A.: Applications of supervised learning techniques on undergraduate admissions data. In: *Proceedings of the ACM International Conference on Computing Frontiers*, pp. 412–417 (2016)
33. Madiega, T.: *Artificial intelligence act*. European Parliament: European Parliamentary Research Service (2021)
34. Martello, S., Toth, P.: *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, Hoboken (1990)
35. Mishra, S., Dutta, S., Long, J., Magazzeni, D.: A survey on the robustness of feature importance and counterfactual explanations. *arXiv preprint [arXiv:2111.00358](https://arxiv.org/abs/2111.00358)* (2021)
36. Mochaourab, R., Sinha, S., Greenstein, S., Papapetrou, P.: Robust counterfactual explanations for privacy-preserving SVM. In: *International Conference on Machine Learning (ICML 2021), Workshop on Socially Responsible Machine Learning* (2021)
37. Nguyen, D., Bui, N., Nguyen, V.A.: Distributionally robust recourse action. *arXiv preprint [arXiv:2302.11211](https://arxiv.org/abs/2302.11211)* (2023)
38. Nguyen, T.D.H., Bui, N., Nguyen, D., Yue, M.C., Nguyen, V.A.: Robust Bayesian recourse. In: *Uncertainty in Artificial Intelligence*, pp. 1498–1508. PMLR (2022)
39. Pawelczyk, M., Bielawski, S., Van den Heuvel, J., Richter, T., Kasneci, G.: Carla: a Python library to benchmark algorithmic recourse and counterfactual explanation algorithms. *arXiv preprint [arXiv:2108.00783](https://arxiv.org/abs/2108.00783)* (2021)
40. Pawelczyk, M., Broelemann, K., Kasneci, G.: On counterfactual explanations under predictive multiplicity. In: *Conference on Uncertainty in Artificial Intelligence*, pp. 809–818. PMLR (2020)
41. Pawelczyk, M., Datta, T., Van den Heuvel, J., Kasneci, G., Lakkaraju, H.: Probabilistically robust recourse: navigating the trade-offs between costs and robustness in algorithmic recourse. In: *The Eleventh International Conference on Learning Representations* (2022)
42. Rawal, K., Kamar, E., Lakkaraju, H.: Algorithmic recourse in the wild: understanding the impact of data and model shifts. *arXiv preprint [arXiv:2012.11788](https://arxiv.org/abs/2012.11788)* (2020)
43. Schut, L., et al.: Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In: *International Conference on Artificial Intelligence and Statistics*, pp. 1756–1764. PMLR (2021)
44. Spooner, T., Dervovic, D., Long, J., Shepard, J., Chen, J., Magazzeni, D.: Counterfactual explanations for arbitrary regression models. *arXiv preprint [arXiv:2106.15212](https://arxiv.org/abs/2106.15212)* (2021)
45. Swist, T., Gulson, K.N.: *School Choice Algorithms: Data Infrastructures, Automation, and Inequality*, pp. 1–19. *Postdigital Science and Education* (2022)
46. Upadhyay, S., Joshi, S., Lakkaraju, H.: Towards robust and reliable algorithmic recourse. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 16926–16937 (2021)
47. Virgolin, M., Fracaros, S.: On the robustness of counterfactual explanations to adverse perturbations. *arXiv preprint [arXiv:2201.09051](https://arxiv.org/abs/2201.09051)* (2022)
48. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL Tech.* **31**, 841 (2017)
49. Xiang, J.: Ai in lending. *The AI Book: The Artificial Intelligence Handbook for Investors, Entrepreneurs and FinTech Visionaries*, pp. 34–38 (2020)



# Towards Non-adversarial Algorithmic Recourse

Tobias Leemann<sup>1,2</sup>(✉) , Martin Pawelczyk<sup>3</sup> , Bardh Prenkaj<sup>2</sup> ,  
and Gjergji Kasneci<sup>2</sup> 

<sup>1</sup> University of Tübingen, Tübingen, Germany  
tobias.leemann@uni-tuebingen.de

<sup>2</sup> Technical University of Munich, Munich, Germany

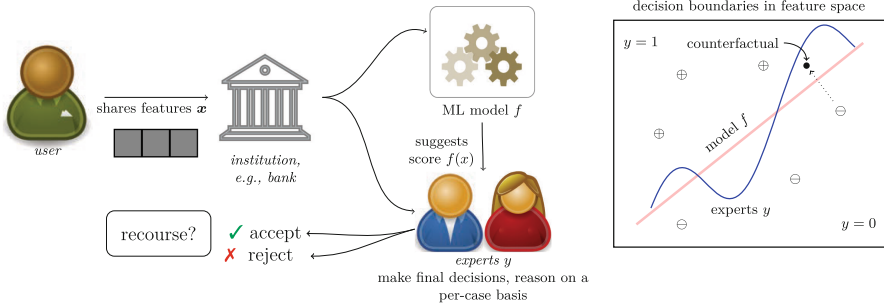
<sup>3</sup> Harvard University, Cambridge, MA, USA

**Abstract.** The streams of research on adversarial examples and counterfactual explanations have largely been growing independently. This has led to several recent works trying to elucidate their similarities and differences. Most prominently, it has been argued that adversarial examples, as opposed to counterfactual explanations, have a unique characteristic in that they lead to a misclassification compared to the ground truth. However, the computational goals and methodologies employed in existing counterfactual explanation and adversarial example generation methods often lack alignment with this requirement. Using formal definitions of adversarial examples and counterfactual explanations, we introduce non-adversarial algorithmic recourse and outline why in high-stakes situations, it is imperative to obtain counterfactual explanations that do not exhibit adversarial characteristics. We subsequently investigate how different components in the objective functions, e.g., the machine learning model or cost function used to measure distance, determine whether the outcome can be considered an adversarial example or not. Our experiments on common datasets highlight that these design choices are often more critical in deciding whether recourse is non-adversarial than whether recourse or attack algorithms are used. Furthermore, we show that choosing a robust and accurate machine learning model results in less adversarial recourse desired in practice.

**Keywords:** Counterfactuals · Adversarials · Algorithmic Recourse

## 1 Introduction

A continuous stream of predominantly independent research in the fields of adversarial examples [26, 58] and counterfactual explanations [47, 61, 63, 65] has sparked an ongoing scholarly discourse on their similarities and differences [23, 46]. While adversarial examples originate from the security literature, characterizing instances capable of deceiving machine-learned classifiers into erroneous decisions, algorithmic recourse has its roots in the trustworthy machine-learning



**Fig. 1. Overview of the realistic decision-making scenario considered in this work.** We consider the relevant case where an institution, e.g., a bank, deploys a machine learning model to support decision-making overseen by human experts that make final, case-based decisions based on the model’s score (left). In such a setting, constructing recourse only based on the scoring model  $f$  may lead to unreliable recourse because the experts’ final  $y$  decision is based on further restrictions, thereby representing a shifted decision boundary (right).

literature. Algorithmic recourse is primarily concerned with providing actionable recommendations for changes that would lead to a different, more favorable outcome for the end user (e.g., changing a loan decision from rejection to acceptance). Despite the apparent differences in goals and associated semantics between adversarial examples and recourse, scholars have observed a strikingly similar algorithmic foundation underpinning these two domains [23, 46].

The current debate surrounding the potential distinctions between these two concepts remains somewhat ambiguous. To provide greater context and significance to this discourse, we establish a tangible connection to a real-world application where the differentiation between counterfactual and adversarial examples becomes intuitive and indispensable. To this end, we slightly modify the established recourse problem in the context of loan assignments [62]. Unlike previous work, which assumes that a machine learning system solely determines loan assignments, we argue that this perspective oversimplifies the real world. Article 22 of the European Union’s General Data Protection Regulation (GDPR) [25], which asserts the right of the data subject “not to be subject to a decision based solely on automated processing which produces legal effects concerning him or her”, thereby suggesting that automated models alone cannot make legally binding decisions. Consequently, we consider a more practical scenario where algorithmic decisions are subject to scrutiny by a human expert panel. This expert panel holds the authority to issue a final, case-specific decision and can override the model’s recommendation. This refined setup is illustrated in Fig. 1.

Complementarily, the GDPR grants individuals who receive an adverse decision the right to receive “meaningful information about the logic involved” which, in a broader context, can be interpreted as the right to “recourse” [64]. When the model exclusively determines decisions, it is evident that recourse can be directly computed from the model itself. However, in the more realistic scenario

considered in this work, where human experts play a role in the final decisions, the model’s output does not fully encapsulate the ultimate decision. This raises the question of appropriate recourse design in such a scenario and how to reconcile these two GDPR principles – the right to receive meaningful recourse and the prohibition of fully automated decision-making.

Under the premise that the model has been mainly distilled from past decisions of the experts, we consider the experts as an imperfect oracle providing ground truth labels,<sup>1</sup> whereas the model returns an imperfect approximation of these labels. While the humans decide on a per-case basis, it is hard to directly ask them for specific thresholds, as the interplay of the features quickly makes the task intractable. Therefore, we are interested in computing counterfactual explanations that do not only change the model’s prediction but also flip the *true* labels. This perspective aligns with the argument made by Freiesleben [23] that a distinctive feature of adversarial examples, as opposed to counterfactual explanations, is their tendency to be misclassified regarding their true labels. Since counterfactual explanations should also change the true label in this case, this gives rise to the term “non-adversarial algorithmic recourse”, i.e., *counterfactual explanations that come with both a change in the model’s prediction and a changed ground-truth label*.

Unlike prior work taking a merely definitional view, this work additionally contributes to implementing non-adversarial algorithmic recourse in practical scenarios. In summary, we propose the following contributions:

- **Introduction of a novel recourse problem:** We introduce a novel recourse problem that addresses real-world decision systems wherein human experts play a pivotal role in making case-based decisions, while also considering input from a machine learning model.
- **Proposing non-adversarial recourse as a solution to the realistic recourse problem:** We consider prior work’s [23] distinction of adversarial examples and counterfactual explanations and suggest a novel formal definition of *non-adversarial algorithmic recourse*, proving a conceptual bridge between the academic discourse on distinguishing adversarial examples from counterfactual explanations and practical decision-making.
- **Promoting non-adversarial recourse theoretically:** After a theoretical analysis of the problem, we derive optimal cost functions that encourage non-adversarial recourse. Our analysis underscores how feature attributions can be leveraged to identify task-relevant features contributing to less “adversarial” recourse.
- **Empirical Insights:** We are the first to consider several other key components practitioners can manipulate to foster non-adversarial algorithmic recourse. These include improving the robustness and accuracy of the machine learning model and the recourse algorithms. In contrast to parts of the literature which argue that cost functions are central, we empirically find that changes in the model are often more significant than the cost function.

---

<sup>1</sup> The oracle is imperfect as some labels are generated from “defaults”, i.e., false positives of expert decisions.

## 2 Related Work

**Human-Assisted Decisions.** In crucial situations, societies rely on human experts for decisions. However, delays and quality issues due to a shortage of experts and a high volume of decisions, e.g., long waits for medical diagnoses, have sparked a debate on when automated or human decision-making should be deployed. A stream of prior works [10, 51, 59] argue that ML models should make decisions in high-stake domains where they have matched or surpassed the average of human performance. Nevertheless, their decisions can still be worse than those of human experts [53] in some cases. In this direction, works such as [13, 14, 43] propose to optimize ML models to operate under different automation levels: i.e., take decisions on a fraction of the given instances and leave the rest to human experts. In line with other works [21], we argue that the human factor in the loop in a human-AI partnership cannot be neglected when considering the application of AI on real-world problems [1, 27]. This position is also cemented in common data protection laws such as the EU’s GDPR [25], which grants individuals a right to object fully automated decision-making. For GDPR-compliant decision-making, human oversight can thus be considered essential. Unlike previous works, we explicitly model a human expert panel in the decision-making setup as depicted in Fig. 1, which makes the generation of reliable recourse much more challenging.

**Counterfactual Explanations.** There is an established literature on the computation of counterfactual explanations [2, 8, 34, 37, 41, 50, 54, 61, 65] in variegated domains. According to Guidotti et al. [28], given a classifier  $f$  that outputs a decision  $f(\mathbf{x}) = y$  for an instance  $\mathbf{x}$ , a counterfactual explanation of  $\mathbf{x}$  is an instance  $\mathbf{x}'$  such that  $f(\mathbf{x}') \neq y$ , and the difference between  $\mathbf{x}$  and  $\mathbf{x}'$  is minimal. Current research streams include the robustness of counterfactual explanations [18, 48, 60] and the compatibility with other GPDR principles [49]. We briefly review this research field in the following but point the reader to recent surveys [28, 52, 63] for a comprehensive overview. Mothilal et al. [41] solve an optimization problem with various constraints, among which user-specified ones for (im)mutable features, to ensure feasibility and diversity when producing a set of counterfactuals for a given input. Carreira-Perpiñán and Hada [8] propose CEODT specifically designed for classification trees, including Oblique Decision Trees (ODTs) [29]. Because the counterfactual optimization problem for ODTs is non-convex, nonlinear, and non-differentiable, CEODT computes an exact solution via the optimization problem within the region represented by each leaf and finally picks the leaf with the best solution. Lastly, Ustun et al. [61] were among the first authors to address the problem of actionability in counterfactual explanations (i.e., recourse). Their method constrains the generated counterfactuals such that manipulations do not change immutable features. Overall, we note that previous literature relies on the common assumption that an automated model acts as a sole decision-maker, which might not be realistic in practical scenarios.

**Adversarial Examples.** Following Szegedy et al. [58], adversarial examples are instances that contain subtle perturbations – usually via adding small amounts of noise – to instances in the training set. These “new” instances, when fed to an underlying ML model, produce an erroneous output with high confidence. It is possible to build an adversarial example  $\mathbf{x}'$  which is indistinguishable<sup>2</sup> from  $\mathbf{x}$  but is classified incorrectly, i.e.,  $f(\mathbf{x}') \neq y$ . Successfully generating such examples gives rise to *adversarial attacks* [5, 26, 39], which can have potentially lethal consequences (e.g., in biosecurity and biotechnology [45], autonomous driving [20, 66], and power grid blackouts [24]). Several methods have been proposed in the literature to generate adversarial examples assuming varying degrees of knowledge/access of the model, training data, and methods for injecting perturbations. Goodfellow et al. [26], Kurakin et al. [35], and Moosavi et al., [40] propose methods with gradient and data access to find the minimum  $\ell_\infty$ -norm (and  $\ell_2$ -norm respectively) perturbations. With only assuming query access to the target classifier, the authors in [11, 44, 57] design adversarial examples to tightly control sparsity. We refer the reader to a well-established survey for a comprehensive overview of adversarial examples [3].

**Linking Counterfactuals and Adversarial Examples.** Strikingly, counterfactual explanations and adversarial examples have conceptual connections and a similar formulation [6, 23, 65] (see also Sect. 3). Freiesleben [22] highlights conceptual differences in aims, formulation, and use-cases between both sub-fields and suggests that the distinctive formal feature of adversarial examples lies in their misclassification concerning the ground truth. Concurrently, there have been proposals to align recourse with a ground truth. König et al. [34] proposes improvement-focused causal recourse, designed to change the true targets instead of the predictions but relies on causal information. Laugel et al. [36] proposes the notion of “justified recourse” that should be close to a correctly classified instance. On the other hand, Browne et al. [6] focus on deep networks and attribute conceptual differences to the interpretation of semantics in the hidden layers of deep networks. Pawelczyk et al. [46] formalize the similarities between popular counterfactual explanations and adversarial example generation methods identifying conditions when they are equivalent. Trying to disentangle and reconcile the various distinctions made in prior works, we provide formal definitions in the next section. Besides König et al. [34], who rely on causal information, there have been few attempts to implement recourse that follows the ground truth. In this work, we provide valuable insights on how to implement non-adversarial recourse in practical decision-making.

### 3 Preliminaries

We first formalize the general problem considered in this work, before we provide the relevant distinctions between adversarial examples and counterfactual explanations.

<sup>2</sup> We invite the reader to think about images in this context, as described in [26]. Additionally, some works analyze perturbations – e.g., adversarial patches – that are perceptually distinguishable by humans but fool the classifier  $f$  [16, 19, 67].

### 3.1 The General Problem

Both recourse and adversarial methods consider a fixed machine learning model  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , where  $\mathcal{X} \subseteq \mathbb{R}^k$ . We usually consider the binary classification problem, where the label is binary, i.e.,  $\mathcal{Y} = \{0, 1\}$  or a numerical score,  $\mathcal{Y} = \mathbb{R}$ .

We suppose there is another function  $y : \mathcal{X} \rightarrow \mathcal{Y}$  that assigns the true labels and represents the human experts in our introductory example. In practice, it is impossible to perfectly learn this function with a model, for instance due to insufficient data coverage or additional circumstances that can be taken into considerations only by the human experts. However, it is possible to query  $y$  a limited number of times, as it is possible to present the experts with an example and ask for their decision. We model the expert predictions  $y$  in the scenario outlined as

$$y(\mathbf{x}) = g(\mathbf{x}, f(\mathbf{x})), \quad (1)$$

where  $g$  models the human expert committee that can recalibrate the score in light of the features in their entirety. However, we suppose that we usually have  $y(\mathbf{x}) \approx f(\mathbf{x})$ , i.e., the original score is only lightly adapted through  $g$ . In practice, models are fitted on a limited number of potential observations of the experts' decisions.

As noted before [46], the classical optimization problem solved by both practical adversarial and counterfactual methods for a model  $f : \mathcal{X} \rightarrow \mathcal{Y}$  a factual input  $\mathbf{x} \in \mathcal{X}$ , and a target label  $y_t \in \mathcal{Y}$  is mathematically similar and can usually be formalized as a special case of the following general optimization problem [23]:

$$\operatorname{argmin}_{\mathbf{x}' \in \mathcal{X}} d_1(\mathbf{x}, \mathbf{x}') + \lambda d_2(f(\mathbf{x}'), y_t), \quad (2)$$

where  $d_1 : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  is a distance metric defined on the input space,  $d_2 : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  is a metric on the output space and  $\lambda \in \mathbb{R}_{\geq 0}$  is a non-negative trade-off parameter. Intuitively, the solution to this problem returns instances, that are close to the factual  $\mathbf{x}$  and have a label that is close (or corresponds exactly) to the target label  $y_t$ .

### 3.2 Algorithms for Computing Counterfactual Explanations and Adversarial Examples

We briefly review the most common strategies to compute counterfactuals and adversarial examples in practice.

*Score CounterFactual Explanations (SCFE).* For a given classifier  $f(h(\mathbf{x}))$  that relies on logit scores  $h(\mathbf{x})$  and a distance function  $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_+$ , Wachter et al. [65] formulate the problem of finding a counterfactual  $\mathbf{x}'$  for  $\mathbf{x}$  as:

$$\operatorname{argmin}_{\mathbf{x}'} (h(\mathbf{x}') - s)^2 + \lambda d(\mathbf{x}, \mathbf{x}'), \quad (3)$$

where  $s$  is the target score for  $\mathbf{x}$ . The problem is solved for different values of  $\lambda$  until  $f(\mathbf{x}') = s$ . More specifically, to arrive at a counterfactual probability of



0.5, the target score for  $h(\mathbf{x})$  for a sigmoid function is  $s = 0$ . Using the inverse logit transform  $h(\mathbf{x}) = \text{invlogit}(f(\mathbf{x}))$ , the first part of the objective can be interpreted as a particular instantiation of  $d_2$  in Eq. (2) when  $\mathcal{Y}$  is taken to be the interval  $[0, 1]$ .

*Diverse Counterfactual Explanations (DiCE)*. As different users may have different preferences (i.e., it might be easier for them to change one feature or another), DiCE [42] generates multiple counterfactuals. An additional loss term is added to the objective in Eq. (3) to encourage diversity. As users will only choose one counterfactual in practice, we usually consider a randomly selected instance of the discovered recourse candidate for evaluation as in [49].

*Actionable Recourse (AR)*. The actionable recourse (AR) method by Ustun et al. [61] sets up the following optimization problem:

$$\min \text{cost}(\boldsymbol{\delta}; \mathbf{x}) \tag{4}$$

$$\text{s.t. } f(\mathbf{x} + \boldsymbol{\delta}) = +1, \boldsymbol{\delta} \in \mathcal{A}(\mathbf{x}), \tag{5}$$

where  $+1$  corresponds to the positive outcome and  $\mathcal{A}$  is an action set  $\mathcal{A}(\mathbf{x})$ . This problem corresponds to Eq. (2) when using a distance function  $d_1$  that returns  $\infty$  once  $\boldsymbol{\delta} \notin \mathcal{A}(\mathbf{x})$  and the cost function otherwise. The distance  $d_2$  can be interpreted as the Dirac-distance, that is  $\infty$  once  $f(\mathbf{x} + \boldsymbol{\delta}) \neq 1$ . They solve the problem using mixed integer linear programming (MIP) for linear models.

Like counterfactual explanations, most adversarial example methods also solve a constrained optimization problem to find perturbations in the input manifold that cause models to misclassify.

*C&W Attack*. For a given input  $\mathbf{x}$  and classifier  $f$ , Carlini and Wagner [7] formulate the problem of finding an adversarial example  $\mathbf{x}' = \mathbf{x} + \boldsymbol{\delta}$  such that  $f(\mathbf{x}') \neq f(\mathbf{x})$  as:

$$\underset{\mathbf{x}' \in \mathcal{X}}{\text{argmin}} c \cdot \ell(\mathbf{x}') + d(\mathbf{x}, \mathbf{x}') \quad \text{s.t. } \mathbf{x}' \in [0, 1]^d, \tag{6}$$

where  $c > 0$  is a suitably chosen hyperparameter, and  $\ell(\cdot)$  is an objective function on the adversarial  $\mathbf{x}'$  s.t.  $f(\mathbf{x}') = y_t$  iff  $\ell(\mathbf{x}') \leq 0$  with  $y_t$  being a target class. The authors choose  $d(\mathbf{x}, \mathbf{x}')$  to be the  $l_p$  norm of  $\boldsymbol{\delta}$ , i.e., minimizing the  $p$ -norm of  $\boldsymbol{\delta}$  is equivalent to minimizing  $d(\mathbf{x}, \mathbf{x}')$ .

*DeepFool Attack*. For a given instance  $\mathbf{x}$ , DeepFool [40] perturbs it by adding small perturbation  $\boldsymbol{\delta}_{\text{DF}}$  at each iteration. The minimal perturbation to change the classification model’s prediction is the solution to the following objective:

$$\boldsymbol{\delta}_{\text{DF}}^*(\mathbf{x}) \in \underset{\boldsymbol{\delta} \text{ s.t. } \mathbf{x} + \boldsymbol{\delta} \in \mathcal{X}}{\text{argmin}} \|\boldsymbol{\delta}\|_2 \quad \text{s.t. } \text{sign}(f(\mathbf{x} + \boldsymbol{\delta})) \neq \text{sign}(f(\mathbf{x})) \tag{7}$$

*PGD Attack.* PGD [38] is a first-order optimization technique. In the context of adversarial examples, it is usually used to maximize<sup>3</sup>, the objective for a specific factual  $\mathbf{x}$ . This is because the objective is typically chosen to be the cross-entropy loss  $\mathcal{L}$ :

$$\underset{\delta \text{ s.t. } \mathbf{x} + \delta \in \mathcal{C}}{\operatorname{argmax}} \mathcal{L}(f(\mathbf{x} + \delta), f(\mathbf{x})) \quad (8)$$

where  $\delta$  is the adversarial perturbation to be added to the factual  $\mathbf{x}$ . PGD maximizes the objective by taking steps along the gradient’s direction. After each update, the current perturbation  $\delta^t$  is projected onto a set of constraints  $\mathcal{C}$ . For instance, the adversarial examples are all constrained to a ball of size  $\epsilon$  around  $\mathbf{x}$ . We argue that the projection of the adversarials  $\mathbf{x}' = \mathbf{x} + \delta$  into an  $\epsilon$ -ball could be interpreted as a  $d_1$  distance function in Eq. (2), that returns an infinite cost value for actions outside the  $\epsilon$ -ball. Meanwhile, the cross-entropy loss subsumes the role of the  $d_2$ -cost function. Therefore, Eq. (8) can be considered as a special case of Eq. (2) transformed into a maximization problem.

We invite the reader to notice that the approaches presented above – whether adversarial attacks or counterfactual explanation methods – solve the same objective. In fact, they can be interpreted as heuristics to optimizing an instance of the formulation in Eq. (2), although pertaining to different “semantics” as argued in [55]. However, a precise distinction between counterfactual explanations and adversarial attack algorithms cannot be derived from their implementations. To this end, we investigate precise definitions for both problems in the next section.

## 4 Definitions

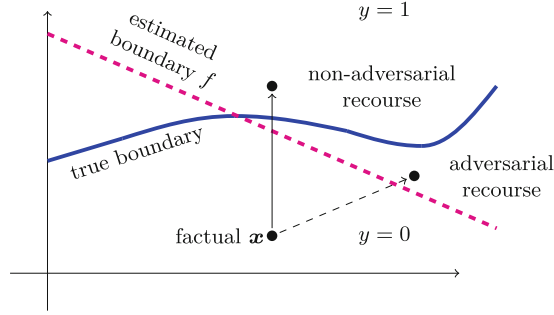
### 4.1 Formalizing Adversarials and Counterfactuals

We take the definition of an adversarial example by Freiesleben [23] as a starting point. It intuitively describes the properties that such instances should have. In other words, they should be close to the original instance, change the model’s predictions and be misclassified. Most notably and in contrast to other works, Freiesleben argues that the misclassification is a distinctive property of adversarial examples. This distinctive property has also previously been mentioned in other works on adversarial examples more or less directly [56], giving rise to the following definition:

**Definition 1 (Adversarial Example [23]).** *An instance  $\mathbf{x}' \in \mathcal{X}$  is an **adversarial example** for a factual  $\mathbf{x} \in \mathcal{X}$  and a classifier  $f : \mathcal{X} \rightarrow \mathcal{Y}$  if the following conditions hold:*

- (1)  $\mathbf{x}'$  is close to  $\mathbf{x}$ , i.e.,  $d_1(\mathbf{x}, \mathbf{x}') < \epsilon$ ;
- (2) the classifier output is changed, i.e.,  $f(\mathbf{x}) \neq f(\mathbf{x}')$ ;
- (3)  $\mathbf{x}'$  is misclassified, i.e.,  $y(\mathbf{x}') \neq f(\mathbf{x}')$ .

<sup>3</sup> Thus, projected gradient ascent is often the more appropriate description for this attack. However, we will follow common practice and refer to the algorithm as PGD.



**Fig. 2. Visualizing our definitions.** The space of valid recourse for a factual  $\mathbf{x}$  changes crosses the classifier  $f$ 's estimated decision-boundary (pink). The experts combine it with their expertise and restrictions into a latent decision boundary (blue). However, some recourse might not change the true label and is therefore considered adversarial (dashed arrow). The challenge is to obtain recourse that convinces the human experts. To this end, we are interested in finding the directions that lead to *non-adversarial recourse* (solid arrow). (Color figure online)

We also consider the definition of recourse (or equivalently, counterfactual examples) by Freiesleben [23], which states that recourse  $\mathbf{x}'$  changes the classification label and is the closest point to the factual that does so. We propose a slight relaxation. In particular, we argue that even points that are not closest to the factual are still valid (though possibly suboptimal) recourse.

**Definition 2 (Recourse).** An instance  $\mathbf{x}' \in \mathcal{X}$  is **recourse** for a factual  $\mathbf{x} \in \mathcal{X}$  with  $f(\mathbf{x}) \neq y_t$ , a classifier  $f : \mathcal{X} \rightarrow \mathcal{Y}$ , and a target label  $y_t \in \mathcal{Y}$  if the following conditions hold:

- (1)  $\mathbf{x}'$  is close to  $\mathbf{x}$ , i.e.,  $d_1(\mathbf{x}, \mathbf{x}') < \epsilon$ ;
- (2) the classifier output is changed to the target, i.e.,  $f(\mathbf{x}') = y_t \neq f(\mathbf{x})$ .

These general definitions cover most definitions explicitly or implicitly used in the literature (see [23] for details). We immediately see that our definition of recourse abandons the final constraint in the definition of adversarial examples, that  $\mathbf{x}'$  should be misclassified. For the two-class problem where  $y_t$  is just the opposite class of  $f(\mathbf{x})$ , according to these definitions, (a) all adversarial examples are recourse<sup>4</sup>, and (b) there is a distinct (though potentially empty) subset of examples, that are recourse, but are not adversarials, as visualized in Fig. 2.

## 4.2 Non-adversarial Algorithmic Recourse

In this work, we place our attention on the examples present in this subset, that are recourse but not adversarial examples. We thus refer to them as *non-adversarial recourse* and introduce a novel definition for this class of instances:

<sup>4</sup> For multi-class problems, all adversarials which are classified as  $y_t$  are recourse.

**Definition 3 (Non-adversarial Recourse).** *An instance  $\mathbf{x}' \in \mathcal{X}$  is **non-adversarial recourse** for a factual  $\mathbf{x} \in \mathcal{X}$  with  $f(\mathbf{x}) \neq y_t$ , target label  $y \in \mathcal{Y}$ , and a classifier  $f : \mathcal{X} \rightarrow \mathcal{Y}$  if the following conditions hold:*

- (1)  $\mathbf{x}'$  is close to  $\mathbf{x}$ , i.e.,  $d_1(\mathbf{x}, \mathbf{x}') < \epsilon$ ;
- (2) the classifier output is changed, i.e.,  $f(\mathbf{x}') = y_t \neq f(\mathbf{x})$ ;
- (3)  $\mathbf{x}'$  is not misclassified,  $f(\mathbf{x}') = y(\mathbf{x}')$ .

We observe that in the considered realistic decision-making scenario, we desire recourse that convinces the human experts, i.e., also changes the true label  $y$ . These correspond exactly to the instances described in the definition of non-adversarial recourse.

## 5 Theoretical Analysis

As outlined in Fig. 2, we are interested in finding changes, or at least directions of change, that lead to non-adversarial recourse efficiently. As it is impossible to precisely model the ground truth  $y$  in our setup (otherwise, there would be no need for an additional human expert), this is challenging in practice. However, we can use some guidance from the model, which approximates the ground truth, to find non-adversarial recourse.

### 5.1 Summarizing Influential Factors for Less Adversarial Recourse

We first take a step back and consider the general formulation of the problem given in Eq. (2). We observe that the problem formulation features three potential factors of influence (the model  $f$ , the distance functions  $d_1$  and  $d_2$ ) and a hyperparameter (the choice of optimization algorithm) that can be changed in practice to arrive at less adversarial recourse. If we follow the usual binary classification setup where we chose  $\lambda > 0$  and  $d_2$  to be the Dirac distance that amounts to infinity if the target label is not met, i.e.,  $d_2(f(\mathbf{x}'), y_t) = \delta_{f(\mathbf{x}')=y_t}$ , there are three remaining factors of influence, that we tackle in this study with different outcomes (discussed in more detail in Sect. 6):

*Machine Learning Model.* Considering the model  $f$  first, we note that there is a simple theoretical solution to non-adversarial recourse: If the model would exactly match the theoretical ground truth, i.e.,  $f \equiv y$ , there would be no adversarial recourse as every instance that leads to a different model prediction also changes the ground truth. However, in the setup we consider, it is impossible to perfectly learn  $y$ . Nevertheless, using the best possible model as close to the ground truth as possible should be fruitful. Another way to improve the model's alignment with the ground truth – in case the truth is known to be smooth in some measure – could be to potentially leverage regularization techniques such as adversarial training [38] to rule out many adversarial instances in the first place. **We empirically find that more accurate and robust models lead to less adversarial recourse.**

*Input Space Distance Function.* The distance function  $d_1$  has been attributed a crucial role when computing recourse or adversarial examples. For instance, Wachter et al. [65] have claimed that, unlike recourse, *none of the standard works on adversarial perturbations use appropriate distance functions*. In this work, we follow the perspective of [6, 23], who argue that the distance metric is not definitional but may still play an essential role in making recourse non-adversarial. Besides standard cost functions like  $p$ -norms such as the  $l_1$ ,  $l_2$ , and  $l_\infty$ , we are interested in how feature weightings may potentially impact recourse. We follow the intuition that some features are discriminative in the ground truth problem, e.g., income determines creditworthiness. However, ML models may rely on many more features, as the model designers cannot precisely specify a priori which features will be relevant for the task. When non-discriminative features are used in the task, they may open the door to adversarial changes as they can be picked up by an ML model regardless of their irrelevance w.r.t. the ground truth. In the next section, we will present an attempt to down-weight the influence of such features by individually assigning a cost to each of them. In particular, we will consider distance functions of the form

$$d_{1,S}(\mathbf{x}, \mathbf{x}') := \boldsymbol{\delta}^\top \mathbf{S} \boldsymbol{\delta}, \mathbf{S} = \text{diag}(\mathbf{s}), \quad (9)$$

where  $\mathbf{S} \in \mathbb{R}^{k \times k}$  is some diagonal matrix with diagonal  $\mathbf{s} = [s_1, \dots, s_k]^\top \in \mathbb{R}_{>0}^k$  and  $\boldsymbol{\delta} := \mathbf{x}' - \mathbf{x}$ . For simple models analytical solutions of algorithmic recourse exist [46]. This allows to set up a nested optimization problem, where besides optimizing the recourse for a specific cost function, we find the cost function such that the resulting optimal recourse remains most non-adversarial. We will introduce the specific objective in the next section. We will see that the problem of finding optimal values for  $\mathbf{s}$  can be solved analytically based on the gradients for linear models. **Surprisingly, we empirically find that the cost function does not play a key role in obtaining non-adversarial recourse.**

*Optimization Routine.* As the general problem is highly non-linear for complex models, it is hard to discover an optimal solution. As a result, algorithms to compute recourse or adversarial examples include different heuristic optimization routines such as stochastic gradient descent (deployed in SCFE, DICE, and C&W), gradient projection (deployed in PGD), or discretization (deployed in AR). The optimization procedure may thus also play a non-negligible role in determining whether the nature of the resulting recourse is adversarial and whether approaches designed for recourse yield fewer adversarial examples than their adversarial counterparts. **In this regard, we find that adversarial methods succeed to compute non-adversarial recourse, but also incur higher costs.**

## 5.2 Optimal Cost Functions Under Linear Models with Noisy Labels

In this section, we will restrict ourselves to the input space distance function  $d_1$  and study its influence on the recourse from a theoretical standpoint.

We first introduce a measure to quantify the extent to which recourse is non-adversarial. To be able to do so, we consider the simplified setup where we have a feature set  $\mathcal{F}$  and a subset of discriminative features  $\mathcal{F}_{\text{disc}} \subset \mathcal{F}$  that contains relevant information affecting the ground truth. The remainder of the features are noise variables. Such features exist for many tasks; however, they may require a change of representation to be axis-aligned. For instance, in image generation models such as StyleGAN [32], the first latent variables control high-level concepts in the generation, whereas the later variables merely add noise that is unimportant for the classification output. Successes with dimensionality reduction techniques through autoencoding [30] also show that important information occupies only a subspace of tabular data. As outlined in Fig. 3, following the discriminative features is essential for obtaining non-adversarial recourse. We can quantify the share of the recourse that lies in the discriminative directions over the entire length of the recourse vector through the following measure.

**Definition 4 (NADV measure).** *Let  $p \in \mathbb{N} \cup \{\infty\}$ . The non-adversarialness measure  $NADV_p$  is defined as*

$$NADV_p(\boldsymbol{\delta}) = \frac{\sum_{i \in \mathcal{F}_{\text{disc}}} |\delta_i|}{\|\boldsymbol{\delta}\|_p}. \quad (10)$$

We consider linear models in our initial analysis, as they are the standard in many industries (e.g., in financial applications such as credit scoring [12]) and are commonly studied in the literature on algorithmic recourse [49, 60]. They model a generative process of the form

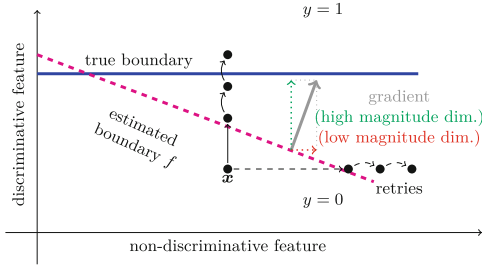
$$y(\mathbf{x}) = \boldsymbol{\beta}^\top \mathbf{x} + \epsilon, \quad (11)$$

where  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  is Gaussian noise of variance  $\sigma^2$  and  $\boldsymbol{\beta} \in \mathbb{R}^k$  denotes the true linear parameter vector. Such a model can be easily adapted to a classification task by introducing a decision threshold, e.g.,  $y(\mathbf{x}) > 0$  indicates a positive outcome. As motivated in the introduction, the noise may represent uncertainty and variance in the human labels. We are interested in weightings  $s_i$  that minimize this measure, potentially leveraging the empirical coefficients  $\hat{\boldsymbol{\beta}}$  obtained when fitting a linear model to the noisy data.

**Theorem 1 (Optimal feature weights for recourse in linear models).** *Suppose the data-generating process in Eq. (11) and that for  $i \notin \mathcal{F}_{\text{disc}}$ , we have  $\beta_i = 0$ , and for  $i \in \mathcal{F}_{\text{disc}}$ ,  $|\beta_i| > \alpha \in \mathbb{R}$ . We can maximize the expected  $NADV_p$  measure for  $p \in \{1, 2, \infty\}$  when using the empirical coefficients  $\hat{\beta}_i$  of the fitted model by setting the weights to*

$$s_i \sim \begin{cases} \left\{ \begin{array}{ll} 1, & \text{if } i = \arg \max_j \mathbf{p}_{\text{disc}}(\hat{\beta}_j), \text{ else } \infty \end{array} \right\} & \text{if } p = 1 \\ \frac{|\hat{\beta}_i|}{\mathbf{p}_{\text{disc}}(\hat{\beta}_i)} & \text{if } p = 2 \\ |\hat{\beta}_i| & \text{if } p = \infty \end{cases},$$

where  $\mathbf{p}_{\text{disc}}(\hat{\beta}_i)$  is a probability of the feature being discriminative dependent on its empirical coefficient, which has a tractable sigmoidal form given in the Appendix.



**Fig. 3. Role of discriminative features in providing non-adversarial recourse.** When features can be discriminative, (i.e., class-relevant) or non-discriminative (i.e., noise features), exploiting the discriminative ones will eventually lead to non-adversarial recourse, whereas solely relying on the non-discriminative ones will result in an adversarial. Nevertheless, even when selecting the correct features, several retry steps in the recourse direction may be required to cross the true decision boundary. To align recourse with discriminative features, the gradients of the model may serve as guidance, as we expect the discriminative dimensions to exhibit a **higher** gradient magnitude.

We provide a proof of this result in Appendix A. This finding highlights that in the case of discriminative and non-discriminative features in the data (even if they are not known), different loss functions affect which share of the recourse is awarded to the discriminative features. It also highlights the effect of the different norms. Optimizing the  $\text{NADV}_1$  measure assigns infinite costs to all but the dimension that is most likely to be discriminative (with the highest absolute coefficient). On the other hand, the  $\text{NADV}_\infty$  measure is maximized if the discriminative features exhibit the maximum change of all features, disregarding changes in non-discriminative features. Therefore, the solution attempts to change all dimensions equally through assigning more discriminative dimensions a proportionally higher cost. This ensures that the less discriminative dimensions are altered as well. We observe that  $p = 2$  seems to constitute a suitable trade-off, where dimensions with low probabilities of being discriminative ( $\mathbf{p}_{\text{disc}}(\hat{\beta}_i) \approx 0$ ) are penalized by high costs, but the changes will otherwise be distributed evenly among the remaining dimensions.

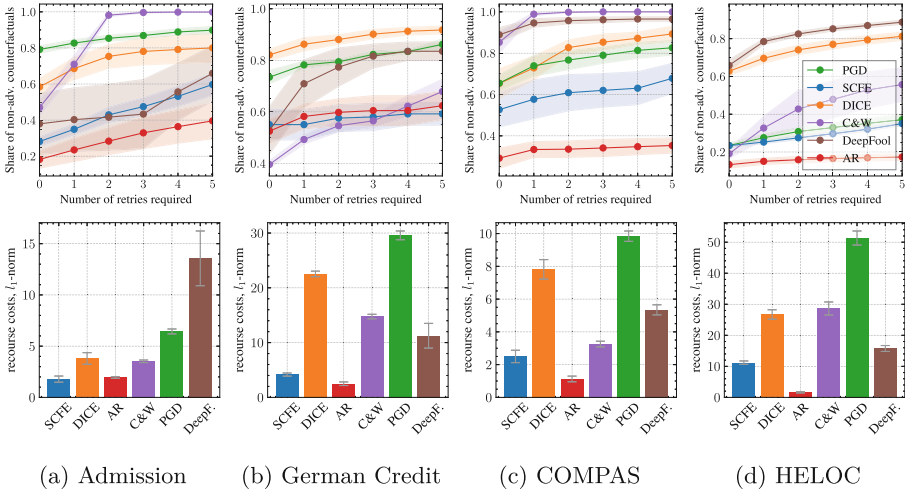
## 6 Experimental Evaluation

### 6.1 Experimental Setup

**Datasets and Preprocessing.** To link to the scenario considered in the introduction, we consider four tabular datasets concerned with high-stakes decision-making scenarios where human oversight may be required.

The Law School Admission data set<sup>5</sup> (“admission”) contains information on students from law schools across the United States. Features are collected

<sup>5</sup> <https://github.com/mkusner/counterfactual-fairness>.



**Fig. 4. Both adversarial and recourse methods can succeed in producing non-adversarial recourse for ANNs.** As it might not always be possible to change the ground truth immediately, we study the share of non-adversarial recourse instances after taking a certain number of retries  $r$  (a higher share is better). We experiment with three recourse methods (SCFE, DICE, AR) and three adversarial methods (C&W, PGD, DeepFool). Our results indicate that DICE and PGD usually perform best in identifying non-adversarial counterfactuals. The other adversarial methods, C&W and DeepFool, often outperform the standard recourse method SCFE regarding non-adversarial recourse. Note that recourse methods strictly optimize for the lowest costs and are therefore less robust than adversarial methods, which incur higher costs.

prior to their entry to law school and include race, sex, entrance exam scores (LSAT), grade-point average (GPA), and regional group. The predicted variable is the z-score of the first-year average grade (ZFYA). The German Credit dataset (“german”) is taken from the UCI machine learning repository<sup>6</sup> and is concerned with credit scoring. It contains the personal data of 1000 individuals with a binary indicator named “credit risk” that serves as a prediction target. The Home Equity Line of Credit (“HELOC”) data set<sup>7</sup> is a large collection of HELOC applications from anonymized homeowners collected by the financial services provider FICO. The target variable RiskPerformance is “Bad” if the applicant was at least 90 days past due within the two years after opening the credit account. The COMPAS data set<sup>8</sup> was initially collected by ProPublica and contains features describing criminal defendants in Broward County, Florida. It also contains their respective recidivism score provided by the COMPAS algorithm and whether or not they reoffended within the following two years. For our

<sup>6</sup> <http://archive.ics.uci.edu/dataset/144/statlog+german+credit+data>.  
<sup>7</sup> <https://community.fico.com/s/explainable-machine-learning-challenge>.  
<sup>8</sup> <https://www.kaggle.com/s/danofer/compass>.



analysis, we only kept features relevant for predicting recidivism within the next two years and dropped irrelevant features such as name, date, sex, and race.

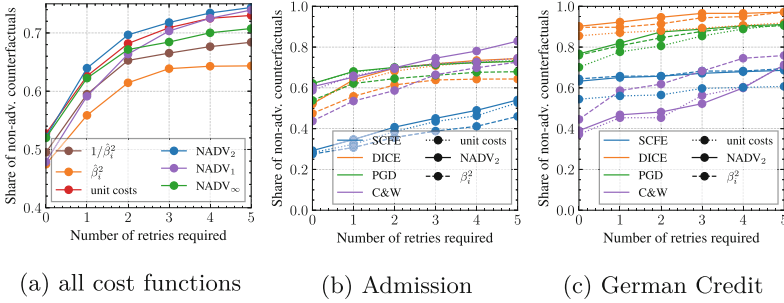
For all datasets, continuous features are standardized. Datasets with continuous labels are used in a binary classification fashion where we only predict if the z-score exceeds the population’s median.

**Machine Learning Models.** We use standard Artificial Neural Networks (ANN) that reflect the implementation of the `sklearn` library but are implemented in the `PyTorch` library to leverage automated differentiation capabilities. We train the ANN model (two fully connected hidden layers of width 30) using stochastic gradient descent with the ADAM optimizer. An overview over implementation parameters is provided in the Appendix.

**Adversarial Attacks and Recourse Algorithms.** We implement three powerful adversarial attacks and three recourse methods to study the problem from a practical perspective. We stick to the methods introduced earlier, which include SCFE [65], which uses a gradient-based objective to find recourses, DICE [42] with an extra diversity constraint, and AR [61], which uses a Mixed-Integer-Program on a discretized action set. Regarding the adversarial attacks, we use C&W [7] that finds the minimum perturbation on the factual instance to make it change class, PGD [38] that uses projected gradients to engender adversarials, and DeepFool [40] that perturbs the input iteratively until the class changes. We adapt the cost function of each optimization algorithm to reflect Eq. (2) and plug in the different cost functions.

**Ground Truth.** Unfortunately, the number of instances with labels on real-world data sets is limited, such that the ground truth function  $y$  is not explicitly available. We, therefore, rely on a simulated ground truth, which uses a subset of the training data that will not be used for model training or testing. We use this data to construct a  $k$  nearest neighbor classifier (with  $k = 5$ ) that uses a subset of features to simulate an expert committee relying on discriminative features and deciding by majority vote. We manually select features that can be considered directly discriminative for the task, which are listed in Table 2. For instance on the COMPAS dataset, we use features such as the number of priors, and whether recidivism has occurred in the last two year. By doing so, we can guarantee that we have discriminative and non-discriminative features. We then use this ground truth  $y$  to predict the remaining instances of the train set. Subsequently, the actual ML model is trained on the remainder of the data and their predictions, making up tuples of the form  $(\mathbf{x}, y(\mathbf{x}))$ .

**Evaluation Measures.** Many recourse (and adversarial) methods are implemented to stop right after the model’s boundary is crossed. However, this might not initially lead to the non-adversarial recourse desired in practice, even if the correct discriminative features are manipulated (see Fig. 3 for an illustration). We argue that in the practical use case, an individual would query the oracle (e.g., submit their application to the bank again) after obtaining recourse. If the recourse was ineffective in changing the loan decision, an individual could continue to move in the given direction (e.g., further increase their savings amount)



**Fig. 5. Cost functions can play a role in generating non-adversarial recourse.** (a) “admission” dataset with ANN model, DICE results shown. (b,c): Our NADV<sub>2</sub> cost function helps in making recourse slightly less adversarial for several and thereby reduces the number of retries required. However, analysing the standard deviations does *not* confirm statistical significance.

until the loan is eventually awarded. We mimic this setup, by increasing the magnitude of  $\delta = \mathbf{x}' - \mathbf{x}$  by 10% in each step, thus considering  $\mathbf{x}'_r = \mathbf{x} + (1.1^r)\delta$  after  $r \geq 0$  retries. We additionally consider the canonical recourse costs in the  $l_1$  and  $l_2$  norm.

### 6.2 Choice of Optimization Algorithm

We first put all six implemented methods to the test and check the adversarialness of their outputs. The results are visualized in Fig. 4. We consider the initial recourse and up to 5 more steps in the initial direction. We observe that DICE and PGD usually perform best in identifying non-adversarial counterfactuals. However, the other adversarial methods, C&W and DeepFool, also often outperform the classical recourse method SCFE regarding non-adversarial recourse. This underlines that, for tabular data, the methods do not reliably produce adversarials. Indeed, they could be considered as recourse methods as well. However, we observe that the adversarial techniques usually result in higher costs, because returning an optimal solution is not their main concern (it just needs to be “close” to the input). In contrast, many recourse methods are designed to provide cost-optimal solutions. Non-adversarial recourse is associated with higher cost, leading us to believe that classical recourse methods may be overly cost sensitive for this purpose. We obtained similar results using L2-costs.

### 6.3 Choice of Cost Function

We now study the different cost functions derived in Sect. 5.2 to actual implementations of both recourse and adversarial methods on real data. In particular, we compute the gradients of the model and use the cost weightings derived earlier as well as the default  $l_2$ -costs, squared gradient costs ( $\beta_i^2$ , should assign low cost to non-discriminative features) and inverse squared costs ( $1/\beta_i^2$ ) as baselines. DeepFool and AR do not allow for the simple, straightforward inclusion

of arbitrary cost functions, so we only consider the four remaining approaches for this experiment and modify their cost-function. The results are shown in Fig. 5. They show that cost weighting can steer the recourses towards the non-adversarial features and align them better with the ground truth. However, in Fig. 5a, the differences remain statistically insignificant. We observe that the  $\text{NADV}_2$  optimal weighting scores best among all costs. Inversely weighting the features (e.g.,  $s_i = \hat{\beta}_i^2$ , which assigns low costs to features with almost zero gradients and high costs to features with high gradients), preventing them from being changed, results in the most adversarial recourse. Even though the gap is small, the improvement seems stable across methods (see Fig. 5b, c) with one exception (C&W on German Credit). In conclusion, while the cost function can help to make recourse less adversarial, its effect seems to be rather subtle.

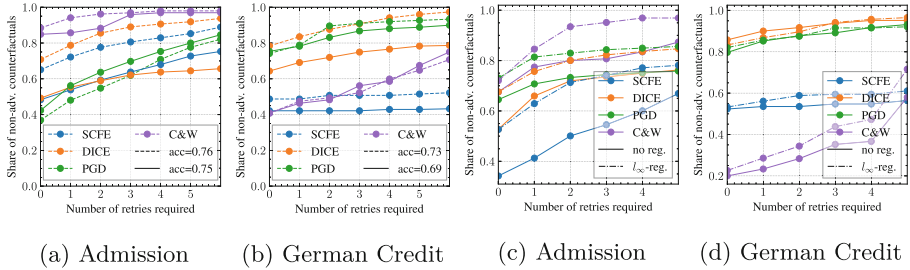
#### 6.4 Choice of Machine Learning Model

In our analysis section, we outlined how the machine learning (ML) model may be crucial in determining whether the outcomes can be considered adversarial. We first study the role of the goodness of the model fit. To this end, we train a model on a version of the dataset, where a random sample of 25% of the data points have flipped labels, which could reflect a realistic use case with noisy human annotations. To rule out other confounding effects to the convexity or smoothness of the model’s decision boundary (models trained on noisy labels may have very sharp and more non-convex decision boundaries), we study logistic regression models in this experiment and report the results in Fig. 6 (a, b). Surprisingly, the drop in accuracy is not very high (it remains in a range of 1.5% to 5%), which we attribute to the datasets being already very noisy previously. Nevertheless, we observe a clear tendency for recourses to be less adversarial for the more accurate models. This trend is stable across datasets and methods.

Adversarial training was proposed by Madry et al. [38] to make models more robust against adversarial attacks. Therefore, it might also offer a suitable way of mitigating adversarial examples in the recourse setup. We study the effect of this form of regularization in an  $l_\infty$ -ball of radius  $\epsilon = 0.2$  in Fig. 6 (c, d). We observe that substantial improvements are possible on the Admission dataset. They are not as pronounced for the remaining datasets but remain visible for most methods. We observe comparable results for the remaining two datasets. Our results highlight that maintaining robust and accurate models is one of the most promising strategies towards non-adversarial recourse.

## 7 Discussion

**Adversarial Methods Compute Recourse on Tabular Data.** Intriguingly, we observe that despite their purpose, many adversarial attacks succeed in computing non-adversarial recourse on tabular data. While many of the methods were arguably designed with other data modalities, e.g., images, in mind, our finding raises the question of how transferable existing attacks are to variants of the canonical attack scenario. This observation is one in a series of recent claims



**Fig. 6. (a, b): More accurate models lead to less adversarial recourse.** We plot the number of retries required to obtain a valid, non-adversarial recourse that changes the ground truth. Logistic Regression Model shown. Results on the remaining datasets can be found in the extended Appendix (<https://arxiv.org/abs/2403.10330>). **(c, d): Regularization through Adversarial Training may improve non-adversarialness.** We robustify models through adversarial training, which improves the share of non-adversarial recourses.

suggesting that current adversarial attacks may not be realistic in the majority of practical use cases [4] or require a fundamental paradigm shift away from norms as cost functions towards realistic measures of detector evasion [15].

**An Implicit Pursuit Towards Non-adversarial Recourse.** The recourse literature suggests several strategies for improving the quality of recourse. Kommiya et al. [33] discovered that feature attributions and feature modifications in recourses only partially agree, raising the question of how they can potentially be used as guidance. Recent takes on robustifying recourse by going further than mandated by the actual decision boundary [48, 60] can be interpreted as another take to reduce the possibility of ending up with an adversarial. Therefore, we conclude that these works seem to have implicitly followed the goal of obtaining non-adversarial recourse and can be interpreted as orthogonal attempts to reach this common goal. We hope that our precise definition of non-adversarial recourse allows for these efforts to be bundled and unified in the future.

**Non-adversarial Recourse via Distributional Constraints.** Another avenue we have not followed in this work considers the feasible set. The feasible set  $\mathcal{X}$  many works have claimed that recourse should be actionable, leading to realistic instances [50, 61]. A fairly general way to arrive at this goal is to constrain the recourse to be in-distribution [17, 31, 47], which can be seen as another strategy towards non-adversarial recourse: For in-distribution examples, every model that is a suitable approximation of the ground truth should result in an above-chance-level agreement between the model and the ground truth. We leave an investigation of this connection to future work.

## 8 Conclusion

In this work, we explored the nuanced differences between adversarial examples and counterfactual explanations, focusing on real-world high-stakes decision-

making processes. For such scenarios, we introduced the desirable concept of non-adversarial recourse, emphasizing that useful counterfactual explanations should not only change the model’s prediction but also align with the ground truth in contrast to adversarial examples.

Our theoretical and experimental analyses on multiple real-world datasets illuminate different ways the model parameters can shape the generation of non-adversarial recourse. Our findings suggest that choosing a suitable model that is highly accurate and robust has more impact on whether recourse can be considered adversarial than the choice of the cost function. For tabular data, adversarial methods also succeed in computing suitable recourse. In summary, we provided valuable insights into generating counterfactuals of reduced adversarialness. Hence, this work lays a foundation for developing resilient recourse models and their deployment in realistic decision-making scenarios.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## A Derivation of Theorem V.I

This section presents the proof of Theorem 1 proof. First, we show how the probability of a relevant feature can be easily estimated in linear models. Suppose we have obtained a data matrix  $\mathbf{X} \in \mathbb{R}^{n \times k}$ . Then, we can obtain the analytical least-squares solution  $\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$ . We can estimate the variance of  $\hat{\boldsymbol{\beta}}$  to be  $\text{Var}[\hat{\boldsymbol{\beta}}] = \sigma^2 (\mathbf{X}^\top \mathbf{X})^{-1}$ . Simplifying through assuming the features in  $\mathbf{x}$  to be independent and of zero-mean,  $\mathbf{X}^\top \mathbf{X}$  is diagonal and we obtain

$$\text{Var}[\hat{\beta}_i] = \frac{\sigma^2}{\sum_{j=1 \dots n} (\mathbf{x}_j)_i^2}. \quad (12)$$

This allows to use of the estimated coefficients to estimate the probability of a feature being relevant,  $\mathbf{p}_{\text{disc}}$  through the following derivation:

$$\mathbf{p}_{\text{disc}}(\hat{\beta}_i) = \mathbf{p}(i \in \mathcal{F}_{\text{disc}} | \hat{\beta}_i) \quad (13)$$

$$= \frac{\mathbf{p}(\hat{\beta}_i, i \in \mathcal{F}_{\text{disc}})}{\mathbf{p}(\hat{\beta}_i, i \in \mathcal{F}_{\text{disc}}) + \mathbf{p}(\hat{\beta}_i, i \notin \mathcal{F}_{\text{disc}})} \quad (14)$$

$$= \frac{\mathbf{p}(\hat{\beta}_i | i \in \mathcal{F}_{\text{disc}})}{\mathbf{p}(\hat{\beta}_i | i \in \mathcal{F}_{\text{disc}}) + \mathbf{p}(\hat{\beta}_i | i \notin \mathcal{F}_{\text{disc}}) \underbrace{\frac{\mathbf{p}(i \notin \mathcal{F}_{\text{disc}})}{\mathbf{p}(i \in \mathcal{F}_{\text{disc}})}}_q} \quad (15)$$

$$= \frac{1}{1 + \frac{\mathbf{p}(\hat{\beta}_i | i \notin \mathcal{F}_{\text{disc}})}{q \cdot \mathbf{p}(\hat{\beta}_i | i \in \mathcal{F}_{\text{disc}})}} \geq \frac{1}{1 + \exp(\alpha^2 - 2\alpha|\hat{\beta}_i| - \log q)} \quad (16)$$

$$= \text{sigmoid}(2\alpha|\hat{\beta}_i| - \alpha^2 + \log q). \quad (17)$$

The above calculation highlights that it is possible to use the coefficients  $\hat{\beta}$  in the linear model as noisy estimates for assessing whether a feature is discriminative.

We combine this insight with the optimal recourse found using a specific cost matrix  $\mathbf{S}$ . To this end, we leverage the analytical solution to this problem [9, Lemma 4, Appendix]:

$$\delta(\mathbf{S}) = \underbrace{\frac{f(\mathbf{x}) - y_t}{\hat{\beta}^\top \mathbf{S}^{-1} \hat{\beta}}}_c \mathbf{S}^{-1} \hat{\beta}. \tag{18}$$

We can then compute the expected value of the measure of non-adversarialness for the recourse that will be found with the corresponding cost function:

$$\mathbb{E}_{\hat{\beta}} [\text{NADV}_p(\mathbf{S})] = \mathbb{E}_{\hat{\beta}} \left[ \frac{\sum_{i \in \mathcal{F}_{\text{disc}}} |\delta_i|}{\|\delta\|_p} \right] = \mathbb{E}_{\hat{\beta}} \left[ \frac{\sum_{i \in \mathcal{F}_{\text{disc}}} |\hat{\beta}_i|}{\|\mathbf{S}^{-1} \hat{\beta}\|_p} \right] \tag{19}$$

$$= \frac{\sum_i p_{\text{disc},i}(\hat{\beta}) |\hat{\beta}_i|}{\|\mathbf{S}^{-1} \hat{\beta}\|_p} = \frac{\mathbf{p}_{\text{disc}}^\top(\hat{\beta})(\mathbf{S}^{-1}|\hat{\beta}|)}{\|\mathbf{S}^{-1} \hat{\beta}\|_p} \tag{20}$$

$$= \frac{\mathbf{p}_{\text{disc}}^\top(\hat{\beta})(\mathbf{S}^{-1}|\hat{\beta}|)}{\|\mathbf{S}^{-1}|\hat{\beta}\|_p} \tag{21}$$

Taking the above expression, we can obtain optimal costs for different values of  $p$  by solving

$$\arg \max \mathbb{E}_{\hat{\beta}} [\text{NADV}_p(\mathbf{S})]. \tag{22}$$

Continuing the calculation separately for the most common values  $p \in \{1, 2, \infty\}$ , we obtain the following cost weights  $s_i$  that depend on the estimated  $\hat{\beta}_i$ :

$p = 1$	$p = 2$	$p = \infty$
implicit		
$\mathbf{S}^{-1} \hat{\beta}  = \kappa e_{\arg \max_i p_{\text{disc}}(\hat{\beta}_i)}$	$\mathbf{S}^{-1} \hat{\beta}  = \kappa \frac{\mathbf{p}_{\text{disc}}(\hat{\beta})}{\ \mathbf{p}_{\text{disc}}(\hat{\beta})\ _2}$	$\mathbf{S}^{-1} \hat{\beta}  = \kappa \mathbf{1}$
explicit		
$s_i \sim \left\{ 1, \text{ if } i = \arg \max_j p_{\text{disc}}(\hat{\beta}_j), \text{ else } \infty \right\}$	$s_i \sim \frac{ \hat{\beta}_i }{p_{\text{disc}}(\hat{\beta}_i)}$	$s_i \sim  \hat{\beta}_i $

## B Experimental Details

We use the following experimental parameters (Table 1):

**Table 1.** Implementation parameters

		Artificial Neural Network	Logistic Regression
Config.	Units	[Input dim., 30, 30, 2]	[Input dim., 1]
	Intermediate activations	ReLU	N/A
	Last layer activations	None	Sigmoid
Training	Learning rate	$10^{-3}$	N/A
	Regularization	None	$l_2$ with pen = 1
	Batch size	32	N/A
	Epochs	$10^3$	$5 \times 10^3$

Method	Optimizer	lr	Iterations	$\lambda$	Additional Comments
SCFE	Adam	$10^{-1}$	100	0.1	step = 0
DiCE	RMSProp	$10^{-1}$	100	-	Two counterfactuals, one is randomly sampled for evaluation
AR	Default as in [61]	-	-	-	Squared loss in cost function
C&W	Gradient-based as in [7]	$10^{-2}$	1000	-	Constant factor $c = 1$
DeepFool	-	-	50	$2 \times 10^{-2}$	Target label for attack directionality [40]
PGD	-	$10^{-1}$	10	$10^{-1}$	$\alpha = 10^{-1}$ , $\epsilon = 2$

**Table 2.** Features that are used by the experts (GT) and total number of features available to adversarial methods and recourse methods on each dataset.

Dataset	GT Features	Tot. features
Admission	ugpa, first_pf	4
German Credit	status, credit-history, employment-duration, housing, number-credits	19
COMPAS	age, two_year_recid, priors_count	5
HELOC	MSinceMostRecentTradeOpen, NumTrades60Ever2DerogPubRec, NumTrades90Ever2DerogPubRec, NumTradesOpeninLast12M, NumInqLast6M, NumInqLast6Mexcl7days, NumRevolvingTradesWBalance, NumInstallTradesWBalance, Num- Bank2NatlTradesWHighUtilization	22

## References

1. Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.: Trends and trajectories for explainable, accountable and intelligible systems: an HCI research agenda. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–18 (2018)
2. Abrate, C., Bonchi, F.: Counterfactual graphs for explainable classification of brain networks. In: KDD (2021)
3. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: a survey. *IEEE Access* **6**, 14410–14430 (2018)
4. Apruzzese, G., Anderson, H.S., Dambra, S., Freeman, D., Pierazzi, F., Roundy, K.: “real attackers don’t compute gradients”: Bridging the gap between adversarial ml research and practice. In: 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML), pp. 339–364. IEEE (2023)
5. Baluja, S., Fischer, I.: Adversarial transformation networks: learning to generate adversarial examples. arXiv preprint [arXiv:1703.09387](https://arxiv.org/abs/1703.09387) (2017)
6. Browne, K., Swift, B.: Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. arXiv preprint [arXiv:2012.10076](https://arxiv.org/abs/2012.10076) (2020)
7. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57. IEEE (2017)
8. Carreira-Perpiñán, M.Á., Hada, S.S.: Counterfactual explanations for oblique decision trees: exact, efficient algorithms. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 6903–6911 (2021)
9. Chen, Y., Wang, J., Liu, Y.: Strategic recourse in linear classification. arXiv preprint [arXiv:2011.00355](https://arxiv.org/abs/2011.00355) **236** (2020)
10. Cheng, J., Danescu-Niculescu-Mizil, C., Leskovec, J.: Antisocial behavior in online discussion communities. In: Proceedings of the International AAAI Conference on Web and Social Media, vol. 9, pp. 61–70 (2015)
11. Croce, F., Hein, M.: Sparse and imperceivable adversarial attacks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4724–4732 (2019)
12. Dastile, X., Celik, T., Potsane, M.: Statistical and machine learning models in credit scoring: a systematic literature survey. *Appl. Soft Comput.* **91**, 106263 (2020)
13. De, A., Koley, P., Ganguly, N., Gomez-Rodriguez, M.: Regression under human assistance. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2611–2620 (2020)
14. De, A., Okati, N., Zarezade, A., Rodriguez, M.G.: Classification under human assistance. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 5905–5913 (2021)
15. Debenedetti, E., Carlini, N., Tramèr, F.: Evading black-box classifiers without breaking eggs. arXiv preprint [arXiv:2306.02895](https://arxiv.org/abs/2306.02895) (2023)
16. Demir, U., Unal, G.B.: Patch-based image inpainting with generative adversarial networks. *CoRR* abs/1803.07422 (2018). <http://arxiv.org/abs/1803.07422>
17. Dhurandhar, A., et al.: Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
18. Dominguez-Olmedo, R., Karimi, A.H., Schölkopf, B.: On the adversarial robustness of causal algorithmic recourse. In: Proceedings of the 39th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 162, pp. 5324–5342. PMLR (2022)



19. Du, A., et al.: Physical adversarial attacks on an aerial imagery object detector. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 1796–1806 (2022)
20. Duan, R., et al.: Adversarial laser beam: effective physical-world attack to DNNs in a blink. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16062–16071 (2021)
21. Ferreira, J.J., de Souza Monteiro, M.: The human-AI relationship in decision-making: AI explanation to support people on justifying their decisions. In: Joint Proceedings of the ACM IUI 2021 Workshops, vol. 2903 (2021)
22. Freiesleben, T.: Counterfactual explanations & adversarial examples—common grounds, essential differences, and potential transfers. arXiv preprint [arXiv:2009.05487](https://arxiv.org/abs/2009.05487) (2020)
23. Freiesleben, T.: The intriguing relation between counterfactual explanations and adversarial examples. *Mind. Mach.* **32**(1), 77–109 (2022)
24. Garcia, L., Brassler, F., Cintuglu, M.H., Sadeghi, A.R., Mohammed, O.A., Zonouz, S.A.: Hey, my malware knows physics! attacking PLCs with physical model aware rootkit. In: NDSS, pp. 1–15 (2017)
25. GDPR: Regulation (EU) 2016/679 of the European parliament and of the council. *Off. J. Eur. Union* (2016)
26. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint [arXiv:1412.6572](https://arxiv.org/abs/1412.6572) (2014)
27. Grudin, J.: AI and HCI: two fields divided by a common focus. *AI Mag.* **30**(4), 48 (2009). <https://doi.org/10.1609/aimag.v30i4.2271>, <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2271>
28. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. *Data Min. Knowl. Discov.* 1–55 (2022)
29. Heath, D., Kasif, S., Salzberg, S.: Induction of oblique decision trees. In: *IJCAI*, vol. 1993, pp. 1002–1007. Citeseer (1993)
30. Ilkhechi, A., et al.: DeepSqueeze: deep semantic compression for tabular data. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pp. 1733–1746 (2020)
31. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. arXiv preprint [arXiv:1907.09615](https://arxiv.org/abs/1907.09615) (2019)
32. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of styleGAN. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8110–8119 (2020)
33. Kommiya Mothilal, R., Mahajan, D., Tan, C., Sharma, A.: Towards unifying feature attribution and counterfactual explanations: different means to the same end. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society, pp. 652–663 (2021)
34. König, G., Freiesleben, T., Grosse-Wentrup, M.: Improvement-focused causal recourse (ICR). In: AAAI Conference on Artificial Intelligence (2023)
35. Kurakin, A., Goodfellow, I., Bengio, S., et al.: Adversarial examples in the physical world (2016)
36. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: unjustified counterfactual explanations. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19) (2019)

37. Ma, J., Guo, R., Mishra, S., Zhang, A., Li, J.: Clear: generative counterfactual explanations on graphs. In: *Advances in Neural Information Processing Systems*, vol. 35, pp. 25895–25907 (2022)
38. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint [arXiv:1706.06083](https://arxiv.org/abs/1706.06083) (2017)
39. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1765–1773 (2017)
40. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2574–2582 (2016)
41. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 607–617 (2020)
42. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT\*)* (2020)
43. Mozannar, H., Sontag, D.: Consistent estimators for learning to defer to an expert. In: *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, pp. 7076–7087 (2020)
44. Narodytska, N., Kasiviswanathan, S.P.: Simple black-box adversarial perturbations for deep networks. arXiv preprint [arXiv:1612.06299](https://arxiv.org/abs/1612.06299) (2016)
45. Pauwels, E.: How to protect biotechnology and biosecurity from adversarial AI attacks? A global governance perspective. In: Greenbaum, D. (ed.) *Cyberbiosecurity*, pp. 173–184. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-26034-6\\_11](https://doi.org/10.1007/978-3-031-26034-6_11)
46. Pawelczyk, M., Agarwal, C., Joshi, S., Upadhyay, S., Lakkaraju, H.: Exploring counterfactual explanations through the lens of adversarial examples: a theoretical and empirical analysis. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 4574–4594. PMLR (2022)
47. Pawelczyk, M., Broelemann, K., Kasneci, G.: Learning model-agnostic counterfactual explanations for tabular data. In: *Proceedings of The Web Conference 2020 (WWW)*. ACM (2020)
48. Pawelczyk, M., Datta, T., den Heuvel, J.V., Kasneci, G., Lakkaraju, H.: Probabilistically robust recourse: navigating the trade-offs between costs and robustness in algorithmic recourse. In: *The Eleventh International Conference on Learning Representations (ICLR)* (2023)
49. Pawelczyk, M., Leemann, T., Biega, A., Kasneci, G.: On the trade-off between actionable explanations and the right to be forgotten. In: *The Eleventh International Conference on Learning Representations (ICLR)* (2023)
50. Poyiadzi, R., Sokol, K., Santos-Rodriguez, R., De Bie, T., Flach, P.: Face: feasible and actionable counterfactual explanations. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 344–350 (2020)
51. Pradel, M., Sen, K.: DeepBugs: a learning approach to name-based bug detection. *Proc. ACM Program. Lang.* **2**(OOPSLA), 1–25 (2018)
52. Prado-Romero, M.A., Prenkaj, B., Stilo, G., Giannotti, F.: A survey on graph counterfactual explanations: definitions, methods, evaluation, and research challenges. *ACM Comput. Surv.* (2023)
53. Raghu, M., Blumer, K., Corrado, G., Kleinberg, J., Obermeyer, Z., Mullainathan, S.: The algorithmic automation problem: prediction, triage, and human effort. arXiv preprint [arXiv:1903.12220](https://arxiv.org/abs/1903.12220) (2019)

54. Rawal, K., Lakkaraju, H.: Beyond individualized recourse: interpretable and interactive summaries of actionable recourses. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 12187–12198 (2020)
55. Sahil, V., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: a review (2010)
56. Stutz, D., Hein, M., Schiele, B.: Disentangling adversarial robustness and generalization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6976–6987 (2019)
57. Su, J., Vargas, D.V., Sakurai, K.: One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.* **23**(5), 828–841 (2019)
58. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint [arXiv:1312.6199](https://arxiv.org/abs/1312.6199) (2013)
59. Topol, E.J.: High-performance medicine: the convergence of human and artificial intelligence. *Nat. Med.* **25**(1), 44–56 (2019)
60. Upadhyay, S., Joshi, S., Lakkaraju, H.: Towards robust and reliable algorithmic recourse. In: *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34 (2021)
61. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT\*)* (2019)
62. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: *Proceedings of the Conference on Fairness, Accountability, and Transparency* (2019). <https://doi.org/10.1145/3287560.3287566>
63. Verma, S., Dickerson, J., Hines, K.: Counterfactual explanations for machine learning: a review. [arXiv:2010.10596](https://arxiv.org/abs/2010.10596) (2020)
64. Voigt, P., Von dem Bussche, A.: The EU general data protection regulation (GDPR). In: *A Practical Guide*, 1st edn. Springer, Cham (2017). **10**, 3152676
65. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. *Harv. J. Law Technol.* **31**(2) (2018)
66. Zhang, J., Lou, Y., Wang, J., Wu, K., Lu, K., Jia, X.: Evaluating adversarial attacks on driving safety in vision-based autonomous vehicles. *IEEE Internet Things J.* **9**(5), 3443–3456 (2022). <https://doi.org/10.1109/JIOT.2021.3099164>
67. Zhao, G., Zhang, M., Liu, J., Li, Y., Wen, J.R.: AP-GAN: adversarial patch attack on content-based image retrieval systems. *GeoInformatica*, 1–31 (2022)



# Communicating Uncertainty in Machine Learning Explanations: A Visualization Analytics Approach for Predictive Process Monitoring

Nijat Mehdiyev<sup>1,2(✉)</sup>, Maxim Majlatow<sup>1,2</sup>, and Peter Fettke<sup>1,2</sup>

<sup>1</sup> German Research Center for Artificial Intelligence (DFKI),  
Campus D 3.2, 66123 Saarbrücken, Saarland, Germany

{nijat.mehdiyev,maxim.majlatow,peter.fettke}@dfki.de

<sup>2</sup> Saarland University, Campus D 3.2, 66123 Saarbrücken, Saarland, Germany

**Abstract.** As data-driven intelligent systems advance, the need for reliable and transparent decision-making mechanisms has become increasingly important. Therefore, it is essential to integrate uncertainty quantification and model explainability approaches to foster trustworthy business and operational process analytics. This study explores how model uncertainty can be effectively communicated in global and local post-hoc explanation approaches. Furthermore, this study examines appropriate visualization analytics approaches to facilitate such methodological integration. By combining these two research directions, decision-makers can not only justify the plausibility of explanation-driven actionable insights but also validate their reliability. Finally, the study includes expert interviews to assess the suitability of the proposed approach and designed interface for a real-world predictive process monitoring problem in the manufacturing domain.

## 1 Introduction

The widespread use of machine learning (ML) models in real-world, high-stakes decision-making requires establishing reliability and understandability, which promote trust among relevant stakeholders [22]. In this regard, numerous explainable artificial intelligence (XAI) techniques have recently been proposed, including post-hoc explanation approaches, to provide global and local explanations of the model behavior [3]. These approaches aim to enhance the transparency and interpretability of the models, enabling stakeholders to comprehend their decision-making processes better.

Uncertainty quantification (UQ) is another emerging ML research field focusing on estimating and communicating the uncertainty associated with models' predictions. It can be regarded as a complementary form of transparency that

---

M. Majlatow and P. Fettke—Contributing Authors.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

L. Longo et al. (Eds.): xAI 2024, CCIS 2155, pp. 420–438, 2024.

[https://doi.org/10.1007/978-3-031-63800-8\\_21](https://doi.org/10.1007/978-3-031-63800-8_21)

can boost the explainability of solutions for decision tasks, which may not be sufficient on their own [4]. Recent studies suggest that effectively communicated and appropriately calibrated quantification of model uncertainty not only enhances stakeholders' trust in model predictions, thereby improving decision automation or augmentation, but also fosters transparency and trust among domain experts by providing insights into model confidence combined with suitable communication strategies [23, 24].

Selecting and deploying appropriate methods to generate explanations or quantify uncertainties is critical in ML model inspection. However, the efficiency of this process might be hampered if the insights are not presented to stakeholders clearly and concisely. Therefore, to accomplish optimal communication of the model's outcomes, it is essential to build interactive interfaces that are customized to the mental models of the intended stakeholders while minimizing their cognitive load. To this end, interactive information visualization has emerged as a promising area of research for enhancing ML models' interpretability, trustworthiness, and reliability by providing users with relevant visual representations [8]. Upon closer examination of the relevant literature, it becomes evident that a considerable amount of research has been conducted on either visual analytics for uncertainty communication [16], or model interpretability [2], but in isolation from each other. Despite the importance of both research directions in facilitating more informed decision-making, there appears to be a gap in integrated approaches that combine information visualization for uncertainty communication with model interpretability for a more holistic understanding.

With the gap mentioned above in mind, we present a novel methodology that aims to integrate model uncertainties into both local and global post-hoc explanations. Our approach is deployed in a visual analytics interface that enables to verify the appropriateness and usability of the generated explanations. To ensure that the solution meets the needs of its intended users, a comprehensive requirements analysis for the design process and evaluation has been conducted within a consortium research project. To summarize our main contributions, we have made the following contributions:

- This study delves into various methods of presenting model uncertainty in individual predictions, focusing particularly on predictive process monitoring. Our investigation spans visual, textual, and tabular formats for effectively conveying uncertainty.
- We develop and rigorously define techniques for incorporating model uncertainty into global post-hoc explanation frameworks. This includes enhancements to Partial Dependence Plots (PDP) and the introduction of a localized variant, Individual Conditional Expectation (ICE) plots, to better articulate model behaviors.
- To facilitate the practical application of our methodologies, we create sophisticated visualization analytics tools. These tools are evaluated extensively, engaging domain experts to address a real-world predictive process monitoring challenge, thereby demonstrating the utility and impact of our approach.

## 2 Background and Related Work

In numerous practical application scenarios, black-box ML algorithms are essential to reach a level of accuracy that conventional, intrinsically interpretable ML approaches cannot. Nevertheless, such opaque approaches frequently fail to explain their working mechanism, making it difficult for analysts to verify their veracity [3]. Providing explanations is an effective method for promoting acceptance of the predictions provided by intelligent systems. As a result, XAI has arisen as a fruitful area of research to enhance the collaboration between AI-based systems and human users by making the underlying non-transparent algorithms understandable [15]. The notion of explainability is intricate and multifaceted, requiring consideration of various factors within the decision-making environment. These factors include the analytical context, user attributes, explanation objectives, and a range of socio-cognitive and process-specific aspects. Therefore, when designing intelligent methods and interfaces, it is crucial to take into account these factors to ensure adequate explainability [19].

Different taxonomies are available for XAI techniques, one of the main categories being post-hoc explanation techniques. These techniques provide explanations for AI model predictions and can be grouped into two categories: local and global explanations. Local techniques (SHapley Additive exPlanations (SHAP), Local Interpretable Model-Agnostic Explanations (LIME), ICE) explain individual predictions, while global methods (PDP, SHAP Summary Plots, Permutation-based Feature Importance) explain the overall behavior of an AI model [17]. Post-hoc explanation techniques are often preferred by domain experts for justification and verification purposes due to their comprehensible nature. However, recent studies have revealed that most of these post-hoc explanation techniques exhibit inconsistency and instability and may fail to provide adequate information regarding their reliability, highlighting the need for integrating model uncertainty estimation [22]. Nonetheless, to date, only a limited number of studies undertook the endeavor to model uncertainty in post-hoc explanations, as we do in this study [5,20,22]. Moreover, there has been a significant lack of attention towards developing user interfaces that incorporate visualization analytics approaches in this intersection [6,7,9,18].

The ML lifecycle encompasses multiple stages, from data collection to model training, each of which introduces inherent uncertainties. As a result, the predictions generated by AI models are subject to various types of uncertainty, such as errors in data collection, model complexity, and algorithmic limitations. Two types of uncertainty that are commonly distinguished in the context of AI models are aleatoric uncertainty, and epistemic uncertainty [13]. Aleatoric uncertainty is related to inherent data variability or the observed phenomena' underlying stochastic nature. In contrast, epistemic uncertainty arises from incomplete or insufficient knowledge about the modeled system or the model's limitations. While aleatoric uncertainty depicts uncertainty that cannot be reduced, usually pertaining to the underlying data, epistemic uncertainty is reducible and either due to incomplete data or a characteristic of the fitted model. Methods used to quantify the uncertainty that capture both categories can be divided into

two main categories: Bayesian approaches and Frequentist approaches [4]. An overview of techniques from both families can be found in this study [13].

Integrating UQ in the context of explainability is a step towards a holistic, trustworthy Artificial Intelligence (AI), especially regarding the user's trust and acceptance of a model's decisions.

### 3 Research Methodology

This section describes the approach to constructing an uncertainty-aware XAI solution with corresponding interfaces. The design science research (DSR) approach is adopted to provide a systematic and rigorous process for conducting applied research [21]. This methodology is particularly suitable for information systems research and involves six steps: problem identification and motivation, defining objectives of a solution, design and development, demonstration, evaluation, and communication.

**Problem Identification and Motivation:** To secure both contextual applicability and methodological precision, our article draws upon inputs from the application domain for contextual relevance and the existing scientific knowledge base for methodological rigor. To ensure relevance, a consortium research method is used to engage practitioners in identifying open issues and defining objectives. Furthermore, an extensive literature analysis was conducted to secure the rigor, and findings were refined through iterative discussions with practitioners.

The primary challenge identified in this study is to develop an approach that can overcome the non-transparent nature of black-box ML techniques. This challenge is particularly relevant in high-stakes decision-making problems. To tackle this issue, the design of such systems should incorporate post-hoc explanation techniques that enable the users to ratify the validity of model decisions. Moreover, to ensure that solutions are not only explainable but also reliable and trustworthy, it is essential to effectively communicate model uncertainty in the explanations generated by the system

**Objective of a Solution:** The next step involves defining the objectives of the solution. The goals have two dimensions. First, a methodological approach is required for communicating model uncertainty and incorporating this information into post-hoc explanations. This would increase the reliability and trust of underlying algorithms. Second, interfaces with relevant visualization techniques should be devised to effectively communicate the outcomes to the system users.

**Design and Development:** Our proposed artifact comprises a deep feedforward neural network for generating predictions, Monte Carlo (MC) dropout to estimate model uncertainty, and ICE plots and PDP approach to generate local and global explanations, respectively. The study's novelty lies in incorporating uncertainty information into these visualization-based post-hoc explanation approaches. With such integration, decision-makers can not only justify the plausibility of explanation-driven actionable insights but also validate their reliability by examining the model confidence.

**Demonstration:** The applicability of the proposed artifact has been examined for a predictive process monitoring problem. Predictive process monitoring

is a branch of process mining that combines advanced computational intelligence methods with process modeling approaches [10]. The objective is to enable continuous business process improvement by extracting predictive, data-driven, process-specific insights from the event logs generated by a process-aware information system (PAIS) [1]. Event logs are an essential enabler for evidence-based process analysis by providing necessary details about process execution.

More specifically, we address a real-world use case scenario in the manufacturing domain. The examined problem pertains to cycle time prediction, with a focus on predicting the duration of individual manufacturing activities required to fulfill orders. The data is obtained from the Manufacturing Execution Systems (MES) of the consortium partner. Prior to implementing our proposed artifact, we conducted a rigorous feature engineering process using process-specific data that had been enriched with customer order data.

**Evaluation:** The evaluation phase involves conducting semi-structured interviews with two domain experts who provide critical and constructive feedback on the system design, usability, and suggestions for improvements. The evaluation results provide valuable insights to refine the system and improve its usability.

**Communication:** Finally, the communication phase involves sharing the findings and approach details through scientific publications and industrial events to a broad audience of researchers and practitioners from different backgrounds.

Through the use of DSR, we can ensure a methodologically sound and systematic process for conducting applied research that is relevant to the application domain. Ultimately, the XAI solution developed using this approach, with a focus on communicating uncertainty, is expected to enhance the trustworthiness of AI systems.

## 4 Uncertainty Estimation in Post-hoc Explanations

In this section, we provide an overview of the mathematical foundations that underpin our proposed novel uncertainty-aware XAI approach.

### 4.1 Uncertainty Quantification with Monte Carlo Dropout

The MC dropout technique is among the cutting-edge approaches for assessing uncertainty within deep learning frameworks and is applied to neural networks to primarily mitigate overfitting through the stochastic deactivation of neurons [12]. Enabling dropout regularization during the deployment phase allows the approach to be interpreted as a Bayesian approximation to the probabilistic deep Gaussian process. By performing  $T \in \mathbb{N}^+$  stochastic forward-passes through the network, the model’s predictive variance can be calculated and, in turn, serves as a measure of uncertainty.

We realize MC dropout by performing  $T \geq 50$  stochastic forward passes through our deep feedforward neural network. This approach allows the model’s



predictions to be mapped to the corresponding variance. In particular, uncertainty profiles can be created for the training data by first sorting the variances in ascending order, then calculating the variance thresholds. For example, thresholds for the 25th and 75th percentile can be calculated, resulting in three uncertainty profiles. The utilization of percentile-based estimations provides an initial foundation for the categorization of model confidence profiles. Ultimately, the domain experts hold the responsibility of determining the final profiles, either by refining the initial data-driven estimations or by defining their own ranges or categories.

## 4.2 Enhancing Partial Dependence Plots with Uncertainty Quantification

Incorporating Uncertainty Quantification (UQ) into Partial Dependence Plots (PDPs) provides an enhanced perspective on the intricate relationship between predictor variables and the target outcome within a model framework. As initially introduced by Friedman [11], PDP serves as a global explanation tool, illustrating the dependency of the target variable's predictions on variations within predictor variables. This method involves generating a new dataset by substituting the values of a set of predictors with those from a specific instance, and then predicting outcomes with the model for this modified dataset. Averaging these predictions across all observations of the feature and plotting these averages against the feature values produce the PDP.

The methodology commences with a dataset  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ , encompassing pairs of predictor variables  $\mathbf{x} = (x_1, \dots, x_p)$  and a response variable  $y$ , with  $N$  indicating the total number of observations. This dataset is segregated into subsets designated for training ( $D_{train}$ ), validation ( $D_{val}$ ), and testing ( $D_{test}$ ), utilized respectively for the purposes of model training, hyperparameter optimization, and evaluation. The predictive function  $\hat{F}(\mathbf{x})$ , trained on the  $D_{train}$  subset, maps predictors to the response. To generate a PDP for selected predictor variables  $\mathbf{x}_S$  and their complement  $\mathbf{x}_C$ , the partial dependence of the model's prediction on  $\mathbf{x}_S$  is articulated through an expectation over  $\mathbf{x}_C$ , integrating both the joint and marginal probability densities of these variables.

The integration of UQ into PDP considers the unique observed values of the predictors  $\mathbf{x}_S$  within the training data. For each value, a modified copy of  $D_{train}$  is created where  $\mathbf{x}_S$  values are replaced by a particular value  $\mathbf{x}_{S,k}$ . The average prediction for this altered dataset is computed alongside a variance vector through multiple stochastic forward passes, reflecting the uncertainty in predictions. This variance information is utilized to construct a visual representation of uncertainty, with the resulting PDP plots colored according to identified uncertainty categories. These plots convey the mean effect of selected features on model predictions and enrich the user's insight by detailing the variance distribution and an uncertainty profile for each plot point, facilitating a deeper exploration of prediction uncertainties.

This refined approach to PDP, augmenting Friedman's original methodology by weaving in UQ, aims to unveil a more comprehensive depiction of the relation-

ship between selected features and the target variable. Additionally, it tackles the challenge that Goldstein et al. [14] posed in capturing local feature-target relationships, advocating for the application of UQ in Individual Conditional Expectation (ICE) plots for an in-depth analysis of these interactions.

### 4.3 Enhancing Individual Conditional Expectation Plots with Uncertainty Quantification

As introduced by Goldstein et al. [14], an Individual Conditional Expectation (ICE) plot offers a detailed global explanation technique by examining the impact of marginal changes in a single feature on the predictions made by a fitted model. This method involves selecting a subset of predictor variables, replicating a specific observation across an artificial dataset multiple times, and systematically altering the values of the chosen predictors to cover their unique observed range. The model then evaluates this dataset, generating a set of prediction scores that correlate with the unique values of the selected predictors, thus facilitating a visual representation of their relationship. If the focus is on a single predictor, this relationship might typically be illustrated using a line plot. Extending this approach to encompass all observations within the dataset allows for the addition of multiple lines to the plot—one for each observation. The average of these lines yields the Partial Dependence Plot (PDP) for the selected features, situating the ICE plot as a method for local explanation when applied to individual observations.

To further this approach by incorporating Uncertainty Quantification (UQ), we propose a refined methodology tailored for local explanations that accentuates the predictive uncertainty for a singular observation and predictor of interest. Consider  $x_S$  as the predictor under scrutiny, characterized by its unique observed values  $\{x_{S,1}, \dots, x_{S,j}\}$ . With  $j$  representing the count of unique values within the training data, and  $(x_S^{(i)}, \mathbf{x}_C^{(i)})$ ,  $i \in \{1, \dots, N\}$  signifying a selected observation from the dataset, the process unfolds as follows: For each unique value  $x_{S,k}$ , a duplicate of the chosen observation is created with  $x_S^{(i)}$  updated to  $x_{S,k}$ . This allows for the calculation of the model's prediction for the modified observation, alongside the variance  $v_k^{(i)}$  across  $T$  stochastic forward passes, encapsulating the predictive uncertainty. The association of  $v_k^{(i)}$  with a specific uncertainty profile is denoted, facilitating the coloring of the plotted pairs  $\{x_{S,k}, \widehat{F}(x_{S,k}, \mathbf{x}_C^{(i)})\}$  based on their uncertainty categorization.

This methodology culminates in the generation of an ICE plot, uniquely colored to reflect the uncertainty associated with each prediction. Additionally, it introduces a novel visual representation through stacked histograms, which categorize and color feature values and predictions according to their uncertainty profile alignment, thereby offering an insightful and comprehensive examination of the model's predictive behavior under uncertainty.

## 5 Interface Overview

The main focus of this section is to introduce the primary interface components of the solution that are specifically designed to effectively communicate uncertainties arising with each model decision, along with corresponding explanations. Apart from a component for a general overview and instance selection (Fig. 1), the three interface components that make up the solution are “Uncertainty Estimation and Visualization,” “Uncertainty Communication in ICE Plots,” and “Uncertainty Communication in PDP.”

The core ML components for this project are built using the “*keras*” library in R. Other key libraries utilized for data preparation, interface creation, and visualization include “*data.table*”, “*dplyr*”, “*ggplot2*”, “*plotly*”, “*vip*”, “*shiny-dashboard*” etc.

### 5.1 Uncertainty Estimation and Visualization

The “Uncertainty Estimation and Visualization” component of our proposed solution is designed to inform system users about model uncertainty using various presentation forms. More specifically, two distinct visualization techniques were utilized, a density plot and a box plot, along with textual and tabular descriptions that convey information on the specific model uncertainty for the selected prediction.

To begin, the first visualization approach presents the distribution of possible model predictions generated using the chosen UQ approach (such as MC dropout) through density plots (Fig. 2, A). This allows users to visually inspect distribution details and understand the ranges where the model predictions are predominantly located. Alternatively, we can use a box plot to visualize the same information (Fig. 2, C), showing the interquartile range within its hinges, a vertical line representing the median, and whiskers extending to the lowest and highest data points within 1.5 times the interquartile range. Both visualization approaches are supplemented with additional information. For instance, prediction intervals are incorporated into the plots, showing the range within which the predictions will fall with a 95% probability. A label below each plot includes the confidence interval as a visual aid, depicted using arrows pointing at the red vertical lines. The plots are colored based on the qualitative uncertainty descriptions, with green, yellow, and red representing “low,” “medium,” and “high” confidence profiles, respectively.

In addition to these visualization approaches, we provide a textual description (Fig. 2, B) that includes information about the UQ method, an explanation of prediction intervals in words, and the uncertainty profile that shows the model confidence for the particular prediction. Finally, a table is presented to the user (Fig. 2, D), showing the model prediction, standard deviation, and other relevant information.

## 5.2 Uncertainty Communication in Individual Conditional Expectation Plots

This component of our proposed uncertainty-aware local explanation approach provides an interface that visualizes prediction scores for new synthetic instances and communicates their uncertainty information (Fig. 3, B for numerical and Fig. 4, B for categorical variables). This is achieved through the use of color-coded confidence intervals within the plot. In addition, the uncertainty for each synthetic instance is communicated through the various presentation forms described in the previous component.

## 5.3 Uncertainty Communication in Partial Dependence Plots

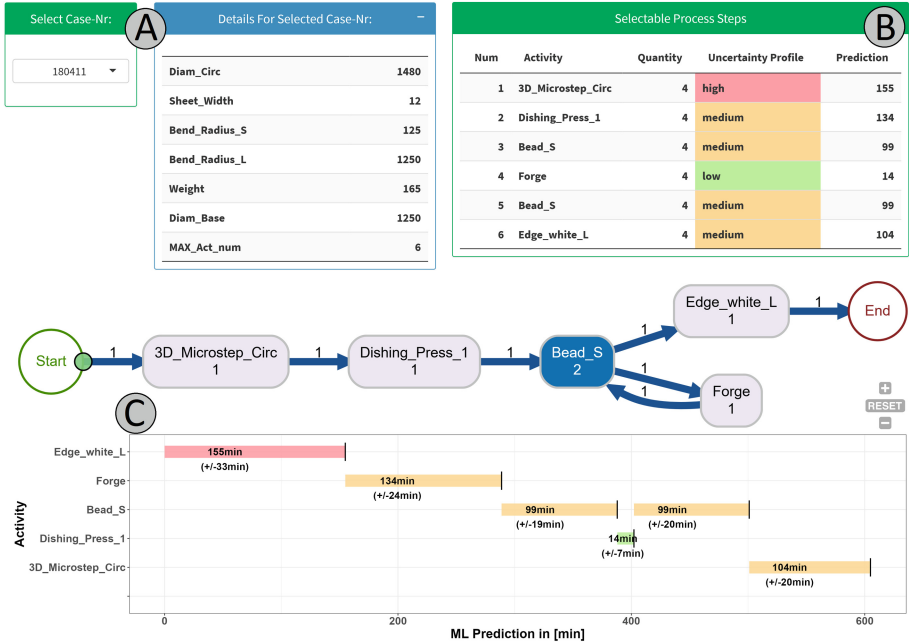
To improve the transparency and interpretability of the PDP, we introduce a new component for the PDP interface that provides two types of uncertainty information (Fig. 5, B). The first type of uncertainty information is presented through a complementary density plot (Fig. 5, C), which displays the distribution of predictions and the 95% confidence level bands around them. While this information could be directly visualized in the main graph, we found that doing so would make it harder for users to read and interpret the plot. The main contribution of our approach is the visualization of model uncertainty for each examined value of the feature of interest within PDP analysis. To achieve this, we use a doughnut chart (Fig. 5, D) that displays the distribution of uncertainty profiles, giving users an overview of the model's global reliability for the examined value.

# 6 Evaluation

## 6.1 Usage Scenario

The effectiveness of the designed interface, which includes visualization analytics components, is showcased in this usage scenario. The interface assists process experts (PE) in estimating the cycle time required for a given customer order. The sequence of required production activities is already predetermined based on the order specifications. However, the PE must still determine the duration of each activity, which can be aided by our data-driven approach. As the timely completion of high-priority orders is critical, the PE are responsible for ensuring the validity of the data-driven guidance provided by the system.

**General Overview:** The system interaction commences with the PE being directed to a dedicated “General Overview” page exclusively designed for the predictive process monitoring use case. The PE can select the relevant case of interest on this page by using a drop-down menu corresponding to a customer order's production activity sequence (Fig. 1, A). The page also highlights the specifications of the customer order, such as product weight and dimensions,

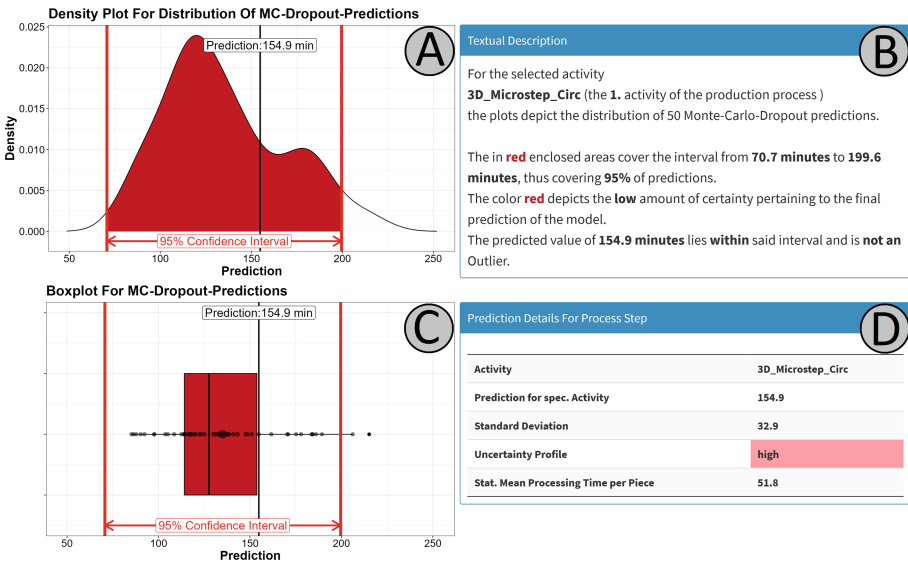


**Fig. 1.** General overview. A: The drop-down menu on the upper left-hand side allows the user to select a production case; corresponding details are displayed on the right-hand side. B: A table view allows further insight into each step of the production case and information concerning activity duration prediction and uncertainty. Process activities are selectable for further analysis. C: An animated process map and a Gantt-Chart depict the planned sequencing and cycle time prediction. (Color figure online)

which are also used as inputs to the ML algorithm. Figure 1, B shows all pertinent information regarding duration prediction for each production step, along with uncertainty profile information. An animated process map and a Gantt-Chart (Fig. 1, C) provide a visual representation of the activity sequence, with the latter featuring the predicted duration for each process step. Each step in the Gantt-Chart is color-coded according to one of three uncertainty categories: “high” is indicated by the color red, “medium” by the color orange, and “low” by the color green.

The production activities in this particular scenario include 3D-cutting (“3D\_Microstep\_Circ”), work at a dishing press (“Dishing\_Press\_1]), beading (“Bead\_S”), shape adjustments in the forge (“Forge”), another beading (“Bead\_S”) and refinement of the edges (“Edge\_white\_L”). Our solution categorizes activities 2, 3, 5, and 6 as having a “medium” uncertainty profile, whereas activity 3 has a “low” uncertainty profile. The first activity, 3D-cutting, is an exception and falls under the “high” uncertainty group. It is predicted to take 155 min with a standard deviation of 33 min, as shown in the Gantt-Chart. The

PE decides to investigate this production activity since any disruption at this early stage can potentially cause a cascading effect on the rest of the processes.



**Fig. 2.** Dashboard interface for uncertainty analysis on an instance-level. A: The density plot depicts the distribution of MC Dropout predictions. B: The same data from A is visualized as a box plot. C: A textual description summarizes important findings from A and B. D: A table displays additional information concerning the examined production activity. The red color coding depicts the “high” level of uncertainty affiliated with the activity duration prediction. (Color figure online)

**Uncertainty Analysis on the Activity Level:** The PE can analyze the uncertainty of individual activities by clicking on the activity of interest, in this case, 3D-cutting. This action displays the distribution of MC dropout predictions as shown in Fig. 2. The model prediction of 154.9 min falls into the highlighted confidence interval of 70.7 min to 199.6 min (Fig. 2, A), which covers 95% of the values and indicates that the model prediction is not an outlier. However, the upper hinge of the box plot (Fig. 2, C) and the upper limit of the confidence interval suggest that a delay of nearly 45 min is not unlikely. A textual description (Fig. 2, B) is also provided to ensure correct interpretation. Finally, Fig. 2, D provides a tabular summary for quick access to duration predictions and uncertainty information for the chosen activity.

**Uncertainty-Informed ICE Plots:** To understand the impact of both numerical and categorical features on model predictions and generate a plan to avoid undesired outcomes, the PE consults the uncertainty-aware ICE plot (Fig. 3, A).

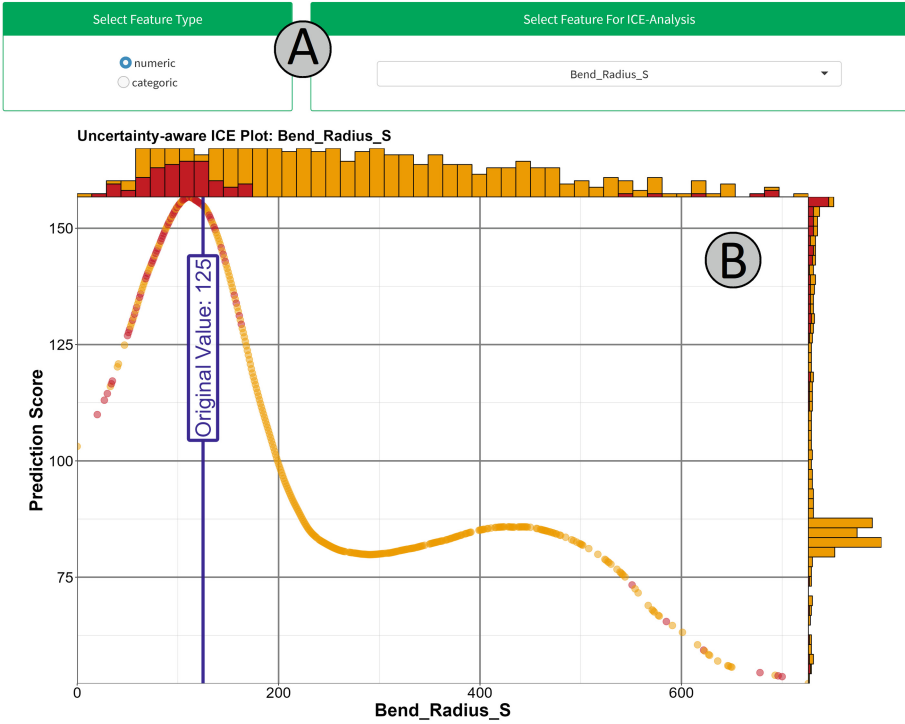
By analyzing variables related to the product specifications, such as the bend radius (“Bend\_Radius\_S”), the PE realizes that the duration prediction of 3D-cutting activity around the original value (125 cm) results in “high” uncertainty predictions (Fig. 3, B). However, increasing the value of this feature reduces the predicted activity duration and increases the model’s confidence in its predictions, resulting in “medium” uncertainty. This uncertainty-aware ICE plot enables the PE to understand the relationship between the feature of interest and model outcomes and comprehend the model confidence. However, the value of this feature can not be altered for the planning process as it is a fixed, pre-determined product specification.

The PE identifies a categorical variable, “Worker,” that can be manipulated without affecting the order requirements (Fig. 4, A) and filters out unavailable personnel. In this scenario, the PE notices that the prediction for the allocated worker by the system (anonymized through the identifier “751”) falls into the “high” uncertainty category (Fig. 4, B), while other workers would be available with a “medium” uncertainty prediction. Within the area with the lowest prediction scores, the PE chooses a group of three other available workers (“114”, “736”, and “797”) whose predicted duration falls into the “medium” uncertainty group to examine them further. The anticipated durations for these three workers are  $\sim 103$ ,  $\sim 118$ , and  $\sim 123$  minutes for “797”, “736,” and “114,” respectively.

The PE validates the accuracy of the model outputs by cross-referencing them with their domain expertise. The improvement in predictions resulting from the alteration of the selected worker is attributed to the greater experience of the alternative workers in performing the production activity. Since the model’s estimated prediction duration and uncertainty align with the PE’s expectations, their confidence in its reliability increases.

**Uncertainty-Informed PDP:** The PE switches to the “Global Explanations” tab to ensure that at least one of the selected replacement workers performs well in general, given the high priority of the order. This tab provides the PE with a permutation-based feature importance plot (Fig. 5, A) which helps in understanding the overall impact of certain variables. Additionally, an uncertainty-aware PDP (Fig. 5, B) is also presented to the PE as a tool for further analysis.

The PE selects the categorical variable “Worker” and iteratively examines each of the three available workers. By using the distribution of the MC dropout predictions (Fig. 5, C), the PE concludes that worker “797” has an upper boundary of the confidence interval that is approximately 30 min lower than the currently allocated worker, with a mean prediction score (82 min) that is 22 min lower. Furthermore, the doughnut chart (Fig. 5, D) indicates that worker “797” is associated with a greater amount of “low” (33.1%) and a smaller amount of “high” (16.6%) uncertainty when compared to the current worker (21.2% “low,” 22.7% “high” uncertainty). Consequently, the PE has sufficient grounds to modify the production plan by substituting the current worker “751” with worker “797” for the given process activity, reducing the predicted lead time and decreasing model uncertainty.



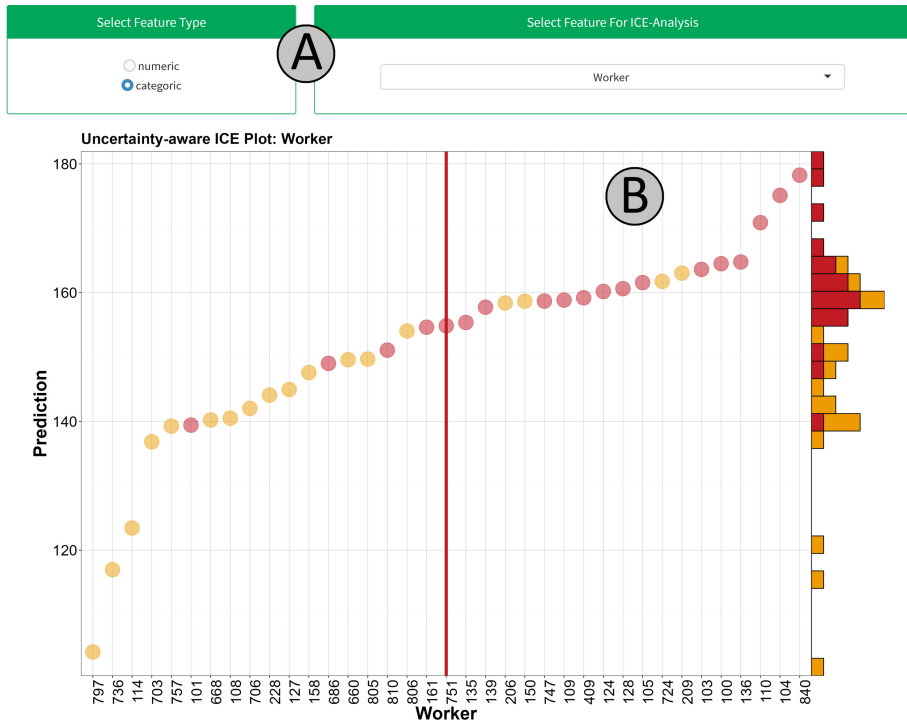
**Fig. 3.** Dashboard interface for uncertainty-aware ICE plots. A: The user selects “numerical” as the variable type on the upper left-hand side, updating the drop-down menu with variables of the chosen type right next to it. B: The proposed uncertainty-aware ICE plot for numerical variables is displayed here.

### 6.2 Expert Interview

During the evaluation of the proposed uncertainty-aware explanation methods, an interview was conducted with two process experts, including a factory manager and a production manager. Both experts possess extensive knowledge of the underlying data and have a deep understanding of the processes involved, as well as expert knowledge regarding the interrelationships between the features under examination and the target.

During the interview, the process experts selected an exemplary Case-ID and explored the visualizations for the corresponding production process to establish a starting point for further analysis. They quickly orientated themselves within this tab and delved into the analysis of individual production steps. Within the exemplary case, they chose a process step and explored the uncertainty analysis in detail. Following that, uncertainty-aware ICE plots were presented and discussed in a similar manner. The global explanations, which contain the uncertainty-aware PDP, were examined next. These steps were repeated for the other process steps within the exemplary case. Next, they performed the same



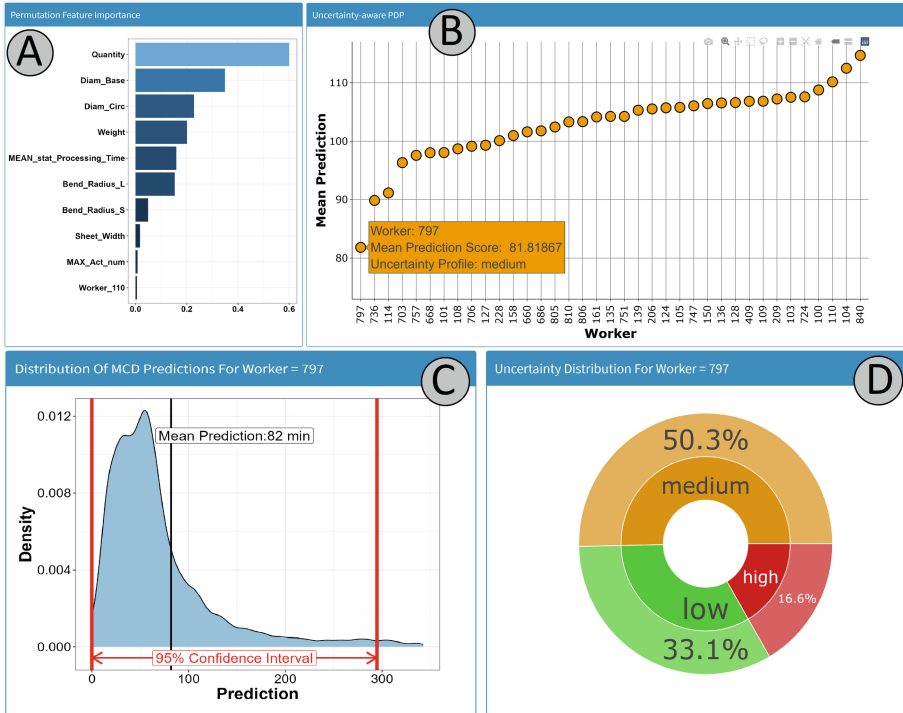


**Fig. 4.** Dashboard interface for uncertainty-aware ICE plots. A: The user selects “categorical” as the variable type on the upper left-hand side, updating the drop-down menu with variables of the chosen type right next to it. B: The proposed uncertainty-aware ICE plot for categorical variables is displayed here, with the distribution of categorical values being omitted since each value only occurs once during plot generation. The red vertical line indicates the original variable value of the analyzed instance. (Color figure online)

steps for other randomly chosen cases and were asked to provide feedback on how they would interact with the dashboard if it were deployed for production planning. Finally, they discussed the dashboard in detail, and the expert feedback was documented. The interview lasted for one hour, and each step described above took approximately 15 min. The interviewed experts provided valuable feedback on the usability and design of the visualization and explanation techniques employed in the proposed uncertainty-aware explanation methods and proposed suggestions for improvement.

**Design and Usability:** In terms of design, the experts expressed positive views toward the proposed uncertainty-aware ICE plots and uncertainty-aware PDP and were able to derive and validate the relationships depicted in these visualizations shortly after their introduction. Furthermore, the integrated color scheme

for faster differentiation of uncertainty profiles was internalized quickly and utilized by the experts as they explored real-world usage scenarios in the third interview step.



**Fig. 5.** Dashboard interface for uncertainty-aware PDP. A: Permutation-based feature importance is displayed as a supporting tool for orientation and global explainability. B: The proposed uncertainty-aware PDP for categorical variables is displayed here. Hovering over a point displays its mean prediction score and the corresponding dominant uncertainty group. C: Clicking on a point in B updates this density plot, which displays the distribution of predicted values. D: Clicking on a point in B updates this doughnut chart, which displays the distribution of uncertainty group membership from the corresponding predicted values.

The density plot was rated as the most accessible and effective visualization method for the distribution of MC dropout predictions, followed by the box plot as a complementary visualization. The additional textual explanations were highlighted as a powerful tool for preventing false interpretations and improving overall user acceptance. The experts found the relationship between the prediction score and the distribution of MC dropout predictions easy to grasp after a brief initial explanation.

**Suggestions for Improvement:** While the experts were generally satisfied with the proposed explanation methods, they expressed a desire for the ability

to compare similar cases. This would enable users to filter the underlying data to construct uncertainty-aware ICE plots and uncertainty-aware PDP restricted based on the filtered data. Incorporating this feature into the proposed dashboard would enhance the user experience and improve the utility of the proposed explanation methods for production planning.

## 7 Discussion and Conclusion

In this work, we presented an approach for integrating and communicating model uncertainty in the context of XAI, focusing on visualization as a medium for conveying information. In particular, we introduced uncertainty-aware ICE plots as a local and uncertainty-aware PDP as a global explanation method, enhanced with various visual properties and functionalities for deeper uncertainty analysis. We presented the efficacy of this approach in a real-world manufacturing scenario, demonstrating its utility in the hands of a process expert. Additionally, an interview with experts evaluated the effectiveness and usability of the presented methods when integrated into a prototypical dashboard. Below we discuss findings, challenges, and future work.

**Flexibility and Transferability:** Our study employed a deep neural network and utilized MC dropout to quantify uncertainties in the model's predictions accurately. The use of data-driven methodologies in these steps is not limited to MC dropout and can be interchanged with other methods, such as Extreme Gradient Boosting (XGBoost) for generating predictions or the bootstrap approach for uncertainty quantification. The flexibility of our approach makes it applicable to a wide range of classification or regression challenges that involve tabular data, not just limited to predictive process monitoring scenarios.

**Combining Local and Global Explanations:** The integration of instance-based and global explanations have been found to increase overall trust and acceptance of AI models. By combining global and local explanations, users can gain a high-level overview of the model and the ability to delve into the details of specific instances. Compared to providing only global or local explanations, this approach is considered more effective in promoting a better understanding and trust in AI models. Our evaluation further underscores the importance of using both explanations for exploring model uncertainty.

**Scalability:** The integration of uncertainty UQ and XAI techniques, such as MC Dropout and ICE and PDP, presents significant computational challenges, particularly concerning scalability. For example, the complexity of generating ICE and PDP plots for models with high-dimensional inputs can result in enormous evaluations, which can be prohibitively expensive when MC Dropout generates multiple predictions for each input. In addition, the computational cost of combining these techniques can make it difficult to scale to larger datasets or more complex models. To mitigate these challenges, several approaches have been proposed. One potential solution is to parallelize computations across multiple GPUs, which can lead to significant speedups. This approach is practical for MC Dropout, where the numerous forward passes required for each input can be

efficiently distributed across multiple processors. Additionally, binned values can be used for XAI techniques instead of all unique values, which can significantly reduce the number of evaluations required.

**Future Work:** A promising direction for future research involves enhancing the dashboard by integrating uncertainty reliability measures, such as Prediction Interval Coverage Probability (PICP) and Mean Prediction Interval Width (MPIW). These measures are crucial in scenarios requiring high-stakes decision-making, where formal guarantees on prediction intervals are indispensable. In our forthcoming work, we aim to incorporate conformal prediction techniques into the UQ component of our proposed approach. Furthermore, exploring various ad-hoc UQ methods and post-hoc calibration techniques in the XAI context could prove beneficial. Additionally, expanding the dashboard to encompass model accuracy metrics like Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for different uncertainty profiles will provide users with a deeper insight into model performance across various levels of uncertainty. Moreover, there's a pressing need for more systematic approaches to evaluate the combination of XAI and UQ, both qualitatively and quantitatively. Investigating the synergies between UQ and XAI methods could shed light on the fairness of algorithmic decision-making in high-stakes contexts. Given the risks of bias and discrimination inherent in AI-based decision systems, this area of research carries profound social and ethical significance. We aim to deepen our understanding of the performance and reliability of UQ and XAI methods by pursuing these avenues. This knowledge will be crucial in developing more robust and equitable decision-making systems.

**Acknowledgement.** The results of this publication were achieved in part by the German Federal Ministry of Education and Research under grant numbers 01IS21006B (project ExPro), 01IS24048C (project EINHORN), and within the AI initiative GreenAI Hub Mittelstand (<https://greenai-hub.de>) of the Federal Ministry for the Environment, Nature Conservation, Nuclear Safety and Consumer Protection (BMUV).

**Disclosure of interest.** The authors have no competing interests to declare that are relevant to the content of this article.

## References


1. van der Aalst, W.: Process mining: overview and opportunities. *ACM Trans. Manag. Inf. Syst. (TMIS)* **3**(2), 1–17 (2012)
2. Alicioglu, G., Sun, B.: A survey of visual analytics for explainable artificial intelligence methods. *Comput. Graph.* **102**, 502–520 (2022)
3. Arrieta, A.B., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* **58**, 82–115 (2020)
4. Bhatt, U., et al.: Uncertainty as a form of transparency: measuring, communicating, and using uncertainty. In: *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 401–413 (2021)
5. Cafri, G., Bailey, B.A.: Understanding variable effects from black box prediction: quantifying effects in tree ensembles using partial dependence. *J. Data Sci.* **14**(1), 67–95 (2016)

6. Chatzimparmpas, A., Martins, R.M., Jusufi, I., Kerren, A.: A survey of surveys on the use of visualization for interpreting machine learning models. *Inf. Vis.* **19**(3), 207–233 (2020)
7. Chatzimparmpas, A., Martins, R.M., Jusufi, I., Kucher, K., Rossi, F., Kerren, A.: The state of the art in enhancing trust in machine learning models with the use of visualizations. In: *Computer Graphics Forum*, vol. 39, pp. 713–756. Wiley Online Library (2020)
8. Choo, J., Liu, S.: Visual analytics for explainable deep learning. *IEEE Comput. Graph. Appl.* **38**(4), 84–92 (2018)
9. Doula, A., Schmidt, L., Mühlhäuser, M., Guinea, A.S.: Visualization of machine learning uncertainty in AR-based see-through applications. In: *2022 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 109–113. IEEE (2022)
10. Fettke, P.: Conceptual modelling and artificial intelligence: overview and research challenges from the perspective of predictive business process management. In: *Companion Proceedings of Modellierung 2020 Short, Workshop and Tools and Demo Papers co-located with Modellierung 2020, Vienna, Austria, 19–21 February 2020*, pp. 157–164 (2020)
11. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**(5), 1189–1232 (2001)
12. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning. ICML'16*, vol. 48, pp. 1050–1059. JMLR.org (2016)
13. Gawlikowski, J., et al.: A survey of uncertainty in deep neural networks. *arXiv abs/2107.03342* (2021)
14. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Stat.* **24**(1), 44–65 (2015)
15. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM Comput. Surv. (CSUR)* **51**(5), 1–42 (2018)
16. Hullman, J., Qiao, X., Correll, M., Kale, A., Kay, M.: In pursuit of error: a survey of uncertainty visualization evaluation. *IEEE Trans. Vis. Comput. Graph.* **25**(1), 903–913 (2018)
17. Islam, M.R., Ahmed, M.U., Barua, S., Begum, S.: A systematic review of explainable artificial intelligence in terms of different application domains and tasks. *Appl. Sci.* **12**(3), 1353 (2022)
18. Maaack, R.G., Scheuermann, G., Hagen, H., Peñaloza, J.T.H., Gillmann, C.: Uncertainty-aware visual analytics: scope, opportunities, and challenges. *Vis. Comput.* **39**(12), 6345–6366 (2023)
19. Mehdiyev, N., Fettke, P.: Explainable artificial intelligence for process mining: a general overview and application of a novel local explanation approach for predictive process monitoring. In: *Interpretable Artificial Intelligence: A Perspective of Granular Computing*, pp. 1–28 (2021)
20. Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., Bischl, B.: Explaining hyperparameter optimization via partial dependence plots. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 2280–2291 (2021)
21. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. *J. Manag. Inf. Syst.* **24**(3), 45–77 (2007)

22. Slack, D., Hilgard, A., Singh, S., Lakkaraju, H.: Reliable post hoc explanations: modeling uncertainty in explainability. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 9391–9404 (2021)
23. Tomsett, R., et al.: Rapid trust calibration through interpretable and uncertainty-aware AI. *Patterns* **1**(4), 100049 (2020)
24. Zhang, Y., Liao, Q.V., Bellamy, R.K.: Effect of confidence and explanation on accuracy and trust calibration in AI-assisted decision making. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 295–305 (2020)



# XAI for Time Series Classification: Evaluating the Benefits of Model Inspection for End-Users

Brigt Håvardstun<sup>1</sup>, Cèsar Ferri<sup>2</sup>, Kristian Flikka<sup>3</sup>,  
and Jan Arne Telle<sup>1</sup>

<sup>1</sup> Department of Informatics, University of Bergen, Bergen, Norway  
{brigt.havardstun, Jan.Arne.Telle}@uib.no

<sup>2</sup> VRAIN, UPV-Polytechnic University of Valencia, Valencia, Spain  
cferri@dsic.upv.es

<sup>3</sup> Eviny AS, Bergen, Norway  
Kristian.Flikka@eviny.no

**Abstract.** We present an XAI tool for time series classification providing model-agnostic instance-based post-hoc explanations, by means of prototypes and counterfactuals. Additionally, our tool allows for model inspection on instances generated by the user, to navigate the boundary between classes. This will allow the user to test and improve their hypotheses when formulating a model of the black box classification. We perform a human-grounded evaluation with forward simulation, to contribute a quantitative end-user evaluation to the field of XAI for time series.

**Keywords:** Time series · Human-grounded evaluation · Instance-based explanations

## 1 Introduction

Temporal data is encountered in many real-world applications ranging from patient data in healthcare [15] to the field of cyber security [20]. Deep learning methods have been successful in time series classification [11, 15, 20], but such methods are not easily interpretable, and often viewed as black boxes, which limits their applications when user trust in the decision process is crucial. To enable the analysis of these black-box models we revert to post-hoc interpretability. Recent research has focused on adapting existing methods to time series, both specific methods like SHAP-LIME [8], Saliency Methods [10] and Counterfactuals [4], and also combinations of these [16].

However, compared to images and text, time series data are not intuitively understandable to humans [18]. This makes interpretability of time series extra demanding, both when it comes to understanding how users will react to the provided explanations and also to the evaluation of its usefulness. Nevertheless,

as humans learn and reason by forming mental representations of concepts based on examples, and any machine learning model has been trained on data, then data e.g. in the form of prototypes and counterfactuals is indeed the natural common language between the user and this model. In addition, several studies have highlighted the need to rethink new ways of interaction with an XAI algorithm, to allow for a dialogue between explainer and explainee, and to enable model inspection at will [1, 13, 17].

Hence, we advance an XAI time series tool that on the one hand provides users with instances of both prototype and counterfactual time series, and on the other hand lets the user generate their own instances for classification. This enables active learning and allows the user to test and improve their hypotheses when formulating a mental model of the black box classification.

We perform a quantitative user evaluation, thereby meeting a demand from the XAI research community [4, 21, 22], to measure how prototypes, counterfactuals and interactivity increases the understanding that the user has of the black box classification. Using the taxonomy of interpretability evaluation [5] what we do is a Human-Grounded Evaluation with Forward Simulation.

We use 3 datasets having a binary classification (e.g. 24-h power demand of a household in Winter versus Summer) and for each dataset we train an ML model. Note that it is generally more difficult to explain the classification of an AI model than the classification of a real-world dataset. The reason for this is that the dataset contains only real instances, whereas the AI model classifies *any* instance, also artificial ones. In this work, for prototypes we use real instances from the dataset, whereas we produce counterfactuals by combining real instances with artificial ones, based on the NativeGuide method [4]. For tests we have chosen to use real instances.

In the remainder of this paper, we first discuss related work, then we give definitions, followed by a description of the tool and a discussion of the user evaluation results.

## 2 Related Work

Research on XAI for time series classification has progressed similarly to XAI in general, with earlier work focused on feature-importance rather than on instance-based methods using prototypes and counterfactuals. A comprehensive survey of XAI methods for time series can be found in [21]. Our own work deals with model-agnostic instance-based post-hoc explanations, by means of prototypes and counterfactuals. In [14], the author offers a survey of work on instance-based explanations for XAI mainly in the image domain, defining and classifying the various approaches in the literature. In this paper we evaluate the use of instance-based explanations for time series by end-users. The work [7] studied how non-experts handled post-hoc example-based explanations, however not in the time series domain. They found that even though these do assist users with correct judgement, people have significant difficulties dealing with misclassifications in an unfamiliar domain. Thus we should maybe not expect very high accuracy



from our own evaluations on end-users, as time series are notoriously hard for humans to interpret [18].

Interactivity and model inspection have recently been seen as important in XAI. The paper [2] argues that interactivity in XAI is a core value in the interface between the model and the user, and that a user study is needed for a qualitative evaluation. In [22], the authors develop an interactive XAI tool for loan applications that allows users to experiment with hypothetical input values and inspect their effect on model outcomes, and perform a user evaluation on MTurk.

User evaluations of XAI systems come in various forms. A taxonomy of interpretability evaluation, from the gold standard of Application-oriented evaluations, to Human-grounded evaluations as we perform in this work, to Functionally-grounded evaluations that do not require human experimentations is developed in [6]. For time series it seems the latter approach is the more common. The authors in [16] apply several XAI methods previously used on image and text domain to time series, and introduces verification techniques specific to times series, in particular a perturbation analysis and a sequence evaluation, but they do not include any user evaluation of their systems. Likewise, [9] presents a Python package to provide a unified interface to interpretation of time series classification, but no user evaluation.

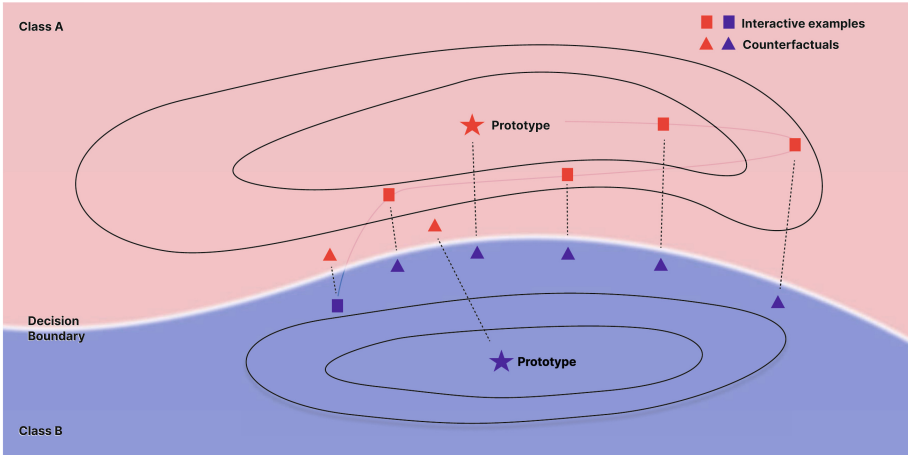
The work of [4] provides a method for generating counterfactuals for time series classifiers, called Native Guide, that applies Class Activation Mappings [23] to select discriminative areas for modification. They end their paper by arguing that ‘Given the ubiquitous nature of time series data and the frequent requirement for explanation, it is clear that experiments with human users and CBR solutions have much to offer in future work.’ The survey [21] says about Native Guide that ‘...evaluating this promising approach involving end users could be promising for future work’. Indeed this is a part of what we do in the current paper as the counterfactuals we show users are exactly the ones generated by Native Guide.

### 3 Definitions

Let us present formal definitions for Time Series Classification (TSC) and recall basic notions.

Staying consistent with earlier notation [4,21] a time series  $T = \{t_1, t_2, \dots, t_m\}$  is an ordered set of  $m$  real-valued observations (or time steps). A time series dataset  $D = \{T_1, T_2, \dots, T_n\} \in R^{n \times m}$  is a collection of such time series where each time series has a class label  $c$  forming a vector of class labels. In this paper we consider only binary classification tasks. Given such a dataset, Time Series Classification is the task of training a mapping  $b$  from the space of possible inputs to a probability distribution over the class values. Thus, a black-box classifier  $b(T)$  takes a time series  $T$  as input and predicts a probability output over the class values. Given a to-be-explained time series  $T$ , with predicted label  $b(T) = c$  from the black-box classifier, a counterfactual explanation aims to find

how  $T$  needs to (minimally) change to some  $T'$  for the system to classify it alternatively, as  $b(T') = c' \neq c$ . We refer to  $T'$  as a counterfactual explanation for  $T$ , without having to specify the target class since we consider only binary classification tasks. The minimality criterion usually refers to a notion of distance (proximity) between time series, but another criteria property can be sparsity (that  $T$  and  $T'$  differ on few data points, or on few contiguous sequences of data points) and plausibility (that the instance is not an outlier).



**Fig. 1.** Explanation types illustrated in a two-class decision space. The two prototypes are representatives of each class, positioned in the middle of the data density contour map. A path of interactive examples and their counterfactuals is shown to illustrate a possible user's journey.

Prototypes are time series exemplifying the main aspects responsible for a classifier's specific decision outcome. It can be a real instance (which is what we opt for in our tool) sampled from the dataset that is important and meaningful because it summarizes the shape of many other similar instances, or a synthetic one, for example a cluster centroid or an instance generated by following some ad-hoc processes. See Fig. 1.

## 4 Prototypes, Counterfactuals, and User-Model Inspection

In this section we introduce our tool for XAI on time series classification providing model-agnostic instance-based post-hoc explanation, by means of prototypes and counterfactuals. Additionally, one of our contributions is to allow for model inspection on instances generated by the explaine. This interactive tool allows the user to themselves navigate the boundary between classes. Starting from a

prototype, while simultaneously seeing a counterfactual (see Fig. 1) the user can change individual time points at will, and see the resulting classification. This will allow the user to test and improve their hypotheses when formulating a model of the black box classification.

In the rest of this section we describe our tool and start by giving an argument for its motivation, coming from the industry side. We then give a description of the algorithms for generating prototypes plus counterfactuals, and a discussion of how to present these to the user on-screen. We end by reporting on the design choices related to the interactive part, that allows the user to change individual time points and do model inspection.

#### 4.1 The Need for Trust

When presenting output from black box ML systems to non-trained users, the need for explanations arises from at least two angles: The user must trust the results, and the user must, to a sufficient degree understand the results. To use a real-world example from energy companies [19], if a system based on charging data tells a car owner that their car seems faulty, or tells a home owner that their power consumption deviates from expectations - they would need to trust their correctness, otherwise the message will be ignored. When trust is established, the next step is usually to fix the situation. If the car owner understands the reason for their car charging being classified as faulty, he/she can bring it to the vendor or repair shop with concrete information that can be used to fix it. Similarly, if the home owner understands why their consumption is classified as deviating - they could perhaps fix a broken appliance - or adjust their consumption to a more favourable price pattern. These are situations encountered at Eviny, a Norwegian energy company collaborating on the present tool. The more complicated the data and classifications are (for example consisting of several series of data measuring power, temperature, etc. - and/or having temporal patterns like slowly decreasing trends), the more challenging will the explanation be. Thus - exploring different approaches and tools for explanations for non-trained users is of great use when companies plan on applying machine learning on complex data. Without trust the models will be ignored, and without an understanding of what the actual problem is no appropriate action can be taken. Of course, one can ask how much benefit non-trained users can gather from example-based explanations in the realm of time series, and this is indeed one of the questions guiding the user evaluations done in this work.

#### 4.2 Generating and Presenting Prototypes and Counterfactuals

We have opted to use cluster centers as our prototypes. To ensure the prototypes we selected have high plausibility, we use a k-medoids algorithm to find the prototypes, see [12]. Specifically we used KMedoids from `sklearn_extra.cluster` package. We used default configuration of KMedoids, with Euclidean distance as the distance metric.

There is a growing consensus that counterfactuals provide robust and informative explanations to a query time series whose classification is to be explained. In the time series domain the visualization of counterfactuals is straightforward. We have opted to use a novel method for generating counterfactuals for time series called Native Guide, developed recently by Delaney et al. [4]. This method extracts counterfactual time series, named Native Guides, starting from initial training data. Starting from the query time series whose classification is to be explained, the Native Guide method starts by finding a counterfactual time series belonging to the dataset that is close to the query. This Native series are then adapted in a Guided way to generate novel counterfactuals, following four identified key properties for good counterfactuals: proximity, sparsity, plausibility, and diversity. The Native Guide counterfactual generation method uses Class Activation Mappings to Guide the counterfactual generation from the Native series. The use of CAM in itself puts some limitations on the AI model used, e.g. having the last layer be global average pooling, so in that sense is not completely model-agnostic. To ensure compatibility between the model doing the classification and the Native Guide counterfactual generator, we therefore closely follow the time series classification model implementation of Delaney et al. available [here](#).

When presenting time series to the user we have opted to use two colours for the two classes, namely Blue and Pink. Since a counterfactual, say Blue, will be used to explain the classification of a given query Pink time series, we have chosen to present both at the same time to the user. Thus, we plot the query with Pink lines, and then show only the deviation of the counterfactual by Blue dotted lines. See Figs. 2, 3, 4 and 5 which show also that the user is allowed to make interactive changes, as described in the next subsection. Note that the y-values in the figures represents the normalized values for each dataset.

### 4.3 Allowing Interaction and Model Inspection

A central aspect of our tool is that it allows the user to alter individual data points and do model inspection. In real-time the tool will update the model classification by changing the color of the time series if the classification changes, showing the model confidence in this classification. It will also update to a new counterfactual. This enables active learning and allows the user to test and improve their hypotheses when formulating a mental model of the black box classification. In Figs. 2, 3, 4 and 5, we see 4 screen shots of an actual session with the tool: Fig. 2) User starts with a Pink prototype and a counterfactual. Figure 3) Makes changes to left end of series so that confidence (top bar) of model classification drops, plus new counterfactual, but same color/class. Figure 4) Makes further changes and now the model classification switches. Figure 5) Last changes made by user and confidence of model classification increases. Note how the user can explore their understanding of the classification by progressive changes to the current time series. Compare also with Fig. 1.

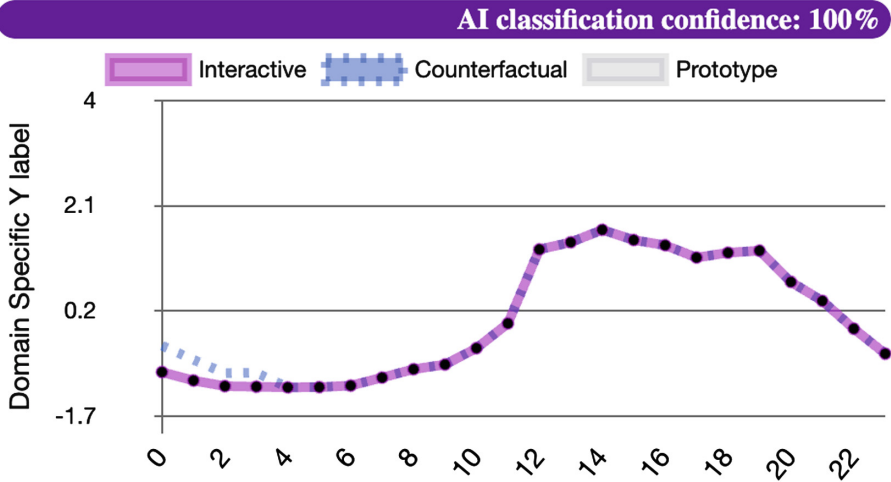


Fig. 2. Start in Pink prototype. (Color figure online)

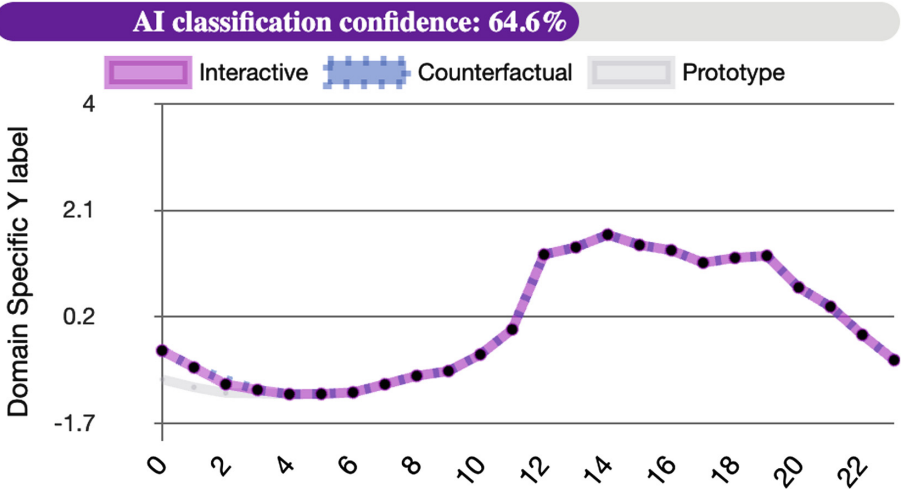


Fig. 3. Changes at left end, still Pink. (Color figure online)

## 5 End-User Evaluation with Forward Simulation

We have not before seen such an interactive tool for time series in the research literature. As time series are notoriously difficult for human users, most of the human evaluations of XAI methods done so far have been qualitative and involving domain experts. Our goal in this work has been to add the new dimension of quantitative user evaluations involving end-users, i.e. non-experts, on XAI for time series. In this section we present this user evaluation. We start by presenting

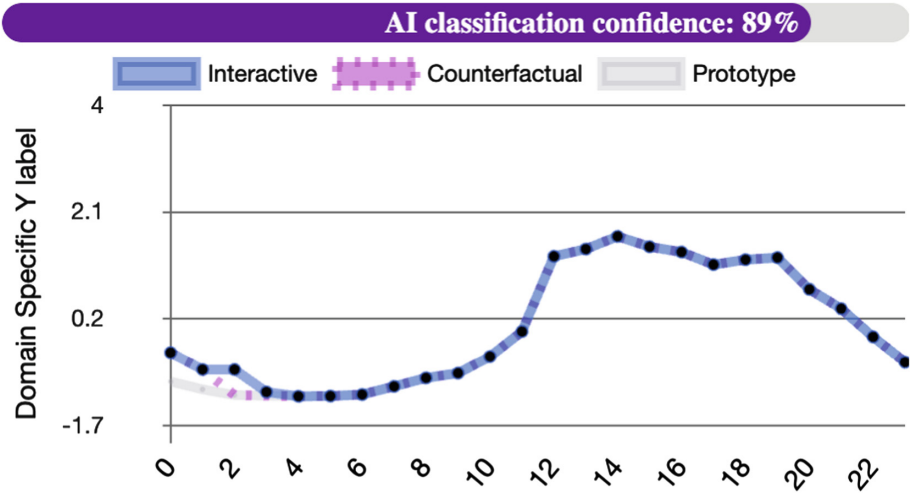


Fig. 4. Further changes, now Blue. (Color figure online)

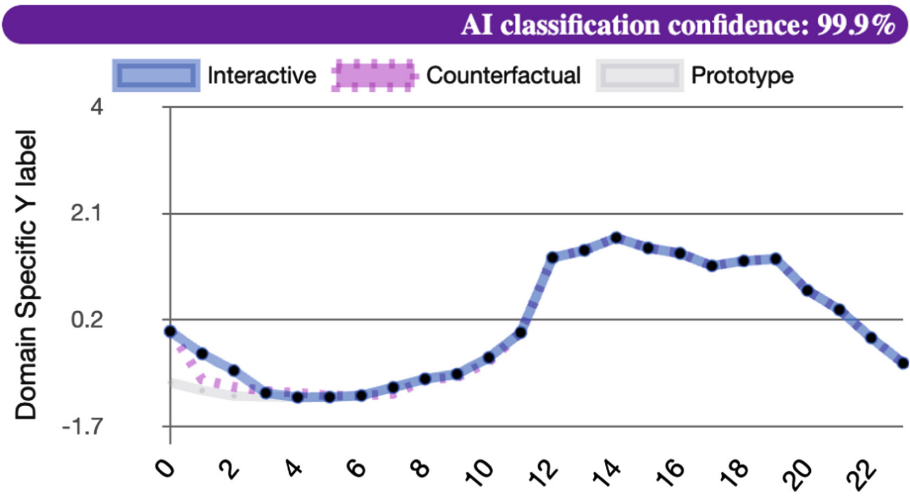


Fig. 5. Last changes, still Blue. (Color figure online)

the discussion leading up to the 3 time series datasets chosen for the evaluation. We then discuss the 3 distinct survey groups that will be given different abilities in the training stage. We end by giving the results of the evaluation, on each pairing of dataset and survey group, and a discussion of their significance.

Our tool is designed for univariate datasets. Moreover, since our test users are not domain experts and time series are notoriously non-intuitive to us humans we had some desiderata when choosing datasets for our survey. Firstly, we wanted univariate datasets with a binary classification, Secondly, we wanted datasets on

time series with not too many data points. Thirdly, we wanted datasets where it is fairly easy for a non-expert to understand the domain. Datasets satisfying these criteria will increase the prospective of providing constructive quantitative feedback, and also they form the more common situation where an explanation would potentially be provided for end-users. However, it is important to note that apart from the above constraints we did not want simple datasets where the binary classification is particularly easy or could be described (e.g. by ourselves) in any straightforward way. We chose the following three datasets satisfying the above criteria:

- From UCR [3]: Italy Power Demand. This dataset shows power demand in Italian households over a 24-h time period, and classifies these into Winter (October-March) and Summer (April-September).
- From UCR [3]: Chinatown. This dataset shows the number of pedestrians on a particular street corner of Chinatown in Melbourne over a 24-h time period, and classifies these into Weekend (Sat-Sun) and Weekdays (Mon-Fri).
- From Eviny: Car Charging. This dataset shows the power demand at a particular charging station for electric vehicles over a 24-h period, and classifies these into Weekend (Sat-Sun) and Weekdays (Mon-Fri).

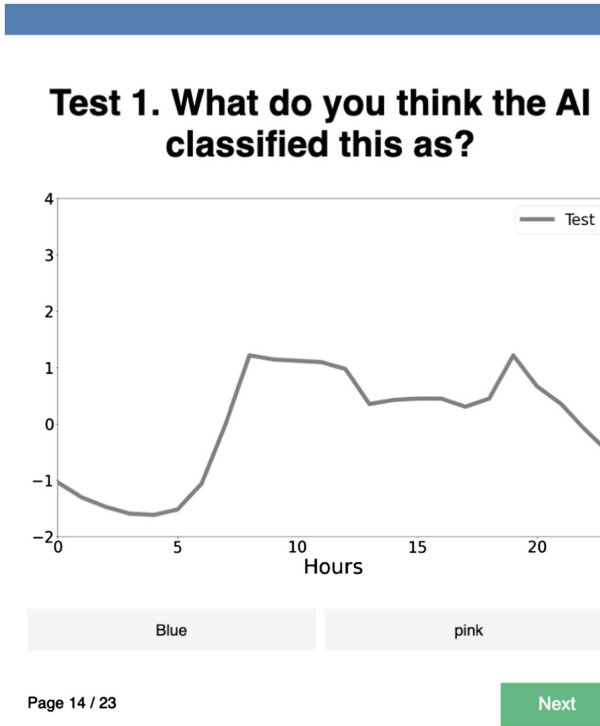
**Table 1.** Information of the datasets employed in the evaluation. We also include information about the proportion of data employed for training and testing the data, and the accuracy obtained by the learned model.

Dataset	Number of instances	Class distribution	Train/Test	Accuracy
ItalyPowerDemand	363	104/259	6%/94%	98%
ChinaTown	1096	547/549	5%/95%	96%
Car Charging	365	269/96	75%/25%	57%

Some details of the datasets employed in the experiments can be found in Table 1. We also include the accuracy obtained by the trained models. In the datasets we used the split between training and test defined and the repository except from Car charging where we employed a random selection of 75% train and 25% test.

As mentioned earlier, the three main components used in our interactive XAI system for time series are prototypes (PT), counterfactuals (CF), and model inspection (MI) with user-generated instances. Our survey groups will enter a 3-stage process, as follows:

- Intro: A short introduction is given to the relevant components, the dataset, training stage, and testing stage.
- Training: Group dependent.
- Testing: 10 randomly selected time series from the dataset are shown, and for each one the user must guess the AI model classification. See Figure below.



**Fig. 6.** Test case for ItalyPowerDemand.

To cover the distribution of time series within each class, we wanted to show users more than one prototype for each class. However, we did not want to overload the user with information, hence we opted to have 6 prototypes in total, three for each class. We will have three survey groups (PT, PT+CF, PT+CF+MI) depending on what is made available to users in the Training stage:

- PT: One at a time, the user is shown 3 prototypes from class A, then 3 from class B, and finally shown a screen with all 6 prototypes.
- PT+CF: One at a time, the user is shown 3 prototypes from class A together with a counterfactual from class B, then 3 converse pairs, and finally shown a screen with all 6 pairs.
- PT+CF+MI: The user is shown same as PT+CF but model inspection is permitted, with prototypes being interactive to allow iterative changes of any chosen time points, and the ensuing classification and also new counterfactual continually updated.

In Figs. 2, 3, 4 and 5 we see how a user in group PT+CF+MI is shown a pair consisting of a prototype and a counterfactual and is allowed to modify data points. A user in group PT+CF is shown a single such pair consisting

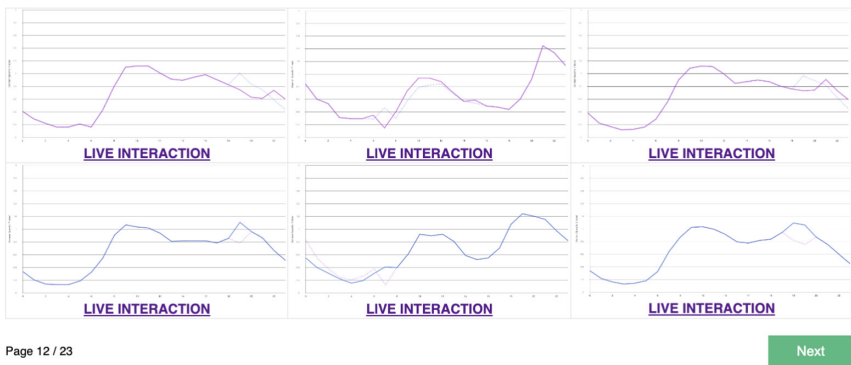


of prototype and counterfactual but without the ability to make modifications, while a user in group PT is shown only the prototype. In the last part of the training stage the users are shown all 6 prototypes/counterfactuals on one screen, to easily make a comparison, as in Fig. 7 for group PT+CF+MI.

## Last part of Training stage

We are now almost done with training and will soon move to testing. Before continuing, have a final careful look and do some interaction because during testing you must decide if a given time series is classified pink or blue by the AI, based only on these training examples and your interaction.

Please spend at least **4 minutes** studying / interacting on this page, to make yourself a rule for classifying blue or pink.



**Fig. 7.** Blue and pink prototypes with counterfactuals, for ItalyPowerDemand. (Color figure online)

## 5.1 Evaluation Results

Let us present the results of the user survey. In total, we had 65 voluntary participants, who were either students in a university-level informatics course, or researchers in informatics, thus with experience in using PCs, none of whom received compensation. The participants were presented with the survey and freely chose to participate. The participants were randomly divided into the 3 survey groups. After introduction and training they were asked to answer 10 test questions from each of up to 3 datasets. Those who spent less than a minimum amount of time (less than 1 min for training and testing combined on a single dataset) were discarded, as we considered that it would be impossible to even read the provided information in that time.

In Table 2 we see the accuracy and number of tests (i.e. number of participants times 10) for each of the three survey groups, on each of the three datasets. Accuracy is the percentage of correct test answers. The rightmost column gives the aggregated information for each survey group, and the bottom row the aggregated information for each dataset. The value in the bottom right corner shows that the overall accuracy was 66.7%, thus clearly better than a random guess, and satisfactory given the complicated nature of these time series classifications.

Let us compare the performance between the three survey groups. Perhaps surprisingly, on aggregate we see that survey group PT that was only shown the prototypes, did slightly better than the other two, with accuracy of 70.2% versus 65.9% and 64%. This could indicate that non-expert users are not necessarily able to use the extra information provided by counterfactuals and model inspection in a meaningful way. It could also mean that a ‘rule-of-thumb’ that appears useful based only on prototypes (or prototypes + counterfactuals) actually works well on a high percentage, say 80%, of test instances. However, after close model inspection a user in group PT+CF+MI may discover that this rule is not precise (as it fails on several instances) leaving the user to discard this rule-of-thumb, and subsequently actually performing worse on the tests. A final possible explanation for why group PT+CF+MI did not achieve higher accuracy is the fact that the average time spent by users in total on all 3 datasets was about the same, around 15 min: group PT 954 s, group PT+CF 934 s, group PT+CF+MI 966 s. However, it seems natural that a user doing model inspection to learn a better rule should have spent more time than one who cannot do model inspection, making us question how careful the users in group PT+CF+MI were. On the other hand, there is not a clear trend when we compare accuracy versus time spent for users in this group.

Let us turn to comparing between the 3 datasets. Interestingly, all 3 survey groups had the highest accuracy on Chinatown and the lowest accuracy on Car-Charging. We asked some users about rules they had been using for their own mental classification, and the most mentioned rule was for Chinatown, something like the following: ‘if there is a sharp dip at the beginning of the series then it is Blue, and otherwise Pink’. Compare this rule to Figs. 2, 3, 4 and 5 from the Chinatown dataset. See also Figs. 7 and 6 which for ItalyPowerDemand gives all prototypes and counterfactuals, plus an example of a test, to get an impression of how difficult the classification indeed is for this dataset. For the test shown in Fig. 6 the average accuracy was 59.3%. Note the users could not navigate back to see the prototypes when answering the tests.

**Table 2.** Overview of accuracy and number of tests by survey group and dataset.

Survey ID	Data	ItalyPowerDemand	Chinatown	CarCharging	All 3 datasets
PT	accuracy	65.9% $\pm$ 10	79.4% $\pm$ 13	65.3% $\pm$ 14	70.2% $\pm$ 14
	tests	170	170	170	510
PT+CF	accuracy	61.9% $\pm$ 17	81.3% $\pm$ 13	54.4% $\pm$ 21	65.9% $\pm$ 20
	tests	160	160	160	480
PT+CF+MI	accuracy	60.2% $\pm$ 16	76.2% $\pm$ 22	55.7% $\pm$ 13	64.0% $\pm$ 20
	tests	240	210	230	680
All groups	accuracy	62.7% $\pm$ 15	79.0% $\pm$ 17	58.4% $\pm$ 17	66.7% $\pm$ 19
	tests	570	540	560	1670

## 6 Conclusion

As companies and controllers must cope with the EU regulations in form of GDPR and the AI Act, explainability that allows model inspection by end-users may very well become the target for developers. In domains where we humans have poor intuition, such as time series, this may pose several challenges. In this work we have contributed a tool for XAI on time series classification that allows such model inspection. The evaluation results show that users can then successfully perform a difficult forward simulation test. However, to attain the full benefits from model inspection for such a complicated domain, it seems necessary to have highly motivated users that are willing to spend more time with such a tool, in order to form better mental models of the black-box classification. As future work, we propose to explore the use of simplified versions of time series prototypes generated by Machine Teaching techniques to explain time series classification models.

**Acknowledgement.** This paper is part of NRF project 329745 Machine Teaching For XAI. We thank the anonymous reviewers for their comments and the volunteers who participated in the experiments. This work was funded by ValGrai, CIPROM/2022/6 (FASSLOW) and IDIFEDER/2021/05 (CLUSTERIA) funded by Generalitat Valenciana, the EC H2020-EU grant agreement No. 952215 (TAILOR), US DARPA HR00112120007 (RECoG-AI) and Spanish grant PID2021-122830OB-C42 (SFERA) funded by MCIN/AEI/10.13039/501100011033 and “ERDF A way of making Europe”.

**Disclosure of interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Abdul, A.M., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.S.: Trends and trajectories for explainable, accountable and intelligible systems: an HCI research agenda. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI 2018, p. 582. ACM (2018)

2. Beretta, I., Cappuccio, E., Manerba, M.M.: User-driven counterfactual generator: A human centered exploration. In: Conference on eXplainable Artificial Intelligence (xAI-2023), pp. 83–88. CEUR Workshop Proceedings (2023)
3. Dau, H.A., et al.: The UCR time series classification archive (2018)
4. Delaney, E., Greene, D., Keane, M.T.: Instance-based counterfactual explanations for time series classification. In: Sánchez-Ruiz, A.A., Floyd, M.W. (eds.) ICCBR 2021. LNCS (LNAI), vol. 12877, pp. 32–47. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-86957-1\\_3](https://doi.org/10.1007/978-3-030-86957-1_3)
5. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. CoRR **abs/1702.08608** (2017)
6. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint [arXiv:1702.08608](https://arxiv.org/abs/1702.08608) (2017)
7. Ford, C., Keane, M.T.: Explaining classifications to non-experts: an XAI user study of post-hoc explanations for a classifier when people lack expertise. In: Rousseau, J.J., Kapralos, B. (eds.) ICPR 2022. LNCS, vol. 13645, pp. 246–260. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-37731-0\\_15](https://doi.org/10.1007/978-3-031-37731-0_15)
8. Guillemé, M., Masson, V., Rozé, L., Termier, A.: Agnostic local explanation for time series classification. In: 31st IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2019, Portland, OR, USA, 4–6 November 2019, pp. 432–439. IEEE (2019). <https://doi.org/10.1109/ICTAI.2019.00067>
9. Höllig, J., Kulbach, C., Thoma, S.: Tsinterpret: a Python package for the interpretability of time series classification. J. Open Source Softw. **8**(87), 5220 (2023)
10. Ismail, A.A., Gunady, M.K., Bravo, H.C., Feizi, S.: Benchmarking deep learning interpretability in time series predictions. In: Annual Conference on Neural Information Processing Systems 2020. NeurIPS 2020, 6–12 December 2020, Virtual (2020)
11. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data Min. Knowl. Disc. **33**(4), 917–963 (2019)
12. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990). <https://doi.org/10.1002/9780470316801>
13. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. **267**, 1–38 (2019)
14. Poché, A., Hervier, L., Bakkey, M.C.: Natural example-based explainability: a survey. In: Longo, L. (ed.) xAI 2023. CCIS, vol. 1902, pp. 24–47. Springer, Cham (2023). [https://doi.org/10.1007/978-3-031-44067-0\\_2](https://doi.org/10.1007/978-3-031-44067-0_2)
15. Rajkomar, A., et al.: Scalable and accurate deep learning with electronic health records. NPJ Digit. Med. **1**(1), 1–10 (2018)
16. Schlegel, U., Arnout, H., El-Assady, M., Oelke, D., Keim, D.A.: Towards a rigorous evaluation of XAI methods on time series. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pp. 4197–4201. IEEE (2019)
17. Shneiderman, B.: Human-centered artificial intelligence: reliable, safe & trustworthy. Int. J. Hum.-Comput. Interact. **36**(6), 495–504 (2020)
18. Siddiqui, S.A., Mercier, D., Munir, M., Dengel, A., Ahmed, S.: Tsviz: demystification of deep learning models for time-series analysis. IEEE Access **7**, 67027–67040 (2019). <https://doi.org/10.1109/ACCESS.2019.2912823>
19. Su, L., Zhang, S., McGaughy, A.J., Reeja-Jayan, B., Manthiram, A.: Battery charge curve prediction via feature extraction and supervised machine learning. Adv. Sci. **10**(26), 2301737 (2023)

20. Susto, G.A., Cenedese, A., Terzi, M.: Time-series classification methods: review and applications to power systems data. In: *Big Data Application in Power Systems*, pp. 179–220 (2018)
21. Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R.: Explainable AI for time series classification: a review, taxonomy and research directions. *IEEE Access* **10**, 100700–100724 (2022)
22. Wang, Z.J., Vaughan, J.W., Caruana, R., Chau, D.H.: GAM coach: towards interactive and user-centered algorithmic recourse. In: *Proceedings of Conference on Human Factors in Computing Systems*, pp. 835:1–835:20. ACM (2023)
23. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016*, pp. 2921–2929. IEEE Computer Society (2016)

# Author Index

## A

Abrate, Carlo 18  
Acar, Erman 143, 350  
Andreoletti, Davide 294  
Ariva, Joonas 39  
Ayoub, Omran 294

## B

Barbiero, Pietro 185  
Bischi, Bernd 85  
Blesch, Kristin 85  
Bonchi, Francesco 18  
Breuer, Nils Ole 143

## C

Carbonelli, Cecilia 232  
Cinquini, Martina 108  
Conati, Cristina 125

## D

Dandl, Susanne 85  
den Hengst, Floris 350  
Dezem, Vinicius 319  
Di Lavore, Elena 185  
Dimitrakakis, Christos 374  
Domnich, Marharyta 39, 60

## E

Engelhardt, Raphael C. 165  
Ezzeddine, Fatima 294

## F

Ferri, Cèsar 439  
Fettke, Peter 420  
Fickett, Dale 319  
Fioravanti, Stefano 185  
Fishman, Dmytro 39  
Flikka, Kristian 439  
Freiesleben, Timo 85

## G

George, Anne-Marie 374  
Giannini, Francesco 185  
Giordano, Silvia 294  
Gjoreski, Martin 294  
Guidotti, Riccardo 108

## H

Håvardstun, Brigit 439  
Hu, Hui 125

## I

Islam, Sheikh Rabiul 334

## J

Jang, Daesik 125

## K

Kapar, Jan 85  
Kasneci, Gjergji 395  
Koenen, Niklas 247  
Konen, Wolfgang 165  
König, Gunnar 85  
Kumar, Neha Mohan 334

## L

Lange, Moritz 165  
Langheinrich, Marc 294  
Leemann, Tobias 395  
Liò, Pietro 185  
Lisa, Fahmida Tasnim 334  
Luengo, David 3

## M

Majlatow, Maxim 420  
Mehdiyev, Nijat 420  
Mohammadi, Majid 143

**P**

Pawelczyk, Martin 395  
Prenkaj, Bardh 395

**R**

Refoyo, Mario 3  
Rizzo, Matteo 125

**S**

Saad, Mirna 294  
Sachan, Swati 319  
Sauter, Andreas 143  
Sbeity, Ihab 294  
Segal, Meirav 374  
Seifi, Sarah 232  
Servadei, Lorenzo 232  
Shvetsov, Dmytro 39  
Siciliano, Federico 18  
Silvestri, Fabrizio 18  
Stramiglio, Alessandra 270  
Strobel, Maximilian 232  
Sukianto, Tobias 232

**T**

Telle, Jan Arne 439  
Tonda, Alberto 185  
Torabian, Alireza 207

**U**

Urner, Ruth 207

**V**

Vicente, Raul 39, 60  
Visbeek, Samantha 350  
Vitali, Fabio 270

**W**

Wille, Robert 232  
Wiskott, Laurenz 165  
Wright, Marvin N. 85, 247

**Y**

Yapicioglu, Fatima Rabia 270  
Yu, Ingrid Chieh 374