Imperial College London
Department of Computing

Robust Counterfactual Explanations in Machine Learning

Junqi Jiang

October 2025

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of Imperial College London
and the Diploma of Imperial College London

Statement of Originality

I declare that this thesis was composed by myself and that the work it presents is my own, except where otherwise stated. I certify that all material in this thesis which is not my own work has been properly acknowledged.

Junqi Jiang

Abstract

Recent advances in machine learning (ML) have led to the widespread deployment of complex black-box models, raising concerns about their trustworthiness. Explainable artificial intelligence (XAI) aims to address these concerns. Within XAI, counterfactual explanations (CEs) have emerged as a prominent paradigm, providing insights into how an input must be modified to alter a model's prediction. CEs are highly valuable in high-stakes domains such as finance and healthcare, where they enable algorithmic recourse, informing users negatively affected by model decisions how to achieve more favourable outcomes. Beyond fundamental requirements like validity and proximity, CEs must ideally be plausible and diverse. However, current CE methods often lack robustness: small perturbations in the generation process can render a previously valid CE invalid, posing significant deployment risks.

This thesis investigates the robustness of CEs. We define novel forms of robustness theoretically, characterise scenarios where robustness is lacking, and introduce methods improving upon the state-of-the-art. Specifically, we study three forms of robustness: against model changes, input changes, and model multiplicity. First, we propose a technique to certify CEs' robustness against changes in the ML model. We then propose new methods that jointly encourage provable robustness and plausibility, advancing state-of-the-art performance across various CE metrics. Next, we address robustness against input changes, ensuring similar inputs receive similar CEs. We propose a comprehensive model-agnostic method to achieve this, leveraging a novel generative model that learns a structured latent space where each class label is represented by Gaussian components with diverse, prototypical centroids. Finally, we study CE robustness under model multiplicity, where multiple trained models may produce conflicting predictions. We formalise this setting as an ensembling problem and develop a method providing robust recourse. Collectively, these contributions advance the theoretical and practical aspects of CE robustness, offering new approaches for building more trustworthy XAI systems.

Copyright Declaration

The copyright of this thesis rests with the author. Unless otherwise indicated, its contents are licensed under a Creative Commons Attribution-Non Commercial 4.0 International Licence (CC BY-NC).

Under this licence, you may copy and redistribute the material in any medium or format. You may also create and distribute modified versions of the work. This is on the condition that: you credit the author and do not use it, or any derivative works, for a commercial purpose.

When reusing or sharing this work, ensure you make the licence terms clear to others by naming the licence and linking to the licence text. Where a work has been adapted, you should indicate that the work has been changed and describe those changes.

Please seek permission from the copyright holder for uses of this work that are not included in this licence or permitted under UK Copyright Law.

Acknowledgements

The PhD has been a rapid and challenging journey, yet it stands as the most rewarding period of my life so far. Reflecting on the past few years, every memory feels remarkable and marks a moment of personal growth. I have been fortunate to receive immense support from many wonderful people, and I am deeply grateful to everyone who has shared this journey with me.

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Francesca Toni. I have consistently been inspired by her critical and creative way of thinking, her deeply professional attitude towards research, and her patient and supportive approach to our day-to-day supervision. As her student, I have been privileged to learn a great deal, both within and beyond the context of scientific research. Her constant support and insightful mentorship through the years have shaped my PhD research. These influences will continue to guide my research career.

I would also like to thank my co-supervisors, Dr. Francesco Leofante and Dr. Antonio Rago (not official, but effectively he is), for their guidance and friendship. It has often been said that I have many supervisors, a fact for which I could not be more grateful. Their invaluable feedback and our frequent, in-depth discussions have taught me to think more creatively and comprehensively. I feel very fortunate to have worked closely with them and to have learned so much about all aspects of life as a researcher.

A big thank you to my PhD examiners, Dr. Matthew Wicker, Dr. Yingzhen Li from Imperial, and Professor Hana Chockler from King's College London, who took the time and brain power to carefully read and evaluate my thesis. The four-hour viva was a remarkably rewarding experience with a challenging yet supportive environment that they fostered. Their thought-provoking questions and expert feedback not only strengthened this thesis but also helped me grow as a researcher.

My PhD research was made possible by the generous funding from the Royal Academy of Engineering and JPMorganChase through Francesca's Research Chair in Argumentation for Interactive Explainable AI project. I am also grateful for the opportunity to have undertaken a research internship at the

AI Research department of JPMorganChase, which was an enjoyable and formative experience. I would like to thank the Trustworthy AI Center of Excellence team (especially those researchers in London - Emanuele, Eoin, Salim, Tom, Saumitra) for creating such a collaborative and intellectually stimulating environment. I would also like to thank other staff members I have worked with, and my fellow interns (Anna H., Jabez, Yizhi) for sharing a memorable experience.

While at Imperial College London, I have been very lucky to be part of Francesca's research group with an exceptionally welcoming and friendly atmosphere. I want to thank my colleagues and friends there for sharing all the highs and lows of the PhD journey, for our fun lunch breaks, after-work drinks, and for the many events that made my time here so enjoyable. They are: Adam D., Adam G., Andria, Anna R., Antonio, Avinash, Deniz, Fabrizio, Francesco, Francis, Gabriel, Guilherme, Hamed, Kai, Lihu, Mansi, Nico, Xiang. I also extend my thanks to all my colleagues and friends at Imperial, especially those in the Department of Computing. My PhD experience would not have been nearly as engaging or fun without them. A special thank you goes to the departmental staff and our PhD administrator, Dr. Amani El-Kholy, for making our research lives much easier.

I would like to thank my friends, both those here in London and those I have known from home. Moving to a new city can be tough, but sharing my time with them has made it much smoother than I imagined. My old friends, now scattered around the world, are a part of who I am today, and I am forever grateful for their support.

My deepest and most heartfelt thanks go to my parents and my family. Their unconditional love and encouragement have been the foundation upon which I could pursue my aspirations in life. I am also deeply thankful to my wonderful parents-in-law, who have provided unwavering support for me. I extend my warmest thanks to my family friends for their kindness and support throughout the journey.

Finally, and most importantly, no words can fully express my gratitude for my incredible wife, Shuxian Nie. This PhD has been our shared journey from the very beginning. Her infinite patience, love, and belief in me were the critical components that made my successes possible. I thank her dearly for the brave and selfless decision she made to move to London and start our future here. Throughout all the times of stress and struggle, it was the idea of building a life for us, as a family, that carried me forward. This achievement is not just mine; it is ours, and I dedicate this thesis to her with all my love.

Contents

Abstract **3**

1 Introduction **22**

1.1 Motivation and Background 22

1.2 Thesis Objectives 26

1.3 Thesis Contributions 26

1.4 Thesis Structure 28

1.5 Thesis Publications 28

2 Background **32**

2.1 Machine Learning Classification 32

2.2 Counterfactual Explanations 34

2.3 Generative Deep Learning 36

2.4 Computational Argumentation 39

2.5 Summary 41

3 Literature Review **42**

3.1 Desirable Properties of Counterfactual Explanations 42

3.2 Robust Counterfactual Explanations 44

3.3 Summary 57

4 Interval Abstractions for Guaranteeing Robustness of Counterfactual Explanations **59**

4.1 Motivation 60

4.2 Interval abstractions and robustness for binary classification . 61

4.3 Interval abstractions and robustness for multi-class classification 67

4.4 Computing Δ-robustness with MILP 71

4.5 Algorithms 74

4.6 Experiments 79

4.7 Summary, Limitations and Discussion 92

5 Provably Robust and Plausible Counterfactual Explanations via Robust Optimisation **94**

5.1 Motivation 95

5.2 Problem Statement 95

5.3 PROPLACE 97

5.4 Experiments 105

5.5 Summary, Limitations and Discussion 108

6 Synthesising Counterfactual Explanations via Label-Conditional Gaussian Mixture Variational Autoencoders 111

6.1 Motivation 112

6.2 Latent Path Counterfactual Explanations 114

6.3 Evaluation 120

6.4 Summary, Limitations and Discussion 127

7 Argumentative Ensembling for Robust Recourse under Model Multiplicity 129

7.1 Motivation 130

7.2 Recourse under Model Multiplicity 131

7.3 Argumentative Ensembling 137

7.4 Empirical Evaluation 154

7.5 Summary, Limitations and Discussion 163

8 Conclusion 166

8.1 Summary of Contribution 166

8.2 Future Work 167

Bibliography 170

Appendices 189

A Appendices for Chapter 4 **189**

A.1 Computation Time of RNCE and the Impact of `robustInit` 189

A.2 Full Experiment Results 191

B Appendices for Chapter 6 **194**

B.1 L-GMVAE in the MNIST Example 194

B.2 L-GMVAE in the CE Evaluation 195

List of Tables

3.1 Robust CE methods, partitioned based on the targeted robustness form. We include: *(i)* the types of models they target (support vector machines (SVM), linear models (LM), neural networks (NN), differentiable models (DM), tree-based models (TM) and model agnostic methods ($*$)); *(ii)* the model access required (white box, gradients and predictions); *(iii)* the method used (gradient descent (GD), robust optimisation (RO), mixed-integer programming (MIP), computational argumentation (CA), formal verification (FV), genetic algorithms (GA), satisfiability solving (SAT) and data augmentation (DA)); *(iv)* whether a robustness guarantee is provided (Deterministic, Probabilistic, for linear models only (LM) and none ($-$)); *(v)* other properties satisfied by the method, including Actionability, Causality, Diversity, and Plausibility. Note that all methods consider validity and proximity by default, thus they are omitted. †These works are part of the technical contributions of this thesis. 45

4.1 Dataset and classifier details. 80

4.2 Properties (validity, proximity, plausibility, robustness) addressed by baselines and whether the methods apply to neural network (NN) or logistic regression (LR), indicated by ✓. . . . 86

4.3 Quantitative evaluation of the compared CE generation methods on neural networks. The ↑ (↓) following each metric indicates the higher (lower) the value, the better. Methods are separated by horizontal lines, indicating non-robust baselines, robust baselines, and our methods with robustness target Δ instantiated by δ_{val} and δ_{inc}, respectively. 87

4.4 Computation time (in seconds) of the compared methods on logistic regression models and neural networks. The adult dataset has 48832 data points with 13 attributes, the compas dataset has 6172 data points with 7 attributes, the gmc dataset has 115527 data points with 10 attributes, the heloc dataset has 9871 data points and 21 attributes. 90

4.5 Quantitative evaluation of the compared methods on neural networks in multi-class classification tasks. The evaluation metrics are the same as in Table 4.3. 92

5.1 Evaluations of PROPLACE (OURS) and baselines on NNs. The ↑ (↓) indicates that higher (lower) values are preferred for the evaluation metric. 107

6.1 Dataset, classifier details, and L-GMVAE utility evaluation. The first column reports the number of features (#continuous + #categorical) of each dataset. The test accuracies of classifiers are included. The Utility columns report the test accuracies of new classifiers following TSTR. 121

6.2 The CE desirable properties addressed by each CE method in our experiments. ✓(*) indicates the property is explicitly taken into account (implicitly encouraged) by the method. . . 123

6.3 Actionability evaluations – portion of test inputs for which the CEs satisfy the actionability constraints while remaining valid. 126

7.1 Augmented ($S^n = \mathcal{M}^n \cup \mathcal{C}^n$) and robust ($S^v = \mathcal{M}^n \cup \mathcal{C}^v$) ensembling, as well as our argumentative approach with the s-preferred extension (defined later in Section 7.3), assessed against the desirable properties defined in Section 7.2.1. Satisfaction of a property is shown by ✓, while partial satisfaction under given conditions is shown by ✓*. 137

7.2 Different semantics applied to Examples 14 and 15. $P^{stable}, P^d, P^s, P^c$ are in each table entry, with argumentative ensembling solutions highlighted within. 143

7.3 Accuracies of trained neural networks of different sizes on each dataset 155

7.4 Average accuracy and simplicity of the models in the ensembling result by each method. "avg" rows report these scores for the models in $\mathcal{M}$. 158

A.1 RNCE computation time (seconds) of generating robust CEs for 100 randomly selected inputs using neural networks (NN) or logistic regressions (LR), when configured with different `robustInit` settings. 190

A.2 RNCE computation time (seconds) of generating robust CEs for all data points in the training dataset that are classified to class 0 using neural networks (NN) or logistic regressions (LR), when configured with different `robustInit` settings. The numbers following the dataset names represent the approximate number of inputs for each dataset. 190

A.3 Standard deviation results of the quantitative evaluation of the compared CE generation methods on neural networks. . . 191

A.4 Quantitative evaluation of the compared CE generation methods on logistic regression models. The evaluation metrics are the same as in Table 4.3. 192

A.5 Standard deviation results of the quantitative evaluation of the compared CE generation methods on logistic regression models. 192

B.1 L-GMVAE training details. The loss weights are the weighting terms for KL(z), KL(z), and reconstruction. 196

List of Figures

1.1 An example where a bank employs three ML models which reject a loan application by naive ensembling. Existing research ignores the need for recourse. 25

2.1 AAF for the loan example where model's predictions for the input x are given as superscripts, i.e., $M_1(x) = 0$, $M_2(x) = 0$ and $M_3(x) = 1$; reciprocal attacks are represented by double-headed red arrows. 39

2.2 An example BAF extending the previous AAF example; reciprocal supports are represented by dual-headed green arrows labelled with + and reciprocal attacks are represented by dual-headed red arrows labelled with −. 40

4.1 Illustration of Example 7. 65

4.2 Visual representation of Definition 5. In (a), $\mathcal{I}_{(\theta,\Delta)}$ classifies an input as 1 because the output range for that input is always greater than 0.5. In (b), the output range includes value 0.5 therefore the classification result is undefined. In (c), in a similar manner to the way in which the input in (a) is classified as 1, the input is classified as 0. 66

4.3 Visual representation of Definition 10 instantiated with three colour-coded classes {'orange', 'green', 'blue'}. In (a), $\mathcal{I}_{(\theta,\Delta)}$ classifies an input as o since the output range for that class is always greater than those of any other classes. In (b), the output ranges for o overlap with b, the classification result is therefore undefined. In (c), in a similar manner to the way in which the input in (a) is classified as o, $\mathcal{I}_{(\theta,\Delta)}$ classifies an input as b. 69

4.4 Illustration for Example 10 70

4.5 RNCE behaviours when `optimal` = `F`, $\Delta = \emptyset$ (4.5a); `optimal` = `T`, $\Delta = \emptyset$ (4.5b); `optimal` = `F`, $\Delta \neq \emptyset$ (4.5c); `optimal` = `T`, $\Delta \neq \emptyset$ (4.5d). Each figure depicts the CE that is chosen (green cross) among a set of candidate CEs (grey crosses) for a given input (red circle) and model M_θ. The continuous curved line represents the decision boundary of M_θ; the dashed line represents a possible change in the decision boundary under Δ. Figures 4.5a and 4.5b show configurations under which RNCE may return CEs that are not Δ-robust, whereas Figures 4.5c and 4.5d depict robust ones. 79

4.6 δ values obtained by retraining on increasing portions of $\mathcal{D}_2$. The red dots highlight the δ_{val} and δ_{inc} values found by the two strategies described in Section 4.6.2. 83

4.7 Evaluation of Δ-validity (against increasing δ values) of the CEs found by state-of-the-art methods (GCE, PROTO, MCE, ROAR) and by our methods (GCE-R, PROTO-R, MCE-R, RNCE). 84

6.1 Illustration of LAPACE in binary classification. Given a dataset with a trained classifier's predictions (Left), a L-GMVAE is first learned with latent clusters (Gaussian components), capturing the data distribution with the classifier's predictions. In this example, we have 6 Gaussian components (Middle, the coloured areas). Prediction label 0 (1) is associated with Clusters 0-2 (3-5). The cluster centroids (learned Gaussian mixture prior) for classes 0 and 1 are marked with crosses and check marks. Assuming we are computing CEs for a negatively classified point (Left, purple star), LAPACE first performs linear interpolations linking the input's latent representation to each class 1 cluster centroid (Middle, dashed lines). These paths are then decoded to the input space to obtain paths of points, where they terminate at the decoded class 1 cluster centroids (Right). 113

6.2 Example CE paths found by LAPACE on MNIST dataset, for an input image of class 5 and a target label of 7. This L-GMVAE has 3 Gaussian clusters per class. Each row is a separate CE path, going from the reconstructions of the original input (the second image from the left, $\tau = 0$) to each reconstructed cluster centroid (the last image, when $\tau = 1$). . 119

6.3 Continued MNIST example for the second cluster (the middle row in Figure 6.2). A requirement that a dash should not appear in the resulting CE images of class 7 is enforced on every image. For an input image size of 28×28, pixel values greater than a threshold of 0.01 at the 13th to 18th rows and the 8th to 14th columns (where the dash appears) are penalised via a loss function. 119

6.4 Quantitative evaluation on validity, proximity, plausibility, and diversity of CEs generated by: **NNCE**, **FACE**, **RobXCE**, **DiCE**, **DRCE**, **LAPACE-First**, **LAPACE-Middle**, **LAPACE-Last**. Each subplot is the quantitative comparison of all methods on one dataset-classifier combination and on one evaluation metric. The arrows following each metric indicate that higher or lower values are considered better. For diversity, the first three methods find only one CE per input for which the evaluation metric cannot be computed. Therefore, they are assigned a value of -1. 124

6.5 Quantitative evaluation of robustness and computation time of CEs generated by: **NNCE**, **FACE**, **RobXCE**, **DiCE**, **DRCE**, **LAPACE-First**, **LAPACE-Middle**, **LAPACE-Last**. Each subplot is the quantitative comparison of all methods on one dataset-classifier combination and on one evaluation metric. The arrows following each metric indicate that higher or lower values are considered better. For diversity, the first three methods find only one CE per input for which the evaluation metric cannot be computed. Therefore, they are assigned a value of -1. 125

7.1 BAF for Example 13 where: models' predictions for the input x are given as superscripts, e.g., $M_1(x) = 0$ but $M_4(x) = 1$; reciprocal supports are represented by dual-headed green arrows labelled with + and standard (reciprocal) attacks are represented by single-headed (dual-headed, respectively) red arrows labelled with −. 141

7.2 Two illustrative examples. 142

7.3 AAF for Example 16 where: models' predictions for the input x are given as superscripts, e.g., $M_1(x) = 0$ and $M_4(x) = 1$; standard (reciprocal) attacks are represented by single-headed (double-headed, respectively) red arrows. 154

7.4 Satisfaction of desirable properties on heloc dataset. The three subplots show results respectively for the two baselines, d-preferred argumentative ensembling, and s-preferred argumentative ensembling. 160

7.5 Satisfaction of desirable properties on compas dataset. The three subplots show results respectively for the two baselines, d-preferred argumentative ensembling, and s-preferred argumentative ensembling. 161

7.6 Satisfaction of desirable properties on credit dataset. The three subplots show results respectively for the two baselines, d-preferred argumentative ensembling, and s-preferred argumentative ensembling. 162

7.7 Average computation time of solving argumentative ensembling for each input. 163

B.1 Reconstructed cluster centroids for the L-GMVAE model on MNIST dataset used in the examples. 195

Chapter 1

Introduction

1.1 Motivation and Background

Classification is one of the fundamental tasks in Machine Learning (ML) and Artificial Intelligence (AI). Typically, a model is trained to assign a discrete class label to an input data point. Common examples range from everyday applications such as spam email detection to high-stakes decisions such as loan approval. The advances of highly complex models, particularly deep neural networks, have led to state-of-the-art performance but have also given rise to the "black-box" problem. Namely, the internal decision-making mechanisms of these models are often considered to be opaque to human understanding because of their vast parameter spaces. The field of explainable AI (XAI) has emerged to develop methods that explain AI models' classification decisions [42]. The motivation behind XAI research is often to achieve better trustworthiness for confidently developing and deploying these powerful tools, with measurable and controllable risks. This need is also supported by emerging regulations. For example, the European Union's General Data Protection Regulation (GDPR) [52], ethics guidelines [2], and AI Act [43] all advocate for model transparency and a "right to explanation" for the end users of AI models, showing the legal and ethical need for explainability.

Counterfactual explanations (CEs) have emerged as one of the dominant methods for explaining AI models [78]. For a given input to a ML model, a CE essentially presents a user with a minimally modified input which results in a different output from the model, thus revealing how a different outcome could be achieved if the proposed changes were to be enacted. While they are sometimes useful for data augmentation [116], model debugging and improvements [1], in this thesis, we focus on CEs for providing algorithmic recourse for users (negatively) affected by the decisions of ML models. For illustration, consider a fictional loan application with features *income* of £50K, *loan term* of 35 months and *loan amount* of £10K being rejected by a model. Here, a CE could demonstrate that increasing the *income* to £55K would result in the application being accepted. This simultaneously serves as a local explanation by highlighting the influence of the income feature, and provides a clear path to recourse.

CEs have been shown to improve human understanding and trust [95], as they provide information about alternative possibilities that humans can use to build rich mental representations [25]. Initial approaches to generate CEs [137, 142] were optimised for *validity*, i.e., the ability of a CE to correctly change the output, and *proximity*, with respect to some distance measure between the original and modified inputs. Since then, additional metrics have been proposed (see [56, 78] for recent overviews), such as *diversity* [99], i.e., how widely CEs differ from one another, and *plausibility* [34], i.e., whether the CEs lie within the data distribution. To illustrate these properties with our loan example, the suggestion to increase income to £55K is both valid, as it flips the outcome, and proximate, as it represents a minimal change. A method providing a diverse set of CEs might also suggest an alternative CE, such as "increase the loan term to 40 months". Finally, these CEs may be considered plausible because they are realistic financial adjustments, whereas a suggestion to increase income to an unattainable £500K, while potentially valid, would be implausible and thus unhelpful.

Given the critical nature of many scenarios in which CEs could be deployed, e.g., in medical or financial settings, it is of utmost importance that the recourse they provide is *robust* to mild perturbations in the CE generation pipeline. However, recent work has demonstrated that state-of-the-art CE

methods host major drawbacks when it comes to robustness. In particular, Pawelczyk et al. [111] showed that popular approaches for generating CEs may return explanations that are indistinguishable from adversarial examples. Broadly speaking, this means that CEs are extremely susceptible to small changes occurring in the setting they were generated for.

In this thesis, we place particular focus on three forms of robustness, namely robustness against *input changes*, *model changes*, and *model multiplicity*. A method with poor *robustness against input changes* might find significantly different CEs for very similar inputs [5, 86], highlighting an instability in the generation method. Further, if certain sensitive or protected features are changed in the CEs for some inputs but not for other similar inputs, it could also cause fairness issues [127]. While recent research attempts to address this form of robustness, they lack a guarantee of stability in the CE generation process [86].

Robustness against model changes considers the cases where changes occur in the underlying ML model, possibly induced by model retraining with newly available data. This form of robustness is of critical importance in practice. Continuing the loan example, assume the bank's ML model is retrained with new data, while, in the meantime, the customer has achieved a raise in salary (to £$55K$, as prescribed by our example CE). The customer may then return to the bank only to find that the previously specified CE is now invalidated by the new model. In this case, the bank could be seen as being responsible by the user and could potentially be legally liable, risking financial and reputational damage to the organisation. The quality of such unreliable CEs is also questionable: Rawal et al. [122] and Dutta et al. [40] have shown that CEs found by existing non-robust methods are prone to such invalidation due to their closeness to the decision boundary. In fact, none of the existing solutions towards this robustness problem provides guarantees of the robust behaviour of their resulting CEs.

The third kind of robustness we consider relates to a common scenario in ML, when multiple equally performing models for the same prediction task are obtained, e.g., when training neural networks with different random seeds. Model multiplicity (MM) is the situation which arises when these competing

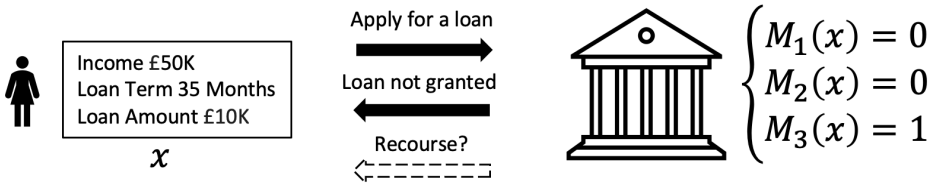


Figure 1.1: An example where a bank employs three ML models which reject a loan application by naive ensembling. Existing research ignores the need for recourse.

models differ in their predictions for the same input. Black et al. [17] argue that MM is a scenario worth addressing because taking into consideration the predictions from multiple independently trained ML models could lead to better-informed decision-making. Consider our loan example and assume the bank has trained a set of three ML models M_1, M_2, M_3[1] to predict whether the loan should be granted or not (Figure 1.1). Even though each model may exhibit good performance overall, their internal differences may lead to conflicts, e.g., if $M_1(x) = M_2(x) = 0$ (i.e., loan rejected), while $M_3(x) = 1$ (i.e., loan accepted). Existing naive solutions often involve ensembling, and they tend to ignore the need to provide avenues for recourse [17]. Traditionally, CE generation methods assume access to a fixed model prediction, which is not directly available given the potentially conflicting predictions under MM. Questions such as how to find CEs that are *robust against MM* have not been studied previously.

Overall, while CEs are a promising tool for algorithmic recourse, their practical reliability is undermined by a lack of robustness, an issue that remains largely underexplored in the literature. As discussed, this is evident in likely real-world scenarios such as input perturbations, model updates, and under the phenomenon of MM.

[1]Technical notations will be formally introduced later in Chapter 2.

1.2 Thesis Objectives

The main aim of this thesis is to identify key robustness challenges for CEs and to develop new methods that address them. These methods are designed to integrate robustness considerations with other essential properties, such as validity, proximity, and plausibility. The ultimate goal is to improve the quality of CEs for user recourse, thus enhancing the reliability of XAI systems that use them. Throughout the thesis, we focus on classification tasks on tabular data. To achieve this, the thesis pursues four concrete objectives:

Objective 1: Characterise. To formally define the critical robustness problems affecting CEs.

Objective 2: Evaluate. To develop novel metrics and frameworks for evaluating the robustness of CEs.

Objective 3: Guarantee. To propose new methods that can generate CEs with provable robustness guarantees.

Objective 4: Integrate. To propose new methods that balance these new robustness properties with existing desirable properties to produce practical and reliable CEs.

1.3 Thesis Contributions

This thesis makes the following technical contributions to achieve the objectives and to push forward the state of the art in robust CEs:

- **Chapter 3: Objective 1.** We wrote a systematic survey summarising the existing forms of CE robustness, including their high-level definitions, quantitative evaluation metrics, and an overview of the algorithms for generating robust CEs.
- **Chapter 4: Objectives 1, 2, 3, 4.** We formally define the problem of CE robustness against model changes for parametric ML models

such as neural networks and logistic regression. We then propose an evaluation method based on interval abstractions of ML models, implemented via mixed integer programming, to certify whether a CE is provably robust. We further propose two distinct algorithms to compute robust CEs. Both methods address validity and proximity, and one of which additionally incorporates plausibility. To the best of our knowledge, this work is the first in the literature to advocate and provide provable robustness guarantees for CEs.

- **Chapter 5: Objectives 3, 4.** While methods in Chapter 4 separate robustness certification and CE generation, here we propose a unified approach using robust optimisation, framing these two objectives as the inner-level and outer-level sub-problems, solved iteratively. This method finds provably robust (against model changes), valid, proximal, and plausible CEs, achieving state-of-the-art performance.

- **Chapter 6: Objectives 3, 4.** We propose a new CE generation method, combining two forms of CE robustness (against model changes and input changes) with most of the existing properties (validity, proximity, plausibility, diversity), while supporting user-specified *actionability* requirements, an additional property of CEs when used to provide recourse, constraining the allowed feature changes in the resulting CEs. First, we introduce the Label-conditional Gaussian Mixture Variational Autoencoder (L-GMVAE), a generative model trained to learn a structured latent space where each class label is represented by a set of Gaussian components with diverse, prototypical centroids. Building on L-GMVAE's latent space, we present our new CE method that synthesises entire paths of CE points. Unlike the methods in the previous two chapters, this new method works on any type of trained classifiers.

- **Chapter 7: Objectives 1, 2, 3.** We formally characterise the problem of providing robust recourse under MM, framing it as *recourse-aware ensembling*. We propose the idea that under MM, CEs for each individual model should be considered alongside their predictions so that the aggregated prediction and recourse are decided in tandem. We propose six desirable properties which the solution to this new

problem should satisfy. Leveraging computational argumentation, a subfield of symbolic AI, we develop a new argumentative ensembling method to satisfy (most of) the properties.

1.4 Thesis Structure

After this introduction, Chapter 2 presents the technical background needed for this thesis. This includes technical notations for ML classification, basic formulations of computing CEs, fundamentals of relevant generative models, and an overview of computational argumentation. Chapter 3 reviews the current literature on CEs for recourse, discussing methods to address existing desirable properties, and surveying the state-of-the-art in CE robustness. Chapters 4 and 5 focus on new methods for robustness against model changes with formal guarantees. Chapter 6 aims to provide a well-rounded CE generation method covering most of the existing properties. Chapter 7 studies the novel scenario of providing recourse when multiple ML models are involved, which potentially give conflicting predictions to the same input. Finally, we conclude in Chapter 8.

1.5 Thesis Publications

This thesis is based on the following publications:

1. Section 3.2: [67] **Junqi Jiang**, Francesco Leofante, Antonio Rago, and Francesca Toni. "Robust counterfactual explanations in machine learning: a survey." In Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI), pp. 8086-8094. 2024.

2. Chapter 4 [64]: **Junqi Jiang**, Francesco Leofante, Antonio Rago, and Francesca Toni. "Formalising the robustness of counterfactual explanations for neural networks." In Proceedings of the AAAI conference on artificial intelligence (AAAI), vol. 37, no. 12, pp. 14901-14909. 2023.

3. Chapter 4 [68]: **Junqi Jiang**, Francesco Leofante, Antonio Rago, and Francesca Toni. "Interval abstractions for robust counterfactual explanations." Artificial Intelligence 336: 104218. 2024.

4. Chapter 5 [65]: **Junqi Jiang**, Jianglin Lan, Francesco Leofante, Antonio Rago, and Francesca Toni. "Provably robust and plausible counterfactual explanations for neural networks via robust optimisation." In Asian Conference on Machine Learning (ACML), pp. 582-597. 2023.

5. Chapter 6 [73]: **Junqi Jiang**, Francesco Leofante, Antonio Rago, and Francesca Toni. "Synthesising Counterfactual Explanations via Label-Conditional Gaussian Mixture Variational Autoencoders." In The Fourteenth International Conference on Learning Representations (ICLR), 2026.

6. Chapter 7 [66]: **Junqi Jiang**, Francesco Leofante, Antonio Rago, and Francesca Toni. "Recourse under Model Multiplicity via Argumentative Ensembling." In Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 954-963. 2024. **Best Student Paper Runner-Up Award.**

7. Chapter 7 [71]: **Junqi Jiang**, Antonio Rago, Francesco Leofante, and Francesca Toni. "Argumentative Ensembling for Robust Recourse under Model Multiplicity." Under Review, 2025.

8. Implementations and benchmarking results for Chapters 6 and 7 [70]: **Junqi Jiang**, Luca Marzari, Aaryan Purohit, and Francesco Leofante. "RobustX: Robust Counterfactual Explanations Made Easy." In Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI), 2025.

Additionally, some other works relevant to XAI and CEs have been published throughout the PhD, but do not contribute directly to this thesis. These include counterfactual explainability in large language models (LLMs) for textual data, interpretability-aided LLM prediction when multiple answers or models are involved (similar to the MM scenario studied in this thesis),

application of XAI and CEs in applied ML, and a position paper discussing contestability, a more user-centric and interactive form of explanations. They are listed below:

9. **Junqi Jiang**, Tom Bewley, Salim I. Amoukou, Francesco Leofante, Antonio Rago, Saumitra Mishra, Francesca Toni. "Representation Consistency for Accurate and Coherent LLM Answer Aggregation." Advances in Neural Information Processing Systems 39 (NeurIPS), 2025. [72]

10. **Junqi Jiang**, Tom Bewley, Saumitra Mishra, Freddy Lecue, Manuela Veloso. "Interpreting Language Reward Models via Contrastive Explanations." In The Thirteenth International Conference on Learning Representations (ICLR), 2025. [69]

11. Ming Pok Ng, **Junqi Jiang**, Gabriel Freedman, Antonio Rago, Francesca Toni. "MArgE: Meshing Argumentative Evidence from Multiple Large Language Models for Justifiable Claim Verification." Under Review, 2025. [102]

12. Xuehao Zhai, **Junqi Jiang**, Adam Dejl, Antonio Rago, Fangce Guo, Francesca Toni, Aruna Sivakumar. "Heterogeneous Graph Neural Networks with Post-hoc Explanations for Multi-modal and Explainable Land Use Inference." Information Fusion 120 (2025): 103057. [151]

13. Jamie Todd, **Junqi Jiang**, Aaron Russo, Steffen Winkler, Stuart Sale, Joseph McMillan, Antonio Rago. "Explainable Time Series Prediction of Tyre Energy in Formula One Race Strategy." In Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC), pp. 1082-1089. 2025. [136]

14. Devin Thomas, **Junqi Jiang**, Avinash Kori, Aaron Russo, Steffen Winkler, Stuart Sale, Joseph McMillan, Francesco Belardinelli, Antonio Rago. "Explainable Reinforcement Learning for Formula One Race Strategy." In Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing (SAC), pp. 1082-1089. 2025. [135]

15. Francesco Leofante, Hamed Ayoobi, Adam Dejl, Gabriel Freedman, Deniz Gorur, **Junqi Jiang**, Guilherme Paulino-Passos, Antonio Rago, Anna Rapberger, Fabrizio Russo, Xiang Yin, Dekai Zhang, Francesca Toni. "Contestable AI needs Computational Argumentation." In Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR), vol. 21, no. 1, pp. 888-896. 2024. [88]

Chapter 2

Background

This chapter covers the technical background essential for the thesis. Section 2.1 introduces the basic preliminaries for ML classification. Section 2.2 provides an overview of existing CE generation algorithms. Contents in these two sections will be the underlying problem setting across the thesis - computing higher-quality CE for classification tasks. Then, Section 2.3 covers the basics of generative deep learning, specifically, the variational auto-encoders (VAEs). It then introduces an advanced VAE method with a latent space regularised in the form of Gaussian Mixture Models, which the methodology in Chapter 6 is based on. Finally, Section 2.4 provides an overview of computational argumentation, a subfield of symbolic AI that is a major component of Chapter 7.

2.1 Machine Learning Classification

General Notations. We use $[k]$ to denote the set $\{1, \ldots, k\}$, for $k \in \mathbb{Z}^+$. We use $|S|$ to denote the cardinality of a set S. Given a vector $x \in \mathbb{R}^n$, we use x_i to denote the i-th component. Similarly, for a matrix $W \in \mathbb{R}^n \times \mathbb{R}^m$, we use $W_{i,j}$ to denote the i, j-th element and $\mathbf{vec}(W)$ to refer to W's vectorisation $\mathbf{vec}(W) = [W_{1,1}, \ldots, W_{n,1}, W_{1,2}, \ldots, W_{n,2}, \ldots, W_{n,m}]$. Finally, $\mathbb{I}(\mathbb{R})$ denotes the set of all closed intervals over $\mathbb{R}$.

ML Classification. Throughout the thesis, we consider a supervised classification problem. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the d-dimensional feature space (the set of all input features) and $\mathcal{Y} = \{0, \ldots, L-1\}$ be a set of L distinct class labels. The goal is to learn a classifier, $M : \mathcal{X} \to \mathcal{Y}$, which predicts label $y \in \mathcal{Y}$ for any given input $x \in \mathcal{X}$, denoted as $M(x) = y$. This model is optimised using a training dataset $\mathcal{D}$ composed of N samples, $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y}$.

ML Models. There exist some established algorithms for classification, which will be used in this thesis. For example, tree-based classifiers learn a set of explicit, rule-based conditions on the input features. Alternatively, parametric models are characterised by a set of equations parameterised by a vector $\theta \in \mathbb{R}^{d_\theta}$, for some $d_\theta > 0$. Here, M_θ refers to a specific concretisation obtained for a given parameter vector θ. Two common examples are logistic regression and neural networks. We illustrate them with binary classification ($L = 2$) in the following examples.

Example 1. *Consider an input $x \in \mathcal{X}$ and assume M_θ implements a* logistic regression classifier*, characterised by the following equation:*

$$M_\theta(x) = \sigma\left(\sum_{i=1}^{d} \theta_i x_i + \theta_0\right)$$

where σ is the Sigmoid function. Then, $\theta = [\theta_0, \ldots, \theta_d]$, θ_0 corresponds to the bias term.

Example 2. *Consider an input $x \in \mathcal{X}$ and assume M_θ implements a* fully connected neural network *with k hidden layers, characterised by the following equations:*

- $V^{(0)} = x$*;*
- $V^{(i)} = \phi(W^{(i)} \cdot V^{(i-1)} + B^{(i)})$ *for $i \in [k]$, where $W^{(i)}$ and $B^{(i)}$ are weights and biases, respectively, associated with layer i, and ϕ is an activation function applied element-wise;*
- $M_\theta(x) = \sigma(V^{(k+1)}) = \sigma(W^{(k+1)} \cdot V^{(k)} + B^{(k+1)})$*, where σ is the Sigmoid function.*

Then, $\theta = [\mathbf{vec}(W^{(1)})\ \ \mathbf{vec}(B^{(1)}) \ldots \mathbf{vec}(W^{(k+1)})\ \ \mathbf{vec}(B^{(k+1)})]$.

Note that many parametric models, including these two examples, initially output a real-valued score rather than a discrete label. For binary classification using the sigmoid function (as in the earlier examples), this output represents the estimated probability of the positive class ($y = 1$). To derive the final class label, this probability is thresholded at 0.5. For the sake of brevity, with a slight abuse of notation, we use $M_\theta(x)$ to refer to both the raw probability output and the thresholded discrete prediction, where the exact meaning will be clear from the context. Then, the discrete prediction yields 1 if the raw output $M_\theta(x) \geq 0.5$, and 0 if the raw output $M_\theta(x) < 0.5$.

2.2 Counterfactual Explanations

In tabular classification, most existing work focuses on computing CE in a binary classification setting for algorithmic recourse purposes. This aims to provide recommended action for data subjects who are negatively affected by a ML decision. In this section, we introduce the notions suitable for binary classification, while later in the technical chapters, we will discuss how the proposed methods are readily extended to multi-class classification.

Consider a classification model M (where clear from context, we drop the parameter subscript θ for notational simplicity) trained to solve a binary classification problem. Assume an input $x \in \mathcal{X}$ is given for which $M(x) = 0$. Intuitively, a CE is a new input x' which is somehow similar to x, e.g., in terms of some specified distance between feature values, and for which $M(x') = 1$. Formally, existing methods in the literature may be understood to compute counterfactuals by solving this optimisation problem:

$$\underset{x' \in \mathcal{X}}{\arg\min} \quad dist(x, x') \tag{2.1a}$$

$$\text{subject to} \quad M(x') = 1 \tag{2.1b}$$

where $dist : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is a distance metric.

A CE thus corresponds to the closest input x' (Eq. 2.1a) belonging to the original input space that makes the classification flip (Eq. 2.1b). Eq. 2.1a and 2.1b are typically referred to as *proximity* and *validity* properties, respectively, of CEs. A common choice for the distance metric *dist* is the normalised L_1 distance [142] for sparse changes in the CEs, which we also adopt in this thesis. Under this choice of *dist*, we note that when M is a piece-wise linear model or a tree-structured model, an exact solution to Eqs. 2.1 a-b can be computed with mixed integer programming – see, e.g., [98]. Such methods to compute CE will require white-box access to the classifier.

For differentiable models such as neural networks, x' can be approximated by solving this optimisation problem via gradient descent [142]:

$$\underset{x' \in \mathcal{X}}{\arg\min} \quad dist(x, x') + \lambda \cdot BCE(M(x'), 1) \tag{2.2}$$

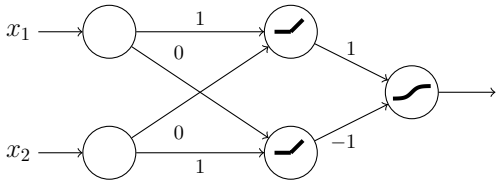
where $\lambda \in [0, 1]$ is a trade-off parameter and BCE is the binary cross-entropy loss.

Alternatively, a simple yet effective model-agnostic approach would be to take the point closest to x from the training dataset, which is predicted with a label 1 [21]. Although this method only uses the black-box prediction function of the classifier, it will require access to the training dataset.

Below, we provide two example CEs following the examples 1 and 2. These will be reused in Chapter 4.

Example 3. *(Continuing from Example 1.) Assume a logistic regression classification model $M_\theta(x) = \sigma(-x_1 + x_2)$ for any input $x = [x_1, x_2]$, where σ is the standard sigmoid function. For a concrete input $x = [0.7, 0.5]$, we have $M_\theta(x) = 0$. A possible CE for x may be $x' = [0.7, 0.7]$, for which $M_\theta(x') = 1$.*

Example 4. *(Continuing from Example 2.) Consider the fully-connected feed-forward neural network M_θ below, where weights are as indicated in the diagram, biases are zero. Hidden layers use ReLU activations, whereas the output node uses a sigmoid function. The network receives a two-dimensional input $x = [x_1, x_2]$ and produces output $M_\theta(x)$.*



The symbolic expressions for the output is $M_\theta(x) = \sigma(\max(0, x_1) - \max(0, x_2))$*, where* σ *denotes a sigmoid function with the usual meaning. Given a concrete input* $x = [1, 2]$*, we have* $M_\theta(x) = 0$*. A possible CE may be* $x' = [2.1, 2]$*, for which* $M_\theta(x') = 1$*.*

2.3 Generative Deep Learning

The CE methods mentioned in Section 2.2 leverage the trained classifier's classification function or the model internals. It is not intuitive to explicitly incorporate the requirement to generate in-distribution CEs into these methods. To capture the distribution pattern of the dataset, another branch of CE research learns an external generative model on the training dataset, which is then used for generating CEs.

In this section, we first introduce VAE [80], which is a popular generative model and is used by several CE methods [74, 108]. Then, we cover a sophisticated version of VAE, Gaussian Mixture VAE (GMVAE), which will be adapted as the generative model in our proposed method in Chapter 6. We introduce these models on the input space $\mathcal{X} \subseteq \mathbb{R}^d$, and show how they can be adapted to capture the class label information in Chapter 6.

2.3.1 Variational Autoencoders

Given some input space $\mathcal{X} \subseteq \mathbb{R}^d$, the core idea of a generative model is to learn the underlying distribution of the data, typically by introducing a latent space $\mathcal{Z} \subseteq \mathbb{R}^h$ that captures the structure of the complex, high-dimensional input space.

VAE-like models assume that the observed variable $x \in \mathcal{X}$ is generated from

a continuous latent variable $z \in \mathcal{Z} \subseteq \mathbb{R}^h$. The standard VAE requires that z is governed by a simple prior distribution, typically a multivariate Gaussian. We denote a Gaussian distribution with mean μ and variance σ^2 as $\mathcal{N}(\mu, \sigma^2)$. The generative process of a VAE is defined as the following:

$$p(x, z) = p(z)\, p(x \mid z) \tag{2.3a}$$
$$p(z) = \mathcal{N}(z \mid 0, I) \tag{2.3b}$$
$$p(x \mid z) = \mathcal{N}(x \mid \mu_x(z), \sigma_x^2(z)) \tag{2.3c}$$

where $p(z)$ is the fixed (Gaussian) prior and $p(x \mid z)$ is the likelihood function, or decoder, parameterised by a neural network.

The inference model $q(z \mid x)$, also known as the encoder, approximates the true but intractable posterior $p(z \mid x)$.

$$q(z \mid x) = \mathcal{N}(z \mid \mu_z(x), \sigma_z^2(x)) \tag{2.4}$$

This approximate posterior is a Gaussian distribution whose parameters, the mean $\mu_x(z)$ and the variance $\sigma_x^2(z)$, are predicted by an encoder neural network. To enable gradient-based optimisation, the sampling of z is performed using the reparameterisation trick, where $z = \mu_z(x) + \sigma_z(x)\, \epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$. The model is trained by maximising the Evidence Lower Bound (ELBO), formulated as:

$$\begin{aligned} L_{ELBO} &= \mathbb{E}_{q(z|x)}[\log \frac{p(x, z)}{q(z \mid x)}] = \mathbb{E}_{q(z|x)}[\log p(x, z) - \log q(z \mid x)] \\ &= \mathbb{E}_{q(z|x)}[\log p(x \mid z)] - \mathbb{E}_{q(z|x)}[\log \frac{q(z \mid x)}{p(z)}] \end{aligned} \tag{2.5}$$

The first term is the reconstruction loss, which encourages the decoder to faithfully reconstruct the input x from its latent representation z. The second term is the Kullback-Leibler (KL) divergence, which acts as a regulariser, pushing the learned posterior distribution $q(z \mid x)$ to be close to the fixed prior distribution $p(z)$.

2.3.2 Gaussian Mixture Variational Autoencoders

The GMVAE model [35, 129] assumes that the latent variable $z \in \mathbb{R}^h$ is governed by a Gaussian Mixture prior which has K Gaussian components. A discrete variable $c \in \mathcal{C} = \{0, \ldots, K-1\}$ controls the cluster assignment for the mixture, and it conforms to a categorical distribution, $\text{Cat}(1/K)$. The generative process of GMVAE is defined as the following:

$$p(x, c, z) = p(c)\, p(z \mid c)\, p(x \mid z) \tag{2.6a}$$

$$p(c) = \text{Cat}(1/K) \tag{2.6b}$$

$$p(z \mid c) = \mathcal{N}(z \mid \mu_z(c), \sigma_z^2(c)) \tag{2.6c}$$

$$p(x \mid z) = \mathcal{N}(x \mid \mu_x(z), \sigma_x^2(z)) \tag{2.6d}$$

The inference model $q(z, c \mid x)$ of the GMVAE is factorised as:

$$q(z, c \mid x) = q(c \mid x)\, q(z \mid x, c) \tag{2.7}$$

where $q(c \mid x)$ is a categorical distribution and $q(z \mid x, c)$, the approximate posterior, is a Gaussian. Both are approximated using neural network models. Practically, the encoder takes in x as an input, then predicts logits over the clusters, which are then concatenated with x (or its processed version) to predict the Gaussian parameters ($\mu_z(c)$, $\sigma_z^2(c)$) for sampling z. The standard reparameterisation trick is also applied at this step for differentiable training. Note that, unlike standard VAE, the prior distribution of z, a GMM, is also parameterised and has an associated learnable network component. This is updated via the KL divergence term (the second term in Equation 2.8 below). The ELBO for this model is:

$$\begin{aligned} L_{ELBO} &= \mathbb{E}_{q(z,c|x)}[log \frac{p(x, c, z)}{q(z, c \mid x)}] = \mathbb{E}_{q(z,c|x)}[log\ p(x, c, z) - log\ q(z, c \mid x)] \\ &= \mathbb{E}_{q(z,c|x)}[log \frac{p(c)}{q(c \mid x)} + log \frac{p(z \mid c)}{q(z \mid x, c)} + log\ p(x \mid z)] \end{aligned} \tag{2.8}$$

In Chapter 6, we will see how the GMVAE model can be adapted to a high-quality CE generator.

2.4 Computational Argumentation

Computational Argumentation provides a formal way of modelling and resolving conflicts between various entities. In this thesis, we use it to address the recourse under model multiplicity problem in Chapter 7. In this section, we provide an overview of relevant background in argumentation.

An *abstract argumentation framework* (AAF) [39] is a tuple $\langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{A}$ is a set of *arguments*, $\mathcal{T} \subseteq \mathcal{A} \times \mathcal{A}$ is a directed relation of *attack*. Given an AAF $\langle \mathcal{A}, \mathcal{T} \rangle$, for any $\alpha_1 \in \mathcal{A}$, we refer to $\mathcal{T}(\alpha_1) = \{\alpha_2 | (\alpha_2, \alpha_1) \in \mathcal{A}\}$ as the *attackers of* α_1. Let us construct an example AAF for the loan example in Figure 1.1 introduced in Section 1.1, with each model as an argument. An attack exists between two models if their predictions for the applicant are different. This AAF is visualised in Figure 2.1.

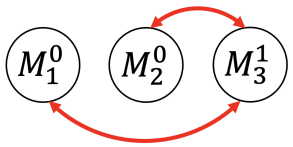


Figure 2.1: AAF for the loan example where model's predictions for the input x are given as superscripts, i.e., $M_1(x) = 0$, $M_2(x) = 0$ and $M_3(x) = 1$; reciprocal attacks are represented by double-headed red arrows.

We use the notions of *acceptability* of sets of arguments in AAFs [39], as follows. A set of arguments $X \subseteq \mathcal{A}$, also called an *extension*, is said to be *conflict-free* iff $\nexists \alpha_1, \alpha_2 \in X$ such that α_1 attacks α_2. A set X *defends* $\alpha_1 \in \mathcal{A}$ iff $\forall \alpha_2 \in \mathcal{A}$, if α_2 attacks α_1 then $\exists \alpha_3 \in X$ such that α_3 attacks α_2. The notion of a set being *admissible* requires that X is conflict-free and defends all its elements. X is said to be *preferred* iff it is the maximal admissible set wrt set-inclusion ($\subseteq$-maximal onwards). The admissible sets in our example are $\{M_1\}, \{M_2\}, \{M_3\}$, $\{M_1, M_2\}$, and the preferred extensions are $\{M_1, M_2\}$ and $\{M_3\}$. Note that each preferred extension groups together the models with the same prediction.

Similarly, a *bipolar argumentation framework* (BAF) [24] is a tuple $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$, where $\langle \mathcal{A}, \mathcal{T} \rangle$ is an AAF and $\mathcal{S} \subseteq \mathcal{A} \times \mathcal{A}$ is a directed relation of *direct support*. Given a BAF $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$, for any $\alpha_1 \in \mathcal{A}$, we refer to $\mathcal{S}(\alpha_1) = \{\alpha_2 | (\alpha_2, \alpha_1) \in \mathcal{S}\}$

as α_1*'s direct supporters*. Then, an *indirect attack* from α_x on α_y is a sequence $\alpha_1, r_1, \ldots, r_{n-1}, \alpha_n$, where $n \geq 3$, $\alpha_1 = \alpha_x$, $\alpha_n = \alpha_y$, $r_1 \in \mathcal{T}$ and $r_i \in \mathcal{S}$ $\forall i \in \{2, \ldots, n-1\}$. Similarly, a *supported attack* from α_x on α_y is a sequence $\alpha_1, r_1, \ldots, r_{n-1}, \alpha_n$, where $n \geq 3$, $\alpha_1 = \alpha_x$, $\alpha_n = \alpha_y$, $r_{n-1} \in \mathcal{T}$ and $r_i \in \mathcal{S}$ $\forall i \in \{1, \ldots, n-2\}$. Straightforwardly, a supported attack on an argument implies a direct attack. Let us construct an example BAF extending the AAF example (Figure 2.1), shown in Figure 2.2. Assume there are three more arguments c_1, c_2 and c_3, respectively having bi-directional supports with M_1, M_2 and M_3. Then, for example, there is an indirect attack from M_3 to c_1, and a supported attack from c_1 to M_3.

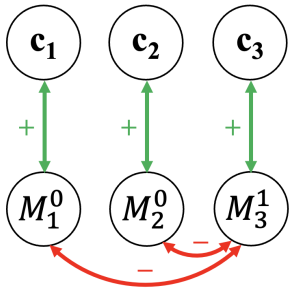


Figure 2.2: An example BAF extending the previous AAF example; reciprocal supports are represented by dual-headed green arrows labelled with + and reciprocal attacks are represented by dual-headed red arrows labelled with −.

We will also use notions of acceptability of sets of arguments in BAFs [24]. A set of arguments (or extension) $X \subseteq \mathcal{A}$ is said to *set-attack* any $\alpha_1 \in \mathcal{A}$ iff there exists an attack (whether direct, indirect or supported) from some $\alpha_2 \in X$ on α_1. Meanwhile, X is said to *set-support* any $\alpha_1 \in \mathcal{A}$ iff there exists a direct support from some $\alpha_2 \in X$ on α_1.[1] For example, in Figure 2.2, $\{c_1, c_2\}$ set-attacks M_3 (because both c_1 and c_2 supported-attack M_3) and set-supports M_2. Then, a set $X \subseteq \mathcal{A}$ *defends* any $\alpha_1 \in \mathcal{A}$ iff $\forall \alpha_2 \in \mathcal{A}$, if $\{\alpha_2\}$ set-attacks α_1 then $\exists \alpha_3 \in X$ such that $\{\alpha_3\}$ set-attacks α_2. Any set $X \subseteq \mathcal{A}$ is then said to be *conflict-free* iff $\nexists \alpha_1, \alpha_2 \in X$ such that $\{\alpha_1\}$ set-attacks α_2, and *safe* iff $\nexists \alpha_3 \in \mathcal{A}$ such that X set-attacks α_3 and either: X set-supports α_3; or $\alpha_3 \in X$. (Note that a safe set is guaranteed to be conflict-free.) In the example, the set $\{c_2, c_3\}$ is conflict-free but is not

[1] In [24], set-supports are defined via sequences of supports, which we do not use here.

safe, because the two arguments, while not directly attacking each other, simultaneously supported-attack and support the same arguments (M_2 and M_3). The notion of a set $X \subseteq \mathcal{A}$ being *stable* requires that X is conflict-free, and that X set-attacks all arguments which are not in X. Similarly, the notion of a set $X \subseteq \mathcal{A}$ being *d-admissible* (based on admissibility in [39]) requires that X is conflict-free and defends all of its elements. In our running BAF example, the set $\{c_3, M_3\}$ is both stable and d-admissible. This notion is extended to account for safe sets: X is said to be *s-admissible* iff it is safe and defends all of its elements. (Thus an s-admissible set is guaranteed to be d-admissible.) Further, X is said to be *c-admissible* iff it is conflict-free, closed for $\mathcal{S}$ and defends all of its elements. Finally, X is said to be *d-preferred* (respectively, *s-preferred*, *c-preferred*) iff it is $\subseteq$-maximal d-admissible (respectively, s-admissible, c-admissible). Continuing with the example, the set $\{c_1, c_2, c_3\}$ is stable and d-preferred, but is not s- or c-preferred because it is not safe or closed for support relations. On the contrary, the set $\{c_1, M_1, c_2, M_2\}$ satisfies the requirements for being stable, d-, s-, and c-preferred.

2.5 Summary

In this section, we covered the technical background needed for later sections in this thesis. Classification (Section 2.1) and CE generation (Section 2.2) are the problem settings, while generative deep learning (Section 2.3) and argumentation (Section 2.4) are tools which we leverage for the new CE methods. Additional tools we use include interval abstraction methods for neural networks (based on mixed-integer programming) and robust optimisation, the optimisation formulations for which will not require heavy preliminaries and will be introduced directly in the corresponding technical chapters.

Chapter 3

Literature Review

This chapter presents an overview of existing CE generation methods. As discussed in Section 2.2, the two fundamental requirements for CEs are validity and proximity. Section 3.1 additionally introduces other traditional desirable properties CEs should satisfy for recourse purposes. Section 3.2 systematically reviews the robustness property of CEs, providing a taxonomy of four types of robustness scenarios and introducing existing CE approaches to address them. For completeness of this literature review, some papers that constitute the technical chapters for this thesis are also positioned in the overview alongside prior work (see Table 3.1).

3.1 Desirable Properties of Counterfactual Explanations

Various methods for generating CEs in classification tasks have been proposed throughout the recent surge in XAI research, often optimising for one or more metrics characterising desirable properties of CEs. [137] focuses on tree-based classifiers and evaluates the CEs' validity, whereas Wachter et al. [142] formulate the CE search problem for differentiable models as a gradient-based optimisation problem and evaluates also CEs' proximity. These two metrics remain a prominent research focus, e.g., [98] treats CE generation

in neural networks as a constrained optimisation problem such that formal validity and proximity guarantees can be given.

Another important property is *plausibility*, requiring that the generated CEs are not outliers of the data distribution behind a dataset. To address this, Brughmans et al. [21] find dataset points which are naturally on the data manifold as CEs. [117] uses density-based metrics to obtain realistic CE points, which are connected to the original input via a path of dataset points. Meanwhile, generative models like VAEs have been used to compute plausible CEs [34, 108, 140, 140]. *Actionability* ensures that the CEs only change the mutable features coherently. This is typically managed by customising constraints in the optimisation process of finding CEs [139]. Mothilal et al. [99] and Dandl et al. [31] build optimisation frameworks for the *diversity* of the generated CEs, aiming to find multiple CEs for one input point which are far away from each other, as measured by some distance metrics.

Additional Considerations. Typically, a CE takes the form of one or more data points per input for the ease of problem formulation and computation. Recent studies also seek more complex forms of CEs for recourse, providing information for the end user richer than a single recourse action. For example, [117] finds a plausible CE with a path of real dataset points linking the input and the CE. Other works view the recourse problem at a more abstract level, finding a single recourse action or counterfactual regions for groups of inputs. [121, 89] adopt frequent itemset mining algorithms to construct global counterfactual rules for summarising a classifier's decision logic. [75, 15] learn decision trees to obtain similar if-then counterfactual rules over subspaces of inputs.

Outside the scope of tabular data classification, studies have also investigated CEs for, e.g., graph data tasks [10, 151], visual tasks [8], textual data [149, 69], time series prediction tasks [33], etc. We refer to [56, 78] for comprehensive overviews.

3.2 Robust Counterfactual Explanations

In this section, we conduct a systematic survey of the existing literature on the robustness of CEs. We first performed keyword searches on Google Scholar using the following patterns for exact matches: *robust counterfactual explanation*, *robustness of counterfactual explanation*, *counterfactual explanation robustness*. For better coverage in each pattern, the words *consistent* (*consistency*) and *stable* (*stability*) were used in addition to *robust* (*robustness*). The phrase *counterfactual explanation* was also interchangeable with *counterfactual explanations, counterfactuals, recourse, algorithmic recourse*. To narrow down the topic, we filtered for technical papers from and after year 2017, when two seminal works in the CE literature were published [137, 142]. For completeness, we also checked in Google Scholar for additional papers that cited the influential early works on robust CEs, namely [109, 138, 130].

Table 3.1 summarises the results of our survey. We found that the type of robustness falls within four separately studied categories, namely robustness against *Model Changes* (MC), *Model Multiplicity* (MM), *Noisy Execution* (NE), and *Input Changes* (IC). For the identified approaches to these problems, we detail the suitable types of models to be explained, the level of access to the model which is required, the computational method used, whether formal guarantees of robustness are given, and other considered properties.

In the next sections, we review the problem definitions, evaluation metrics for robustness, solutions and theoretical results for each category of robustness.

		Models	Access	Method	Guarantee	Properties
MC	Mochaourab et al. [96]	SVM	Pred.	Search	-	-
	Upadhyay et al. [138]	LM	Grad.	GD+RO	-	Ac
	Black et al. [18]	NN	Grad.	GD	-	-
	Bui et al. [22]	LM	Grad.	GD	Prob.	Di
	da Silva and Bertini [29]	$*$	Pred.	Search	-	Pl
	Dutta et al. [40]	TM	Pred.	Search	Prob.	Pl
	Ferrario and Loi [45]	$*$	White	DA	-	-
	Forel et al. [50]	TM	White	MIP	Prob.	Ac/Pl
	Nguyen et al. [104]	DM	Grad.	GD	-	Pl
	Bui et al. [23]	$*$	Pred.	GD, MIP	-	Ac/Pl
	Guo et al. [57]	DM	White	GD+RO	-	Ac
	Hamman et al. [61]	NN	Pred.	Search	Prob.	Pl
	†Jiang et al. [65]	NN	White	MIP+RO	Det.	Ac/Pl
	†Jiang et al. [64]	NN	White	MIP	Det.	-
	Krishna et al. [83]	LM	Grad.	GD	Det.	-
	Nguyen et al. [103]	$*$	Pred.	MIP	Det. / LM	Ac
	Pawelczyk et al. [113]	DM	Grad.	GD	-	-
	Wang et al. [143]	$*$	Pred.	Search	-	Ac/Di/Pl
MM	Pawelczyk et al. [109]	$*$	Pred.	Search	-	Pl
	Leofante et al. [87]	NN	White	MIP	Det.	-
	†Jiang et al. [66]	$*$	Pred.	CA	Det.	-
NE	Hada and Carreira-Perpiñán [60]	TM	White	MIP	-	-
	Dominguez-Olmedo et al. [37]	DM	Grad.	GD+RO	LM	Ac/Ca
	Sharma et al. [128]	$*$	Pred.	Search	-	Di/Pl
	Guyomard et al. [59]	DM	Grad.	GD	Prob.	-
	Leofante and Lomuscio [85]	NN	White	FV	Det.	Ac
	Maragno et al. [91]	NN, TM	White	MIP+RO	Det.	Ac/Pl
	Pawelczyk et al. [112]	DM	Grad.	GD	Prob.	Ac
	Raman et al. [120]	DM	Grad.	Sampling	Prob.	Ac/Ca/Di/Pl
	Virgolin and Fracaros [141]	$*$	Pred.	GA	-	Pl
IC	Artelt et al. [5]	$*$	White	MIP	Prob. / LM	Pl
	Slack et al. [130]	DM	Grad.	GD	-	-
	Wang et al. [144]	$*$	Pred.	SAT	-	Di
	Zhang et al. [152]	$*$	Pred.	Search	-	Pl
	Leofante and Potyka [86]	$*$	Pred.	Search	Det.	Di

Table 3.1: Robust CE methods, partitioned based on the targeted robustness form. We include: *(i)* the types of models they target (support vector machines (SVM), linear models (LM), neural networks (NN), differentiable models (DM), tree-based models (TM) and model agnostic methods ($*$)); *(ii)* the model access required (white box, gradients and predictions); *(iii)* the method used (gradient descent (GD), robust optimisation (RO), mixed-integer programming (MIP), computational argumentation (CA), formal verification (FV), genetic algorithms (GA), satisfiability solving (SAT) and data augmentation (DA)); *(iv)* whether a robustness guarantee is provided (Deterministic, Probabilistic, for linear models only (LM) and none ($-$)); *(v)* other properties satisfied by the method, including Actionability, Causality, Diversity, and Plausibility. Note that all methods consider validity and proximity by default, thus they are omitted. †These works are part of the technical contributions of this thesis.

3.2.1 Robustness to Model Changes

As shown in Table 3.1, the majority of approaches on robustness of CEs in the literature focus on robustness against MC. Model changes are typically defined as modifications in the parameters of a ML model that do not alter its architecture. These changes are typically assumed to result from retraining on inputs drawn from a slightly shifted data distribution [122]. However, model changes resulting from data deletion queries have also been studied [113, 83].

Different characterisations of the notion of change have been proposed. [138] first considered *plausible model changes*, which are defined as updates to model parameters whose magnitude is bounded by a small constant. The same notion is also considered in [18, 65, 64]. The assumption on bounded updates is lifted in [61], where *naturally-occurring model changes* are studied in the context of neural networks. In a nutshell, this notion allows for unbounded changes as long as the updated model preserves a similar behaviour around the input being explained.

The robustness against MC can be summarised as follows.

> **Robustness against MC:** Assume an input x and a model M. Let x' be a CE for x. Robustness against MC requires that whenever the model M **changes** to M', and this change is **sufficiently small**, then $M(x') = M'(x')$.

Despite different definitions of what constitutes a (small) change in this setting, e.g., some approaches require explicitly that $M(x) = M'(x)$, all existing works agree that a lack of such *robustness against MC* could be a problem for both the user and the explanation providers. Consider the loan example: after being rejected, the applicant would expect that when applying again, achieving a £55K annual salary would result in the application being successful, as captured by the CE. However, a lack of robustness to MC may result in this CE being invalidated by small updates to the ML model (e.g., by retraining with new data). Ultimately, this may lead to the application being rejected despite the applicant having implemented the CE. On the

contrary, a robust CE should preserve its validity in such a scenario.

3.2.1.1 Robustness Metrics

A commonly used metric to assess the robustness of CEs against MC is *Validity after Retraining* (VaR). Roughly speaking, VaR measures the percentage of CEs that remain valid after the model for which they were generated is updated. Common update strategies involve retraining from scratch using shifted datasets [138] or different portions of the original dataset [61], incremental retraining [64], and retraining with varying initialisation conditions [18]. This metric is also referred to as *recourse outcome instability* [113] and *recourse reliability* [49], the latter being defined for CEs in a multi-agent multi-step setting. A probabilistic formulation of this property is also given in [22], where the authors compute lower and upper bounds on the probability of a CE remaining valid after retraining of linear models.

To quantify robustness against plausible model changes, the Δ-*robustness* metric was proposed in [64]. Intuitively, a CE is said to be Δ-robust if its validity is preserved across the whole family of models that can be obtained after applying bounded model changes. Given a CE x', this can be can be captured by requiring that $M(x') = M'(x')$ for all M' such that $\|Param(M') - Param(M)\|_p \leq \delta\}$ for fixed p, δ and $Param$ being the vectorisation of a model's parameters.

Counterfactual Stability (CS) is a robustness metric proposed to capture naturally-occurring model changes. A formulation of CS was introduced in [40] for tree ensemble models to capture the intuition that *i.* the class score for a robust CE should be high and that *ii.* the class score of the CE's neighbouring input should also be high with a small variance. In practice, for a given CE, the CS score evaluates the mean and standard deviation of the class scores of samples from a Gaussian distribution centred at the CE. The metric was later specialised in [61] to target naturally-occurring models changes in neural networks.

3.2.1.2 Algorithms

Robust optimisation approaches. Targeting plausible model changes, [138] solve a min-max problem that generates the best CE under the largest admissible model change. At each step, an inner maximisation routine finds the weight perturbation vector that increases the prediction loss to the greatest extent. Then, an outer minimisation loop updates the CE which optimises the overall loss (Eq (2.2) under the previously computed worst-case perturbation). Although this formulation natively supports linear models only, the authors show that it can also be applied to non-linear models by leveraging linear surrogate models obtained, e.g., using LIME [123][1].

Similarly to this approach, [65] present a constrained optimisation formulation of the same problem in the context of feed-forward neural networks with piece-wise activations. Using a MIP encoding instead of gradient-based optimisation, the authors are able to provide stronger robustness guarantees than [138]. However, this comes with higher computational costs.

Increasing class scores. Other approaches generate robust explanations by guiding the search towards CEs that result in high class scores. For example, [83] leverage leave-k-out analysis to approximate the solutions to the robust CEs problem by finding CEs with higher class scores on the original model. [50] increase the required class scores in the CE search and give probabilistic guarantees for ensembles of convex base learners. Similarly, [29] target CEs with high class scores by exploring the input's neighbouring points using k-associated optimal graph analysis. Meanwhile, [96, 143] construct application-dependent CE prototypes and select those with higher class scores as robust CEs.

Complementing these results, [18] show that for complex non-linear models, higher class scores alone may not be sufficient to ensure robustness. Therefore, additional requirements need to be imposed on the search to promote robustness. For instance, [18] require that the CEs be located in a region

[1]Note that in practice, all CE methods applicable to linear models can be also used for non-linear models by approximating them with linear surrogate models, but any theoretical properties could be lost.

with a low Lipschitz constant and propose a method that leverages this result during CE search. Instead, [40, 61] generate robust CEs by requiring high class scores not only for the CE, but also for its neighbouring points. A first search algorithm exploiting this idea is presented in [40] in the context of decision trees; a gradient-based version is later developed in [61]. Both methods provide theoretical probabilistic guarantees for their CEs' VaR Finally, [64] propose to strengthen the class score requirement with a robustness test based on the notion of Δ-robustness. The authors propose an algorithm that iteratively generates CEs with higher class scores until the candidate solution is certified to be robust. For the certification step, a MIP encoding is used to explore the space of model changes exhaustively.

Probabilistic modelling. Model changes can also be approximated using probabilistic modelling techniques as done in [104], where kernel density estimation techniques are used to model the data distributions, along with a Gaussian mixture ambiguity set to capture the model shifts. A min-max optimisation problem is then formulated to maximise the worst-case validity probability of CEs and solved using gradient-based optimisation. A similar approach is taken in [103] where the space of possible model shifts is modelled using Gaussian mixtures over model parameters. Finally, [22] obtain a lower bound on the probability of VaR by modelling the mean and covariance matrix of some nominal distribution assumed for the model parameters. Their lower bound can be relaxed to a differentiable form, and thus can be added as an additional term to the loss function of the gradient-based CE search.

(Re-)Training for robustness. Training methods have been proposed to robustify the CE generation process. Augmentation-based training is proposed in [45], where CEs are used alongside training instances to train neural network classifiers. The authors show empirical results pointing to the fact that augmentation increases the chances of CEs remaining valid under successive shifts in the data-generating distribution. However, this technique may cause imbalances in the data, since the CEs are not naturally-collected data points from the same distribution as other training data. [57]

formulate a tri-level optimisation problem which aims to simultaneously train an accurate neural network for prediction purposes, for which robust CEs can be generated by passing any input into another jointly trained neural network. Finally, [23] sample data points (with their prediction results) centred at the original model's decision boundary near the input, and perturb the data covariance matrix of these samples. Then, linear surrogate models which are aware of the potential decision boundary shifts in the original model can be trained using the data shifts. They show that CEs generated on such linear surrogates using non-robust methods are robust.

3.2.2 Robustness against Model Multiplicity

MM (also called *predictive multiplicity*) refers to the phenomenon that for a single ML task, multiple near-optimal models can be trained with similar test accuracies [19, 93, 17]. MM can be dealt with by aggregating the models towards a single *aggregated prediction* for each input [16, 17]. However, recent results have shown that these models could have distinct behaviours for the same individuals in terms of predictions and their explanations [124, 27]. This has important implications for CEs, with one being that a CE for an aggregated prediction may not even be valid for all of the aggregated models, which would likely confound users' expectations, leading to the following high-level definition:

> **Robustness against MM:** Assume an input x and a set of models $\mathcal{M}$. Let $agg(\mathcal{M}, x)$ be an **aggregated prediction** for x, and x' be a CE for $agg(\mathcal{M}, x)$. Robustness against MM requires that x' is **valid across some subset** $\mathcal{M}' \subseteq \mathcal{M}$, i.e., for any $M_i \in \mathcal{M}'$, $M_i(x') \neq agg(\mathcal{M}, x)$.

In the CE literature, [109] observe that traditional CEs generated on one original model have a high probability of being invalidated by other models resulting from MM. In this setting, it is assumed that *the prediction result is fixed to one produced by one of the original models*, and the robustness target is to guard the CE validity against potential future changes in the classifier: a robust CE should ideally remain valid under the alternative models. This

form of robustness is very similar to robustness against MC. The differences are that the MM problem does not assume a data distribution shift causing the model shifts, and there are no assumptions on the model types and architectures for the original model on which the CEs are generated and for the alternative models.

Recent advances have also raised concerns about *when the prediction result for one input is pending* and to be decided by all the models under MM, instead of assuming a result from one original model [16]. [66] investigate the problem of deciding the prediction result together with the CEs, with the robustness of the latter driving the former. That is, the solution should be a subset of models under MM and their (valid) CEs, from which the prediction results are decided.

3.2.2.1 Robustness Metrics

The robustness evaluation for the fixed prediction scenario is very similar to the VaR for robustness against MC problem, using validity under alternative models. These models could be arbitrary models trained on (different portions of) the same dataset with high accuracy. Thus, the same validity metric gives an intuitive evaluation of how robust the CEs are.

For the pending prediction scenario, [66] propose desirable properties to characterise the optimal resulting set of models and CEs. *Non-emptiness* requires that the resulting set be not empty, and ideally it should contain more than one model and CE such that the prediction result is collectively determined by multiple models with their explanations (*non-triviality*). *Model agreement* states that the models included in the solution set should have the same prediction result for the input. If these results are the same as the prediction results by the majority of models in the MM model set, then the solution is said to satisfy *majority vote*. Meanwhile, *counterfactual validity* requires that every selected CE remains valid on every selected model. Finally, *counterfactual coherence* enforces that if a model is selected as part of the solution set, then its corresponding CE should also be included, and vice versa.

3.2.2.2 Algorithms

Fixed prediction scenario. [109] provide important theoretical results demonstrating that the plausible CEs within the data manifold are more likely to stay valid under alternative models, compared with the closest CEs found by optimising Eqs (2.1) or (2.2), and that there is a plausibility-cost tradeoff, indicating that the more robust CEs will admit higher costs. CEs empirically show increased (but much lower than 100%) validity under alternative models for an existing plausible CE method compared with non-plausible methods, along with increased costs. However, the method is not deployed to produce robust CEs. [87] propose a solution to generating CEs that are guaranteed to be valid for a set of ReLU neural networks. Their method builds a product construction combining all specified models into a single neural network, on which CEs can be generated by MIP. They also prove that finding a CE that is valid across a set of piece-wise linear models is NP-complete.

Pending prediction scenario. Leveraging methods in computational argumentation, [66] present a novel argumentative ensembling method for finding a subset of the models and their CEs which satisfies the properties mentioned in Section 3.2.2.1 (with the exception of majority vote). In essence, this consists in grouping similarly-behaving models together to eliminate conflicts therebetween, then finding the maximal group of models and CEs as the final output. Alternatively, one could first decide the prediction result for the input using ensembling methods like majority voting or the method of [16], then select the group of models and CEs with this prediction as the final solution set.

3.2.3 Robustness against Noisy Executions

Traditional methods compute CEs under the assumption that the user receiving them will follow them to the letter. In practice, achieving the prescribed feature values exactly might not always be feasible, especially for the continuous features. Continuing from the example in Section 3.2.1, a CE

may suggest a salary increase of £5342.04 but achieving this exact change could be beyond the loan applicant's control. In some other cases, even if the prescribed change is achievable, the user may inadvertently introduce some noise in the implementation of the recourse. Broadly speaking, robustness to NE requires that a CE be invariant to these small changes

> **Robustness against NE:** Assume an input x and a model M. Let x' be a CE for x. Robustness against NE requires that whenever a **small perturbation** σ is applied to x', validity is not affected, i.e., $M(x') = M(x' + \sigma)$.

Different characterisations of σ have been proposed in the literature. For instance, [112, 120] assume noise could be sampled from some probability distribution. A more conservative formulation is used in [37, 85], where the authors take an adversarial ML and/or verification angle, and define σ as a p-norm ball around the CE. A specialised version of the latter is given in [141], in which a set of feature-specific perturbation vectors is obtained from domain knowledge.

3.2.3.1 Robustness Metrics

[112] formalise the notion of *invalidation rate* (IR) of a CE as the expectation of the difference between the predicted label of the CE and those predicted for its noisy variants, i.e., $\mathbb{E}_\sigma[M(x') - M(x' + \sigma)]$. When used to empirically evaluate CEs, the expectation component can be approximated by sampling a certain number of noise vectors.

Alternatively, inspired by the adversarial robustness literature, [37] evaluate robustness by adding perturbation vectors (found by adversarial attack methods) to CEs to invalidate them, and measuring the minimum magnitude of the successful attack. Then, the CEs are said to be robust to perturbation vectors of up to this magnitude. Similarly, [85] employ formal verification tools to check the *local robustness* of a CE. In particular, a verification query is formulated to check if a CE remains valid within an ∞-norm ball around it. These notions are more conservative than the IR and complement it by

capturing worst-case scenarios.

3.2.3.2 Algorithms

Robust optimisation and class scores. [37] propose different solutions for both linear and non-linear models. For the linear case, the authors prove that increasing class scores is a sufficient condition for finding robust CEs. Similar remarks are also made in [60, 128]. In the non-linear case, a robust optimisation approach is proposed. This is similar to [138] for the MC problem (Section 3.2.1.2), except that the worst-case perturbation now applies to the CE itself instead of the model parameters. [91] provide a similar formulation using a MIP encoding, finding CEs with robustness guarantees for both neural networks and tree ensemble models.

Verification-based approach. [85] show that local robustness queries as commonly phrased in adversarial ML can be used to evaluate a given CE's robustness. Then, the authors provide an iterative algorithm to quantify robustness by embedding a local robustness check into a binary search procedure.

Novel loss functions. [112] propose a method that allows the end user to specify the IR of their CEs. The intuition behind this is that robust CEs typically have higher cost, which the user may not be willing to sustain. Specifying a threshold on the IR allows the end user to control the level of risk they intend to take, thus implicitly reducing the cost from the most robust CEs. The authors give a differentiable first-order approximation of the IR for linear models and Gaussian noise applied to the CE. This term can then be added to the loss function of Eq (2.2) and optimised with gradient descent. Under specific assumptions on the choices of models and CE generation algorithms (e.g., logistic regression models and using the method of [142] as the base CE method), IR can be expressed analytically, and a probabilistic robustness guarantee in terms of IR can be obtained through setting the class score. To tackle these strong assumptions and their inherent limitations on practicality, [59] relax the definition of IR to capture

class scores with which they provide a tighter upper bound on the exact IR. Then, the upper bound approximation by Monte Carlo estimation is plugged into the loss function and CEs are optimised in the same manner as in [112]. In [141], a robustness term which is similar to IR is added to Eq (2.1), and then minimised via a genetic algorithm.

Probabilistic approaches. The methods of [37, 91, 85] effectively obtain a region containing valid CE points. Such regions could also be characterised by probability distributions, from which CEs could be sampled. [120] take a Bayesian hierarchical modelling approach where noise is explicitly modelled by random variables with Gaussian or scaled Dirichlet distributions, for continuous and categorical variables respectively. Their method outputs a posterior distribution which estimates the counterfactual distribution, and a Hamiltonian Monte Carlo sampling method is applied to generate diverse CEs.

3.2.4 Robustness against Input Changes

Unlike the previous notions of robustness which mostly focus on the validity of CEs under certain perturbations, robustness against IC requires that CEs generated for similar inputs are consistent. A first argument in favour of this property is given by [62], who advocates that explanations generated for similar inputs should not differ radically to improve the justifiability of explanations. However, [130] found that traditional CE generation algorithms may fail to satisfy this property, raising fairness concerns. As an example, the paper shows that neural networks can be trained in such a way that individuals belonging to a non-protected group can always obtain a lower cost recourse when compared to protected group. Following this example, we can formulate robustness against IC as follows.

Robustness against IC: Assume two inputs $x^{(1)}, x^{(2)}$ and a model M such that $M(x^{(1)}) = M(x^{(2)})$. Let $x'^{(1)}, x'^{(2)}$ be CEs for $x^{(1)}, x^{(2)}$, respectively. Robustness against IC requires that whenever $x^{(1)}, x^{(2)}$ are **similar**, then $x'^{(1)}, x'^{(2)}$ are also **similar**.

Similarity between inputs is typically defined in terms p-norm balls, e.g., given an input $x^{(1)}$, an input $x^{(2)}$ is similar to $x^{(1)}$ if $\|x^{(1)} - x^{(2)}\|_p \leq \delta$. Then, one could use this formulation to derive a similarity notion for CEs, bounding the maximum distance therebetween. For example, [86] discuss the case where $\|x'^{(1)} - x'^{(2)}\|_p \leq k \ \|x^{(1)} - x^{(2)}\|_p, k \in \mathbb{R}^+$. A special case of robustness against IC is encountered when dealing with inputs with missing feature values, where $x^{(2)}$ differs from $x^{(1)}$ only in the missing attributes. The returned CEs should ideally capture validity for any missing feature values, as discussed in [76].

3.2.4.1 Robustness Metrics

Robustness measures are usually characterised by the expected distance between the CEs of similar inputs, $\mathbb{E}_{x^{(2)} \sim S}[dist(x'^{(1)}, x'^{(2)})]$, where S denotes a set of inputs that are similar to $x^{(1)}$. This quantity is identified as the *local instability* of CEs [5], and is targeted by most of the surveyed studies. If the method generates a diverse set of CEs for a single input, then local instability needs to be generalised to account for distances between two sets of points, as in [144, 86].

3.2.4.2 Algorithms

Adversarial manipulations and defenses. [130] propose an adversarial training framework whereby the cost of CE changes can be made to change drastically for across protected and unprotected subgroup of data points. They demonstrate that such instabilities against small perturbations in the input are evident for traditional gradient-based CE methods, and discuss possible mitigation strategies. These include randomising the CE search

initialisation, reducing the number of features used for CE search, or reducing the model size to limit overfitting.

Plausibility and robustness. [5] give some theoretical results about the upper bound of the local instability of CEs for linear binary classifiers under Gaussian and Uniform noise in the input. They show that plausible CEs, i.e., CEs that lie within the data manifold, exhibit a higher degree of invariance to input perturbations. This insight is also the intuition behind the method by [152] which aims to maximise robustness to input changes by finding more plausible CEs. Similarly, [144] proposes a boolean satisfiability approach to generate plausible CEs, which also demonstrate a lower local instability than other traditional CE methods.

Robustness via diversity. The above methods do not explicitly target the local instability of CEs. [86] show that satisfying robustness notions such as $\|x'^{(1)} - x'^{(2)}\|_p \leq k \, \|x^{(1)} - x^{(2)}\|_p, k \in \mathbb{R}^+$ may be impossible for traditional CE methods in the general case. Therefore, the authors propose to move away from single-instance CEs and instead consider (diverse) sets of CEs. Under some assumptions, the authors prove that a relaxed form of robustness can be satisfied whereby sets of explanations generated for similar inputs are guaranteed to contain similar CEs.

3.3 Summary

This chapter has reviewed the current literature on CEs, with a special focus on the emerging landscape of robust methods. When compared with the existing research, the technical chapters of this thesis all provide unique contributions:

First, Chapters 4 and 5 introduce methods with rigorous robustness guarantees, a feature largely overlooked by prior work. Second, the new method in Chapter 6 is designed to be a comprehensive CE method that balances two forms of robustness with a large range of existing desirable properties, while

being able to obtain multiple paths of CEs in a computationally efficient way and being model-agnostic. Finally, Chapter 7 addresses a novel problem setting related to MM, which has not been formally characterised before.

Chapter 4

Interval Abstractions for Guaranteeing Robustness of Counterfactual Explanations

As discussed in Section 3.2, CEs found by existing methods often become invalid when slight changes occur in the parameters of the model they were generated for. Methods recently introduced to target this problem typically induce robustness either by using dedicated continuous optimisation procedures [104, 138], or by applying post-hoc refinements to candidate CEs found by non-robust CE generation methods [18, 40, 61]. However, due to their approximate nature, these methods are unable to provide formal robustness guarantees, which are advocated as being vital towards achieving trustworthy AI [92].

In this chapter, we fill this gap by presenting a method which provides formal robustness guarantees for CEs under model changes. After discussing the motivations in Section 4.1, Sections 4.2 and 4.3 introduce the formalisations of our targeted type of model changes and the method that can certify CEs' robustness. Then, Section 4.4 details the practical implementation of the certification process. Two CE generation algorithms are then presented in Section 4.5, which are extensively evaluated through experiments in Section 4.6. We finally conclude in Section 4.7.

4.1 Motivation

We target plausible model shifts [138] that can be encoded by a set Δ of norm-bounded perturbations to the parameters of ML models. Given this setting, we propose a novel *interval abstraction* technique, which over-approximates the output ranges of parametric ML models (including neural networks and logistic regressions) when subject to the model shifts encoded in Δ. This technique is inspired by the abstraction method presented in [118], originally proposed for estimating neural networks' outputs by grouping weight edges into weight intervals. We show that using our interval abstraction, a precise notion of robustness for CEs under Δ can be stated and formally verified. In the following, we refer to such notion of robustness as the Δ-robustness of CEs. Unlike most previous robust CE methods which only apply to binary classification, our focus on computing output ranges allows our method to also work on multi-class classification.

The set Δ is inherently constrained by a radius parameter δ, and focusing on this parameter as our primary robustness metric is motivated by its dual theoretical and practical utility. First, δ bounds the worst-case magnitude of model parameter changes, serving as a strictly conservative measure: any CE certified for a given δ is mathematically guaranteed to remain valid under all possible parameter shifts within that radius. Second, despite its conservative theoretical nature, this bound translates naturally to practical reliability. As we demonstrate in Section 4.6, our guaranteed Δ-robustness under increasing small δ values nicely represents the empirical robustness observed when models undergo actual updates or retraining. Finally, to ensure these bounds are practically meaningful without yielding overly pessimistic recourse costs, we detail our principled approach to determining suitable and realistic δ values in Section 4.6.2

Using interval abstractions, we propose two procedures to generate provably robust CEs: an iterative algorithm augmenting existing CE generation methods, and a sound and complete Robust Nearest-neighbour Counterfactual Explanations (RNCE) method. Since this work uses Mixed Integer Linear Programming (MILP) to practically test Δ-robustness, the ensuing presentation will only focus on ML models whose forward pass can be encoded

into a MILP program.

In an extensive empirical study, we demonstrate the effectiveness of our CE generation algorithms in a benchmarking study against seven existing methods. Notably, one configuration of our iterative algorithm finds Δ-robust CEs with the lowest costs among all the robust baselines, and our new RNCE algorithm achieves perfectly robust results while finding CEs close to the data manifold.

The core contributions of this chapter are summarised as follows:

- We propose a novel interval abstraction method to test Δ-robustness, formally verifying whether a CE is robust against plausible model changes.
- Our method explicitly characterises robustness for CEs in multi-class classification.
- We present a principled workflow demonstrating the usefulness of Δ-robustness for evaluating and generating robust CEs in practice.
- We introduce an iterative algorithm and the RNCE algorithm to generate provably robust CEs, which are demonstrated to have superior performances against seven baselines.

4.2 Interval abstractions and robustness for binary classification

In this section, we introduce a novel interval abstraction technique inspired by INNs to reason about the provable robustness guarantees of CEs, under the possibly infinite family of classification models obtained by applying a pre-defined set of plausible model changes, Δ. We formalise the novel notion of Δ-robustness, which, once satisfied, will ensure that the validity of CEs is not compromised by any model parameter change encoded in Δ.

4.2.1 Plausible model changes Δ

First, we formalise the type of model changes central to our method. We begin by defining a notion of distance between two concretisations of a parametric classifier.

Definition 1. *Let M_θ and $M_{\theta'}$ be two instances of the same parametric classification model architecture, parameterised by $\theta \in \mathbb{R}^{d_\theta}$, $\theta' \in \mathbb{R}^{d_{\theta'}}$, and $d_\theta = d_{\theta'}$. For $0 \leq p \leq \infty$, the* p-distance between M_θ and $M_{\theta'}$ *is defined as:*

$$dist_p(M_\theta, M_{\theta'}) = \|\theta - \theta'\|_p$$

.

Example 5. *Consider two models $M_\theta = \sigma(-x_1 + x_2)$ and $M_{\theta'} = \sigma(0.8 \cdot x_1 + x_2)$. Assume $p = \infty$. Then, their p-distance is $dist_p(M_\theta, M_{\theta'}) = \max_i |\theta_i - \theta'_i| = 1.8$.*

Intuitively p-distance compares M_θ and $M_{\theta'}$ in terms of their parameters and computes the distance between them as the p-norm of the difference of their parameter vectors. Using this notion, we next characterise a model shift as follows.

Definition 2. *Given $0 \leq p \leq \infty$, a* model shift *is a function S mapping a classification model M_θ into another $M_{\theta'} = S(M_\theta)$ such that:*

- *M_θ and $M_{\theta'}$ are instances of the same model architecture, parameterised by $\theta \in \mathbb{R}^{d_\theta}$, $\theta' \in \mathbb{R}^{d_{\theta'}}$, and $d_\theta = d_{\theta'}$;*
- *$dist_p(M_\theta, M_{\theta'}) > 0$.*

Model shifts are typically observed in real-world applications when a model is regularly retrained to incorporate new data. In such cases, models are likely to see only small changes at each update. In the same spirit as [138], we capture this as follows.

Definition 3. *Given a classification model M_θ, a threshold $\delta \in \mathbb{R}_{>0}$ and $0 < p \leq \infty$, the* set of plausible model shifts *is defined as:*

$$\Delta = \{S \mid dist_p(M_\theta, S(M_\theta)) \leq \delta\}.$$

When a set of plausible model shifts Δ is applied on M_θ, the resulting maximum change on each model parameter in the original model is conservatively upper-bounded:

Lemma 1. *Consider a classification model M_θ and a set of plausible model shifts Δ with threshold δ and $0 < p \leq \infty$. Then, $\forall M_{\theta'} = S(M_\theta)$, $S \in \Delta$, and $\forall \theta_i$, θ'_i, $i \in [d]$, we have $\theta'_i \in [\theta_i - \delta, \theta_i + \delta]$.*

Proof. combining Definition 3 with Definition 1, we obtain:

$$\left(\sum_{i=1}^{d} |\theta_i - \theta'_i|^p\right)^{\frac{1}{p}} \leq \delta$$

We raise both sides to the power of p:

$$\sum_{i=1}^{d} \left(|\theta_i - \theta'_i|^p\right) \leq \delta^p$$

where the inequality is preserved as both sides are always positive. We now observe this inequation bounds each addend from above, i.e.,

$$|\theta_i - \theta'_i|^p \leq \delta^p$$

Solving the inequation for each addend we obtain $\theta'_i \in [\theta_i - \delta, \theta_i + \delta]$, which gives a conservative upper-bound on the maximum change that can be applied to each parameter in $M_{\theta'}$. □

4.2.2 Interval abstractions

To guarantee robustness to the plausible model changes Δ, methods are needed to compactly represent and reason about the behaviour of a CE under the potentially infinite family of models originated by applying each $S \in \Delta$ to M_θ. In the following, we introduce an abstraction framework that can be used to this end.

Definition 4. *Consider a classification model M_θ with $\theta = [\theta_1, \ldots, \theta_d]$. Given a set of plausible model shifts Δ, we define the* interval abstraction of M_θ under Δ *as $\mathcal{I}_{(\theta,\Delta)} : \mathcal{X} \to \mathbb{I}(\mathbb{R})$ such that:*

- *M_θ and $\mathcal{I}_{(\theta,\Delta)}$ have the same model architecture;*
- *$\mathcal{I}_{(\theta,\Delta)}$ is parameterised by an interval-valued vector $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_d]$;*
- *$\boldsymbol{\theta}_i$, for $i \in [d]$, encodes the range of possible changes induced by the application of any $S \in \Delta$ to M_θ such that $\boldsymbol{\theta}_i = [\theta_i - \delta, \theta_i + \delta]$, where δ is the maximum shift obtainable as per Definition 3.*

Lemma 2. *$\mathcal{I}_{(\theta,\Delta)}$ over-approximates the set of models $M_{\theta'}$ that can be obtained from M_θ via Δ.*

Proof. Lemma 2 states that $\mathcal{I}_{(\theta,\Delta)}$ captures all models $M_{\theta'}$ that can be obtained from M_θ applying a $S \in \Delta$, and possibly more models that violate the plausibility constraint (Definition 3, the p-distances are upper-bounded by δ). The former can be seen by observing that each parameter θ_i in $\mathcal{I}_{(\theta,\Delta)}$ is initialised in such a way to contain all possible parameterisations obtainable starting from the initial value θ_i and perturbing it up to $\pm\delta$ (Lemma 1). As a result, $\mathcal{I}_{(\theta,\Delta)}$ captures all shifted models by construction. $\mathcal{I}_{(\theta,\Delta)}$ also captures more concrete models for which the p-distance is greater than δ, as the bounds used in the interval abstraction, $\boldsymbol{\theta}_i = [\theta_i - \delta, \theta_i + \delta]$, are conservative. □

The following two examples demonstrate the interval abstraction in the context of different classification models.

Example 6. *Continuing from Example 3 (Section 2.2). Consider the same model M_θ and assume a set of plausible model shifts $\Delta = \{S \mid dist_\infty(M_\theta, S(M_\theta)) \leq 0.1\}$. The interval abstraction $\mathcal{I}_{(\theta,\Delta)}$ is defined by the model equation $\mathcal{I}_{(\theta,\Delta)}(x) = \sigma(\boldsymbol{\theta}_1 x_1 + \boldsymbol{\theta}_2 x_2)$, where $\boldsymbol{\theta}_1 = [-1-0.1, -1+0.1]$ and $\boldsymbol{\theta}_2 = [1-0.1, 1+0.1]$. Then, $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]$.*

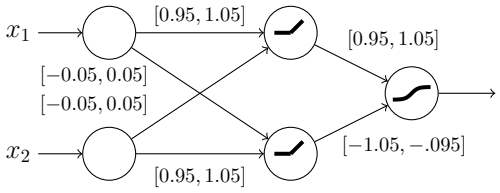


Figure 4.1: Illustration of Example 7.

Example 7. *Continuing from Example 4 (Section 2.2). Consider the same model M_θ and assume a set of plausible model shifts $\Delta = \{S \mid dist_\infty(M_\theta, S(M_\theta)) \leq 0.05\}$. The resulting interval abstraction $\mathcal{I}_{(\theta,\Delta)}$ is represented in Figure 4.1, where the symbolic expression of the output interval is $\sigma([0.95, 1.05] \cdot \max(0, [0.95, 1.05] \cdot x_1 + [-0.05, 0.05] \cdot x_2) + [-1.05, -0.95] \cdot \max(0, [-0.05, 0.05] \cdot x_1 + [0.95, 1.05] \cdot x_2))$.*

Given an input, our interval abstraction maps it to an output interval, constructed by taking the lowest and the highest values from all the possible classification outcomes by any shifted model $M_{\theta'} \in \Delta$. This output interval therefore over-approximates the outputs obtainable from any shifted models in Δ. The classification semantics of the interval abstraction thus departs from those with a single classifier and needs to be generalised to account for this new behaviour.

Definition 5. *Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of a classification model M_θ. Given an input $x \in \mathcal{X}$, let $[l, u]$ be the output interval obtained by applying $\mathcal{I}_{(\theta,\Delta)}$ to x. We say that $\mathcal{I}_{(\theta,\Delta)}$* classifies *$x$ as 1 if $u \geq l \geq 0.5$, 0 if $l \leq u < 0.5$, and* undefined *if $l < 0.5 \leq u$.*

We will slightly abuse notation and use $\mathcal{I}_{(\theta,\Delta)}(x) = 1$ (respectively, $\mathcal{I}_{(\theta,\Delta)}(x) = 0$) to denote the case where the abstraction classifies an input as 1 (respectively, 0). A visual representation of this interval-based classification

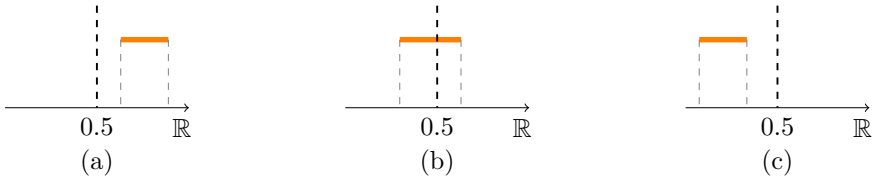


Figure 4.2: Visual representation of Definition 5. In (a), $\mathcal{I}_{(\theta,\Delta)}$ classifies an input as 1 because the output range for that input is always greater than 0.5. In (b), the output range includes value 0.5 therefore the classification result is undefined. In (c), in a similar manner to the way in which the input in (a) is classified as 1, the input is classified as 0.

semantics is given in Figure 4.2. In this work, we are interested in the conservative worst-case robustness of CEs. Therefore, we leave for future work (Section 4.7) the explorations of the undefined case, as in Figure 4.2 (b).

4.2.3 Δ-robustness

Using the interval abstraction as a tool to represent infinite families of shifted models, we now present Δ-robustness, a notion of robustness to model changes which is central to this contribution. First, we define the conditions within which the robustness of a counterfactual can be assessed by introducing a notion of prediction preservation as follows.

Definition 6. *Consider an input $x \in \mathcal{X}$ and a model M_θ such that $M_\theta(x) = 0$. Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ for a set of plausible model shifts Δ. We say that Δ* is prediction-preserving for *x iff $\mathcal{I}_{(\theta,\Delta)}(x) = 0$.*

In other words, prediction preservation requires that shifts in Δ do not alter the class predicted for the original input x. This is a safety requirement that we introduce to ensure consistency in the predictions produced by the interval abstraction.

We then can reason about the robustness properties defined as follows.

Definition 7. *Consider an input $x \in \mathcal{X}$ and a model M_θ such that $M_\theta(x) = 0$. Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ for a set of plausible model shifts Δ. We say that a counterfactual explanation x' is:*

- Δ-robust *iff* $\mathcal{I}_{(\theta,\Delta)}(x') = 1$ *and*
- strictly Δ-robust *iff it is* Δ*-robust &* Δ *is prediction-preserving for* x*.*

The prediction preservation of Δ ensures the theoretical correctness of Δ-robustness. We point out that, in practice, this requirement may be relaxed such that for any input x labelled as class 0 by the original model, one can find a CE x' that is classified as class 1 by all the model shifts in Δ. In this case, strictly speaking, x' cannot be said to be *valid* for Δ because it is possible that for some M_i induced by Δ, $M_i(x) = M_i(x') = 1$. However, in many applications prediction preservation with respect to Δ is not required (as are the setups in e.g., [138, 40]), in which case computing Δ-robust CEs may be more practical than computing those which are strictly Δ-robust.

We conclude with an example summarising the main concepts presented in this section.

Example 8. *(Continuing from Examples 3 and 6). We observe that* $\mathcal{I}_{(\theta,\Delta)}(x)$ *produces an output interval* $[0.42, 0.48]$*, i.e.,* $\mathcal{I}_{(\theta,\Delta)}(x) = 0$*. Since* $M_\theta(x) = 0$*, we can conclude that the set of model shifts* Δ *is prediction-preserving. We then check whether the previously computed counterfactual explanation* $x' = [0.7, 0.7]$ *is* Δ*-robust. Running* x' *through* $\mathcal{I}_{(\theta,\Delta)}$ *we obtain an output interval* $[0.45, 0.53]$*, i.e.,* $\mathcal{I}_{(\theta,\Delta)}(x) = 0$*. Therefore, the counterfactual is not* Δ*-robust. Assume a new counterfactual is obtained as* $x'' = [0.7, 0.86]$*. The output interval produced by* $\mathcal{I}_{(\theta,\Delta)}$ *is now* $[0.5, 0.58]$*, i.e.,* $\mathcal{I}_{(\theta,\Delta)}(x'') = 1$*. Thus, the counterfactual* x'' *is strictly* Δ*-robust.*

4.3 Interval abstractions and robustness for multi-class classification

Next, we introduce the notion of Δ-robustness in multi-class classification settings. Unlike the binary case, where it is usually assumed that class 0 (unwanted class) is the original prediction result for the input and class 1 is the desirable target class for the CE to achieve, in the multi-class settings,

the classes need to be explicitly specified [31, 99]. Given a set of labels $\{1, \ldots, \ell\}$, CEs can be defined as follows.

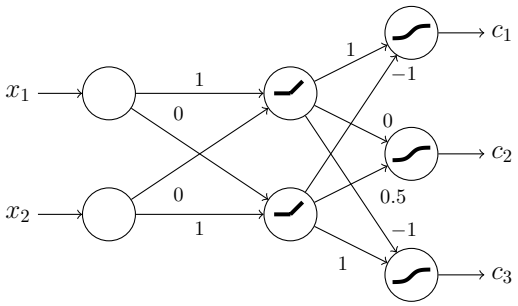
Definition 8. *Given input $x \in \mathcal{X}$ and model M_θ, we say that M_θ* classifies *x as $c \in \{1, \ldots, \ell\}$ if $M_\theta(x)_c \geq M_\theta(x)_{c'}$, for all $c' \in \{1, \ldots, \ell\}$ such that $c' \neq c$.*

Definition 9. *Consider an input $x \in \mathcal{X}$ and a classification model M_θ s.t. $M_\theta(x) = c \in \{1, \ldots, \ell\}$. Assume a desirable target class $c' \in \{1, \ldots, \ell\}$ such that $c' \neq c$ and given a distance metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, a* counterfactual explanation *for class c' is any x' such that:*

$$\begin{aligned} &\arg\min_{x' \in \mathcal{X}} \quad d(x, x') \\ &\textit{subject to} \quad M_\theta(x') = c' \end{aligned}$$

Example 9 illustrates Definition 9 on a toy example where the classifier is a neural network.

Example 9. *Consider the fully-connected feed-forward neural network M_θ below, where weights are as indicated in the diagram, biases are zero. Hidden layers use ReLU activations, whereas the output layer uses a softmax function (activation functions are depicted in the circles). Output classes are $\{c_1, c_2, c_3\}$.*



The symbolic expressions for the output is $M_\theta(x) = Softmax([\max(0, x_1) - \max(0, x_2), 0.5\max(0, x_2), \max(0, x_2) - \max(0, x_1)])$. Given a concrete input $x = [2, 2]$, we have $M_\theta(x) = c_2$. A possible counterfactual explanation for class c_1 may be $x' = [3, 1]$, for which $M_\theta(x') = c_1$.

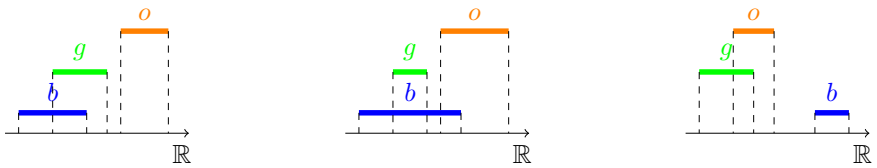


Figure 4.3: Visual representation of Definition 10 instantiated with three colour-coded classes {'orange', 'green', 'blue'}. In (a), $\mathcal{I}_{(\theta,\Delta)}$ classifies an input as o since the output range for that class is always greater than those of any other classes. In (b), the output ranges for o overlap with b, the classification result is therefore undefined. In (c), in a similar manner to the way in which the input in (a) is classified as o, $\mathcal{I}_{(\theta,\Delta)}$ classifies an input as b.

Then, the classification semantics of an interval abstraction for multiclass problems can be obtained by extending Definition 5 as follows.

Definition 10. *Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of a classification model M_θ. Given an input $x \in \mathcal{X}$, let $[l_i, u_i]$ be the output interval obtained for each $i \in \{1, \ldots, \ell\}$ by applying $\mathcal{I}_{(\theta,\Delta)}$ to x. We say that $\mathcal{I}_{(\theta,\Delta)}$* classifies *$x$ as class c, if $l_c \geq u_{c'}$ for all $c' \in \{1, \ldots, \ell\} \setminus \{c\}$. Otherwise we say the classification result from $\mathcal{I}_{(\theta,\Delta)}$ is* undefined *if $\forall c' \in \{1, \ldots, \ell\}$, $\exists c'' \in \{1, \ldots, \ell\} \setminus \{c'\}$ such that $u_{c''} \geq l_{c'}$.*

Figure 4.3 intuitively illustrates the classification semantics of interval abstractions in multi-class settings. Next, we formalise Δ-robustness for CEs.

Definition 11. *Consider an input $x \in \mathcal{X}$ and a model M_θ such that $M_\theta(x) = c \in \{1, \ldots, \ell\}$. Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ for a set of plausible model shifts Δ. We say that Δ* is prediction-preserving for *x iff $\mathcal{I}_{(\theta,\Delta)}(x) = c$.*

Definition 12. *Consider an input $x \in \mathcal{X}$ and a model M_θ such that $M_\theta(x) = c \in \{1, \ldots, \ell\}$. Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ for a set of plausible model shifts Δ. Assume a desirable target class $c' \in \{1, \ldots, \ell\}$ such that $c' \neq c$, we say that a counterfactual explanation x' is:*

- Δ-robust *for class c' iff $\mathcal{I}_{(\theta,\Delta)}(x') = c'$ and*

- strictly Δ-robust *for class* c' *iff it is* Δ*-robust &* Δ *is prediction-preserving for* x*.*

The definition of Δ-robustness transfers to the multi-class semantics by estimating lower and upper bounds for each output class and then checking the robustness property. In Example 10, we instantiate an interval abstraction for a neural network and show how the output intervals can be calculated, thus how Δ-robustness can be examined. We introduce a more general solution for testing Δ-robustness in the next section.

Example 10. *Continuing from Example 9. Consider the same model* M_θ *and assume a set of plausible model shifts* $\Delta = \{S \mid dist_\infty(M_\theta, S(M_\theta)) \leq 0.05\}$*. The resulting interval abstraction* $\mathcal{I}_{(\theta,\Delta)}$ *is defined as in Figure 4.4.*

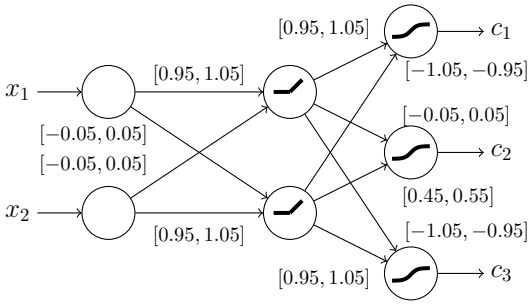


Figure 4.4: Illustration for Example 10

The symbolic expression of the output interval is obtained by applying softmax activation on a vector of intervals $[M_\theta(x)_{c_1}, M_\theta(x)_{c_2}, M_\theta(x)_{c_3}]$ *where each entry is the pre-softmax output interval for output classes* $\{c_1, c_2, c_3\}$*, then sorting the corresponding class interval for each class. We denote the node interval for the hidden nodes as* $M_\theta(x)_{h_1} = \max(0, [0.95, 1.05] \cdot x_1 + [-0.05, 0.05] \cdot x_2)$ *and* $M_\theta(x)_{h_2} = \max(0, [-0.05, 0.05] \cdot x_1 + [0.95, 1.05] \cdot x_2)$*. Then, the output node intervals can be expressed as* $M_\theta(x)_{c_1} = [0.95, 1.05] \cdot M_\theta(x)_{h_1} + [-1.05, -0.95] \cdot M_\theta(x)_{h_2}$*,* $M_\theta(x)_{c_2} = [-0.05, 0.05] \cdot M_\theta(x)_{h_1} + [0.45, 0.55] \cdot M_\theta(x)_{h_2}$*,* $M_\theta(x)_{c_3} = [-1.05, -0.95] \cdot M_\theta(x)_{h_1} + [0.95, 1.05] \cdot M_\theta(x)_{h_2}$*.*

For the input $x = [2, 2]$*, we observe that* $\mathcal{I}_{(\theta,\Delta)}(x)$ *produces output intervals* $[0.114, 0.322], [0.356, 0.51], [0.114, 0.322]$ *respectively for each class, therefore the lower bound of class output node* c_2 (0.356) *is greater than the upper*

bound of the other two classes (0.322)*, i.e.,* $\mathcal{I}_{(\theta,\Delta)}(x) = c_2$*. Since* $M_\theta(x) = c_2$*, we can conclude that the set of model shifts* Δ *is prediction-preserving for* c_2*. We then check whether the previously computed counterfactual explanation* $x' = [3, 1]$ *is* Δ*-robust. Running* x' *through* $\mathcal{I}_{(\theta,\Delta)}$ *we obtain output intervals* $[0.617, 0.91], [0.08, 0.345], [0.01, 0.038]$*, i.e.,* $\mathcal{I}_{(\theta,\Delta)}(x) = c_1$*. Therefore, the counterfactual is* Δ*-robust for class* c_1*. Furthermore, this CE is strictly* Δ*-robust.*

4.4 Computing Δ-robustness with MILP

To determine whether a CE x' is Δ-robust, we are interested in the output ranges of the interval abstraction $\mathcal{I}_{(\theta,\Delta)}$ when x' is passed in. [118] proposed an approach to compute the output ranges of an INN, which we adapt in this section to compute the reachable intervals for the output node in our interval abstraction $\mathcal{I}_{(\theta,\Delta)}$. The output range estimation problem can be encoded in MILP; in the following, we instantiate the encoding for fully connected neural networks with k hidden layers (as first introduced in Example 2) with ReLU activation functions. The encoding introduces:

- a real variable $v_j^{(0)}$ for $j \in [|V^{(0)}|]$, used to model the input of $\mathcal{I}_{(\theta,\Delta)}$;
- a real variable $v_j^{(i)}$ to model the value of each hidden and output node $V^{(i)}$, for $i \in [k+1]$ and $j \in [|V^{(i)}|]$;
- a binary variable $\xi_j^{(i)}$ to model the activation state of each node in $V^{(i)}$, for $i \in [k]$ and $j \in [|V^{(i)}|]$.

Then, for each layer index $i \in [k]$ and neuron index $j \in [|V^{(i)}|]$, the following set of constraints are asserted:

$$
\begin{aligned}
C_j^{(i)} = \Big\{ & v_j^{(i)} \geq 0, v_j^{(i)} \leq \mathcal{M}(1-\xi_j^{(i)}), \\
& v_j^{(i)} \leq \sum_{l=1}^{|V^{(i-1)}|} (W_{j,l}^{(i)} + \delta) v_l^{(i-1)} + (B_j^{(i)} + \delta) + \mathcal{M}\xi_j^{(i)}, \\
& v_j^{(i)} \geq \sum_{l=1}^{|V^{(i-1)}|} (W_{j,l}^{(i)} - \delta) v_l^{(i-1)} + (B_j^{(i)} - \delta) \Big\}
\end{aligned}
\tag{4.2}
$$

where $\mathcal{M}$ is a sufficiently large constant, and δ is the magnitude of model shifts in Δ. Each $C_j^{(i)}$ uses the standard big-M formulation to encode the ReLU activation [90] and estimate the lower and upper bounds of nodes in the INN.

Then, constraints pertaining to the output layer $k+1$ are asserted for each class $j \in |V^{(k+1)}|$.

$$
\begin{aligned}
C_j^{(k+1)} = \Big\{ & v_j^{(k+1)} \leq \sum_{l=1}^{|V^{(k)}|} (W_{j,l}^{(k+1)} + \delta) v_l^{(k)} + (B_j^{(k+1)} + \delta), \\
& v_j^{(k+1)} \geq \sum_{l=1}^{|V^{(k)}|} (W_{j,l}^{(k+1)} - \delta) v_l^{(k)} + (B_j^{(k+1)} - \delta) \Big\}
\end{aligned}
\tag{4.3}
$$

For more details about the encoding and its properties, we refer to the original work [118]. Note that, instead of directly computing the model output, $M_\theta(x) = \sigma(V^{(k+1)})$, our MILP program analyses the interval of the final-layer node values before applying the final sigmoid or softmax function, e.g., $V^{(k+1)} = W^{(k+1)} \cdot V^{(k)} + B^{(k+1)}$. The model output interval can be subsequently obtained by applying these final activation functions.

For the case of binary classification (e.g., as in Example 4), testing Δ-robustness of a CE amounts to solving one optimisation problem as shown in the following.

Definition 13. *Consider an input $x \in \mathcal{X}$ and a fully connected neural network M_θ with k hidden layers, and $M_\theta(x) = 0$. Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ for a set of plausible model shifts Δ. Let ${v_1^{(k+1)}}_{lb}$ be the*

solution of the optimisation problem

$$\begin{aligned} &\underset{v,\xi}{min} && v_1^{(k+1)} \\ &\textit{subject to} && C_j^{(i)},\ i \in [k+1], j \in [|V^{(i)}|] \end{aligned}$$

We say that a counterfactual explanation x' is Δ-robust *if* ${v_1^{(k+1)}}_{lb} \geq 0$.

Indeed, if for a CE x' the lower bound of the output node satisfies ${v_1^{(k+1)}}_{lb} \geq 0$, then after the final sigmoid function, $\sigma({v_1^{(k+1)}}_{lb}) \geq 0.5$, meaning that $\mathcal{I}_{(\theta,\Delta)}(x') = 1$, x' is thus Δ-robust.

For multi-class classification (e.g., as in Example 9), the exact output range for each class $j \in |V^{(k+1)}|$ can be computed by solving two optimisation problems that minimise, respectively maximise, variable $v_j^{(k+1)}$ subject to constraints 4.2-4.3. Referring to Definitions 10 and 12, we are interested in comparing the lower bound of the desirable target class output interval with the upper bound of output intervals from other classes. Therefore, testing Δ-robustness in multi-class classification amounts to solving $|V^{(k+1)}|$ optimisation problems.

Definition 14. *Consider an input $x \in \mathcal{X}$ and a fully connected neural network M_θ with k hidden layers with $M_\theta(x) = c \in \{1, \ldots, \ell\}$ and the desirable target class $c' \in \{1, \ldots, \ell\}$ such that $c' \neq c$. Let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ for a set of plausible model shifts Δ. Let ${v_{c'}^{(k+1)}}_{lb}$ be the solution of the optimisation problem*

$$\begin{aligned} &\underset{v,\xi}{min} && v_{c'}^{(k+1)} \\ &\textit{subject to} && C_j^{(i)},\ i \in [k+1], j \in [|V^{(i)}|] \end{aligned}$$

For each class $c'' \in \{1, \ldots, \ell\}$ such that $c'' \neq c'$, let ${v_{c''}^{(k+1)}}_{ub}$ be the solution of the optimisation problems

$$\begin{aligned} &\underset{v,\xi}{max} && v_{c''}^{(k+1)} \\ &\textit{subject to} && C_j^{(i)},\ i \in [k+1], j \in [|V^{(i)}|] \end{aligned}$$

We say that a counterfactual explanation x' is Δ-robust *for class c' if* $v_{c'}^{(k+1)}{}_{lb} \geq v_{c''}^{(k+1)}{}_{ub}$.

Similar to the binary case, if a CE passes the above Δ-robustness test, then after applying the final softmax function, the predicted class probability of the desirable target class will always be greater than that of the other classes.

In practice, with white-box access to the classifier, the above optimisation problems can be conveniently encoded using any off-the-shelf optimisation solver. In this work we use the Gurobi engine[1].

4.5 Algorithms

We now show how the Δ-robustness tests introduced above can be leveraged to generate CEs with formal robustness guarantees.

4.5.1 Embedding Δ-robustness tests in existing algorithms

We propose an approach (Algorithm 1) that can be applied on top of any CE generation algorithms which explicitly parameterise the tradeoff between validity and cost. For example, if the method is optimising a loss function containing a validity loss term and a cost loss term with a tradeoff hyperparameter (similar to [142]), then modifying the hyperparameter to allow more costly CE with better validity (higher class probability) could lead to more robust CEs. Such heuristics, identified as necessary conditions for more robust CEs, are discussed in several recent studies [18, 40, 61, 83].

The algorithm proceeds as follows. First, an interval abstraction is constructed for the classifier M_θ and targeted plausible model change Δ; the latter is then optionally checked for prediction preservation (Definition 7) using the same MILP programs for Δ-robustness tests, depending on whether

[1] https://www.gurobi.com/solutions/gurobi-optimizer/

Algorithm 1 Iterative algorithm embedding Δ-robustness tests

Require: Classifier M_θ, input x such that $M_\theta(x) = 0$,
set of plausible model shifts Δ and threshold t
Step 1: build encoding for interval abstraction $\mathcal{I}_{(M,\Delta)}$.
Step 2 (Optional): check whether Δ is prediction-preserving
while iteration number $< t$ **do**
    Step 3: compute CE x' for x and M_θ using base method
    if $\mathcal{I}_{(\theta,\Delta)}(x') = 1$ **then**
        return x'
    else
        Step 4: increase allowed distance of next CFX
        Step 5: increase iteration number
return the last x' found

we are aiming at finding strict or non-strict robust CEs. Then, the algorithm performs a Δ-robustness test for the CE generated by the base method. If the test passes, then the algorithm terminates and returns the solution. Otherwise, the search continues iteratively, at each step the hyperparameter of the base method is modified such that CEs of increasing distance can be found. These steps are repeated until a threshold number of iterations t is reached. As a result, the algorithm is incomplete, as it may report that no Δ-robust CE can be found within t steps (while one may exist for larger t). In such cases, the last CE found before the algorithm terminates is returned, assuming the CE found at each iteration is more robust than the previous iteration. As we will see in Section 4.6, we have identified a configuration of our iterative algorithm with a MILP-based method [98] as the base CE generation method, which empirically overcomes the above limitations and is always able to find Δ-robust CEs.

4.5.2 Robust Nearest-neighbour Counterfactual Explanations (RNCE)

We also propose a robust and plausible CE algorithm shown in Algorithm 2, which is complete under mild assumptions. After some initialisation steps, an interval abstraction is constructed for the model M_θ and set Δ in Step 1 (Alg. 2, 6). Similarly to Algorithm 1, the step checking whether Δ is

prediction-preserving is optional. The algorithm then moves on to Step 2 where the dataset $\mathcal{D}$ used to train M_θ is traversed to identify potential CEs for x and filter out unsuitable inputs as described in Algorithm 3. In a nutshell, Algorithm 3 iterates through $\mathcal{D}$ and picks instances that satisfy the counterfactual requirement, parameterised according to a robustness criterion specified by the user via the `robustInit` parameter. When `robustInit` is `T` (`True`), the interval abstraction is used to check, for every instance in the dataset, whether it satisfies Δ-robustness (Definition 7) prior to adding it to $\mathcal{S}$ (Alg. 3, 9). Alternatively, when setting `robustInit` to `F` (`False`), instances are added to the set of candidate CEs $\mathcal{S}$ as long as their predicted label differs from that of x (Alg. 3, 8). In the latter case, the Δ-robustness guarantees are postponed to Step 4.

Once the set $\mathcal{S}$ of potential candidates is obtained from $\mathcal{D}$ (Alg. 2, 8), we fit a k-d tree $\mathcal{T}$ to efficiently store spatial information about them (Step 3). The algorithm then enters its final step, where the best CE is selected from $\mathcal{S}$ as described in Algorithm 4 and returned to the user (Alg. 2, 10). First, $\mathcal{T}$ is queried to obtain the closest robust NNCE x'_{nn}. When Algorithm 3 is instantiated with `robustInit` = `T`, the first nearest neighbour returned from $\mathcal{T}$ is guaranteed to be robust. Otherwise, several queries might be needed to obtain an instance from $\mathcal{T}$ that also satisfies Δ-robustness (Alg. 4, 4-6).[2] Thus, instantiations of different `robustInit` settings help to find the nearest point satisfying Δ-robustness. As the nearest neighbour may not be optimal in terms of proximity to the original instance x, further optimisation steps can be triggered by setting the parameter `optimal` to `T`. This starts a line search to find the closest robust CE to x (Alg. 4, 7-10). The CE computed by Algorithm 4 is then returned to the user.

We stress that when `robustInit`=`T` and `optimal`=`F`, once $\mathcal{T}$ has been fitted, the CE generation time (Alg. 2, 10) for any input is $O(log(|\mathcal{D}|))$, and $\mathcal{T}$ can be used to query CEs for any number of inputs efficiently. When the number of inputs requiring CEs is far smaller than $|\mathcal{D}|$, `robustInit`=`F` may result in faster computation than `robustInit`=`T` (including time for obtaining $\mathcal{T}$), see Appendix A.1 for more details.

[2]For clarity, Algorithm 4 describes the latter case; the robustness check in omitted in the code when `robustInit` is `T`.

Algorithm 2 RNCE: main routine

Require: input x, model M_θ,
1: training dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$,
2: (prediction-preserving) set of plausible model shifts Δ,
3: parameters `robustInit`, `optimal` $\in$ {`F, T`}
4: **Init**: $c \leftarrow M_\theta(x)$; $x' \leftarrow \emptyset$
5: **Step 1**: build encoding for interval abstraction $\mathcal{I}_{(M,\Delta)}$
6: **Step 2**: select candidate counterfactuals
7: $\mathcal{S} \leftarrow$ `getCandidates`$(x, M_\theta, \mathcal{D}, \mathcal{I}_{(\theta,\Delta)}$, `robustInit`$)$
8: **Step 3**: fit k-d tree $\mathcal{T}$ on $\mathcal{S}$
9: **Step 4**: $x' \leftarrow$ `getRobustCE`$(x, M_\theta, \mathcal{I}_{(\theta,\Delta)}, \mathcal{T}$, `optimal`$)$
10: **return** x'

Algorithm 3 RNCE: `getCandidates`

Require: input x, model M_θ,
1: dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^{N}\}$
2: interval abstraction $\mathcal{I}_{(\theta,\Delta)}$
3: parameter `robustInit` $\in$ {`F, T`}
4: **Init**: $c \leftarrow M_\theta(x)$, $\mathcal{S} \leftarrow \{\}$
5: **Step 1**: find candidate subset of the training dataset
6: **for** $(x^{(i)}, y^{(i)}) \in \mathcal{D}$ **do**
7: **if** `robustInit` is `F` **then**
8: Add $(x^{(i)}, y^{(i)})$ to $\mathcal{S}$ if $M_\theta(x_i) = 1 - c$
9: **else** Add $(x^{(i)}, y^{(i)})$ to $\mathcal{S}$ if $\mathcal{I}_{(\theta,\Delta)}(x_i) = 1 - c$
10: **return** $\mathcal{S}$

Remark 1. *RNCE is sound.*

Soundness of the procedure is guaranteed by construction, as solutions can only be chosen among the set of instances for which the classification label flips (see Algorithm 3). This is also true when RNCE is instantiated with `optimal` set to `T`, as an additional check is performed in line 10 of Algorithm 4 to ensure that only valid CEs are returned.

Remark 2. *RNCE is complete if there exists an* $(x', y') \in \mathcal{D}$ *such that* $\mathcal{I}_{(\theta,\Delta)}(x') \neq M_\theta(x)$.

Completeness of the procedure is conditioned on the existence of at least one Δ-robust instance x' in $\mathcal{D}$ whose label differs from that of the original

Algorithm 4 RNCE: `getRobustCE`

Require: input x, model M_θ, interval abstraction $\mathcal{I}_{(\theta,\Delta)}$,
1: k-d tree $\mathcal{T}$, parameter `optimal` $\in \{$`F, T`$\}$
2: **Init**: $c \leftarrow M_\theta(x)$; $x' \leftarrow \emptyset$; $a \leftarrow 1$; $s \leftarrow 0.05$
3: **Step 1**: find robust CE
4: $x'_{nn} \leftarrow \mathcal{T}$.`queryNextNeighbour`$(x)$
5: **while** x'_{nn} is not $\emptyset$ **do**
6: **if** $\mathcal{I}_{(\theta,\Delta)}(x'_{nn}) = 1 - c$ **then** $x' \leftarrow x'_{nn}$; **break**
7: $x'_{nn} \leftarrow \mathcal{T}$.`queryNextNeighbour`$(x)$
8: **if** `optimal` is `T` **then**
9: **while** $a > 0$ **do**
10: $x'_{line} \leftarrow ax' + (1-a)x$; $a \leftarrow a - s$
11: **if** $\mathcal{I}_{(\theta,\Delta)}(x'_{line}) = 1 - c$ **then** $x' \leftarrow x'_{line}$
12: **return** x'

input x. Completeness is not affected by the configurations of parameters in RNCE. When `robustInit` is `T`, Algorithm 3 is guaranteed to identify such input as a candidate CE and add it to set $\mathcal{S}$; when `robustInit` is `F`, the multiple queries of the next nearest neighbour and the following Δ-robustness test (Alg. 4, lines 4-6) will also identify a feasible result. For the parameter `optimal`, Algorithm 4 will always return one among x' or an optimised version of it (in terms of distance) for which robustness is still guaranteed (Alg. 4, 10). Our experimental analysis in Section 4.6 shows that our approach is always able to find Δ-robust CEs.

Remark 3. *RNCE is equivalent to a plain NNCE algorithm if* `optimal` *is* `F` *and* $\Delta = \emptyset$.

RNCE collapses to a plain NNCE algorithm if the set of plausible model shifts Δ is an empty set, thus $\mathcal{I}_{(\theta,\Delta)}$ will be equivalent to the original model, M_θ. Then, the choice of the parameter `robustInit` makes no impact as lines 8 and 9 of Alg. 3 become identical (the multiple queries specified in Alg. 4, 4-6 are also not needed). Note that the line search controlled by `optimal` can also be performed when $\Delta = \emptyset$.

Figure 4.5 shows a pictorial representation of the different behaviours that can be obtained from RNCE based on the configuration of the parameter `optimal`

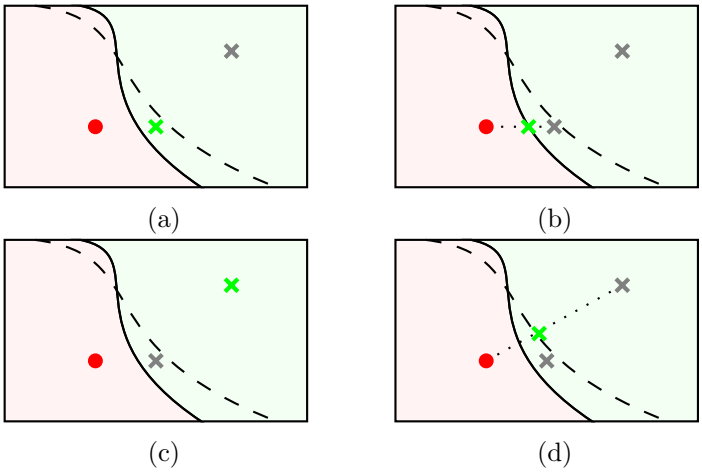


Figure 4.5: RNCE behaviours when `optimal` = `F`, $\Delta = \emptyset$ (4.5a); `optimal` = `T`, $\Delta = \emptyset$ (4.5b); `optimal` = `F`, $\Delta \neq \emptyset$ (4.5c); `optimal` = `T`, $\Delta \neq \emptyset$ (4.5d). Each figure depicts the CE that is chosen (green cross) among a set of candidate CEs (grey crosses) for a given input (red circle) and model M_θ. The continuous curved line represents the decision boundary of M_θ; the dashed line represents a possible change in the decision boundary under Δ. Figures 4.5a and 4.5b show configurations under which RNCE may return CEs that are not Δ-robust, whereas Figures 4.5c and 4.5d depict robust ones.

and whether Δ is empty. When $\Delta = \emptyset$ and `optimal` is `F` (Figure 4.5a), the algorithm behaves like a standard algorithm producing NNCEs and returns the closest counterfactual instance. However, this CE may not be robust. Analogous results may be obtained when $\Delta = \emptyset$ and `optimal` is `T` (Figure 4.5b). Conversely, when $\Delta \neq \emptyset$ (Figures 4.5c and 4.5d), RNCE will only operate on robust instances as candidates, thus guaranteeing the CEs' Δ-robustness.

4.6 Experiments

In this section, we demonstrate through experiments how Δ-robustness can be used in practice. We first described the experimental setup used for our experiments, and present two strategies to find realistic magnitudes in the plausible model changes given a dataset and the corresponding original

classifier. Then, using these magnitudes, we construct interval abstractions to test the robustness of existing CE generation methods, showing a lack of Δ-robustness in state-of-the-art methods. Finally, we benchmark the proposed algorithms to compute Δ-robust CEs with evaluation metrics for proximity, plausibility, and robustness, showing the effectiveness of our methods.

Additionally, we demonstrate the applicability of our methods to multi-class classification tasks, while most existing methods focus on obtaining robust CEs only for binary classifiers. Finally, we present experiments aimed at benchmarking the runtime performance of our procedures and compare them with existing approaches.

The code for our implementation and experiments is available at:

`https://github.com/junqi-jiang/interval-abstractions`.

4.6.1 Setup and evaluation metrics

dataset	**data points**	**attributes**	**NN accuracy**	**LR accuracy**
adult	48832	13	.847±.006	.828±.004
compas	6172	7	.844±.010	.833±.011
gmc	115527	10	.860±.001	.852±.002
heloc	9871	21	.728±.015	.725±.013

Table 4.1: Dataset and classifier details.

We experiment on four popular tabular datasets for benchmarking performances of CE generation algorithms, *adult* income dataset [38], *compas* recidivism dataset [4], give me some credit (*gmc*) dataset [28], Home equity line of credit (*heloc*) dataset [46], from the CARLA library [110]. All datasets contain min-max scaled continuous features, one-hot encoded binary discrete features, and two output classes. Class 0 is the unwanted class while class 1 is the target class for CEs. We train neural networks with two hidden layers and 6 to 20 neurons in each layer, and logistic regression models (used in Section 4.6.4) on the datasets. Table 4.1 reports the dataset sizes, the number of attributes of the datasets, and the 5-fold cross-validation accuracy of the classifiers obtained.

We randomly split each dataset $\mathcal{D}$ into two halves, $\mathcal{D}_1$ and $\mathcal{D}_2$, each including a training and a test set. We use $\mathcal{D}_1$ to train the classifiers as stated above, and we call these original models M_1. $\mathcal{D}_2$ is used to simulate scenarios with incoming data after M_1 are deployed, explained in the next sections.

4.6.2 Identifying δ values

Recall from Definition 3 that δ upper-bounds the magnitude of the model shifts in Δ, as measured by the p-distance (Definition 1). When practically using Δ-robustness, δ, values, regarded as the hyperparameter in our method, need to be first determined. We now propose two realistic strategies for obtaining δ values, depending on how the underlying classifier is retrained with new data in the application.

Incremental retraining This refers to the case where M_1 is periodically fine-tuned by gradient descent on some portions of $\mathcal{D}_2$ with a small number of iterations. In this scenario, depending on how many data points in $\mathcal{D}_2$ are used for retraining, the magnitude of parameter changes could be small. Therefore, the δ values can be directly estimated by calculating the p-distance between M_1 and the updated classifiers. By observing the p-distances when retraining on various portions of $\mathcal{D}_2$, the model developers could potentially link the estimated δ values to time intervals in real-world applications, depending on the frequency at which new data are collected.

Full or leave-one-out retraining *Full retraining* refers to a scenario where a model is retrained with the same hyperparameter setting as M_1, using the concatenation of $\mathcal{D}_1$ and $\mathcal{D}_2$. *Leave-one-out retraining* concerns obtaining a new model using a subset of $\mathcal{D}_1$ with 1% of data points removed. As mentioned in [61], when retraining from scratch using the concatenation of $\mathcal{D}_1$ and $\mathcal{D}_2$, it becomes unrealistic to upper-bound the weights and biases differences in classifiers as the p-distance can be arbitrarily large. In this case, we can treat δ as a hyperparameter and empirically find the minimum value which can ensure robustness on a validation set. Similarly to the standard train-validation-test split for evaluating the accuracy of ML models,

we propose to use a held-out validation set to estimate δ values which lead to sufficient robustness under such retraining scenarios. The procedure can be summarised as follows:

1. Retrain from scratch some new classifiers using $\mathcal{D}_1$ and $\mathcal{D}_2$[3], set initial δ value to a sufficiently small value;
2. Generate Δ-robust CEs using the current δ;
3. Evaluate the percentage of the explanations which are valid under all the retrained models;
4. Examine the above empirical validity, if it has not reached 100% then increase the δ value and repeat steps 2-3. Choose the smallest δ value which results in 100% validity.

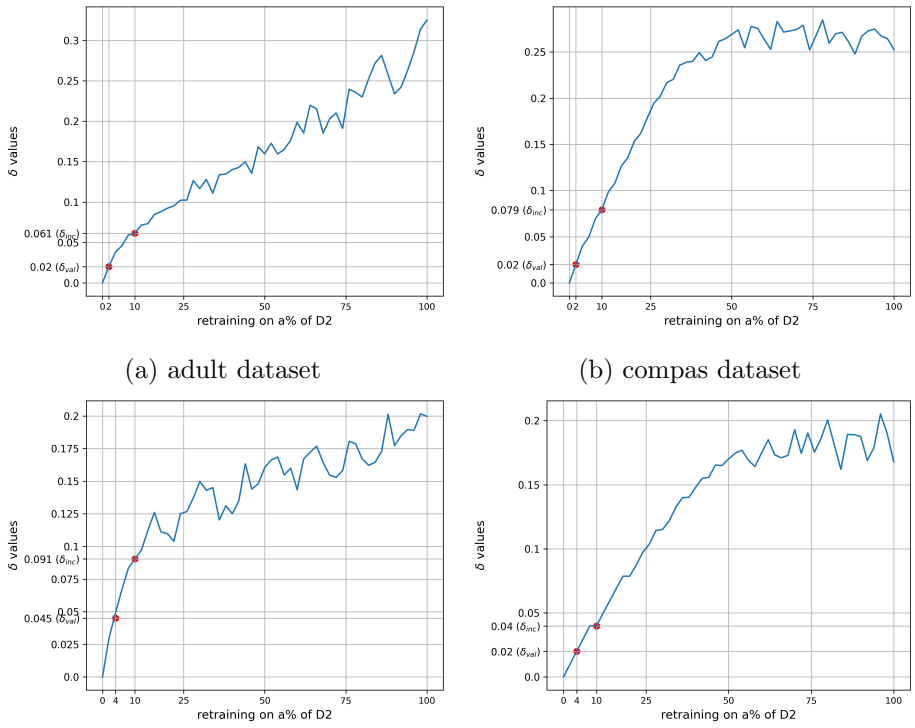
The termination condition in step 4 balances the robustness-cost tradeoff. Once the empirical validity on multiple retrained models reaches 100% for the validation set, increasing δ further will negatively affect the proximity evaluations. It is also expected that when finding Δ-robust CEs with the same δ value for the test set, a similar level of robustness can be observed.

We report the δ value results using both strategies in Figure 4.6, referred to as δ_{inc} (incremental retraining) and δ_{val} (validation set). Specifically, for the first scenario, we record and plot δ values as the average ∞-distances (in the experiments, we use $p = \infty$) between M_1 and five incrementally retrained[4] models using increasing sizes of the retraining dataset (a% of $\mathcal{D}_2$). As can be observed, δ values increase with slight fluctuations as incrementally retraining on more data. The magnitudes are classifier- and dataset-dependent, though in our setting we obtain values from 0 to about 0.3. The δ_{inc} values are obtained by recording the δ values when retraining on 10% of $\mathcal{D}_2$, and use them as one robustness target in the next experiments.

[3]Full retraining is possible in an experimental environment. In practice, however, if $\mathcal{D}_2$ is not available at the time of robust CE generation, leave-one-out retraining on $\mathcal{D}_1$ could be a viable option in step 1.

[4]We re-implemented the partial_fit function in Scikit-learn library: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier.partial_fit

For the second scenario, we use five fully retrained and five leave-one-out retrained classifiers as the new models in Step 1. We use RNCE-FF to find Δ-robust CEs in Step 2. We record the obtained δ_{val} as another robustness target, and we highlight these values in the same plots in Figure 4.6. After matching their magnitudes to the ones obtained for incremental retraining, we identify that these δ_{val} are much smaller than δ_{inc} and they correspond to the magnitudes of retraining on only 2% of $\mathcal{D}_2$. The fact that being Δ-robust against a very small magnitude results in 100% empirical robustness (validity) in a validation set confirms the conservative nature of Δ-robustness, as it guarantees the CE's robustness against all possible model shifts entailed by Δ (Lemma 2).



(a) adult dataset

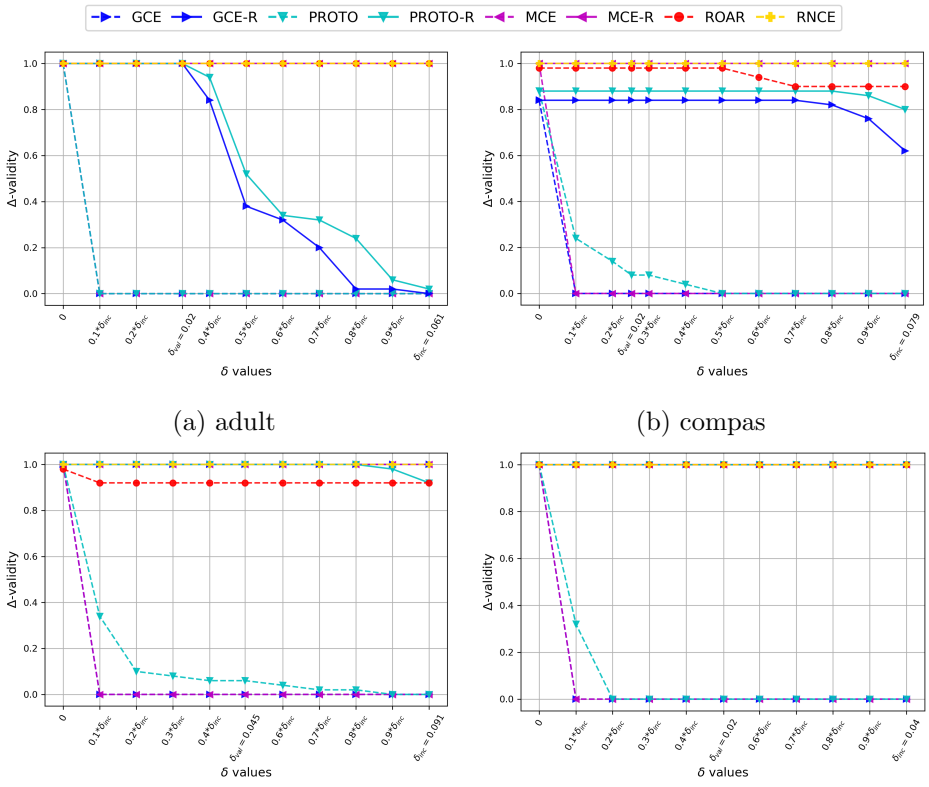
(b) compas dataset

(c) gmc dataset

(d) heloc dataset

Figure 4.6: δ values obtained by retraining on increasing portions of $\mathcal{D}_2$. The red dots highlight the δ_{val} and δ_{inc} values found by the two strategies described in Section 4.6.2.

4.6.3 Verifying Δ-robustness



(a) adult (b) compas

(c) gmc (d) heloc

Figure 4.7: Evaluation of Δ-validity (against increasing δ values) of the CEs found by state-of-the-art methods (GCE, PROTO, MCE, ROAR) and by our methods (GCE-R, PROTO-R, MCE-R, RNCE).

In this experiment, we demonstrate how our MILP procedure can be used as an evaluation tool to examine the robustness of CEs generated by SOTA algorithms. We consider three traditional non-robust baselines, namely a gradient-descent based method *GCE* similar to [142], a plausible method using gradient descent *Proto* [140], and a MILP-based method (referred to as MCE) inspired by [98]. We also include *ROAR* [138], a SOTA framework specifically designed to generate robust CEs, focusing on robustness against the same notion of plausible model changes[5]. For our algorithms,

[5]We use the implementation of ROAR in CARLA library [110] which allows comprehensive hyperparameter tuning.

using the iterative algorithm (Algorithm 1) proposed in Section 4.5, we devise robustified versions of the non-robust baselines, which we call *GCE-R*, *PROTO-R*, and *MCE-R*. We also include our RNCE-`FF` algorithm (Algorithm 2, `robustInit=False`, `optimal=False`) to show the guaranteed robustness results of this method.

For each dataset, we randomly select 50 test inputs for which we use the above baselines to generate CEs. We evaluate their robustness against model shifts of magnitudes up to δ_{inc} using Δ-*validity*, the percentage of test inputs whose CEs are Δ-robust. For all robust methods, their targeted Δ values are instantiated with δ_{inc}.

Figures 4.7 (a-d) report the results of our analysis for the four datasets. As we can observe, all methods generate CEs that tend to be valid for the original model. For the non-robust baselines, Δ-validities soon drop to value 0 as small model shifts are applied, revealing a lack of robustness for these baselines. ROAR exhibits a higher degree of Δ-robustness, as expected. However, its heuristic nature does not allow to reason about all possible shifts in Δ, which affects the Δ-robustness of CFXs as δ grows larger. Also, the fact that it uses a local surrogate model to approximate the behaviour of neural networks could negatively affect the results. For compas and gmc datasets, its Δ-robustness stays lower than 100%.

For the two gradient-based non-robust baselines, our robustified versions (GCE-R, PROTO-R) successfully improved their robustness against small model shifts with lower δ values, however, the Δ-robustness tends to drop (drastically for adult dataset) when it increases near δ_{inc}. This is likely due to the vulnerability to local optimum solutions for the gradient-descent algorithms. The MILP-based method MCE-R which gives more exact solutions for the problem always finds robust CEs with 100% Δ-validity. With proven robustness guarantees, our RNCE algorithm also finds CEs that are 100% robust.

Method	Properties				Classifiers	
	Validity	Proximity	Plausibility	Robustness	NN	LR
GCE [142]	✓	✓			✓	
PROTO [140]	✓	✓	✓		✓	
MCE [98]	✓	✓			✓	
NNCE [21]	✓	✓	✓		✓	✓
RBR [104]	✓	✓	✓	✓	✓	
ROAR [138]	✓	✓		✓	✓	✓
RobXCE [61]	✓	✓	✓	✓	✓	
GCE-R (ours)	✓	✓		✓	✓	
PROTO-R (ours)	✓	✓	✓	✓	✓	
MCE-R (ours)	✓	✓		✓	✓	
RNCE (ours)	✓	✓	✓	✓	✓	✓

Table 4.2: Properties (validity, proximity, plausibility, robustness) addressed by baselines and whether the methods apply to neural network (NN) or logistic regression (LR), indicated by ✓.

4.6.4 Generating Δ-robust CEs

Next, we rigorously benchmark the performance of our CE generation methods against various robust and non-robust baselines. Apart from the CE methods used in Section 4.6.3, we additionally include NNCE [21], *RBR* [104], and *RobXCE* [61]. [6] We also instantiate RNCE-`FT` as one of our methods. The properties of all compared methods are summarised in Table 4.2.

For each dataset, we randomly select 20 test points from the test set to generate CEs using each method. We repeat the process five times with different random seeds and report the mean and standard deviation of the results. The CEs are evaluated against three aspects using the standard metrics in the literature. For proximity, we calculate the average ℓ_1 *distance* between the test input and its CE, which captures both closeness of CEs and sparsity of changes [142]. For plausibility, we report the average local outlier factor score *lof* [20] which quantifies the local data density. A lof score close to value 1 indicates an inlier; the more it deviates from 1, the less plausible the CE. For robustness, we report validity after retraining *vr*, i.e., the percentage of CEs correctly classified to class 1, under 15 retrained

[6] We only included methods which either have open-source implementations or could be re-implemented without requiring excessive effort.

	vr↑	**v**Δ_{val}↑	**v**Δ_{inc}↑	ℓ_1↓	**lof**↓	**vr**↑	**v**Δ_{val}↑	**v**Δ_{inc}↑	ℓ_1↓	**lof**↓
	adult					**compas**				
GCE	51%	0%	0%	.016	1.29	26.5%	0%	0%	.039	3.05
PROTO	61.1%	1%	0%	.011	1.44	50.7%	6.6%	0%	.144	1.66
MCE	48.5%	0%	0%	.009	1.41	25.6%	0%	0%	.019	1.79
NNCE	76.1%	2%	2%	.032	1.34	43.3%	8%	0%	.028	1.30
ROAR	100%	100%	94.8%	.877	12.5	100%	100%	94.7%	.388	8.44
RBR	90.1%	0%	0%	.025	1.33	98.3%	38%	0%	.038	1.53
RobXCE	98.7%	20%	4%	.046	1.27	99.9%	74%	0%	.039	1.23
	target $\delta_{val} = 0.02$					target $\delta_{val} = 0.02$				
GCE-R	100%	100%	0%	.048	1.47	86.9%	87%	0%	.055	3.04
PROTO-R	100%	69%	0%	.042	1.68	91%	91%	0%	.049	1.74
MCE-R	100%	100%	0%	.021	1.65	99.8%	100%	0%	.035	1.68
RNCE-FF	100%	100%	4%	.057	1.32	100%	100%	0%	.039	1.26
RNCE-FT	100%	100%	0%	.049	1.28	100%	100%	0%	.037	1.33
	target $\delta_{inc} = 0.061$					target $\delta_{inc} = 0.079$				
GCE-R	100%	100%	0%	.051	1.65	87%	87%	67%	.109	3.62
PROTO-R	100%	100%	9%	.072	2.36	91%	91%	84%	.108	1.99
MCE-R	100%	100%	100%	.051	2.91	100%	100%	100%	.096	2.81
RNCE-FF	100%	100%	100%	.122	2.78	100%	100%	100%	.088	1.11
RNCE-FT	100%	100%	100%	.095	2.70	100%	100%	100%	.088	1.11
	gmc					**heloc**				
GCE	75.8%	0%	0%	.022	1.52	20.5%	0%	0%	.019	1.18
PROTO	89.1%	1%	0%	.023	1.36	39.5%	0%	0%	.024	1.16
MCE	63.3%	0%	0%	.016	1.40	22%	0%	0%	.014	1.40
NNCE	88.9%	22%	1%	.029	1.23	35.9%	0%	0%	.053	1.05
ROAR	99.3%	98%	98%	.199	23.1	100%	100%	100%	.454	6.94
RBR	100%	62%	0%	.034	1.55	58.7%	0%	0%	.038	1.08
RobXCE	100%	92%	6%	.041	1.10	100%	40%	0%	.078	1.04
	target $\delta_{val} = 0.02$					target $\delta_{val} = 0.02$				
GCE-R	100%	100%	0%	.032	1.79	100%	100%	0%	.049	1.32
PROTO-R	100%	100%	0%	.036	1.45	100%	100%	11%	.079	1.62
MCE-R	100%	100%	0%	.022	1.48	100%	100%	0%	.031	1.94
RNCE-FF	100%	100%	7%	.040	1.07	100%	100%	0%	.083	1.04
RNCE-FT	100%	100%	0%	.035	1.36	100%	100%	0%	.080	1.04
	target $\delta_{inc} = 0.091$					target $\delta_{inc} = 0.04$				
GCE-R	100%	100%	100%	.053	3.43	100%	100%	100%	.109	2.07
PROTO-R	100%	100%	100%	.118	2.49	100%	100%	100%	.163	2.41
MCE-R	100%	100%	100%	.032	1.86	100%	100%	100%	.049	3.04
RNCE-FF	100%	100%	100%	.084	1.22	100%	100%	100%	.150	1.13
RNCE-FT	100%	100%	100%	.084	1.22	100%	100%	100%	.150	1.13

Table 4.3: Quantitative evaluation of the compared CE generation methods on neural networks. The ↑ (↓) following each metric indicates the higher (lower) the value, the better. Methods are separated by horizontal lines, indicating non-robust baselines, robust baselines, and our methods with robustness target Δ instantiated by δ_{val} and δ_{inc}, respectively.

classifiers using respectively full retraining, leave-one-out retraining, and incremental retraining (with 10% new data). We use the same δ_{val} and δ_{inc} obtained from Section 4.6.2 to instantiate Δ with different model shift

magnitudes and report the respective Δ-robustness, termed $v\Delta_{val}$ and $v\Delta_{inc}$.

Table 4.3 reports the mean results for neural network classifiers of the benchmarking study. See Appendix A.2 for the standard deviation results and the evaluations for logistic regression classifiers. Next, we analyse the results by their properties, and for our methods, we first consider the results when targeting δ_{val}.

Our methods produce the most robust CEs Our RNCE algorithm (both configurations) generates the most robust CEs among the compared methods, showing 100% vr and 100% targeted Δ-validity. For the robustified methods using Algorithm 2, MCE-R is the most robust, finding perfectly Δ-robust and 100% empirically robust CEs in most experiments. As discussed in Section 4.6.3, the limited search space might be the cause of the reduced robustness for GCE-R and PROTO-R in two datasets. As a result, when compared with the robust baselines, RNCE and MCE-R both give better robustness than ROAR, RobXCE, and RBR. All robust methods have better robustness performances than the non-robust baselines, as expected.

Cost-robustness tradeoff This tradeoff has been discussed in several other studies [138, 112], which we have also empirically observed. The non-robust baselines always find the most proximal CEs (lowest $\ell 1$ costs). Apart from them, the next best proximity results were obtained by MCE-R (when targeting the smaller δ_{val}) among all the robust methods. Considering that MCE-R also finds near-perfectly Δ-robust CEs, it can be concluded that MCE-R effectively balances the cost-robustness tradeoff. RBR also finds CEs with low costs, but their method is not as robust. Similar remarks can be made for RobXCE as this method moves more towards the robustness end of the tradeoff. Our methods GCE-R, PROTO-R, RNCE demonstrate similar $\ell 1$ costs. Setting `optimal=True` in our RNCE algorithm slightly improves the proximity, as can be seen when comparing RNCE-`FF` and RNCE-`FT`. ROAR results in CEs with high $\ell 1$ cost, and the method has been identified as being overly costly [103] due to their gradient-based robust optimisation procedure.

Plausibility results RobXCE has the best lof scores among all the methods, followed by RNCE which yields comparable results. This is because these two methods select CEs from in-manifold dataset points and are thus unlikely to return outliers. By inherently addressing data density estimation, RBR also finds plausible CEs. The plausibility performances of our three robustified methods and ROAR are not as good due to less regulated search spaces or lack of plausibility constraints, with ROAR having the worst lof results.

The effects of robustification For Algorithm 1, robustifying GCE, PROTO, and MCE resulted in improved empirical and $\Delta-$robustness, but this negatively affected the proximity and plausibility results. Targeting larger-magnitude plausible model shifts (instantiated with δ_{inc}) pushes these tradeoffs further. For RNCE, similar trends can be identified when compared with NNCE, but in most cases, the lof score improves. When targeting δ_{inc} instead of δ_{val}, the cost also increases together with robustness, but plausibility stays comparable.

Concluding remarks From the analysis above, we can conclude that MCE-R achieved the best robustness-cost tradeoff, finding the most proximal CEs among the robust baselines while showing near-perfect robustness results, outperforming all robust baselines. RNCE, on the other hand, finds CEs with even stronger robustness guarantees and great plausibility, at slightly higher costs. However, less costly methods than RNCE are not as robust.

4.6.5 Computation time analysis

In this section, we discuss the computation time required by each method to obtain the results in Section 4.6.4. The average computation times for generating CEs for 20 test points are presented in Table 4.4.

While model complexity impacts all methods, the runtimes of gradient- and MILP-based optimisation methods that are not wrapped in an iterative approach (GCE, PROTO, MCE, RBR) are mostly sensitive to the number

of attributes in the dataset, and those iteratively computed for robustness (ROAR, GCE-R, PROTO-R, MCE-R) are also easily affected by their hyperparameters concerning robustness. The methods that require traversing (part of) the training dataset (NNCE, RNCE, RobXCE) are sensitive to the number of data points in each dataset.

Model	**Method**	**adult**	**compas**	**gmc**	**heloc**
Logistic Regression	NNCE	0.039	0.024	0.636	0.003
	ROAR	11.70	11.49	4.958	13.71
	RNCE-FF δ_{val}	0.410	0.195	0.755	27.71
	RNCE-FF δ_{val}	1.554	1.035	1.722	29.75
	RNCE-FF δ_{inc}	0.627	0.948	0.754	1.534
	RNCE-FF δ_{inc}	1.779	2.061	1.681	3.522
Neural Network	GCE	27.23	29.06	25.15	25.08
	PROTO	604.3	555.8	542.2	607.6
	MCE	0.934	0.340	0.163	1.221
	NNCE	0.095	0.018	0.514	0.039
	ROAR	20.03	2.535	1.740	2.007
	RBR	187.2	118.4	112.2	110.5
	RobXCE	0.579	1.29	0.347	1.419
	GCE-R δ_{val}	101.7	56.97	86.32	67.33
	PROTO-R δ_{val}	1602	756.9	690.5	965.2
	MCE-R δ_{val}	18.77	2.364	1.281	13.05
	RNCE-FF δ_{val}	15.64	4.125	3.447	32.76
	RNCE-FT δ_{val}	27.96	6.744	5.307	43.50
	GCE-R δ_{inc}	155.2	117.08	116.01	120.9
	PROTO-R δ_{inc}	1920	1824	1625	1469
	MCE-R δ_{inc}	37.45	9.152	2.771	19.43
	RNCE-FF δ_{inc}	218.3	48.07	163.2	670.6
	RNCE-FT δ_{inc}	229.9	51.51	165.5	681.6

Table 4.4: Computation time (in seconds) of the compared methods on logistic regression models and neural networks. The adult dataset has 48832 data points with 13 attributes, the compas dataset has 6172 data points with 7 attributes, the gmc dataset has 115527 data points with 10 attributes, the heloc dataset has 9871 data points and 21 attributes.

MCE-R is the fastest among the methods we devised, showing better runtime than ROAR and RBR when targeting a smaller validation δ, and producing close CEs with robustness guarantees. On the other hand, our PROTO-R method is the slowest, while it also fails to give the best CEs in terms of the desired properties (Table 4.3). Similar remarks can be made for GCE-R,

which has comparable runtime as RBR. These three robustified methods (GCE-R, PROTO-R, MCE-R) largely inherit the runtime of their respective base method (GCE, PROTO, MCE). For RNCE, the number of attributes and model complexity largely impact the time required to compute the solution for a single MILP program, while the number of dataset points and targeted δ value jointly impact the number of MILP programs that need to be checked. Therefore, the RNCE runtime varies to a large extent in our experiments. On logistic regression classifiers, our approach is faster than the other robust baseline, ROAR. On neural networks, for a smaller validation δ, the runtimes are in the same order of magnitude as ROAR and are much faster than RBR, but they deteriorate when switching to a larger δ_{inc}.

We conclude that our best performing methods, MCE-R and RNCE, when targeting the δ values found by the validation set method, are able to find better-quality and more robust CEs than the robust baselines while demonstrating comparable runtimes. When trying to achieve robustness guarantees for higher δ values, worse runtimes can be observed. This result is expected, given that proving robustness with respect to model changes has been shown to be an NP-complete problem [94].

4.6.6 Multi-Class Classification

Next, we demonstrate the applicability of our RNCE method to multi-class classification settings, which is not supported by any robust CE generation baselines listed in Table 4.2. We use the Iris dataset [3, 47], a small-scale dataset for three-class classification tasks, and the California Housing dataset [106], in which we transform the regression labels into three classes. Both datasets are available in sklearn [115]. We trained two neural network models with cross validation accuracy of 0.93 and 0.70, respectively. Again, we randomly select (five times with different random seeds) 20 test instances from the test sets which are classified to class 0, and we specify a desired label 2 for CEs. Following the same experimental protocol described in Section 4.6.4, we use the validation set strategy to identify δ values as our targeted plausible model changes magnitude, and we report the same

evaluation metrics for NNCE and RNCE.

Dataset	Method	vr↑	vΔ_{val}↑	ℓ_1↓	lof↓
iris $\delta_{val} = 0.015$	NNCE	75%±.004	0%±.0	.393±.002	1.48±.026
	RNCE-FF	100%±.0	100%±.0	.438±.003	1.50±.002
	RNCE-FT	100%±.0	100%±.0	.438±.003	1.50±.002
housing $\delta_{val} = 0.040$	NNCE	18.1%±.035	0%±.0	.032±.003	1.10±.020
	RNCE-FF	100%±.0	100%±.0	.053±.004	1.33±.037
	RNCE-FT	100%±.0	100%±.0	.052±.004	1.31±.035

Table 4.5: Quantitative evaluation of the compared methods on neural networks in multi-class classification tasks. The evaluation metrics are the same as in Table 4.3.

Table 4.5 presents the results we obtained. Similarly to the results for binary classification, the CEs computed by the NNCE method are not robust against realistically retrained models (indicated by the low **vr**). Measured by the Δ-validity, they also fail to satisfy the Δ-robustness tests for multi-class classifications. With a slight tradeoff with ℓ_1 costs and lof scores, our two configurations of RNCE are both able to find perfectly robust CEs in this study. This demonstrates that the Δ-robustness notion can be used to evaluate provable robustness guarantees for CEs, and our proposed algorithms succeed at finding more robust CEs in the multi-class classification setting.

4.7 Summary, Limitations and Discussion

Summary. Despite the recent advances in achieving robustness against model changes for CEs, state-of-the-art lack rigorous robustness guarantees on the CEs they produce. By introducing a formal robustness notion, Δ-robustness, and a novel interval abstraction technique, we provided the first method in the literature to obtain CEs with certified robustness guarantees. We showed how such Δ-robustness can be practically tested in binary and multi-class classification settings by solving optimisation problems via MILP. Furthermore, we proposed two algorithms to generate CEs that are provably Δ-robust. To demonstrate how our methods can be used in practice, two strategies for identifying the appropriate hyperparameters in our method

have been investigated, linking to three model retraining strategies. We then presented an extensive empirical evaluation involving eleven algorithms for generating CEs, including seven algorithms specifically designed to generate robust explanations. Our results show that our MCE-R algorithm finds CEs with the lowest costs along with the best robustness results, and our RNCE outperforms existing approaches and is able to generate CEs that are both provably robust and plausible, achieving a balance in the robustness-cost and plausibility-cost tradeoffs. We also compared the runtimes and observed that our best-performing methods have comparable runtimes with the robust baselines when targeting small δ values. We see these outcomes as important contributions towards complementing existing formal approaches for XAI and making them applicable in practice.

Limitations. We identified a number of limitations of our approach. Firstly, white-box access to both the training dataset and the classifiers is required. As a result, our approach is primarily aimed at expert users, e.g., model developers. Secondly, our method only works for parametric ML classifiers whose forward pass can be precisely represented by a MILP program. We have considered two examples in this chapter, but other models could be considered. Thirdly, the computation time for our method, when targeting high δ values on complex models and large datasets, can be high as that for providing exact robustness guarantees for CEs under model changes, which is an NP-complete problem [94]. This is partly due to the fact that methods proposed in this chapter treat robustness verification and robust CE generation as two separate steps. We address this limitation in Chapter 5, where the two steps are integrated into a single bi-level optimisation procedure that is (conditionally) sound and complete and is guaranteed to converge. Nevertheless, in this work we demonstrated that our method leads to better-quality CEs than existing methods.

Discussion. Next, in Chapter 5, we will further propose a principled approach to generate plausible and provably Δ-robust CEs. We delegate the discussions on future works in Section 5.5.

Chapter 5

Provably Robust and Plausible Counterfactual Explanations via Robust Optimisation

This chapter aims to improve upon the key limitations of the CE generation methods proposed in Chapter 4. The previous methods generate provably robust CEs via iteratively tuning the hyperparameters of an arbitrary non-robust CEs method and testing for robustness. As a result, they cannot always guarantee soundness and might not be complete, which is also the case for the popular method in [138]. The separation of certifying robustness for a candidate CE and obtaining the actual robust CE as two distinct steps also limits the scalability of those approaches. Also, among all the existing approaches, only RNCE simultaneously addresses plausibility and provable robustness against model changes. However, it relies on retrieving real dataset points as CEs, risking exposing existing data, thus limiting its practical applicability.

We propose a new method in this chapter, which formulates a single robust optimisation problem for computing provably robust (against model changes) and plausible CEs. After a brief introduction in Section 5.1, we discuss the problem formulation in Section 5.2. Then, Section 5.3 introduces the main method, which is then tested in Section 5.4. Section 5.5 concludes this

chapter.

5.1 Motivation

In this chapter, we introduce the robust optimisation problem for computing CEs with proximity property as the objective, and robustness and plausibility properties as constraints. We will present Provably RObust and PLAusible CEs (PROPLACE), a method leveraging robust optimisation techniques to address the abovementioned limitations in the literature. We show the (conditional) soundness and completeness of our method, and give a bi-level optimisation procedure that will converge and terminate. Finally, in our experiments, we compare PROPLACE with six existing CE methods, five of which target robustness, on four benchmark datasets. The results show that our method achieves the best robustness and plausibility, while demonstrating superior proximity among the most robust baselines.

5.2 Problem Statement

The minimum distance objective targets the minimum effort by the end user to achieve a change, which corresponds to the basic requirement of proximity. In the literature, normalised L_1 distance is often adopted as the distance metric because it induces changes in fewer features in the CE [142]. However, methods that find such plain CEs usually result in unrealistic combinations of features, or outliers to the underlying data distribution of the training dataset. A plausible CE avoids these issues and is formally defined as follows:

Definition 15. (Plausible CE) *Given a neural network (NN) M_θ and an input $x \in \mathcal{X}$ such that $M_\theta(x) < 0$[1], a distance metric dist $: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$*

[1]In this Chapter, $M_\theta(x)$ also (with an abuse of notation) refer to the pre-sigmoid (logit) value in the NN.

and some plausible region $\mathcal{X}_{plaus} \subseteq \mathbb{R}^d$*, a* plausible CE *is an* x' *such that:*

$$\begin{aligned} &\arg\min_{x'} \quad dist(x, x') \\ &\textit{subject to} \quad M_\theta(x') \geq 0, \quad x' \in \mathcal{X}_{plaus} \end{aligned}$$

The plausible region $\mathcal{X}_{plaus}$ may be used to eliminate any unrealistic feature values (e.g., a value of 0.95 for a discrete feature), or to indicate a densely populated region that is close to the data manifold of the training dataset. Additionally, it may also include some actionability considerations, such as restricting immutable attributes (e.g., avoiding suggesting changes in gender) or specifying some relations between input features (e.g., obtaining a doctoral degree should also cost the user at least 4 years).

Robustness of Counterfactual Explanations. Studies have shown that CEs found by the above formulations are readily invalidated when small changes occur in the model parameters of the NNs. To reason about this formally, we rely on the interval-abstraction machinery developed in Chapter 4, which we recall here rather than restate: the *p-distance* between two models (Definition 1), a *model shift* S and the *set of plausible model shifts* Δ (Definitions 2 and 3), the *interval abstraction* $\mathcal{I}_{(\theta,\Delta)}$ of a model under Δ together with its *classification semantics* (Definitions 4 and 5), and the notion of a Δ*-robust CE* (Definition 7). We instantiate these for NNs and assume $p = \infty$, so that $\mathcal{I}_{(\theta,\Delta)}$ exactly captures the output range of every shifted model in Δ [64]. Throughout this chapter $M_\theta(x)$ denotes the pre-sigmoid (logit) output, so the classification threshold of 0.5 in Definition 5 becomes 0: $\mathcal{I}_{(\theta,\Delta)}$ classifies x as class 1 iff $l \geq 0$ (abbreviated $\mathcal{I}_{(\theta,\Delta)}(x) \geq 0$), and a CE x' is Δ*-robust* iff $\mathcal{I}_{(\theta,\Delta)}(x') \geq 0$, i.e., it stays valid under every shifted model in Δ. With a slight abuse of notation we also write $M_{\theta'} \in \Delta$ for a shifted model $M_{\theta'} = S(M_\theta)$ obtained from some $S \in \Delta$.

Checking whether a CE x' is Δ-robust requires the calculation of the lower bound l of the pre-sigmoid output node interval $[l, u]$ of $\mathcal{I}_{(\theta,\Delta)}$. This process can be encoded as a MILP program (see Chapter 4).

Optimising for Robustness and Plausibility. Now we introduce the targeted provably robust and plausible optimisation problem based on Definitions 15 and 7, by taking inspiration from the robust optimisation technique [13].

Definition 16. (Provably robust and plausible CE) *Given a NN M_θ, an input $x \in \mathcal{X}$ such that $M_\theta(x) < 0$, a distance metric $dist : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ and some plausible region $\mathcal{X}_{plaus} \subseteq \mathbb{R}^d$, let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ under the bounded model shifts Δ. Then, a* provably robust and plausible CE *$x' \in \mathcal{X}$ is such that:*

$$\arg\min_{x'} \quad dist(x, x') \tag{5.1a}$$

$$\textit{subject to} \quad \mathcal{I}_{(\theta,\Delta)}(x') \geq 0, \tag{5.1b}$$

$$x' \in \mathcal{X}_{plaus} \tag{5.1c}$$

The optimisation problem (5.1) can be equivalently rewritten as follows:

$$\arg\min_{x'} \quad dist(x, x') \tag{5.2a}$$

$$\text{subject to} \quad \max_{M_{\theta'} \in \Delta} [-M_{\theta'}(x')] \leq 0, \tag{5.2b}$$

$$x' \in \mathcal{X}_{plaus} \tag{5.2c}$$

We show next a novel approach for solving this robust optimisation problem (5.2).

5.3 PROPLACE

The procedure for computing robust and plausible CEs, solving the optimisation problem (5.2) , is summarised in Algorithm 5. We will first introduce how the plausible region $\mathcal{X}_{plaus}$ is constructed in Section 5.3.1 (corresponding to Line 3, Algorithm 5). Then, in Section 5.3.2 we will present the bi-level

optimisation method (corresponding to Lines 4-5, Algorithm 5) to solve the robust optimisation problem (5.2). In Section 5.3.2 we will also instantiate the full bi-level optimisation formulations (in MILP form) of our method for NNs with ReLU activation functions. Finally, in Section 5.3.3 we discuss the soundness and completeness of Algorithm 5 and prove its convergence.

5.3.1 Identifying Search Space $\mathcal{X}_{plaus}$

As mentioned in Section 5.1, points from the training dataset (especially k-nearest-neighbours) are frequently utilised in the literature to induce plausibility. In this work, we propose to use a more specialised kind of dataset point, k Δ-*robust nearest-neighbours*, to construct the search space for CEs that is both plausible and robust.

Definition 17. (k Δ-robust nearest-neighbours) *Given a NN M_θ and an input $x \in \mathcal{X}$ such that $M_\theta(x) < 0$, a distance metric dist : $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$, a dataset $\mathcal{D} \subseteq \mathbb{R}^{d2}$ on which M_θ is trained, and a set of bounded model shifts of interest Δ, let $\mathcal{I}_{(\theta,\Delta)}$ be the interval abstraction of M_θ under Δ. Then, the* k Δ-robust nearest-neighbours *of x is a set $S_{k,\Delta} \subseteq \mathcal{D}$ with cardinality $|S_{k,\Delta}| = k$ such that:*

- $\forall x' \in S_{k,\Delta}$, *$x'$ is Δ-robust, i.e., $\mathcal{I}_{(\theta,\Delta)}(x') \geq 0$,*

[2]In this chapter, a dataset refers to only the collection of input points without their corresponding labels, for simplicity.

Algorithm 5 PROPLACE

Require: input x, model M_θ,
1: training dataset $\mathcal{D}$,
2: set of bounded model shifts Δ,
3: plausible region to be used as CEs search space $\mathcal{X}_{plaus}$.
4: **Init**: $x' \leftarrow \emptyset$; $\Delta' \leftarrow \{M_\theta\}$
5: **Repeat**
6: $x' \leftarrow$ `Outer_minimisation`$(M_\theta, x, \mathcal{X}_{plaus}, \Delta')$
7: $M_{\theta'} \leftarrow$ `Inner_maximisation`(x', Δ', Δ)
8: $\Delta' \leftarrow \Delta' \cup \{M_{\theta'}\}$
9: **Until** $(-M_{\theta'}(x')) \leq 0$
10: **return** x'

- $\forall x'' \in \mathcal{D} \setminus S_{k,\Delta}$, *if* x'' *is* Δ*-robust,* $dist(x, x'') \geq \max_{x' \in S_{k,\Delta}} dist(x, x')$.

The first constraint enforces the Δ-robustness, and the second states that the points contained in the set are the k nearest points to the input x amongst all the Δ-robust dataset points. In practice, in order to compute the k Δ-robust nearest-neighbours, we fit a k-d tree on the dataset points that are classified to the desired class, then iteratively query the k-d tree for the nearest neighbour of an input, until the result satisfies the Δ-robustness test (Definition 7). This definition is very similar to the CEs found by the RNCE approach in Chapter 4, with an extension of obtaining multiple Δ-robust points.

Restricting the CE search space within the convex hull of these robust neighbours will likely induce high plausibility (and robustness). However, because these points are deep within parts of the training dataset that are classified to another class, they may be far from the model's decision boundary, therefore resulting in large distances to the inputs. In fact, [40, 61] adopted similar robust nearest neighbours (using other notions of robustness tests) as the final CEs, and poor proximity was observed in their experiment results. They have also shown that finding CEs using line search between proximal CEs and these robust neighbours can slightly improve proximity.

In our case, since the validity of the CEs can be guaranteed from the optimisation procedures (Section 5.3.2), we expand the plausible search space across the decision boundary by taking the input into consideration, which is assumed to also be inside the data distribution.

Definition 18. (Plausible region) *Given an input* $x \in \mathbb{R}^d$ *and its* k Δ*-robust nearest neighbours* $S_{k,\Delta}$*, the* plausible region $\mathcal{X}_{plaus}$ *is the convex hull of* $S_{k,\Delta} \cup \{x\}$.

By restricting the CE search space to such convex hull, the method has the flexibility to find close CEs (with x as a vertex), or robust and plausible CEs (with the robust neighbours as other vertices). This $\mathcal{X}_{plaus}$ ensures the soundness and completeness of our method (Section 5.3.3).

5.3.2 Bi-level Optimisation Method with MILP

5.3.2.1 Outer and Inner Optimisation problems

We separate the robust optimisation problem (5.2) to solve into outer minimisation and inner maximisation problems, as specified in Definitions 19 and 20.

Definition 19. *Given a NN M_θ and an input $x \in \mathcal{X}$ such that $M_\theta(x) < 0$, a distance metric $dist : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ and some plausible region $\mathcal{X}_{plaus} \subseteq \mathbb{R}^d$, let Δ' be a set of shifted models. Then, the* outer minimisation problem *finds a CE x' such that:*

$$\arg\min_{x'} \quad dist(x, x') \tag{5.3a}$$

$$\textit{subject to} \quad -M_{\theta'}(x') \leq 0, \ \textit{for each } M_{\theta'} \in \Delta', \tag{5.3b}$$

$$x' \in \mathcal{X}_{plaus} \tag{5.3c}$$

Definition 20. *Given a CE $x' \in \mathcal{X}$ found by the outer minimisation problem, the set of bounded model shifts Δ, the* inner maximisation problem *finds a shifted model $M_{\theta'}$ such that:*

$$\arg\max_{M_{\theta'}} \quad -M_{\theta'}(x') \tag{5.4a}$$

$$\textit{subject to} \quad M_{\theta'} \in \Delta \tag{5.4b}$$

The outer minimisation problem relaxes the constraint that a CE should be robust to all possible model shifts in the set Δ; instead, it requires robustness wrt a subset of the model changes $\Delta' \subset \Delta$. Δ' is initialised with the original classification model M_θ. At the first execution, the outer minimisation finds the closest CE x' valid for that model. Then, x' is passed to the inner maximisation problem to compute the model shift $S(M_\theta)$ that produces the lowest model output score. This model shift is considered to be the

worst-case perturbation on the model parameters in the set Δ, and is added to Δ'. In the next iterations, x' is updated to the closest CE valid for all the models in Δ' (outer), which is being expanded (inner), until convergence.

5.3.2.2 MILP Formulations

The proposed bi-level optimisation method in Section 5.3.2.1 is independent of specific NN structures. In this section, we take NNs with ReLU activation functions as an example to further elaborate the method. We denote the total number of hidden layers in an NN M_θ as h. We call N^0, $N^{(h+1)}$, and $N^{(i)}$ the sets of input, output, and hidden layer nodes for $i \in [h]$, and their node values are $V^{(0)}$, $V^{(h+1)}$, and $V^{(i)}$. For hidden layer nodes $V^{(i)} = \text{ReLU}(W^{(i)} \cdot V^{(i-1)} + B^{(i)})$, and for output layer nodes $V^{(h+1)} = W^{(h+1)} \cdot V^{(h)} + B^{(h+1)}$, where $W^{(i)}$ is the weight matrix connecting nodes at layers $i-1$ and i, and $B^{(i)}$ is the bias vector of nodes N_i. We instantiate the formulations using normalised L_1, while our method PROPLACE can accommodate arbitrary distance metrics.

The outer minimisation problem is equivalent to the following MILP program, where the subscripts j on weight matrices and bias vectors indicate they are

model parameters of the j-th model $M_{\theta,j} \in \Delta'$:

$$\min_{x',\gamma,\lambda} \quad \|x - x'\|_1 \tag{5.5a}$$

$$\text{s.t.} \quad V_j^{(0)} = x', \tag{5.5b}$$

$$\gamma_j^{(i)} \in \{0,1\}^{|N_i|},\ i \in [h], j \in [|\Delta'|] \tag{5.5c}$$

$$0 \leq V_j^{(i)} \leq \mathcal{M}\gamma_j^{(i)},\ i \in [h], j \in [|\Delta'|] \tag{5.5d}$$

$$W_j^{(i)}V_j^{(i-1)} + B_j^{(i)} \leq V_j^{(i)} \leq (W_j^{(i)}V_j^{(i-1)} + B_j^{(i)}) + \mathcal{M}(1 - \gamma_j^{(i)}),$$
$$i \in [h], j \in [|\Delta'|] \tag{5.5e}$$

$$W_j^{(h+1)}V_j^{(h)} + B_j^{(h+1)} \geq 0, j \in [|\Delta'|] \tag{5.5f}$$

$$\lambda_l \in [0,1],\ l \in [|S_{k,\Delta} \cup \{x\}|], \quad \sum_{l=1}^{|S_{k,\Delta}\cup\{x\}|} \lambda_l = 1$$

$$x' = \sum_{l=1}^{|S_{k,\Delta}\cup\{x\}|} \lambda_l x'_l, x'_l \in S_{k,\Delta} \cup \{x\} \tag{5.5g}$$

Constraints (5.5b) - (5.5f) and constraint (5.5g) correspond respectively to the robustness and plausibility requirement in (5.3b) - (5.3c).

The inner maximisation program can be formulated as the following MILP program, where the subscripts 0 on weight matrices and biases indicate they are model parameters of the original model M_θ, and δ is the bound of model magnitude change specified in Δ:

$$\max_{W,B,\gamma} \quad V^{(h+1)} \tag{5.6a}$$

$$\text{s.t.} \quad V^{(0)} = x', \tag{5.6b}$$

$$\gamma^{(i)} \in \{0,1\}^{|N_i|},\ i \in [h] \tag{5.6c}$$

$$0 \leq V^{(i)} \leq \mathcal{M}\gamma^{(i)},\ i \in [h] \tag{5.6d}$$

$$W^{(i)}V^{(i-1)} + B^{(i)} \leq V^{(i)} \leq (W^{(i)}V^{(i-1)} + B^{(i)}) + \mathcal{M}(1 - \gamma^{(i)}),\ i \in [h] \tag{5.6e}$$

$$V^{(h+1)} = W^{(h+1)}V^{(h)} + B^{(h+1)} \tag{5.6f}$$

$$W_0^{(i)} - \delta \leq W^{(i)} \leq W_0^{(i)} + \delta,\ i \in [h+1] \tag{5.6g}$$

$$B_0^{(i)} - \delta \leq B^{(i)} \leq B_0^{(i)} + \delta,\ i \in [h+1] \tag{5.6h}$$

Due to the flexibility of such MILP programs, the framework accommodates continuous, ordinal, and categorical features [98]. Specific requirements like feature immutability or associations between features can also be encoded [139]. These MILP problems can be directly solved using off-the-shelf solvers such as Gurobi [58].

5.3.3 Soundness, Completeness and Convergence of Algorithm 5

We now discuss the soundness and completeness of our method by restricting the search space for the CE to the plausible region $\mathcal{X}_{plaus}$. From its definition, the vertices (except the input x) of $\mathcal{X}_{plaus}$ are Δ-robust, which thus satisfies the robustness requirement of our target problem (Definition 16). This means that there exist at least k points in the search space satisfying constraint (5.2c) that also satisfy constraint (5.2b), making these points feasible solutions for the target problem. We may thus make the following proposition:

Proposition 1. *Algorithm 5 is sound and complete if* $\exists\ x' \in \mathcal{D}$ *such that* x' *is* Δ*-robust.*

Soundness follows from the termination condition of Algorithm 5: the loop only exits when $-M_{\theta'}(x') \leq 0$ for the worst-case model shift $M_{\theta'}$ returned by the inner maximisation over the entire set Δ (Definition 20). Hence the returned x' satisfies $\max_{M_{\theta'} \in \Delta}[-M_{\theta'}(x')] \leq 0$, i.e., it is Δ-robust, and it is plausible since the outer minimisation constrains $x' \in \mathcal{X}_{plaus}$. Completeness follows from the feasibility argument above: whenever a Δ-robust instance exists in $\mathcal{D}$, the search space $\mathcal{X}_{plaus}$ contains at least one Δ-robust point, so the outer minimisation is feasible and returns a solution.

Next, we adapt the method in [100, Section 5.2] to provide an upper bound on the maximum number of iterations of Algorithm 5.

Proposition 2. *Given the requirements of Algorithm 5, assume the classifier* M_θ *is Lipschitz continuous in* x'*. Then, the maximum number of iterations before Algorithm 5 terminates is bounded.*

Proof. Firstly, we assume two small tolerance variables $\sigma > t > 0$ and modify the robustness constraint (5.2b) of Definition 16 to: $\max_{M_{\theta'} \in \Delta} [-M_{\theta'}(x') + \sigma] \leq t$, such that the correctness of the robustness guarantee is not affected. The termination condition for Algorithm 5 therefore becomes $-M_{\theta'}(x') + \sigma \leq t$.

Consider the plausible CE problem (Definition 15 with the validity constraint modified to $-M_\theta(x') + \sigma \leq t$), which is the problem solved by the first execution (iteration 1) of the outer minimisation problem in Algorithm 5. We denote its feasible region as $\mathcal{F}$. Suppose M_θ is a ReLU NN without the final (softmax or sigmoid) activation layer, then M_θ is Lipschitz continuous. Let $f(x', M_\theta) := -M_\theta(x') + \sigma$, then f is Lipschitz continuous in x' over $\mathcal{F}$ with some Lipschitz constant L. For a distance metric $dist : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$, and any $x_1, x_2 \in \mathcal{F}$, we have:

$$|f(x_1, M_\theta) - f(x_2, M_\theta)| \leq L \cdot dist(x_1, x_2) \tag{5.7}$$

At iteration m, we denote the CE found by the outer minimisation as $x'^{(m)}$, and the shifted model found by the inner maximisation as $M_\theta^{(m)}$. Then, $f(x'^{(m)}, M_\theta^{(m)}) := -M_\theta^{(m)}(x'^{(m)}) + \sigma$. Assume at step m the algorithm has not terminated, then

$$f(x'^{(m)}, M_\theta^{(m)}) > t \tag{5.8}$$

For the iteration steps $n > m$, $x'^{(n)}$ is required to be valid on $M_\theta^{(m)}$ as specified in the outer minimisation problem, we therefore have:

$$f(x'^{(n)}, M_\theta^{(m)}) \leq 0 \tag{5.9}$$

Combining (5.8) and (5.9) yields

$$f(x'^{(m)}, M_\theta^{(m)}) - f(x'^{(n)}, M_\theta^{(m)}) > t \tag{5.10}$$

Further combining (5.10) with (5.7), for the iteration steps $n > m$,

$$dist(x'^{(m)}, x'^{(n)}) > \frac{t}{L} \tag{5.11}$$

Consider the balls $B^{(i)}, i = 1, \ldots, m$, of diameter $\frac{t}{L}$ centred at each intermediate result of the outer minimisation problem, $x'^{(i)}$. From (5.11), it can be concluded that for any two intermediate $x'^{(i)}$, $x'^{(j)}$, $1 < i, j < m$, $dist(x'^{(i)}, x'^{(j)}) > \frac{t}{L}$, and $x'^{(i)}$ and $x'^{(j)}$ are the centres of the balls $B^{(i)}$ and $B^{(j)}$. Therefore, any two circles will not intercept. The total volume of these balls is thus $m \times U \times (\frac{t}{L})^d$, where U is the unit volume in $\mathbb{R}^d$.

Consider a ball that encompasses the feasible solution region $\mathcal{F}$ and let R be its radius. We know that $x'_i, i = 1, \ldots, m$, are all within the feasible region $\mathcal{F}$, therefore, the ball B that has a radius $R + \frac{t}{2L}$ will cover the spaces of the small balls $B^{(i)}, i = 1, \ldots, m$. Also, the volume of B is $U \times (2R + \frac{t}{L})^d$ and will be greater than the total volume of the small balls, which means:

$$U \cdot \left(2R + \frac{t}{L}\right)^d > m \cdot U \cdot \left(\frac{t}{L}\right)^d \quad \Longrightarrow \quad m < \left(\frac{2RL}{t} + 1\right)^d$$

It can be concluded that the step number at which Algorithm 5 has not terminated is bounded above by the $(\frac{2RL}{t} + 1)^d$. □

5.4 Experiments

In this section, we demonstrate that our proposed method achieves state-of-the-art performances compared with existing robust CEs generation methods.

Datasets and Classifiers. Our experiments use four benchmark datasets in financial and legal contexts: the adult Income (adult) [38], compas [4], Give Me Some Credits (gmc) [28], and heloc [46] datasets. We adopt the pre-processed versions available in the CARLA library [110] where each dataset contains binarised categorical features and min-max scaled continuous features. Labels 0 and 1 are the unwanted and the desired class, respectively. We split each dataset into two halves. We use the first half for training NNs with which the robust CEs are generated, and the second half for model retraining and evaluating the robustness of the CEs.

For making predictions and generating CEs, the NNs contain two hidden

layers with ReLU activation functions. They are trained using the Adam optimiser with a batch size of 32, and under the standard $80\%, 20\%$ train-test dataset split setting. The classifiers achieved 84%, 85%, 94%, and 76% accuracies on the test set of adult, compas, gmc, and heloc datasets, respectively.

The retrained models have the same hyperparameters and training procedures as the original classifiers. Following the experimental setup in previous works [40, 45, 104, 18, 138], for each dataset, we train 10 new models using both halves of the dataset to simulate the possible retrained models after new data are collected. We also train 10 new models using 99% of the first half of the dataset (different 1% data are discarded for each training), to simulate the leave-one-out retraining procedures. The random seed is perturbed for retraining. These 20 retrained models are used for evaluating the robustness of CEs.

Evaluation Metrics. The CEs are evaluated by the following metrics for their proximity, plausibility, and robustness.

- ℓ_1 measures the average L_1 distance between a CE and its corresponding input.
- lof is the average 10-Local Outlier Factor [20] of the generated CEs, which indicates to what extent a data point is an outlier wrt its k nearest neighbours in a specified dataset. lof values close to 1 indicate inliers, larger values (especially if greater than 1.5) indicate outliers.
- vr, the validity of CEs on the retrained models, is defined as the average percentage of CEs that remain valid (classified to class 1) under the retrained models.
- $v\Delta$ is the percentage of CEs that are Δ-robust. The bound of model parameter changes δ is specified to be the same as the value used in our algorithm.

Baselines. We compare our method with six state-of-the-art methods for generating CEs, including five which target robustness. GCE [142] is

the first method to generate CEs for NNs, which minimises the ℓ_1 distance between the CEs and the inputs. Robust Bayesian Recourse (RBR) [104] addresses the proximity, robustness, and plausibility of CEs. RobXCE [40] is a nearest-neighbour-based method that focuses on a different notion of robustness to model changes. Robust Algorithmic Recourse (ROAR) [138] optimises for proximity and the same Δ notion of robustness. Proto-R and MILP-R are the methods proposed by [64] which embed the Δ-robustness test into the base methods of [140] and [98]. For all methods including ours, we tune their hyperparameters to maximise the validity after retraining vr.

	adult				compas				gmc				heloc			
	vr↑	**vΔ**↑	ℓ_1↓	**lof**↓	**vr**↑	**vΔ**↑	ℓ_1↓	**lof**↓	**vr**↑	**vΔ**↑	ℓ_1↓	**lof**↓	**vr**↑	**vΔ**↑	ℓ_1↓	**lof**↓
GCE	89	78	.175	1.59	57	0	.170	1.81	84	18	.148	2.80	49	0	.045	1.16
RBR	100	0	.031	1.28	100	62	.043	1.34	90	0	.050	1.31	80	0	.038	1.10
RobXCE	100	82	.064	1.28	100	82	.050	1.11	100	96	.073	1.35	100	30	.073	1.04
ROAR	99	98	.279	1.96	98	100	.219	2.84	96	88	.188	4.22	98	98	.109	1.57
PROTO-R	98	55	.068	1.60	100	100	.084	1.36	100	100	.066	1.49	100	100	.057	1.21
MILP-R	100	100	.024	1.69	100	100	.040	1.71	100	100	.059	2.08	100	100	.044	2.48
OURS	100	100	.046	1.22	100	100	.039	1.24	100	100	.058	1.24	100	100	.057	1.04

Table 5.1: Evaluations of PROPLACE (OURS) and baselines on NNs. The ↑ (↓) indicates that higher (lower) values are preferred for the evaluation metric.

Results. We randomly select 50 test points from each dataset that are classified to be the unwanted class, then apply our method and each baseline to generate CEs for these test points. Results are shown in Table 5.1.

As a non-robust baseline, the GCE method is the least robust while producing high ℓ_1 costs and poor plausibility. Though RBR shows the lowest ℓ_1 results on three datasets, it has only moderate robustness against the naturally retrained models and is not Δ-robust on any dataset. The rest of the baselines all show strong robustness on at least three datasets, with our method having slightly better vr and $v\Delta$ results, evaluated at 100% in every experiment. This indicates that our method PROPLACE can not only guarantee robustness under bounded model parameter changes but also induce reliable robustness against unbounded model changes. In terms of plausibility, our method has the best lof scores among all baselines. Therefore, our method has addressed the limitation in the literature that

no method optimises for guaranteed Δ-robustness and plausibility. Though the two properties have established trade-offs with proximity [109, 112, 138], our method still shows ℓ_1 costs lower than all methods except RBR, which is significantly less robust, and MILP-R, which finds outliers. For the compas dataset, our method has the best proximity result among all baselines.

Note that the PROTO-R baseline from the work which proposed certification for Δ-robustness failed to find Δ-robust CEs on the adult dataset, as was the case in their results (see Table 1, [64]). This is due to the fact that their method rely heavily on a base method to find CEs, and it is not straightforward to be always able to direct the hyperparameters search for optimising Δ-robustness. With improved soundness and completeness (Proposition 1), PROPLACE always finds provably robust results.

5.5 Summary, Limitations and Discussion

Summary. In this chapter, we proposed a robust optimisation framework PROPLACE to generate provably robust and plausible CEs for neural networks. The method addresses the limitation in the literature that existing methods lack formal robustness guarantees to bounded model parameter changes and do not generate plausible CEs. We proved the soundness, completeness, and convergence of PROPLACE. Through a comparative study, we show the efficacy of our method, demonstrating the best robustness and plausibility results with better proximity than the most robust baselines. Despite the specific form of robustness we target, PROPLACE is also empirically robust to model retraining with unbounded parameter changes.

Limitations. PROPLACE comes with two main limitations. First, the notion of Δ-robustness employed here is the same as in Chapter 4 and is thus equally conservative: it guarantees validity against every model shift in Δ, including worst-case perturbations that may not occur in realistic retraining. As discussed in Chapter 4, investigating relaxed forms of robustness would therefore again be a desirable direction towards less conservative guarantees. Second, the MILP formulation requires the classifier to be representable in an exact, deterministic form, which can be difficult to obtain, especially for larger

models. Following the practice of existing works [138] (see also discussions in Section 3.2.1.2), it would be valuable to investigate the empirical robustness performance of our method when the target model is approximated by a surrogate model amenable to such a representation.

Discussion. In Chapters 4 and 5, we focused on finding CEs with provable robustness guarantees against model changes for parametric models. These contributions open up several promising avenues for future work.

One such direction is to investigate relaxed forms of robustness, e.g., when the output intervals in $\mathcal{I}_{(\theta,\Delta)}$ for different classes overlap, with similar interval abstraction techniques. In this thesis, we considered deterministic robustness guarantees, aiming at ensuring CEs' validity against even worst-case model parameter perturbations encoded by Δ. However, we observed that this notion is conservative in that the worst-case perturbations might not always occur in realistic model retraining. One potential way to overcome this is to determine the hyperparameter δ which controls the magnitude of model changes in Δ using a validation set, as illustrated in Section 4.6.2. However, that requires additional computational efforts. Therefore, probabilistic relaxations of our Δ-robustness notion allowing more fine-grained analyses would be valuable.

Another possibility is to investigate the Δ-robustness of CEs in a causal setting. CEs which conform to some structural causal models [114] are usually within data distribution, and they reflect the true causes of predictions, making causality an additional desirable property [77]. It has also been found that plausible CEs are likely to be more robust against model shifts [109]. Therefore, there could be intrinsic links between causality and robustness. Dominguez-Olmedo et al. [37] have proposed a framework to compute CEs that are robust to noisy executions under a causal setting. It would be interesting to also investigate the interplay of Δ-robustness and causality towards facilitating the development of higher-quality CEs.

Finally, it would be possible to apply similar interval abstraction techniques to provide guarantees for other forms of CE robustness. In our Δ-robustness tests, a fixed-value CE is fed into the interval-valued abstraction of a classification model. Intuitively, if instead an interval-valued CE is passed

as input into a fixed-valued classification model, it would be similar to reasoning about the robustness of CEs against noisy executions. Going further, it would be interesting to investigate training techniques incorporating Δ-robustness notions to train models which can produce robust CEs by standard gradient-based optimisation methods.

Chapter 6

Synthesising Counterfactual Explanations via Label-Conditional Gaussian Mixture Variational Autoencoders

In the previous chapters, we studied CEs' robustness against model changes. According to findings in other concurrent research, this form of robustness is also similar to robustness against noisy execution by users of recourse (see discussions in Section 3.2).

In this chapter, we switch the focus and look at how a distinct form, robustness against input changes, can be guaranteed through a new method based on an advanced generative model, while covering most of the established CE desirable properties. Following the motivations in Section 6.1, this generative model is introduced in Section 6.2.1. Sections 6.2.2 and 6.2.3 will present the new CE methods. Both the generative model and the CE methods are evaluated in Section 6.3. Section 6.4 concludes this chapter with discussions on limitations.

6.1 Motivation

As discussed in Section 3.2, research in robust CEs has been fragmented, often addressing specific robustness types in isolation or without considering their interplay with other key criteria [67]. This has resulted in a lack of methods that can generate CEs integrating multiple properties, such as validity, plausibility, diversity, and multiple forms of robustness, within a single framework. This chapter introduces a novel approach to bridge this gap, with a specific focus on (i) encouraging robustness to model changes, such that the recourse stays valid under model parameter changes (such as regular retraining), and (ii) guaranteeing robustness to input perturbations, which amounts to the stability of generated CEs against small perturbations applied to the input [5, 86]. Indeed, if a method finds CEs that are readily invalidated under some model updates, or provides significantly different CEs for nearly identical inputs, the CEs are likely based on arbitrary model behaviour and algorithmic artefacts, rather than meaningful features. This particularly undermines CEs' trustworthiness for recourse purposes.

A recent state-of-the-art method enhances robustness to input changes by leveraging diverse instances of nearest neighbours in the training dataset from the target class [86]. However, this method remains heuristic and cannot guarantee perfect stability given the randomness in its distance-based thresholding process, and risks exposing sensitive training data. We argue that a more principled approach to robustness involves identifying a set of diverse, prototypical recourse points for the target class and then guiding all generated CEs to converge towards these points. While prior works have used generative models like standard variational autoencoders (VAEs), normalising flows, or diffusion models with similar intuitions, their primary goal was to improve plausibility by obtaining a realistic data manifold [108, 148, 101]. A key limitation is that they are typically unconditional, ignoring the information encoded in a classifier's predicted labels that are readily available during recourse generation. As a result, sampling with randomness, or complex algorithms such as latent space gradient optimisation, is needed to perform the CE search in the latent space, without guaranteeing their validity.

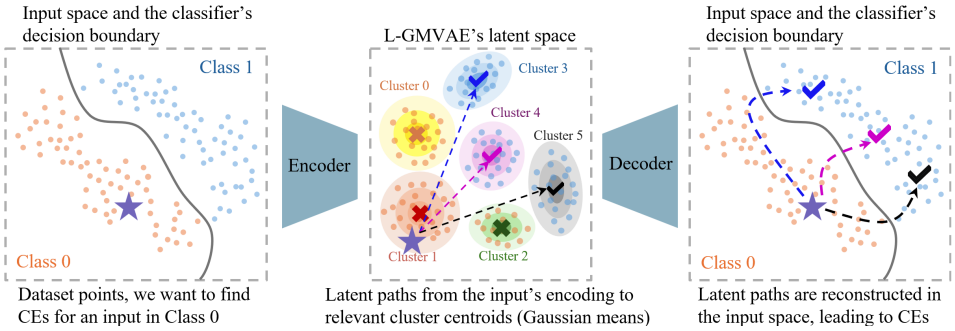


Figure 6.1: Illustration of LAPACE in binary classification. Given a dataset with a trained classifier's predictions (Left), a L-GMVAE is first learned with latent clusters (Gaussian components), capturing the data distribution with the classifier's predictions. In this example, we have 6 Gaussian components (Middle, the coloured areas). Prediction label 0 (1) is associated with Clusters 0-2 (3-5). The cluster centroids (learned Gaussian mixture prior) for classes 0 and 1 are marked with crosses and check marks. Assuming we are computing CEs for a negatively classified point (Left, purple star), LAPACE first performs linear interpolations linking the input's latent representation to each class 1 cluster centroid (Middle, dashed lines). These paths are then decoded to the input space to obtain paths of points, where they terminate at the decoded class 1 cluster centroids (Right).

Based on these insights, we propose Label-conditional Gaussian Mixture VAE (**L-GMVAE**), a novel generative model for recourse. We adapt the standard GMVAE [81, 35, 129] to be label-conditional [131] by explicitly mapping each class label to a dedicated set of Gaussian clusters in the latent space. After training, the mean of each Gaussian component serves as a latent centroid. As these centroids are learned through the VAE's objective, their decodings act as diverse, plausible, and robust prototypes for their associated class, making them ideal targets for recourse.

Building on this model, we introduce **LAPACE** (LAtent PAth Counterfactual Explanations), a simple yet effective method for generating paths describing smooth trajectories that connect inputs to their CEs. This method is illustrated in Figure 6.1. For any input, LAPACE finds a latent representation and generates multiple paths by linearly interpolating to each of the target class's latent centroids. When decoded, these latent trajectories form paths of high-quality CEs in the input space. Because all paths for a given target

class converge to the same set of fixed prototypes, LAPACE achieves perfect robustness to input perturbations. Furthermore, providing paths of CEs allows users to choose between solutions that are close to their original input and those that are more robust but further away. In addition, LAPACE is model-agnostic, requiring only black-box access to the classifier's prediction function. The synthetic CEs also do not expose real dataset points.

Overall, this chapter makes the following contributions:

- We propose L-GMVAE, a new generative model tailored for generating high-quality CEs.
- We introduce LAPACE, a novel model-agnostic CE generation method that uses the L-GMVAE to produce paths of CEs that are robust to input perturbations while also addressing validity, proximity, plausibility (our strong plausibility also leads to robustness against model changes), and diversity.
- We show how user-specified actionability constraints can be easily incorporated into LAPACE.
- We conduct comprehensive experiments that validate the competitive performance of LAPACE across eight quantitative metrics.

6.2 Latent Path Counterfactual Explanations

In this section, we first present L-GMVAE, discussing how it is inherently suitable for synthesising CEs. Then, we present LAPACE, and show how actionability requirements can be accommodated via gradient updates through the decoder network.

6.2.1 Label-Conditional GMVAE

L-GMVAE. Throughout the CE generation pipeline, the classifier-predicted label for the input and the desirable label for CEs are known. For computing

CEs, it would therefore make sense to explicitly incorporate label information into the cluster assignments. Following GMVAE ([129]'s version, see Section 2.3.2), the latent variable $z \in \mathbb{R}^h$ governs the data generation process. z is regulated by a Gaussian Mixture with K components, with a discrete variable $c \in \mathcal{C} = \{0, \ldots, K-1\}$ controlling the cluster assignment. In L-GMVAE, we partition the Gaussian clusters $\mathcal{C}$ using the possible L labels in $\mathcal{Y}$ as: $\mathcal{C} = \mathcal{C}_0 \cup \mathcal{C}_1 \cup \ldots \cup \mathcal{C}_{L-1}$, where each $\mathcal{C}_i \subset \mathcal{C}$, and for all $\mathcal{C}_i, \mathcal{C}_j$ with $i \neq j$ and $i, j \in \mathcal{Y}$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$. The most intuitive approach for partitioning is to uniformly assign L/K clusters for each class. Then, for a given class label y, its corresponding clusters are $\mathcal{C}_y$.

The generative component of L-GMVAE, $p(x, c, z \mid y)$, is characterised as:

$$p(x, c, z \mid y) = p(c \mid y)\, p(z \mid c)\, p(x \mid z) \tag{6.1a}$$

$$p(c \mid y) = \frac{1}{|\mathcal{C}_y|} \text{ if } c \in \mathcal{C}_y \tag{6.1b}$$

$$p(z \mid c) = \mathcal{N}(z \mid \mu_z(c), \sigma_z^2(c)) \tag{6.1c}$$

$$p(x \mid z) = \mathcal{N}(x \mid \mu_x(z), \sigma_x^2(z)) \tag{6.1d}$$

The inference model $q(z, c \mid x, y)$ of the GMVAE is factorised as:

$$q(z, c \mid x, y) = q(c \mid x, y)\, q(z \mid x, c, y) \tag{6.2}$$

We derive the evidence lower bound (ELBO) for this model, starting with

the data log likelihood:

$$
\begin{aligned}
log\, p(x) &= log \int_z \sum_{c \in \mathcal{C}_y} p(x, z, c \mid y)\, dz = log \int_z \sum_{c \in \mathcal{C}_y} q(z, c \mid x, y)\, \frac{p(x, z, c \mid y)}{q(z, c \mid x, y)}\, dz \\
&\geq \int_z \sum_{c \in \mathcal{C}_y} q(z, c \mid x, y)\, log\, \frac{p(x, z, c \mid y)}{q(z, c \mid x, y)}\, dz \qquad \text{(Jensen's Inequality)} \\
&= \mathbb{E}_{q(z,c|x,y)}[log\, p(x, z, c \mid y) - log\, q(z, c \mid x, y)] \\
&= \mathbb{E}_{q(z,c|x,y)} \left[\underbrace{\log \frac{p(c \mid y)}{q(c \mid x, y)}}_{\text{-KL}(c)} + \underbrace{\log \frac{p(z \mid c)}{q(z \mid x, c, y)}}_{\text{-KL}(z)} + \underbrace{\log\, p(x \mid z)}_{\text{Reconstruction}} \right] \\
&\coloneqq L_{\text{ELBO, LGMVAE}}
\end{aligned}
\tag{6.3}
$$

Model architecture and training. In practice, the L-GMVAE consists of two neural network components, the inference model and the generative model, matching the above definitions. The inference model takes x and the known y label as inputs, and outputs the probability distributions over the clusters. These are then concatenated together to predict z. The generative model produces a reconstructed input using z, and learns the Gaussian mixture prior ($p(z \mid c)$). The ELBO objective has three terms, KL divergence for c and z as regularisers, and a reconstruction term. The KL terms are expressed analytically, while the reconstruction term is the MSE loss between x and the reconstructed x'. The practical loss function is a negated, weighted sum of these ELBO terms. See Appendices B.1 and B.2 for more details of our trained L-GMVAE models.

Handling categorical data. Additionally, L-GMVAE can accommodate one-hot-encoded (OHE) categorical data by simply adding a sigmoid function at the decoder's final layer dimensions which correspond to the categorical dimensions. During training, the reconstruction loss for these dimensions uses binary cross-entropy loss. During inference, a rounding operation is placed after the sigmoid output to determine an integer value of either 0 or 1 for each OHE dimension.

We next discuss three key desiderata of a well-optimised L-GMVAE which would make it an ideal tool as a recourse generator, motivating LAPACE.

A recourse task is typically instantiated on a dataset $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1,\ldots,N}$ and a trained classifier M. In our framework, L-GMVAE is trained and tested using the predicted dataset by the classifier, $\{x^{(i)}, M(x^{(i)})\}_{i=1,\ldots,N}$, instead of the original dataset, in order for the explanations to be faithful to M.

Meaningful centroids. Each cluster centroid (the mean of its Gaussian prior) evolves into a representative point of its assigned class in the latent space. The synthetic point should also be representative of the class after being decoded into the input space. This is enforced by two components – the reconstruction loss ensures that the decoded centroid is a valid class instance, while the KL of z loss positions the centroid at the geometric centre of all data points assigned to that cluster.

This characteristic motivates using these reconstructed centroids as part of the CE generation method, because they guarantee the validity of the generated CEs. Furthermore, these points should be well within the data manifold. Therefore, if a classifier is updated, these prototypes are more likely to retain their predicted class, promoting plausibility and robustness to model changes for CEs. Finally, if we enforce that all CEs converge to the vicinity of the centroids of one class, it would also enhance the robustness of CEs to input changes. These properties are all quantitatively measured in our experiments in Section 6.3.

Diverse representations. The model learns varied prototypes for each class, implicitly encouraged by the two regularising terms. The KL of c penalises deviation from the uniform prior over a class's assigned clusters, promoting the use of all available clusters. Concurrently, the KL of z term incentivises separation, as each cluster learns a unique prior mean to best model the data routed to it. This encourages the reconstructed centroids to be distinctly different from each other. We would therefore have a diverse set of CEs (also quantitatively evaluated in experiments) if all prototypes were involved in CE generation.

Smooth latent manifold. As is standard in VAEs, proximity in the latent space corresponds to perceptual similarity in the input space. The KL of z and the reconstruction loss are responsible for this property, respectively

regularising an organised latent space and forcing the decoder to be a smooth, continuous function to accurately reconstruct inputs from their latent representations. Therefore, closeness in the latent space would translate to closeness in the input space, which is evaluated as the proximity between the CE and the original input.

6.2.2 Latent Paths and Counterfactual Explanations

Next, we describe how L-GMVAE can be used to generate paths of CEs leading to high-quality recourse. After learning, the L-GMVAE's latent cluster centroids are fixed and available through the optimised prior distribution $p(z \mid c)$ by taking the mean of the Gaussian component, denoted as z_{c_i} for each $c_i \in \mathcal{C}$. We refer to the L-GMVAE's decoder reconstruction functionality as $Dec : \mathbb{R}^h \to \mathbb{R}^d$. Then, for a target counterfactual class y' and its associated clusters $\mathcal{C}_{y'}$, $Dec(z_{c_j})$ is the reconstructed cluster centroid for each $c_j \in \mathcal{C}_{y'}$. From the discussions in Section 6.2.1, these $Dec(z_{c_j})$ points carry suitable properties for recourse purposes because they are valid, plausible, robust, and diverse. Additionally, they do not risk exposing existing data points given their synthetic nature.

Algorithm 6 LAPACE

Require: x, y, y', z_{c_j} for $c_j \in \mathcal{C}_{y'}$, Enc, Dec, interpolation steps $T = \{0, \ldots, 1\}$.

1: **Init:** $P \leftarrow \{\}$, $z_x \leftarrow Enc(x, y)$
2: **for** $c_j \in \mathcal{C}_{y'}$ **do**
3: $\quad \text{path}_j \leftarrow []$
4: $\quad$ **for** $\tau \in T$ **do**
5: $\quad\quad z_{x \to c_j, \tau} \leftarrow (1 - \tau) z_x + \tau z_{c_j}$
6: $\quad\quad$ Add $Dec(z_{x \to c_j, \tau})$ to path_j
7: $\quad$ Add path_j to P
8: **return** P

LAPACE is described in Algorithm 6. For input x with predicted label $y = M(x)$ and a target class y' for its CE, we first obtain its latent representation z_x via the *Enc*oder of L-GMVAE. (Line 1). Then, latent paths are obtained from z_x to each z_{c_j}, $c_j \in \mathcal{C}_{y'}$ via linear interpolation, denoted as $z_{x \to c_j, \tau} :=$

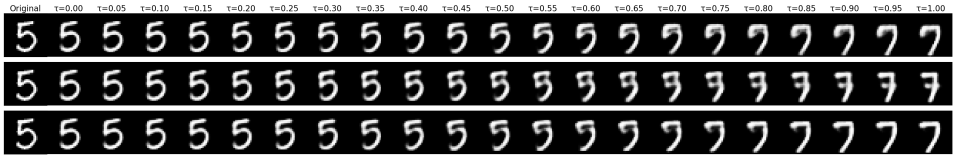


Figure 6.2: Example CE paths found by LAPACE on MNIST dataset, for an input image of class 5 and a target label of 7. This L-GMVAE has 3 Gaussian clusters per class. Each row is a separate CE path, going from the reconstructions of the original input (the second image from the left, $\tau = 0$) to each reconstructed cluster centroid (the last image, when $\tau = 1$).

$(1 - \tau)\, z_x + \tau\, z_{c_j}$ with finite steps $\tau \in [0, 1]$. During the walk in the latent space, we obtain paths of CE points as $\{Dec(z_{x \to c_j, \tau})\}$, for each τ value and each $c_j \in \mathcal{C}_{y'}$ (Lines 5 and 6). Points along the paths can then be tested with M to obtain their predicted labels. Once the L-GMVAE is trained, LAPACE is simple and effective, eliminating the need for gradient-based optimisation in the latent space and avoiding a complex hyperparameter-tuning process (as the step size is relatively trivial) for the CE generation process.

For intuitive visualisation, we show a CE example with image data on MNIST dataset [84] in Figure 6.2. Details of this trained L-GMVAE are in Appendix B.1. We observe that the L-GMVAE learns diverse writing styles of digit 7 as the reconstructed latent cluster centroids. The reconstructed paths are also plausible evolutions from digits 5 to 7. Later points in the path (e.g., roughly when $\tau \geq 0.7$) are more likely to be classified as the target class by a classifier. In Section 6.3, we quantitatively demonstrate LAPACE's competitive performance on tabular datasets.

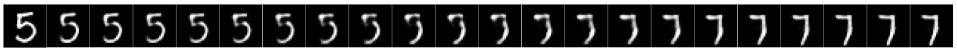


Figure 6.3: Continued MNIST example for the second cluster (the middle row in Figure 6.2). A requirement that a dash should not appear in the resulting CE images of class 7 is enforced on every image. For an input image size of 28×28, pixel values greater than a threshold of 0.01 at the 13th to 18th rows and the 8th to 14th columns (where the dash appears) are penalised via a loss function.

6.2.3 Accommodating Actionability Constraints

LAPACE also supports actionability constraints. We define a set of constraints in the input space using a differentiable function $g(x)$, where $g(x) \leq 0$ indicates the constraint is satisfied. For example, for a feature i that must not be greater than or equal to a value A, the function is simply $g(x) = x_i - A$. Constraints on multiple features can be intuitively summed together.

At each τ step during the latent space interpolation, we evaluate whether the constraints are satisfied via $g(Dec(z_{x \to c_j,\tau}))$. If not, we iteratively correct $z_{x \to c_j,\tau}$ via gradient descent until the constraints are satisfied or up to a maximum number of corrections $N_{correct}$: $z_{x \to c_j,\tau} \leftarrow z_{x \to c_j,\tau} - \eta \cdot \nabla_z g(D(z_{x \to c_j,\tau}))$, where η is a small learning rate and ∇ is the gradient.

In our MNIST example (Figure 6.2), the second cluster learns a dash for digit 7. If we decide that the dash is unwanted in our CE, we can enforce such a constraint, as illustrated by Figure 6.3. In the new paths, high pixel values at the dash positions are mitigated, including for the intermediate images.

6.3 Evaluation

In this section, we quantitatively evaluate the utility of L-GMVAE, and benchmark the quality of LAPACE along the common desirable properties against state-of-the-art baselines.

6.3.1 Experiment Setup

Datasets and classifiers. We run our experiments on four datasets, *heloc* credit [46], *wine* quality [26], *adult* income [12], and *compas* recidivism [4]. We split each dataset into a train and a test set for training classifiers. To demonstrate the model-agnostic nature of our approach, we train a random forest and a neural network for each dataset, which we use to compute the CEs. Characteristics of the datasets and models are summarised in Table 6.1.

Dataset	Classifier	Utility: Train on Real vs. Synthetic	Centroid Acc.
heloc 23=23+0	RF 73.05%	73.97%$_{\pm 1.07\%}$ / 71.07%$_{\pm 1.24\%}$	100%
	NN 73.58%	91.82%$_{\pm 0.55\%}$ / 89.99%$_{\pm 0.50\%}$	100%
wine 11=11+0	RF 75.13%	89.70%$_{\pm 1.66\%}$ / 87.42%$_{\pm 1.46\%}$	100%
	NN 76.95%	85.88%$_{\pm 0.85\%}$ / 84.21%$_{\pm 1.68\%}$	100%
adult 13=6+7	RF 81.15%	93.82%$_{\pm 0.94\%}$ / 81.13%$_{\pm 3.81\%}$	100%
	NN 82.04%	88.83%$_{\pm 1.16\%}$ / 75.86%$_{\pm 3.31\%}$	100%
compas 7=3+4	RF 78.17%	90.79%$_{\pm 1.63\%}$ / 85.03%$_{\pm 1.54\%}$	100%
	NN 77.70%	92.26%$_{\pm 1.70\%}$ / 84.00%$_{\pm 3.26\%}$	100%

Table 6.1: Dataset, classifier details, and L-GMVAE utility evaluation. The first column reports the number of features (#continuous + #categorical) of each dataset. The test accuracies of classifiers are included. The Utility columns report the test accuracies of new classifiers following TSTR.

L-GMVAE training and evaluation. The train set is further split into a train and a test set for L-GMVAE training and evaluation. An L-GMVAE is obtained for each dataset-classifier pair, learning the classifier's predictions on the new training set. Each L-GMVAE has 5 clusters per class. See Appendix B.2 for more details. Following the *Train on Synthetic, Test on Real (TSTR)* Protocol commonly employed to evaluate the utility of generative models [133], we sample a new synthetic dataset from the L-GMVAE matching the size of the new train set. Then, we train new random forest classifiers respectively using the new train set and the new synthetic train set, and evaluate their accuracy on the new test set. Well-optimised L-GMVAEs should result in a synthetic train set with a similar distribution to the original data, yielding a small gap between the test accuracies of the new classifiers, indicating good utility. Additionally, we evaluate whether the original classifier indeed classifies each reconstructed cluster centroid as its dedicated class. This is crucial for guaranteeing that the CEs found by LAPACE are valid. The results are shown in Table 6.1. L-GMVAE shows promising utility (up to 2.8% accuracy gap) on continuous-only datasets. When more categorical features are involved, larger gaps become apparent. Nonetheless, the centroid accuracies are consistently 100%. Also, as we will see by the plausibility evaluation in Section 6.3.2, the synthetic data from all trained L-GMVAEs remains realistic. These results indicate that our L-GMVAEs are suitable for recourse generation purposes.

CE evaluation procedure and metrics[1]**.** For each dataset-classifier pair, we obtain a CE test set of 100 test points predicted with label 0. We then use each method to generate CEs and evaluate on the following metrics. This process is repeated 5 times on different test sets. The procedures are in line with the existing literature. *Validity*: the portion of test points receiving valid CEs. *Proximity*: the average L1 distance between a test point and its CE(s). Lower distances indicate that the recourse prescribed by the CE would be easier to achieve by the end user, therefore it is more preferred [142]. *Plausibility*: the average local outlier factor (lof) [20] of each CE. Lof identifies outliers by measuring the local density deviation of a data point with respect to its neighbours, and lower values indicate better proximities to realistic data, therefore better plausibility. *Diversity*: if a CE algorithm finds multiple CEs per test input, diversity is calculated as the average pair-wise (L1) distance between these CEs. Higher distances indicate that the CEs are more diverse. *Robustness to Model Changes*: we obtain 20 retrained classifiers using different selections of the dataset as the training set, and test on average how many CEs are valid on these new classifiers. *Robustness to Input Changes* evaluates the stability of generated CEs when small norm-bounded perturbations are applied to the test input. We perturb the test input 10 times with small noises, generate new CEs for each of the perturbations, and report the average distance between the new CEs and the original CE (or max-set-distance [86] if the method supports finding multiple CEs per input). Smaller distances would mean better stability - finding similar CEs for similar inputs.

Additionally, on the two continuous-only datasets, we manually specify some actionability constraints (10 and 18 for wine and heloc) in the style of previous work in tabular data generation [132]. Each constraint specifies either a range for a feature value or the relation between two features (one greater than another) for the resulting CE. We randomly select 5 constraints for each test input, which are then enforced on a path of CEs and check if there is at least one CE satisfying all constraints while staying valid.

[1]The evaluation metrics and baseline method implementations are adapted from CARLA library [110] and RobustX library [70]. All experiments were performed on a linux machine with an Intel Xeon w5-2455X CPU with 128GB RAM, and 2x24GB NVIDIA GeForce RTX 4090 GPUs.

Method	Val.	Prox.	Plaus.	Div.	M Rob.	In. Rob.
NNCE [21]	✓	✓	✓	-	-	✓*
FACE [117]	✓	✓*	✓	-	-	-
RobXCE [40]	✓	✓*	✓	-	✓	✓*
DiCE [99]	✓	✓	-	✓	-	-
DRCE [86]	✓	✓	✓	✓	-	✓
LAPACE - Ours	✓	✓	✓	✓	✓*	✓

Table 6.2: The CE desirable properties addressed by each CE method in our experiments. ✓(*) indicates the property is explicitly taken into account (implicitly encouraged) by the method.

Baselines. We include state-of-the-art baselines covering the desirable properties of CEs, summarised in Table 6.2. Because LAPACE generates a path of CE points, to enable direct comparison with the baselines, we take CE points at three different locations in the path. *LAPACE-First* refers to the first CE point which turns the prediction label, when moving from the original input to the reconstructed centroids. *LAPACE-Last* refers to the last points in each CE path – the reconstructed centroids. Then, *LAPACE-Middle* is generated by decoding the latent vector located at the midpoint of the linear path connecting the latent representations of the previous two points.

For the actionability evaluation, we compare our path CEs with two path-based variants of FACE [117], an algorithm designed to obtain in-distribution CEs. FACE-Interpolation simply interpolates between the input and the FACE CE with 20 steps. FACE-Neighbours greedily selects the nearest neighbours from the training dataset, forming a path of 20 points from the input to the FACE CE. For these two baselines, due to the lack of an explicit way to incorporate constraint requirements, we directly modify the feature values to satisfy them and observe whether any CEs stay valid. In contrast, LAPACE allows building the constraints into CE generation, as introduced in Section 6.2.3 (LAPACE-constrained). To allow for a direct comparison to the baselines, we also directly modify the feature values in unconstrained LAPACE generation (LAPACE-naive).

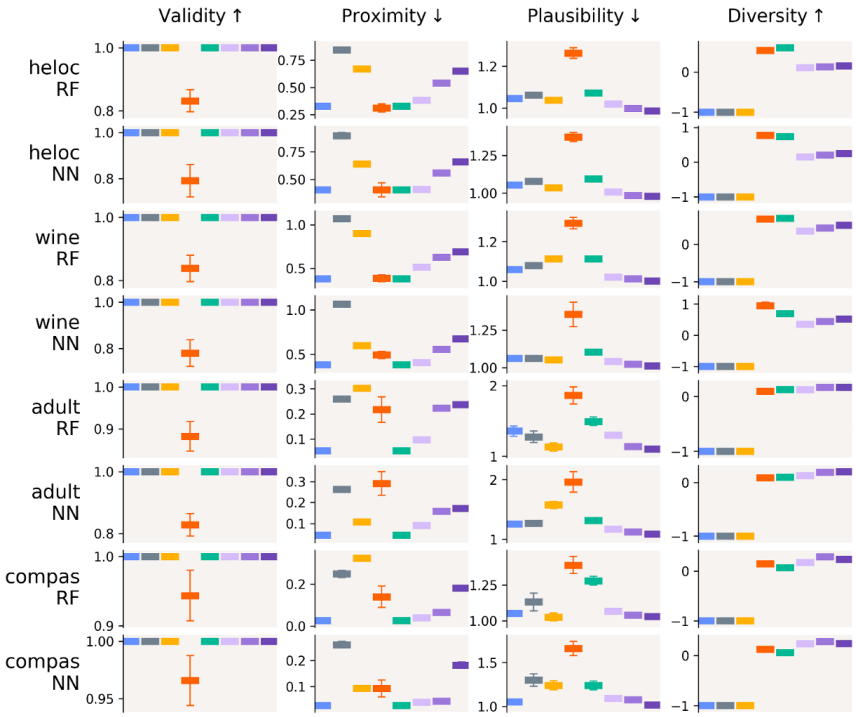


Figure 6.4: Quantitative evaluation on validity, proximity, plausibility, and diversity of CEs generated by: **NNCE**, **FACE**, **RobXCE**, **DiCE**, **DRCE**, **LAPACE-First**, **LAPACE-Middle**, **LAPACE-Last**. Each subplot is the quantitative comparison of all methods on one dataset-classifier combination and on one evaluation metric. The arrows following each metric indicate that higher or lower values are considered better. For diversity, the first three methods find only one CE per input for which the evaluation metric cannot be computed. Therefore, they are assigned a value of -1.

6.3.2 Evaluation Results

The benchmarking results are visualised in Figures 6.4 and 6.5. All included methods, apart from DiCE (due to the possible convergence failure in their optimisation procedures), always find valid CEs. In terms of proximity, NNCE and DRCE consistently perform the best, with DiCE showing competitive performances on continuous-only datasets because of their nearest-neighbour search mechanism over the training dataset. Our methods, especially LAPACE-First, show comparable performances. The observation that the first reconstructed latent CE with a flipped classifica-

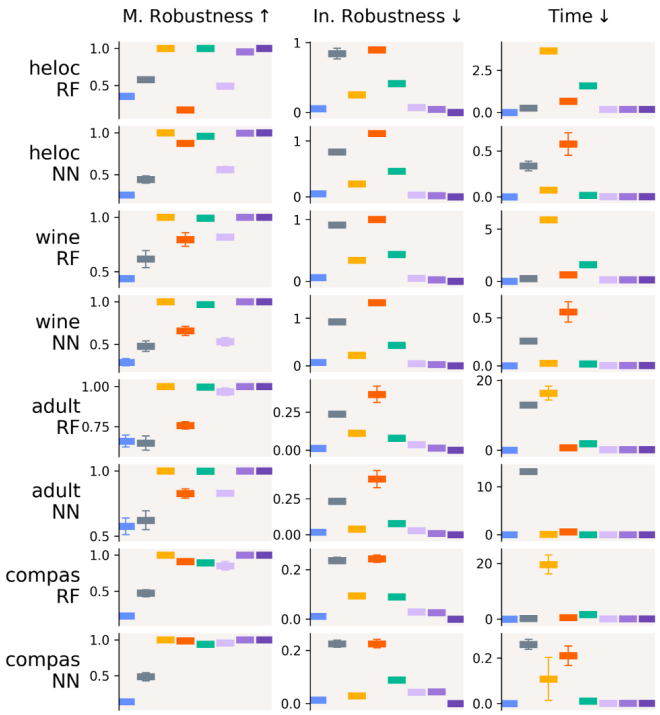


Figure 6.5: Quantitative evaluation of robustness and computation time of CEs generated by: **NNCE**, **FACE**, **RobXCE**, **DiCE**, **DRCE**, **LAPACE-First**, **LAPACE-Middle**, **LAPACE-Last**. Each subplot is the quantitative comparison of all methods on one dataset-classifier combination and on one evaluation metric. The arrows following each metric indicate that higher or lower values are considered better. For diversity, the first three methods find only one CE per input for which the evaluation metric cannot be computed. Therefore, they are assigned a value of -1.

tion (LAPACE-First) matches the proximity of the nearest-neighbour CEs confirms that the learned latent space by the L-GMVAE is well-optimised with respect to the training data. Notably, LAPACE achieves the best plausibility results across all datasets and models, effectively managing the known tradeoff between plausibility and proximity. In terms of diversity, our methods outperform DiCE and DRCE on datasets with mixed data types but perform worse on the two continuous-only datasets.

Our method demonstrates strong performance on two robustness metrics. LAPACE-Last achieves perfect (100%) robustness to model changes and is

guaranteed to be perfectly robust to input changes, as it always generates the same cluster centroid as CE for any input. LAPACE-Last is also competitive when compared with the robust baselines. It matches the proximity of RobXCE (robust to model changes) while achieving better plausibility, diversity, and a more efficient runtime. Compared to DRCE (robust to input changes), LAPACE-Last has notably better robustness and plausibility. Among our method's variants, robustness to model changes steadily increases from First to Last. This is expected, as the CE moves from a point near the original input to a more central point within the desired class. This shift improves plausibility and suggests that newly trained classifiers will likely predict it correctly. Finally, LAPACE is one of the fastest methods (second only to NNCE) because of its amortised inference nature. After a one-time training cost for the main model (L-GMVAE), generating a counterfactual for any new input is very fast, requiring only a few forward passes. It also offers a key advantage over nearest-neighbour-based methods (NNCE, RobXCE, DRCE) by generating synthetic CEs, protecting the privacy of the original dataset.

Dataset	**FACE-Interpolate**	**FACE-Greedy**	**LAPACE-Naive**	**LAPACE-constrained**
heloc-RF	0.92	0.96	1.00	1.00
heloc-NN	0.82	1.00	1.00	1.00
wine-RF	0.74	0.88	1.00	1.00
wine-NN	0.90	1.00	1.00	1.00

Table 6.3: Actionability evaluations – portion of test inputs for which the CEs satisfy the actionability constraints while remaining valid.

Table 6.3 reports the actionability results of LAPACE against two FACE variants. We used a combination of constraints on 5 features per dataset, which is around 20% and 50% of total features on heloc and wine, respectively. Under these conditions, both LAPACE variants consistently found valid counterfactuals that satisfied all constraints, outperforming the baseline methods.

In summary, LAPACE generates multiple paths of CEs that allow users to choose between solutions with high proximity and those with high plausibility and robustness. Actionability constraints can also be reliably addressed. The method is fast to compute, guarantees perfect robustness to input changes, and delivers the best-balanced performance across all metrics. Furthermore,

the very low standard deviation across multiple runs confirms the stability of our approach.

6.4 Summary, Limitations and Discussion

Summary. In this work, we introduce L-GMVAE, a generative model inherently suitable for recourse, and LAPACE, a novel, computationally efficient, and model-agnostic algorithm for generating CEs. LAPACE leverages the structured latent space of the L-GMVAE to produce diverse paths of CEs that are plausible, thus are robust to model changes. By design, all paths for a given target class converge to the same set of prototypical points, guaranteeing robustness against input perturbations. We also demonstrate that actionability constraints can be easily incorporated. Our comparative experiments using eight quantitative metrics validate LAPACE's competitiveness in generating high-quality CEs.

Limitations. LAPACE has limitations worth discussing. First, the guaranteed validity of its CEs is contingent upon a successful L-GMVAE training. This requires a validation step beyond monitoring the loss, where one must confirm that the decoded cluster centroids are correctly classified by the original classifier. Second, in line with other VAE-based methods, our L-GMVAE is less effective on datasets with a large number of categorical features, as is reflected in utility results (Table 6.1). Nevertheless, the performance of our CEs on these heterogeneous datasets remains highly competitive.

Discussion. The L-GMVAE and LAPACE methods in Chapter 6 open exciting avenues for future work. Links to the synthetic tabular data generation literature [133] are identified since this is how we view CE generation in this work. For example, dataset-level causal requirements between features (beyond user-specified actionability requirements) could be incorporated, possibly through adding specialised neural network layers into the training process of L-GMVAE [132]. This is further linked to the causality aspects discussed in Section 5.5. It would be interesting to explore structural causal models for the CE generation setting.

Furthermore, the Gaussian mixture-regularised latent space learned by our model offers a powerful tool for extended forms of counterfactual explainability. Instead of generating point-based CEs for one input at a time, the structured manifold could be used to derive region-based explanations, providing insights into the model's behaviour for entire sub-groups of the data [121, 15].

Chapter 7

Argumentative Ensembling for Robust Recourse under Model Multiplicity

In this chapter, we focus on a more abstract problem setting, MM, which occurs when multiple, equally performing models give conflicting predictions for the same ML task [17]. These models may be obtained, e.g., from different random seeds, and may differ in their model internals, e.g., model architectures, model types or high-level properties like fairness and robustness. MM has gained increasing attention when considering the trustworthiness of ML models, especially in scenarios where their predictions can substantially influence humans, like healthcare or finance [51].

After motivating in Section 7.1, we first characterise the problem setting of providing recourse (via CEs) under MM (7.2), discussing the desirable properties of its potential solutions, and analysing some naive existing methods. Then, Section 7.3 proposes our new method based on computational argumentation, with comprehensive theoretical analyses regarding the properties. Section 7.4 evaluates the practical effectiveness of our new approach. We finally conclude with discussions on limitations in Section 7.5.

7.1 Motivation

Recent research advocates taking into account the predictions from models under MM for inputs and reporting newly proposed metrics for evaluating MM severity, as an additional dimension of uncertainty that relevant stakeholders should be aware of [125, 63, 147]. Additionally, ensembling techniques are commonly used to deal with MM scenarios [17, 16]. A standard such technique is *naive ensembling* [16], where the predictions of several models are aggregated to produce a single outcome that reflects the opinion of a majority of models. For instance, naive ensembling applied to our running example would result in a rejection, as a majority of the models agree that the loan should not be granted (Figure 1.1). These efforts augment the standard ML practice of selecting the single most accurate model on a held-out test set, promoting better-informed predictions. However, their application to consequential decision-making tasks raises some important challenges.

Indeed, these methods tend to ignore the need to provide avenues for recourse to users negatively impacted by the models' outputs, which the ML literature typically achieves via the provision of CEs for the predictions. This is likely because dealing with MM while also taking CEs into account is non-trivial. Traditionally, CE generation methods assume access to a fixed model prediction, which is not directly available given the potentially conflicting predictions under MM. One apparent way for extending existing ensembling methods to account for recourse is to first retrieve an aggregated prediction, then compute CEs for this prediction. However, CEs for single models typically fail to produce recourse recommendations that are valid across the ensembled models [109], and those that are designed to be robustly valid will be more difficult to achieve for the end user [87]. Such naively extended ensembling also ignores the rich information carried by the CEs about each model's local decision boundaries, which could further reflect similarities of subsets of models under MM CEs. Instead, as we show later in this chapter, determining the aggregated prediction and recourse in tandem effectively addresses these limitations.

Another challenge is that naive ensembling ignores other meta-evaluation

aspects of models, like fairness, robustness, and interpretability, ignoring the fact that models under MM can demonstrate substantial differences in these regards [27, 124, 30] and users may have strong preferences for some of these aspects, e.g., they may prefer model fairness to accuracy [17].

This chapter makes the following contributions.

- We propose the first formalisation of the problem, which we name *recourse-aware ensembling* (RAE), along with six desirable properties which the solutions to RAE should satisfy.
- We present two natural extensions of naive ensembling to accommodate CEs for solving RAE, and theoretically show that they violate some important desirable properties.
- We then propose *argumentative ensembling*, a novel technique rooted in computational argumentation (see [7, 11] for overviews). Specifically, we model the RAE problem into Bipolar Argumentation Frameworks (BAF), then apply four argumentation semantics [24] to obtain the solution. Through extensive theoretical analysis, we show that it is able to solve RAE effectively while also naturally incorporating user preferences.
- We empirically evaluate eight instantiations of argumentative ensembling against the extended naive ensembling baselines. We show that our framework is effective in satisfying the desirable properties and addressing the model preferences without compromising the task performance.

7.2 Recourse under Model Multiplicity

As mentioned in Section 7.1, a common way to deal with MM in practice is to employ ensembling techniques, where the prediction outcomes of several models are aggregated to produce a single outcome. Aggregation can be performed in different ways, as discussed in [17, 16]. In the following, we

formalise a notion of *naive ensembling*, adopted in [17], and also known as *majority voting*, which will serve as a baseline for our analysis.[1]

Definition 21. *Given an input x, a set of models $\mathcal{M}$ and a set of labels $\mathcal{Y}$, we define the set of* top labels *$\mathcal{Y}_{max} \subseteq \mathcal{Y}$ as:*

$$\mathcal{Y}_{max} = argmax_{\ell_i \in \mathcal{Y}} |\{M_j \in \mathcal{M} | M_j(x) = \ell_i\}|.$$

Then we then use $\mathcal{M}^n(x) \in \mathcal{Y}_{max}$ to denote the aggregated classification by naive ensembling. *In the cases where $|\mathcal{Y}_{max}| > 1$, we select $\mathcal{M}^n(x)$ from $\mathcal{Y}_{max}$ randomly. With an abuse of notation, we also let $\mathcal{M}^n = \{M_j \in \mathcal{M} | M_j(x) = \mathcal{M}^n(x)\}$ denote the set of models that agree on the aggregated classification.*

For illustration, coming back to our loan example where M_1 and M_2 reject the loan ($M_1(x) = M_2(x) = 0$) while M_3 accepts it ($M_3(x) = 1$), we obtain $\mathcal{M}^n(x) = 0$ and $\mathcal{M}^n = \{M_1, M_2\}$.

Naive ensembling is known to be an effective strategy to mediate conflicts between models and is routinely used in practical applications. However, in this chapter, we take an additional step and aim to generate CEs providing recourse for a user that has been impacted by $\mathcal{M}^n(x)$. Recent work by [87] has shown that standard algorithms designed to generate CEs for single models typically fail to produce recourse recommendations that are robust across $\mathcal{M}^n$. One natural idea to address this would be to extend naive ensembling to account for CEs. Next, we formalise this idea in terms of several properties that we deem important in this setting. We then analyse two concrete methods extending naive ensembling in terms of the properties.

7.2.1 Problem Statement and Desirable Properties

Consider a non-empty set of models $\mathcal{M}=\{M_1, \ldots, M_m\}$ and, for an input x, assume a set $\mathcal{C}(x)=\{c_1, \ldots, c_m\}$ where each $c_i \in \mathcal{C}(x)$ is a CE for x[2], given

[1] It should be noted that, in [17], the case where there is no majority is not discussed.

[2] In this chapter, we use c to denote a CE point for better clarity, while in the previous chapters a CE is written as x'.

M_i. In the rest of this chapter, wherever it is clear that we refer to a given x, we use $\mathcal{C}$ and omit its dependency on x for readability. Our aim is to solve the problem outlined below.

Problem: Recourse-Aware Ensembling (RAE)
Input: input x, set $\mathcal{M}$ of models, set $\mathcal{C}$ of CEs
Output: "optimal" set $S \subseteq \mathcal{M} \cup \mathcal{C}$ of models and CEs.

To characterise optimality, we propose a number of desirable properties for the outputs of ensembling methods. We refer to these outputs as *solutions* for RAE. The most basic requirement requires that both models and CEs in the output are non-empty.

Definition 22. *An ensembling method satisfies* non-emptiness *iff for any given input x, set $\mathcal{M}$ of models and set $\mathcal{C}$ of CEs, any solution $S \subseteq \mathcal{M} \cup \mathcal{C}$ is such that $S \cap \mathcal{M} \neq \emptyset$ and $S \cap \mathcal{C} \neq \emptyset$.*

Specifically, non-emptiness ensures that the RAE method returns some models and some CEs. We then look to ensure that the RAE method returns a non-trivial set of models, as formalised next.

Definition 23. *An ensembling method satisfies* non-triviality *iff for any given input x, set $\mathcal{M}$ of models and set $\mathcal{C}$ of CEs, any solution $S \subseteq \mathcal{M} \cup \mathcal{C}$ is such that $|S \cap \mathcal{M}| > 1$.*

Clearly, the returned models should not disagree amongst themselves on the classification, which leads to our next requirement.

Definition 24. *An ensembling method satisfies* model agreement *iff for any given input x, set $\mathcal{M}$ of models and set $\mathcal{C}$ of CEs, any solution $S \subseteq \mathcal{M} \cup \mathcal{C}$ is such that $\forall M_i, M_j \in S \cap \mathcal{M}$, $M_i(x) = M_j(x)$.*

The next property, which itself requires model agreement to be satisfied, checks whether the set of returned models is among the largest of the agreeing sets of models, a motivating property of naive ensembling.

Definition 25. *An ensembling method satisfies* majority vote *iff it satisfies model agreement and for any given input x, set $\mathcal{M}$ of models, set $\mathcal{C}$ of CEs*

and set $\mathcal{Y}$ of labels, any solution $S \subseteq \mathcal{M} \cup \mathcal{C}$ is such that, letting $\ell_i = M_j(x)$ for all $M_j \in S \cap \mathcal{M}$, $\nexists \ell_k \in \mathcal{Y} \setminus \{\ell_i\}$ such that $|\{M_l \in \mathcal{M} | M_l(x) = \ell_k\}| > |\{M_l \in \mathcal{M} | M_l(x) = \ell_i\}|$.

Next, we consider the robustness of recourse. Previous work [87] considered a very conservative notion of robustness whereby explanations are required to be valid for all models in $\mathcal{M}$. While this might be desirable in some cases, we highlight that satisfying this property may not always be feasible in practice. We therefore propose a relaxed notion of robustness, which requires that CEs are valid only for the models that support them.

Definition 26. *An ensembling method satisfies* counterfactual validity *iff for any given input x, set $\mathcal{M}$ of models and set $\mathcal{C}$ of CEs, any solution $S \subseteq \mathcal{M} \cup \mathcal{C}$ is such that $\forall M_i \in S \cap \mathcal{M}$ and $\forall c_j \in S \cap \mathcal{C}$, $M_i(c_j) \neq M_i(x)$.*

While counterfactual validity is a fundamental requirement for any viable ensembling method, one also needs to ensure that the solutions it generates are coherent, as formalised below.

Definition 27. *An ensembling method satisfies* counterfactual coherence *iff for any given input x, set $\mathcal{M}$ of models and set $\mathcal{C}$ of CEs, any solution $S \subseteq \mathcal{M} \cup \mathcal{C}$, where $\mathcal{M} = \{M_1, \ldots, M_m\}$ and $\mathcal{C} = \{c_1, \ldots, c_m\}$, is such that $\forall i \in \{1, \ldots, m\}$, $M_i \in S$ iff $c_i \in S$.*

Intuitively coherence requires that *(i)* a CE is returned only if it is supported by a model and *(ii)* when the CE is chosen, then its corresponding model must be part of the support. This ultimately guarantees strong justification as to why a given recourse is suggested since selected models and their reasoning (represented by their CEs) are assessed in tandem.

The properties defined above may not be all satisfiable at the same time in practice. Next, we discuss two methods extending naive ensembling towards solving RAE, and explore their satisfaction (or otherwise) of the properties.

7.2.2 Extending Naive Ensembling for Recourse

We now present two strategies that leverage naive ensembling to solve RAE. In particular, we use the relationship between models in the ensemble $\mathcal{M}^n$ and their corresponding CEs as follows.

Definition 28. *Consider an input x, a set $\mathcal{M}$ of models and a set $\mathcal{C}$ of CEs. Let $\mathcal{M}^n \subseteq \mathcal{M}$ be the set of models obtained by naive ensembling. We define the set of* naive CEs *as:*

$$\mathcal{C}^n = \{c_i \in \mathcal{C} \mid M_i \in \mathcal{M}^n\};$$

and the set of valid CEs *as:*

$$\mathcal{C}^v = \{c_i \in \mathcal{C} \mid M_i \in \mathcal{M}^n \wedge \forall M_j \in \mathcal{M}^n, M_j(c_i) \neq M_j(x)\}.$$

Then, two possible solutions to RAE are $S^n = \mathcal{M}^n \cup \mathcal{C}^n$, named augmented ensembling*, or $S^v = \mathcal{M}^n \cup \mathcal{C}^v$, named* robust ensembling.

Intuitively, augmented ensembling suggests taking all the CEs in $\mathcal{C}$ that correspond to the models in $\mathcal{M}^n$. Meanwhile, robust ensembling extends augmented ensembling by enforcing the additional constraint that CEs are selected only if they are valid for all models in $\mathcal{M}^n$. We now provide an illustrative example to clarify the results produced by the two strategies.

Example 11. *Consider $\mathcal{M} = \{M_1, M_2, M_3, M_4, M_5\}$ and an input x such that $M_1(x) = M_2(x) = M_3(x) = 0$ and $M_4(x) = M_5(x) = 1$. Let $\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5\}$ be the set of CEs generated for x, i.e., $M_1(c_1) = M_2(c_2) = M_3(c_3) = 1$, while $M_4(c_4) = M_5(c_5) = 0$. Applying naive ensembling to $\mathcal{M}$ yields $\mathcal{M}^n = \{M_1, M_2, M_3\}$ and $\mathcal{M}^n(x) = 0$. Then, the set of naive CEs is $\mathcal{C}^n = \{c_1, c_2, c_3\}$, and thus augmented ensembling gives $S^n = \{M_1, M_2, M_3, c_1, c_2, c_3\}$. Now, assume that c_1 is invalid for M_2 (i.e., $M_2(c_1) = 0$), c_2 is invalid for M_1, c_3 is invalid for M_2, and all three CEs are otherwise valid for all models in $\mathcal{M}^n$. Then, then the set of valid CEs is empty, i.e., $\mathcal{C}^v = \emptyset$, and thus robust ensembling gives $S^v = \{M_1, M_2, M_3\}$.*

This example shows that both methods host major drawbacks: augmented ensembling may produce CEs which are invalid and thus it is not robust to MM, while robust ensembling is prone to returning no CEs. We now present a theoretical analysis to assess the extent to which augmented and robust ensembling are able to satisfy the properties given in Definitions 22 to 27.

Theorem 1. *Augmented ensembling satisfies non-emptiness, model agreement, majority vote and counterfactual coherence. It satisfies non-triviality if* $|\mathcal{M}|>2$. *It does not satisfy counterfactual validity.*

Proof. By Definition 21, it can be seen by inspection that $|\mathcal{M}^n|>0$. Thus, non-emptiness is satisfied. Again by inspection of the same definition, $\forall M_i, M_j \in \mathcal{M}^n$, $M_i(x) = M_j(x)$. Thus, model agreement is satisfied. We can also see that $\nexists \ell_i \in \mathcal{Y} \setminus \{\mathcal{M}^n(x)\}$ such that $|\{M_j \in \mathcal{M} | M_j(x) = \ell_i\}| > |\mathcal{M}^n|$. Thus, majority vote is satisfied. By Definitions 21 and 28, it can be seen that $\forall M_i \in \mathcal{M}$ and $\forall c_i \in \mathcal{C}$, $M_i \in S$ iff $c_i \in S$. Thus, counterfactual coherence is satisfied.

Example 11 shows that counterfactual validity is not satisfied by providing a counterexample.

Finally, the partial satisfaction of non-triviality can be proven by contradiction: assume $|\mathcal{M}|=n$, $n>2$ but $|\mathcal{M}^n|=1$. By Definition 21, $\mathcal{M}^n$ is the largest subset of $\mathcal{M}$ containing models with the same classification outcome. However, for binary classification, this implies that the remaining $n-1$ all agree on the opposite classification, i.e., $|\mathcal{M}\setminus\mathcal{M}^n|>|\mathcal{M}^n|$, which leads to a contradiction. □

Theorem 2. *Robust ensembling satisfies model agreement, majority vote and counterfactual validity. It satisfies non-triviality if* $|\mathcal{M}| > 2$. *It does not satisfy non-emptiness or counterfactual coherence.*

Proof. The proofs for model agreement, majority vote and non-triviality are analogous to those in Theorem 1 and so are omitted.

It can be seen by inspection of Definition 28 that $\forall M_i \in \mathcal{M}^n$ and $\forall c_j \in \mathcal{C}^v$, $M_i(c_j) \neq M_i(x)$. Thus, counterfactual validity is satisfied.

	S^n	S^v	$S^{a,s}$
non-emptiness	✓		✓
non-triviality	✓*	✓*	✓*
model agreement	✓	✓	✓
majority vote	✓	✓	
counterfactual validity		✓	✓
counterfactual coherence	✓		✓

Table 7.1: Augmented ($S^n = \mathcal{M}^n \cup \mathcal{C}^n$) and robust ($S^v = \mathcal{M}^n \cup \mathcal{C}^v$) ensembling, as well as our argumentative approach with the s-preferred extension (defined later in Section 7.3), assessed against the desirable properties defined in Section 7.2.1. Satisfaction of a property is shown by ✓, while partial satisfaction under given conditions is shown by ✓*.

Example 11 shows that non-emptiness and counterfactual coherence are not satisfied by providing a counterexample. □

These results, summarised in Table 7.1, demonstrate that there may exist cases in which both augmented and robust ensembling fail to solve RAE satisfactorily. This has strong implications on the quality of the results obtained in practice, as we will show experimentally in Section 7.4. Further, these methods provide no way to take into account users' preferences over the models. As previously mentioned (see Section 7.1, there could be different characteristics among the models in $\mathcal{M}$ in terms of meta-evaluation aspects, e.g., a model's fairness, robustness, and simplicity. Depending on the task, a model's fairness might be specified as being more important than its robustness or simplicity. In such cases, it would be desirable to have a principled way to rank models according to the preference specification, as promoted in [17]. The combination of these deficiencies motivates the need for a richer ensembling framework to solve RAE while incorporating user preferences, given next.

7.3 Argumentative Ensembling

In this section, we introduce our method for ensembling models and CEs

which inherently supports specifying preferences over models. We first present general definitions in Section 7.3.1 with illustrative examples, then we instantiate argumentative ensembling with four different argumentation semantics and undertake theoretical analyses of their properties (Section 7.3.2 to Section 7.3.4). We finally discuss alternative definitions of argumentative ensembling in Section 7.3.5.

7.3.1 Definition

We start by formalising ways to incorporate the aforementioned preferences over models. These preferences could be obtained by any information, e.g., meta-rules over models as suggested in [17], but we will assume that they are extracted wrt properties of the models, e.g., their accuracy or (a metric representing) their simplicity.

Definition 29. *Given a set $\mathcal{M}$ of models, a set $\mathcal{P}$ of* properties *is such that $\forall \pi \in \mathcal{P}$, $\pi : \mathcal{M} \to \mathbb{R}$ is a total function.*

We then define a preference over the properties such that users can impose a ranking of priority over them. Here and onwards, for simplicity we use total orders, denoted $\preceq$, over any set S such that, as usual, for any $s_i, s_j \in S$, $s_i \prec s_j$ iff $s_i \preceq s_j$ and $s_i \not\succeq s_j$. Also as usual, we say that $s_i \simeq s_j$ iff $s_i \preceq s_j$ and $s_i \succeq s_j$.

Definition 30. *Given a set $\mathcal{P}$ of properties, a* property preference $\preceq_{\mathcal{P}}$ *is a total order over $\mathcal{P}$.*

Model preferences can be defined using property preferences. In the following example, we define one way for doing so.

Example 12. *Consider the same models as in Example 11 and a set of properties $\mathcal{P}$=$\{\pi_1, \pi_2\}$ where π_1, π_2 represent model accuracy and simplicity, respectively, with a property preference $\preceq_{\mathcal{P}}$ such that $\pi_1 \succ_{\mathcal{P}} \pi_2$ and values for the satisfaction of properties as follows.*

A simple model preference $\preceq_{\mathcal{M}}$ over $\mathcal{M}$ may be such that, for any $M_i, M_j \in \mathcal{M}$, $M_i \succ_{\mathcal{M}} M_j$ iff: (i) $\pi_1(M_i) > \pi_1(M_j)$; or (ii) $\pi_1(M_i) = \pi_1(M_j)$ and

	M_1	M_2	M_3	M_4	M_5
π_1 (accuracy)	0.85	0.87	0.86	0.86	0.87
π_2 (simplicity)	0	0.75	1	0.5	0.75

$\pi_2(M_i) > \pi_2(M_j)$. *This* $\preceq_{\mathcal{M}}$ *is a total order over* $\mathcal{M}$ *and results in* $M_2 \simeq_{\mathcal{M}} M_5 \succ_{\mathcal{M}} M_3 \succ_{\mathcal{M}} M_4 \succ_{\mathcal{M}} M_1$.

Other ways to define preferences over models from preferences over properties of models can be defined, e.g., based on more sophisticated notions of dominance. We will assume some given notion of total preference over models, as follows, ignoring how it is obtained.

Definition 31. *Given a set* $\mathcal{M}$ *of models, a* model preference $\preceq_{\mathcal{M}}$ *over* $\mathcal{M}$ *is a total order over* $\mathcal{M}$.

How can we incorporate these model preferences into the ensembling, while still satisfying the properties defined in Section 7.2.1? To tackle this problem, we use bipolar argumentation[3].

We refer to Section 2.4 for relevant details of computational argumentation. We will denote the set of all preferred extensions for an AAF as P. For BAFs, we refer to the set of all d-preferred (respectively, s-preferred, c-preferred) extensions as P^d (respectively, P^s, P^c). We write the set of all stable extensions as P^{stable}. Finally, for σ amongst any BAF semantics (d-preferred, s-preferred, c-preferred, and stable), we use P^σ to denote the set of all σ extensions.

Definition 32. *The* BAF corresponding to *input* x, *set* $\mathcal{M}$ *of models, set* $\mathcal{C}$ *of CEs and model preference* $\preceq_{\mathcal{M}}$ *is* $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ *with:*

- $\mathcal{A} = \mathcal{M} \cup \mathcal{C}$;
- $\mathcal{T} \subseteq (\mathcal{M} \times \mathcal{M}) \cup (\mathcal{M} \times \mathcal{C}) \cup (\mathcal{C} \times \mathcal{M})$ *where:*
  - $\forall M_i, M_j \in \mathcal{M}$, $(M_i, M_j) \in \mathcal{T}$ *iff* $M_i(x) \neq M_j(x)$, $M_i \succeq_{\mathcal{M}} M_j$;

[3]We discuss the similarities and differences to abstract argumentation in Section 7.3.5.

- $\forall M_i \in \mathcal{M}$ *and* $c_j \in \mathcal{C}$ *where* $M_i(c_j) = M_i(x)$, $(M_i, c_j) \in \mathcal{T}$ *iff* $M_i \succeq_{\mathcal{M}} M_j$ *and* $(c_j, M_i) \in \mathcal{T}$ *iff* $M_j \succeq_{\mathcal{M}} M_i$;

- $\mathcal{S} \subseteq (\mathcal{M} \times \mathcal{C}) \cup (\mathcal{C} \times \mathcal{M})$ *where for any* $M_i \in \mathcal{M}$ *and* $c_j \in \mathcal{C}$, $(M_i, c_j), (c_j, M_i) \in \mathcal{S}$ *iff* $i = j$.

Here, a model attacks another model if they disagree on the prediction and the latter is not strictly preferred to the former. This means that models which are outperformed with regards to preferences must be defended by more-preferred, agreeing models in order to be considered acceptable. Models and CEs are treated similarly, with the CEs inheriting the preferences from the models by which they were generated, and attacks being present between them when the model considers the CE invalid. This, along with the fact that models and their CEs support one another, ensures that the models are inherently linked to their reasoning, in the form of their CEs, and conflicts are drawn not only when two models' predictions differ, but also when their reasoning differs.

Next, we define argumentative ensembling for solving the MM problem described in the corresponding BAF $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$.

Definition 33. *Consider an input x, a set $\mathcal{M}$ of models, a set $\mathcal{C}$ of CEs, a set $\mathcal{Y}$ of labels and a model preference $\preceq_{\mathcal{M}}$. Let $S^{a,\sigma}$ be a cardinality-maximal element of P^σ for the corresponding BAF $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$, according to some semantics σ, i.e.,*

$$S^{a,\sigma} \in argmax_{X \in P^\sigma} |X|.$$

Then, a solution to RAE by σ argumentative ensembling *is $S^{a,\sigma}$.*

The intuition is that, after modelling the conflicts under MM with a BAF, acceptability-based argumentation semantics can be applied to the BAF to obtain subsets of models and CEs which satisfy some argumentative properties, such as conflict-freeness. While these are satisfied by all extensions in P^σ according to the semantics σ, in Definition 33 we only return the cardinality-maximal extension in the spirit of better satisfaction of majority

vote. Note that there may be multiple extensions which are all cardinality-maximal ($|P^\sigma| > 1$, e.g., when one extension is $\{M_1, M_2, c_1, c_2\}$ and the other is $\{M_3, M_4, c_3, c_4\}$ under a four-model scenario). In this case, we randomly choose one as the solution to RAE. Alternatively, we could choose to report all viable resulting ensembles in P^σ, potentially allowing for more informed decisions to be made by relevant stakeholders. We leave this to future work.

The following example demonstrates how quickly the problem, when preferences are included, can become complex. This is the case even with only five models, far fewer than usual in MM. For illustrative purposes, we instantiate the argumentative ensembling solutions with the s-preferred semantics.

Example 13. *The BAF corresponding to input, models and CEs as in Example 12 is* $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ *with (see Fig. 7.1):* $\mathcal{A} = \{M_1, M_2, M_3, M_4, M_5, c_1, c_2, c_3, c_4, c_5\}$; $\mathcal{T} = \{(M_2, M_4), (M_2, M_5), (M_2, c_1), (M_2, c_3), (M_3, M_4), (M_3, c_4), (M_4, M_1), (M_5, M_1), (M_5, M_2), (M_5, M_3), (c_2, M_1)\}$; *and* $\mathcal{S} = \{(M_1, c_1), (M_2, c_2), (M_3, c_3), (M_4, c_4), (M_5, c_5), (c_1, M_1), (c_2, M_2), (c_3, M_3), (c_4, M_4), (c_5, M_5)\}$. *With the s-preferred semantics, this leads to* $P^s = \{\{M_2, c_2\}, \{M_4, M_5, c_4, c_5\}\}$, *and thus* $S^{a,s} = \{M_4, M_5, c_4, c_5\}$, *the aggregated prediction* $M_4(x) = M_5(x) = 1$.

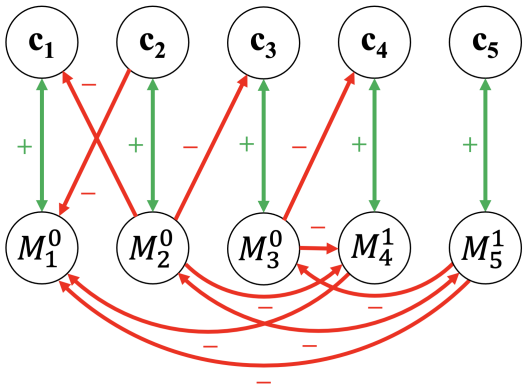
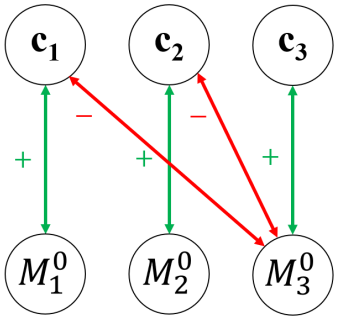
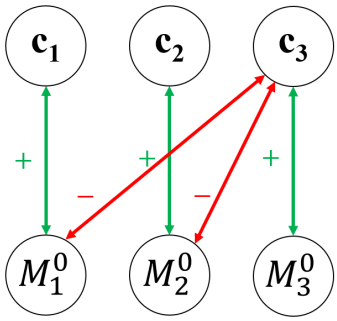


Figure 7.1: BAF for Example 13 where: models' predictions for the input x are given as superscripts, e.g., $M_1(x) = 0$ but $M_4(x) = 1$; reciprocal supports are represented by dual-headed green arrows labelled with + and standard (reciprocal) attacks are represented by single-headed (dual-headed, respectively) red arrows labelled with −.

This example shows how the use of CEs in the ensembling directly results



(a) Graphical illustration for Example 14



(b) Graphical illustration for Example 15

Figure 7.2: Two illustrative examples.

in the prediction being reversed, relative to the other ensembling methods, violating majority vote. This is due to the fact that, while the preferences over the two sets of models are roughly similar, the validity of the CEs for the models selected by the augmented and robust ensemblings is very poor. This means that when a CE that is valid for all models is required, some compromise must be made on the model selection, as we demonstrated.

Definition 33 remains agnostic to the argumentative semantics applied to the BAF. In this work, we consider the acceptability-based semantics in [24], and illustrate the similarities and differences of *d-preferred*, *stable*, *s-preferred*, and *c-preferred* semantics with Examples 14 and 15, two simple yet representative scenarios where three models agree on the prediction result but there are CE validity-related conflicts between them.

Example 14. *Consider* $\mathcal{M} = \{M_1, M_2, M_3\}$ *and an input* x *such that* $M_1(x) = M_2(x) = M_3(x) = 0$. *Let* $\mathcal{C} = \{c_1, c_2, c_3\}$ *be the set of CEs generated for* x, *i.e.,* $M_1(c_1) = M_2(c_2) = M_3(c_3) = 1$. *The three models are equally preferred. Assume the CEs* c_1, c_2 *are not valid on* M_3, *the BAF corresponding to this scenario is:* $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ *with:* $\mathcal{A} = \{M_1, M_2, M_3, c_1, c_2, c_3\}$; $\mathcal{T} = \{(c_1, M_3), (M_3, c_1), (c_2, M_3), (M_3, c_2)\}$; *and* $\mathcal{S} = \{(M_1, c_1), (M_2, c_2), (M_3, c_3), (c_1, M_1), (c_2, M_2), (c_3, M_3)\}$. *See Figure 7.2a for a graphical illustration.*

Example 15. *Consider* $\mathcal{M} = \{M_1, M_2, M_3\}$ *and an input* x *such that* $M_1(x) = M_2(x) = M_3(x) = 0$. *Let* $\mathcal{C} = \{c_1, c_2, c_3\}$ *be the set of CEs*

generated for x, i.e., $M_1(c_1) = M_2(c_2) = M_3(c_3) = 1$. The three models are equally preferred. Assume the CE c_3 is not valid on M_1 and M_2, the BAF corresponding to this scenario is: $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ with: $\mathcal{A} = \{M_1, M_2, M_3, c_1, c_2, c_3\}$; $\mathcal{T} = \{(c_3, M_1), (M_1, c_3), (c_3, M_2), (M_2, c_3)\}$; and $\mathcal{S} = \{(M_1, c_1), (M_2, c_2), (M_3, c_3), (c_1, M_1), (c_2, M_2), (c_3, M_3)\}$. See Figure 7.2b for a graphical illustration.

	Example 14	Example 15
stable	$S^{a,stable} = \{M_1, M_2, c_1, c_2\}$, $\{M_1, M_2, c_3\}$, $\{M_3, c_3\}$	$S^{a,stable} = \{M_1, M_2, c_1, c_2\}$, $\{M_3, c_1, c_2\}$, $\{M_3, c_3\}$
d-preferred	$S^{a,d} = \{M_1, M_2, c_1, c_2\}$, $\{M_1, M_2, c_3\}$, $\{M_3, c_3\}$	$S^{a,d} = \{M_1, M_2, c_1, c_2\}$, $\{M_3, c_1, c_2\}$, $\{M_3, c_3\}$
s-preferred	$S^{a,s} = \{M_1, M_2, c_1, c_2\}$, $\{M_3, c_3\}$	$S^{a,s} = \{M_1, M_2, c_1, c_2\}$, $\{M_3, c_3\}$
c-preferred	$S^{a,c} = \{M_1, M_2, c_1, c_2\}$, $\{M_3, c_3\}$	$S^{a,c} = \{M_1, M_2, c_1, c_2\}$, $\{M_3, c_3\}$

Table 7.2: Different semantics applied to Examples 14 and 15. $P^{stable}, P^d, P^s, P^c$ are in each table entry, with argumentative ensembling solutions highlighted within.

We list P^{stable}, P^d, P^s and P^c according to each semantics, as well as the corresponding argumentative ensembling results in Table 7.2. We observe that all semantics return the same set $\{M_1, M_2, c_1, c_2\}$ as the ensembling solution. Thus, argumentative ensembling methods based on these semantics would potentially satisfy all the requirements described in the desirable properties (Section 7.2.1). Therefore, intuitively, all semantics can lead to meaningful solutions. However, the accepted sets, which are the candidates for the final ensembling solution, are different. It should be noted that, since the stable and d-preferred extensions accept $\{M_1, M_2, c_3\}$, argumentative ensembling methods computed using these extensions would satisfy all (requirements of the) properties except counterfactual coherence.

In the following subsections, we perform a theoretical analysis examining four argumentative semantics applied to Definition 33 against the desirable properties from Section 7.2.1.

7.3.2 D-preferred and Stable Argumentative Ensembling

We start with the least strict d-preferred argumentative ensembling. We show that although it satisfies important properties of model agreement and counterfactual validity, but that it does not satisfy the rest.

Theorem 3. *D-preferred argumentative ensembling satisfies model agreement and counterfactual validity. It does not satisfy non-emptiness, non-triviality, majority vote or counterfactual coherence.*

Proof. Let us prove model agreement by contradiction. Assume that $\exists M_i, M_j \in \mathcal{M}$ such that $M_i(x) \neq M_j(x)$ and $\exists P_i^d \in P^d$ such that $M_i, M_j \in P_i^d$. By Definition 32, it follows that $\exists (M_i, M_j) \in \mathcal{T}$ or $\exists (M_j, M_i) \in \mathcal{T}$, which cannot be the case in a d-preferred set, which must be conflict-free (see Section 2.4), and so we have the contradiction.

Let us prove counterfactual validity by contradiction. Assume that $\exists M_i \in \mathcal{M}$ and $\exists c_j \in \mathcal{C}$ such that $M_i(c_j) = M_i(x)$ and $\exists P_i^d \in P^d$ such that $M_i, c_j \in P_i^d$. By Definition 32, it can be seen that $\exists (M_i, c_j) \in \mathcal{T}$ or $\exists (c_j, M_i) \in \mathcal{T}$, which cannot be the case in a d-preferred set, which, again, must be conflict-free, and so we have the contradiction.

We show that non-emptiness is not satisfied by providing the following counterexample. Consider $\mathcal{M} = \{M_1, M_2\}$ and an input x such that $M_1(x) = 0, M_2(x) = 1$. Let $\mathcal{C} = \{c_1, c_2\}$ be the set of CEs generated for x, with $M_1(c_1) = 1, M_2(c_2) = 0$, but $M_1(c_2) = 1, M_2(c_1) = 0$. The two models are equally preferred. The BAF corresponding to this scenario is: $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ with: $\mathcal{A} = \{M_1, M_2, c_1, c_2\}$; $\mathcal{T} = \{(M_1, M_2), (M_2, M_1)\}$; and $\mathcal{S} = \{(M_1, c_1), (M_2, c_2), (c_1, M_1), (c_2, M_2)\}$. Then, $P^d = \{\{c_1, c_2\}, \{M_1, c_1\}, \{M_2, c_2\}\}$. The d-preferred argumentative ensembling solution $S^{a,d}$ can therefore be $\{c_1, c_2\}$, which, by Definition 22, violates non-emptiness as no model is included.

Non-triviality is violated with the same counterexample as counterfactual coherence.

We show that majority vote is not satisfied by providing the following counterexample. Consider the BAF in Example 13. With d-preferred

extension, $S^d = \{M_4, c_4, M_5, c_5\}$, which is also the d-preferred argumentative ensembling solution. In this case, $\forall M_i \in S^d \cap \mathcal{M},\ M_i(x) = 1$. However, $|\{M_j \in \mathcal{M} | M_j(x) = 0\}| > |\{M_i \in \mathcal{M} | M_i(x) = 1\}|$. Therefore, by Definition 25, majority vote is violated.

We show that counterfactual coherence is not satisfied by providing the following counterexample. Consider Example 14 but additionally c_2 is not valid on M_1. The BAF corresponding to this scenario is: $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ with: $\mathcal{A} = \{M_1,\ M_2,\ M_3,\ c_1,\ c_2,\ c_3\}$; $\mathcal{T} = \{(c_2, M_1),\ (M_1, c_2),\ (c_1, M_3),\ (M_3, c_1),\ (c_2, M_3),\ (M_3, c_2)\}$; and $\mathcal{S} = \{(M_1, c_1),\ (M_2, c_2),\ (M_3, c_3),\ (c_1, M_1),\ (c_2, M_2),\ (c_3, M_3)\}$. Then, $P^d = \{\{M_1, c_1\}, \{M_2, c_2\}, \{M_3, c_3\}, \{M_1, c_3\}, \{M_2, c_1\}, \{M_2, c_3\}\}$. The d-preferred argumentative ensembling solution $S^{a,d}$ can therefore be any of $\{M_1, c_3\}, \{M_2, c_1\}, \{M_2, c_3\}$, all of which violate counterfactual coherence (Definition 27).

□

The violation of counterfactual coherence and non-emptiness hinders the reliability of d-preferred argumentative ensembling. Although practically, any CE and model can be included in the solution set as long as the CE is valid on that model, this might create imbalances in the number of models and CEs, causing the exclusion of any model (e.g., as in the counterexample for non-emptiness in the proof of Theorem 3), or fewer CEs, causing the set of explanations to be less diverse. However, our empirical evaluations in Section 7.4 find that these properties are almost always satisfied in practice.

For an extension to be stable, in addition to being conflict-free, it also needs to set-attack all arguments that it does not contain. Next, we show that under our definition of BAF, this stricter requirement is automatically satisfied by the d-preferred semantics. Therefore, the results in Theorem 3 also apply to stable argumentative ensembling.

Theorem 4. *Stable argumentative ensembling is equivalent to d-preferred argumentative ensembling.*

Proof. Let us prove by contradiction that d-preferred extensions are also stable under our BAF $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$. Let $P_i^d \in P^d$ be a d-preferred extension,

which, by definition (Section 2.4), means it is conflict-free. Assume that P_i^d is not stable, i.e., it does not attack all elements that it does not contain: $\exists \alpha \in \mathcal{A}$ s.t. $\alpha \notin P_i^d$, and P_i^d does not set-attack α. By Definition 32, no argument can self-attack, so $(\alpha, \alpha) \notin \mathcal{T}$.

Let us first consider the case where $\{\alpha\}$ set-attacks at least one argument in P_i^d. Because P_i^d is d-preferred, it defends all its elements. Therefore, P_i^d must set-attack α, so we have the contradiction.

Let us consider the case where $\{\alpha\}$ does not set-attack any argument in P_i^d. Because P_i^d does not set-attack α, $P_i^d \cup \{\alpha\}$ is conflict-free. If α has no attacker, we know that $P_i^d \cup \{\alpha\}$ defends all its elements, and so it must contain α. Thus, P_i^d cannot be a d-preferred extension because it is not $\subseteq$-maximal. Then we have the contradiction. Contrastively, let us consider when α has one or more attackers, written as $X \subseteq \mathcal{A}$. If we consider any $x \in X$, we need to consider three scenarios depending on the relation between x and P_i^d. When there exist attacks from x to any argument in P_i^d, there must also be an attack from P_i^d to x (self-defence). Therefore, $P_i^d \cup \{\alpha\}$ defends all its elements. P_i^d cannot be a d-preferred extension because it is not $\subseteq$-maximal. Then we have the contradiction. When there are attacks from P_i^d to x, the d-preferred extension must defend all its elements and so we have the same contradiction as the previous case. Finally, when there are no attack relations between x and P_i^d, P_i^d is not a d-preferred extension because it does not contain the arguments in X and so, again, it is not $\subseteq$-maximal and we have the contradiction.

We can therefore conclude that d-preferred extensions are also stable under our BAF definitions and that stable argumentative ensembling is equivalent to d-preferred argumentative ensembling. □

7.3.3 S-preferred and c-preferred Argumentative Ensembling

We now undertake a theoretical analysis of s-preferred argumentative ensembling, demonstrating some of the desirable behaviours thereof via properties introduced in Section 7.2.1.

Theorem 5. *S-preferred argumentative ensembling satisfies non-emptiness, model agreement, counterfactual validity and counterfactual coherence. It satisfies non-triviality if for some $M_i \in \mathcal{M}$, where $\nexists M_j \in \mathcal{M} \setminus \{M_i\}$ such that $M_j \succ_{\mathcal{M}} M_i$, $\exists M_k \in \mathcal{M}$ such that $M_k(x) = M_i(x)$, $M_k(c_i) \neq M_k(x)$ and $M_i(c_k) \neq M_i(x)$. It does not satisfy majority vote.*

Proof. Let us first prove counterfactual coherence. By Definition 32, $\forall M_i \in \mathcal{M}$ and $\forall c_j \in \mathcal{C}$, $(M_i, c_j), (c_j, M_i) \in \mathcal{S}$. These are the only support relations in our BAF. Thus, there exists an indirect attack on any $M_i \in \mathcal{M}$ iff there exists a direct attack on $c_i \in \mathcal{C}$. Likewise, there exists an indirect attack on any $c_i \in \mathcal{C}$ iff there exists a direct attack on $M_i \in \mathcal{M}$. Then, letting $\mathcal{M} = \{M_1, \ldots, M_m\}$ and $\mathcal{C} = \{c_1, \ldots, c_m\}$, since we know any $P_i^s \in P^s$ is $\subseteq$-maximal by Definition 33, P_i^s must be such that $\forall i \in \{1, \ldots, m\}$, $M_i \in S$ iff $c_i \in S$.

Let us prove non-emptiness by contradiction. Assume that $\exists P_i^s \in P^s$ such that $P_i^s \cap \mathcal{M} = \emptyset$ or $P_i^s \cap \mathcal{C} = \emptyset$. We know from the above proof that, $\forall P_i^s \in P^s$, $P_i^s \cap \mathcal{M} \neq \emptyset$ iff $P_i^s \cap \mathcal{C} \neq \emptyset$. Then, by the definition of s-preferred extensions (see Section 2.4), it must be the case that $\forall P_i^s \in P^s$, $P_i^s = \emptyset$. Based on the fact that, by Definition . 31, $\preceq_{\mathcal{M}}$ is a total ordering and thus transitive, this is not possible as it will always be the case that $\exists M_i \in \mathcal{M}$ such that $\nexists M_j \in \mathcal{M}$ where $M_j \succ_{\mathcal{M}} M_i$. Thus, by Definition 32, either $\mathcal{T}(M_i) \cup \mathcal{T}(c_i) = \emptyset$ or $\forall \alpha_k \in \mathcal{T}(M_i) \cup \mathcal{T}(c_i)$, $M_i \in \mathcal{T}(\alpha_k)$ or $c_i \in \mathcal{T}(\alpha_k)$, meaning $\{M_i, c_i\}$ is either unattacked or is able to defend itself, therefore would be acceptable in at least one s-preferred extension, and we have the contradiction.

By definition, s-preferred extensions are also d-preferred (see Section 2.4). Therefore, model agreement and counterfactual validity are satisfied (Theorem 3).

Let us prove the conditional satisfaction of non-triviality by contradiction. From Definition 33 and the proof for counterfactual coherence above, for $|S^{a,s} \cap \mathcal{M}| = 1$, it must be that $\forall P_i^s \in P^s$, $|P_i^s \cap \mathcal{M}| = 1$. However, from the above assumptions we can see that for some $M_i \in \mathcal{M}$, where $\nexists M_j \in \mathcal{M} \setminus \{M_i\}$ such that $M_j \succ_{\mathcal{M}} M_i$, $\exists M_k \in \mathcal{M}$ such that $M_k(x) = M_i(x)$,

$M_k(c_i) \neq M_k(x)$ and $M_i(c_k) \neq M_i(x)$. Then, to avoid a contradiction, it must be that $\exists \alpha_l \in \mathcal{M} \cup \mathcal{C}$ such that $(\alpha_l, M_k) \in \mathcal{T}$ or $(\alpha_l, c_k) \in \mathcal{T}$, and $(M_i, \alpha_l), (c_i, \alpha_l) \notin \mathcal{T}$. We know that if $\alpha_l \in \mathcal{M}$, and $M_l(x) \neq M_k(x) = M_i(x)$, then it cannot be the case that $(M_l, M_i) \in \mathcal{T}$ and $(M_i, M_l) \notin \mathcal{T}$ since, by Definition 32, $M_i \succeq_{\mathcal{M}} M_l$. Likewise, if $\alpha_l \in \mathcal{C}$ and $M_l(x) \neq M_k(x) = M_i(x)$, then it cannot be the case that $(M_l, M_i) \in \mathcal{T}$ and $(M_i, M_l) \notin \mathcal{T}$ by the same reasoning. Finally, if $\alpha_l \in \mathcal{C}$ and $M_l(x) = M_k(x) = M_i(x)$, then it must be the case that $M_i(c_l) = M_i(x)$ or $M_l(c_i) = M_l(x)$, then $(M_i, c_l) \in \mathcal{T}$ (a contradiction), and we have the contradiction in all cases.

Finally, Example 13 provides a counterexample which shows that majority vote is not satisfied. □

S-preferred argumentative ensembling improves upon augmented, robust, and d-preferred argumentative ensembling by satisfying non-emptiness, counterfactual validity and counterfactual coherence. Majority vote is sacrificed in order to achieve this behaviour. In Section 7.4 we will assess the impact of not guaranteeing majority vote on argumentative ensembling's accuracy, along with other metrics.

Next, we show that under our definition of BAF, c-preferred and s-preferred extensions are equivalent. Therefore, the theoretical analysis of s-preferred argumentative ensembling above also applies to c-preferred.

Theorem 6. *C-preferred argumentative ensembling is equivalent to s-preferred argumentative ensembling.*

Proof. By definition, each c-admissible set is also d-admissible (see Section 2.4). S-admissible sets are d-admissible, and we show in Theorem 4 that d-preferred sets are equivalent to stable sets, therefore, by Property 2 of [24], our s-admissible sets are closed for support relations $\mathcal{S}$, therefore are c-admissible. □

7.3.4 Discussion on Argumentative Ensembling Behaviours

In this section, we consider more formal results of argumentative ensembling describing their behaviours. As stable and c-preferred argumentative ensemblings are respectively equivalent to their d-preferred and s-preferred counterparts, we will only discuss the latter two onwards. Any results will apply to the former two. For the remainder of the section, we assume as given an input x, a set $\mathcal{M}$ of models, a set $\mathcal{C}$ of CEs, a model preference $\preceq_{\mathcal{M}}$ and a corresponding BAF $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$.

Theorem 7. *If* $\forall M_i, M_j \in \mathcal{M}$, $M_i \simeq_{\mathcal{M}} M_j$, *and* $\forall M_k \in \mathcal{M}$ *and* $c_l \in \mathcal{C}$ *where* $M_k(x) = M_l(x)$, $M_k(c_l) \neq M_k(x)$, *then augmented, robust, and s-preferred argumentative ensembling are equivalent.*

Proof. If $\forall M_i, M_j \in \mathcal{M}$, $M_i \simeq_{\mathcal{M}} M_j$, then it can be seen from Definition 32 that $(M_i, M_j), (M_j, M_i) \in \mathcal{T}$ iff $M_i(x) \neq M_j(x)$. Note that, by Definition 28, augmented and robust ensembling are equivalent since $\forall M_k \in \mathcal{M}$ and $c_l \in \mathcal{C}$, where $M_k(x) = M_l(x)$, $M_k(c_l) \neq M_k(c_l)$. Also by the assumptions, it can be seen from Definition 32 that $\forall M_k \in \mathcal{M}$ and $\forall c_l \in \mathcal{C}$, $(M_k, c_l), (c_l, M_k) \in \mathcal{T}$ iff $M_k(c_l) = M_k(x)$ and $M_k(x) \neq M_l(x)$ due to the assumptions in the theorem, meaning any attack is reciprocated and all arguments defend themselves. Then, $\forall M_m, M_n \in \mathcal{M}$ such that $M_m(x) = M_n(x)$, $(M_m, M_n) \notin \mathcal{T}$. Also, any s-preferred extension satisfies counterfactual coherence. Thus, $P^s = \{\{M_o \in \mathcal{M}, c_o \in \mathcal{C} | M_o(x) = 0\}, \{M_p \in \mathcal{M}, c_p \in \mathcal{C} | M_p(x) = 1\}\}$. By Definitions 21, 28 and 33, all forms of ensembling select from the same two sets of models and CEs in the same manner and are thus equivalent. □

Note that this result does not hold for d-preferred argumentative ensembling as it does not satisfy non-emptiness. Specifically, its solution can contain only CEs and no models. Due to the same reason, most theoretical findings in this section do not apply to d-preferred argumentative ensembling. We therefore omit relevant discussion unless otherwise mentioned.

We also provide a number of theoretical results concerning the behaviour of s-preferred argumentative ensembling, first relating to the preferences.

The first result demonstrates how a completely dominant model wrt the preferences will be present in all s-preferred sets.

Proposition 3. *If* $\exists M_i \in \mathcal{M}$ *such that* $\forall M_j \in \mathcal{M}$, $M_i \succ_{\mathcal{M}} M_j$, *then* $\forall P_i^s \in P^s$, $M_i \in P_i^s$.

Proof. If $\forall M_j \in \mathcal{M}$, $M_i \succ_{\mathcal{M}} M_j$ then, by Definition 32, $\mathcal{T}(M_i) = \mathcal{T}(c_i) = \emptyset$ and $\forall M_j \in \mathcal{M}$ where $M_j(x) \neq M_i(x)$, $M_i \in \mathcal{T}(M_j)$. Similarly, $\forall c_k \in \mathcal{C}$ where $M_i(c_k) = M_i(x)$, $M_i \in \mathcal{T}(c_k)$, and so M_i indirectly attacks M_k. Then, $\nexists P_i^s \in P^s$ such that $M_j \in P_i^s$ or $M_k \in P_i^s$ and thus, $\forall P_i^s \in P^s$, $M_i \in P_i^s$. □

We also show how, for any two s-preferred sets, there exists some trade-off between their models wrt the preferences.

Lemma 3. *Given two s-preferred sets* $P_i^s, P_j^s \in P^s$, $\nexists M_i \in (P_i^s \cap \mathcal{M}) \setminus P_j^s$ *such that* $M_i \succ_{\mathcal{M}} M_j$ *for all* $M_j \in P_j^s$.

Meanwhile, in the non-strict case, a model which is not outperformed by any other model wrt the preferences will be present in at least one s-preferred set.

Proposition 4. *If* $\exists M_i \in \mathcal{M}$ *such that* $\forall M_j \in \mathcal{M}$, $M_i \succeq_{\mathcal{M}} M_j$, *then* $\exists P_i^s \in P^s$ *such that* $M_i \in P_i^s$.

Proof. If $\forall M_j \in \mathcal{M}$, $M_i \succeq_{\mathcal{M}} M_j$ then, by Definition 32, $\nexists \alpha_k \in \mathcal{T}(M_i) \cup \mathcal{T}(c_i)$ such that $M_i \notin \mathcal{T}(\alpha_k)$ and $c_i \notin \mathcal{T}(\alpha_k)$. Thus, it must be the case that $\exists P_i^s \in P^s$ such that $M_i \in P_i^s$. □

We now show that if a model is outperformed wrt the preferences by all other models, then the outperformed model cannot exist in an s-preferred set unless it is defended by a more preferred model.

Proposition 5. *For any* $M_i \in \mathcal{M}$, *if* $M_j \succ_{\mathcal{M}} M_i$ *for all* $M_j \in \mathcal{M} \setminus \{M_i\}$ *and* $\exists P_i^s \in P^s$ *such that* $M_i \in P_i^s$, *then,* $\forall M_k \in \mathcal{M}$ *where* $M_k(x) \neq M_i(x)$, $\exists M_l \in P_i^s$ *such that* $M_l \succeq_{\mathcal{M}} M_k$.

Proof. By Definition 32, $M_k \in \mathcal{T}(M_i)$ and $\{M_m \in \mathcal{M} | M_i \in \mathcal{T}(M_m) \vee c_i \in \mathcal{T}(M_m)\} = \emptyset$. Then, given that $\exists P_i^s \in P^s$ such that $M_i \in P_i^s$, we know that, $\forall M_k \in \mathcal{T}(M_i)$ $\exists M_l \in X$ where $M_l(x) = M_i(x)$ and $M_l \in \mathcal{T}(M_k)$. Thus, by Definition 32, $M_l \succeq_{\mathcal{M}} M_k \succ_{\mathcal{M}} M_i$. □

Propositions 4 and 5 also apply to d-preferred argumentative ensembling with the same proof.

We also consider the behaviour of argumentative ensembling wrt the selected CEs, demonstrating that those from disagreeing models are guaranteed not to be included in any s-preferred set.

Proposition 6. *Any s-preferred set $P_i^s \in P^s$ is such that $\nexists c_i, c_j \in P_i^s \cap \mathcal{C}$ where $M_i(x) \neq M_j(x)$.*

Proof. Let us prove by contradiction, assuming that $\exists c_i, c_j \in P_i^s \cap \mathcal{C}$. Counterfactual coherence (Theorem 5) requires that $M_i, M_j \in P_i^s$. However, by Definition 32, $M_i(x) \neq M_j(x)$ requires that $M_i \in \mathcal{T}(M_j)$ or $M_j \in \mathcal{T}(M_i)$, and so we have the contradiction. □

7.3.5 Additional Discussion on Abstract vs. Bipolar Argumentation

In this section, we discuss the similarities and differences between our proposed argumentative ensembling approach instantiated using bipolar argumentation and an alternative approach instantiated using abstract argumentation. In our previous definition of a CE (Section 2.4), we required that $M(c) \neq M(x)$, i.e., the CE for a model prediction is *valid* on the model. Under this assumption, we obtain the following equivalence result:

Theorem 8. *Consider input x, set $\mathcal{M}$ of models, set $\mathcal{C}$ of CEs such that $\forall c_i \in \mathcal{C}$, $\forall M_i \in \mathcal{M}$, $M_i(c_i) \neq M_i(x)$. Let $\preceq_{\mathcal{M}}$ be the model preferences, and $\langle \mathcal{A}, \mathcal{T}, \mathcal{S} \rangle$ be the corresponding BAF with P^s the set of all s-preferred extensions. Let $\langle \mathcal{A}', \mathcal{T}' \rangle$ be an AAF with:*

- $\mathcal{A}' = \mathcal{M} \times \mathcal{C}$;

- $\mathcal{T}' \subseteq \mathcal{A}' \times \mathcal{A}'$ *where* $\forall (M_i, c_i), (M_j, c_j) \in \mathcal{A}', ((M_i, c_i), (M_j, c_j)) \in \mathcal{T}'$ *iff* $M_i \succeq_{\mathcal{M}} M_j$ *and:*
  - $M_i(x) \neq M_j(x)$*; or*
  - $M_i(c_j) = M_i(x)$*; or*
  - $M_j(c_i) = M_j(x)$.

Let P *be the set of all preferred extensions of* $\langle \mathcal{A}', \mathcal{T}' \rangle$*. Then,*

- $P^s = \{f(P_i) | P_i \in P\}$
- $P = \{f^{-1}(P_i^s) | P_i^s \in P^s\}$

where f *is a function such that* $f(P_i) = \{M_i, c_i | (M_i, c_i) \in P_i\}$

and f^{-1} *is such that* $f^{-1}(P_i^s) = \{(M_i, c_j) \in \mathcal{M} \times \mathcal{C} | M_i, c_j \in P_i^s, i = j\}$.

Proof. First, we prove $\forall P_i^s \in P^s$, $f^{-1}(P_i^s) \in P$. We show $f^{-1}(P_i^s)$ is conflict-free by contradiction. Assume $f^{-1}(P_i^s)$ is not conflict-free, then $\exists (M_i, c_i), (M_j, c_j) \in f^{-1}(P_i^s)$, such that $((M_i, c_i), (M_j, c_j)) \in \mathcal{T}'$. Then, by the AAF definition in Theorem 8, it must be that $M_i \succeq_{\mathcal{M}} M_j$ and, either $M_i(x) \neq M_j(x)$, or $M_i(c_j) = M_i(x)$, or $M_j(c_i) = M_j(x)$. By Definition 32, in the first case, $(M_i, M_j) \in \mathcal{T}$; in the second and third cases, $(M_j, c_j) \in \mathcal{T}$. Then, P_i^s cannot be conflict-free, but it is an s-preferred extension, so we have the contradiction. We now prove $f^{-1}(P_i^s)$ defends all its elements by contradiction. Assume $f^{-1}(P_i^s)$ does not defend all its elements, then, $\exists (M_i, c_i) \in f^{-1}(P_i^s)$, such that $\exists (M_k, c_k) \in \mathcal{A}' \setminus f^{-1}(P_i^s)$ where $((M_k, c_k), (M_i, c_i)) \in \mathcal{T}'$, and $\nexists (M_j, c_j) \in f^{-1}(P_i^s)$ s.t. $((M_j, c_j), (M_k, c_k)) \in \mathcal{T}'$. Mapping this back to the BAF by Definition 32, we know $M_i, c_i \in P_i^s$, $M_k, c_k \in \mathcal{A} \setminus P_i^s$, and either $(M_k,\ M_i) \in \mathcal{T}$, $(M_k, c_i) \in \mathcal{T}$, or $(c_k, M_i) \in \mathcal{T}$, and $\nexists M_j, c_j \in P_i^s$ s.t. $(M_j,\ M_k) \in \mathcal{T}$, $(M_j, c_k) \in \mathcal{T}$, or $(c_j, M_k) \in \mathcal{T}$. This cannot be the case since P_i^s is safe and so we have the contradiction. Therefore, $f^{-1}(P_i^s)$ is conflict-free and it defends all its elements in AAF, i.e., it is a preferred extension in the AAF.

Next, we prove $\forall P_i \in P, f(P_i) \in P^s$. We first show $f(P_i)$ is safe by contradiction. Assume $f(P_i)$ is not safe, then $\exists \alpha_1 \in \mathcal{A}, f(P_i)$ set-attacks α_1,

and either $\alpha_1 \in f(P_i)$, or $f(P_i)$ set-supports α_1. For the former case, we know P_i is conflict-free in the AAF. Mapping the conditions of attacks in the AAF (Theorem 8 above) to the conditions of attacks in BAF (Definition 32), it means $\nexists M_i, M_j \in f(P_i)$ s.t. $(M_i, M_j) \in \mathcal{T}$. Also, $\nexists M_i, c_j \in f(P_i)$ s.t. $(M_i, c_j) \in \mathcal{T}$ or $(c_j, M_i) \in \mathcal{T}$. Therefore, we have a contradiction in the former case. There is also a contradiction in the second case, because by Definition 32, support relations can only exist between a model and its CE. Therefore, $f(P_i)$ is safe. Next, we prove $f(P_i)$ defends all its elements. We know that P_i defends all its elements in the AAF because it is a preferred extension. This means $\forall (M_k, c_k) \in \mathcal{A} \setminus P_i$, if (M_k, c_k) attacks an element in P_i, then $\exists (M_i, c_i) \in P_i$ s.t. $((M_i, c_i), (M_k, c_k)) \in \mathcal{T}'$. Translating the conditions of attacks in the AAF to the conditions of attacks in the BAF, this indicates that in the BAF, one of (M_k, c_k) must be directly attacked and the other one is supported attacked, because they support each other. Therefore, any attacker of $f(P_i)$ is also attacked by $f(P_i)$ in the BAF. Therefore, $f(P_i)$ is an s-preferred extension in the BAF because it defends all its elements is safe.

□

Example 16. *The AAF corresponding to input, models and CEs as in Example 12 is* $\langle \mathcal{A}', \mathcal{T}' \rangle$ *with (see Fig. 7.3):* $\mathcal{A}' = \{(M_1, c_1), (M_2, c_2), (M_3, c_3), (M_4, c_4), (M_5, c_5)\}$*;* $\mathcal{T}' = \{((M_2, c_2), (M_1, c_1)), ((M_2, c_2), (M_3, c_3)), ((M_2, c_2), (M_4, c_4)), ((M_2, c_2), (M_5, c_5)), ((M_3, c_3), (M_4, c_4)), ((M_4, c_4), (M_1, c_1)), ((M_5, c_5), (M_1, c_1)), ((M_5, c_5), (M_2, c_2)), ((M_5, c_5), (M_3, c_3))\}$*, this leads to* $P = \{\{(M_2, c_2)\}, \{(M_4, c_4), (M_5, c_5)\}\}$*, and the cardinality-maximal preferred extension, which becomes the solution to the RAE problem, is* $\{(M_4, c_4), (M_5, c_5)\}$*, the aggregated prediction is* $M_4(x) = M_5(x) = 1$.

Example 16 (Figure 7.3), the AAF equivalent of Example 13 (Figure 7.1), demonstrates the intuitions behind the equivalence between BAF and AAF.

In practice, however, CEs may not always be valid on the model they are generated for. This is especially the case for gradient-based methods because no validity guarantees are given [142, 56]. When this happens, grouping each model and the corresponding CE can become problematic because of the invalidity. Then, the BAF provides a more expressive way to represent

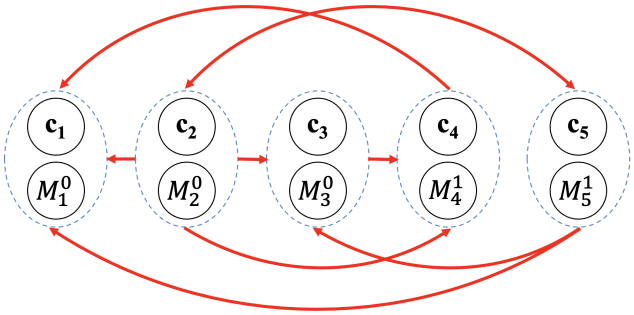


Figure 7.3: AAF for Example 16 where: models' predictions for the input x are given as superscripts, e.g., $M_1(x) = 0$ and $M_4(x) = 1$; standard (reciprocal) attacks are represented by single-headed (double-headed, respectively) red arrows.

the conflicts than the AAF. For example, the support relations between a model and its CE can be removed or changed into attack relations. We leave the investigation of such scenarios to future work.

7.4 Empirical Evaluation

We now examine the effectiveness of our approach using three real-world datasets. Specifically, we empirically evaluate the extent to which each of the ensembling methods introduced in Section 7.2.2 and Section 7.3 satisfy the desirable properties defined in Section 7.2.1. We instantiate four variations of argumentative ensembling with different model preferences over two types of model properties $\mathcal{P}$ and demonstrate the usefulness of incorporating model preferences. We also show that the task performance (accuracy) by our methods remain comparable to the baselines. Finally, we report the compute time for solving BAFs against increasing model sizes in the problem setting.

7.4.1 Experiment Setup

7.4.1.1 Datasets and Models

We apply all ensembling methods on three datasets in the legal and financial contexts: loan approval (heloc) [46], recidivism prediction (compas) [4], and credit risk (credit) [38]. Due to neural networks' sensitivity to randomness at training time, they suffer severely from MM and are frequently targeted when investigating this research topic. Therefore, we focus on neural networks for our experiments. Note that the problem setting focuses on the conflicts between classifiers' predictions and CE validities and is agnostic to the types of models used. Our method is also model agnostic and can be applied to model sets consisting of even different types of classifiers.

	(15, 15)	(20, 15)	(20, 20)	(30, 20)	(30, 25)
heloc	.730±.015	.738±.009	.741±.008	.734±.007	.736±.009
compas	.844±.013	.841±.013	.845±.011	.840±.012	.845±.012
credit	.709±.028	.699±.007	.705±.018	.697±.011	.696±.021

Table 7.3: Accuracies of trained neural networks of different sizes on each dataset

For each dataset, we train-test 150 classifiers with five different hidden layer sizes using 80% of the dataset (this 80% is train-test split for training each model). The neural networks have two hidden layers of sizes {(15, 15), (20, 15), (20, 20), (30, 20), (30, 25)}. The five different sizes are for obtaining five levels of model simplicity, which serves as a toy property evaluation to demonstrate the usefulness of supporting model preferences in our framework. Their 5-fold cross-validation accuracies on the training set are reported in Table 7.3. We observe no significant changes in test accuracies across the different model sizes, as the accuracy variations fall within the range of standard deviation. The classifiers are trained using different random seeds for parameter initialisation and different train-test splits (within the train-test 80% of the dataset), forming a pool of possible models under MM from which we sample multiple sets $\mathcal{M}$ of models to which we apply our ensembling methods. We use the remaining 20% of each dataset as test inputs for the ensembling methods (limited to 500 inputs test

set if larger).

7.4.1.2 Experimental Procedure

We instantiate the RAE problem by randomly sampling sets $\mathcal{M}$ with 10, 20 or 30 models from the model pool of the 150 trained classifiers. We then feed each input to the models to receive their predicted labels and generate one CE from each model. For each size $|\mathcal{M}| = (10, 20, 30)$, we repeat the above process five times and record the average results.

To instantiate RAE at scale, we adopt the Nearest Neighbour CEs (NNCE)[21] method. Because the RAE framework requires valid CEs but remains agnostic to the specific generation technique, NNCE serves as an ideal choice: it inherently guarantees the discovery of a valid CE for every input under any model, while empirically demonstrating high computational efficiency, close proximity, and strong plausibility through adherence to the data manifold [78]. The theoretical validity of this approach relies on two highly realistic assumptions. First, we assume access to the training dataset, which is a practical expectation given that CE computation is typically performed by the model provider. Second, we assume the underlying classification models are sufficiently well-optimised such that they do not trivially predict the same class label for all data points. Provided these conditions hold, for any given input, the nearest training data point (as measured by a specified distance metric) that is predicted to belong to a different class can always be reliably returned as a valid CE. In the rare case where a CE cannot be found for a model, that model can be omitted from the model set.

As concerns model preferences, we focus on two measurable model properties: accuracy of the trained classifiers over the (20%) test inputs and model structure simplicity. For accuracy, we manually made sure that every model in $\mathcal{M}$ has a different test accuracy and thus no pairs of models can make identical predictions to every test input. For the latter, we assign the models, from the most complex to the simplest (depending on the number of neurons in the hidden layers), scores of $\{0, 0.25, 0.5, 0.75, 1\}$ such that higher values imply simpler models. Note that multiple models in $\mathcal{M}$ may have the same simplicity scores as we adopt only five different model structures to obtain

150 neural networks for each dataset. Using the two metrics, we specify four types of model preferences to be used in our method, discussed more later in Section 7.4.1.4.

7.4.1.3 Evaluation Metrics

Each ensembling method is evaluated against the following metrics: aggregated prediction accuracy over the test set, average model simplicity in the ensemble, and the percentage of test points for which each property is satisfied. We also report the average test set accuracies of the models in $\mathcal{M}$. Finally, we report the average computation time needed for argumentative ensembling when the BAF is built to show the scalability of the method.

7.4.1.4 Ensembling Methods

We use augmented ensembling S^n and robust ensembling S^v as baselines. We experiment with both d-preferred and s-preferred argumentative ensembling ($S^{a,d}$ and $S^{a,s}$ respectively). For each type of argumentative ensembling, we use four variations with different preferences: $S^{a,\sigma}$ ($\mathcal{P} = \emptyset$), $S^{a,\sigma}$-A ($\mathcal{P} = \{accuracy\}$), $S^{a,\sigma}$-S ($\mathcal{P} = \{simplicity\}$) and $S^{a,\sigma}$-AS ($\mathcal{P} = \{accuracy, simplicity\}$, with $accuracy \simeq_{\mathcal{P}} simplicity$).

In our implementation of argumentative ensembling, we add simple mechanisms for handling edge cases. Firstly, when there are multiple cardinality-maximum extensions, we return the one that gives the same prediction as majority vote. If there are still multiple sets, we then randomly choose one. Additionally, for d-preferred argumentative ensembling, when there are multiple cardinality-maximum extensions, where possible, we select from the ones that contain at least one model and one CE. The implementations are adapted from [66, 70].

Dataset	Method	$\|\mathcal{M}\| = 10$		$\|\mathcal{M}\| = 20$		$\|\mathcal{M}\| = 30$	
		Acc.	**Simp.**	**Acc.**	**Simp.**	**Acc.**	**Simp.**
heloc	avg	.710	.485	.710	.500	.710	.510
	S^n	.714	.503	.716	.508	.716	.508
	S^v	.714	.503	.716	.508	.716	.508
	$S^{a,d}$	.714	.507	.709	.505	.710	.501
	$S^{a,d} - A$	.720	.478	.726	.547	.728	.507
	$S^{a,d} - S$	.710	.606	.712	.542	.711	.520
	$S^{a,d} - AS$	.716	.529	.711	.520	.709	.506
	$S^{a,s}$	.714	.511	.709	.500	.710	.502
	$S^{a,s} - A$	.720	.478	.726	.547	.728	.507
	$S^{a,s} - S$	.710	.607	.712	.541	.711	.520
	$S^{a,s} - AS$	.716	.529	.711	.520	.709	.507
compas	avg	.855	.570	.855	.558	.855	.533
	S^n	.856	.495	.858	.490	.859	.493
	S^v	.856	.495	.858	.490	.859	.493
	$S^{a,d}$	.860	.492	.858	.479	.858	.480
	$S^{a,d} - A$	.863	.436	.866	.619	.869	.583
	$S^{a,d} - S$	.861	.699	.857	.586	.856	.551
	$S^{a,d} - AS$	.864	.567	.856	.519	.856	.518
	$S^{a,s}$	.860	.492	.858	.476	.858	.479
	$S^{a,s} - A$	.863	.436	.866	.619	.869	.583
	$S^{a,s} - S$	.861	.698	.857	.587	.856	.545
	$S^{a,s} - AS$	.864	.585	.856	.529	.856	.532
credit	avg	.660	.505	.660	.510	.660	.520
	S^n	.696	.509	.705	.533	.709	.545
	S^v	.696	.509	.705	.533	.709	.545
	$S^{a,d}$	.685	.525	.683	.556	.682	.563
	$S^{a,d} - A$	.700	.457	.706	.522	.710	.554
	$S^{a,d} - S$	.670	.610	.686	.564	.685	.568
	$S^{a,d} - AS$	.683	.537	.685	.559	.684	.566
	$S^{a,s}$	.685	.522	.683	.555	.682	.563
	$S^{a,s} - A$	.700	.437	.706	.521	.710	.552
	$S^{a,s} - S$	.670	.611	.686	.563	.685	.568
	$S^{a,s} - AS$	.683	.535	.685	.559	.684	.566

Table 7.4: Average accuracy and simplicity of the models in the ensembling result by each method. "avg" rows report these scores for the models in $\mathcal{M}$.

7.4.2 Results on the Usefulness of Preferences

We first report the accuracy and average model simplicity of the ensembles by each method in Table 7.4 (note that these are obtained on subsets different than the ones we used to obtain cross-validation accuracies in Table 7.3). With test accuracy specified as model preference, $S^{a,\sigma}$-A shows the best accuracy in all experiments (across d- and s-preferred). This validates

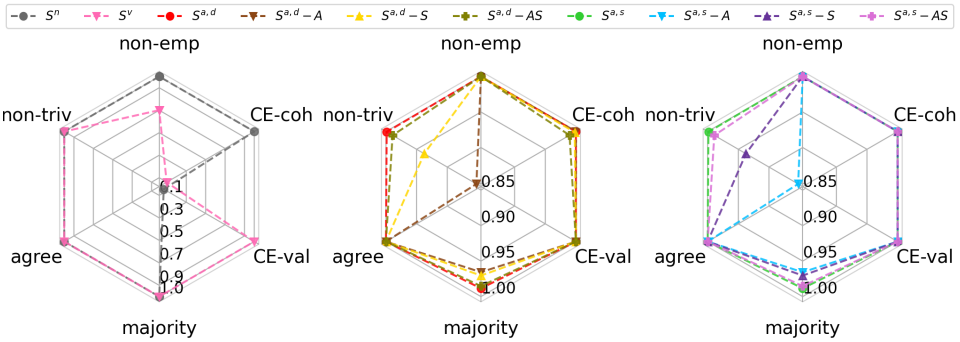
Proposition 3, because, assuming that the accuracy for every model in $\mathcal{M}$ is different, for $S^{a,\sigma}$-A, there exists a model in $\mathcal{M}$ that is the most preferred and is included in the ensemble. Similarly, $S^{a,\sigma}$-S shows the best simp. scores in all experiments. However, since simplicity scores are not unique for each model, usually a single most preferred model does not exist, therefore an optimal simp. evaluation is not guaranteed. When specifying both properties as model preferences ($S^{a,\sigma}$-AS), at least one of the two metrics is improved compared with $S^{a,\sigma}$.

It can be concluded that any ensembling method improves prediction accuracy when compared with the average single-model accuracies, as better evaluations can almost always be observed. While $S^{a,\sigma} - A$ configurations always give the highest accuracy, the majority vote accuracy remains competitive. For four out of nine sets of experiments (heloc $|\mathcal{M}| = 10$, compas $|\mathcal{M}| = 10, 20, 30$), argumentative ensembling accuracies when no preferences are specified are higher or comparable with S^n and S^v, while they are marginally lower in other cases.

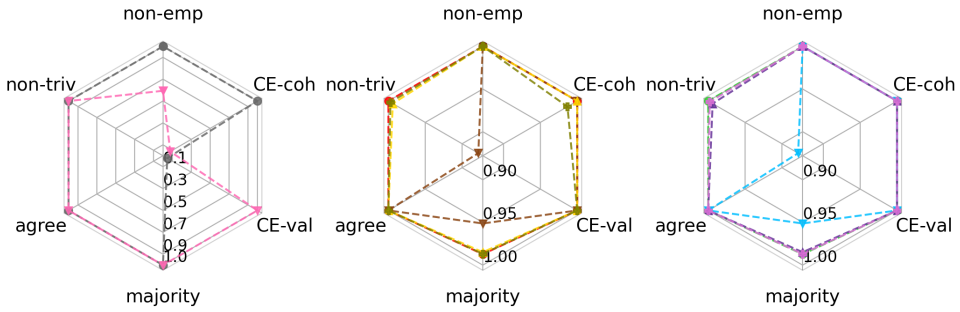
7.4.3 Results on Desirable Properties Satisfaction

Property satisfaction rates for the three datasets are visualised in Figures 7.4 to 7.6. As characterised by our earlier theoretical analysis, both baselines, S^n and S^v, have significant limitations in terms of satisfying important properties (see Table 7.1) for them to be used in practice. S^n demonstrates near-0 counterfactual validity, indicating that for most of the time, there are some CEs in the ensemble solution which are not valid on other models. S^v produces valid CEs over models in the ensemble (100% counterfactual validity), but they do not always exist, as shown by the low non-emptiness scores. In fact, for up to 70% of test inputs, robust ensembling does not find any CEs ($S^v \cap \mathcal{C} = \emptyset$), confirming its violation of non-emptiness. As $|\mathcal{M}|$ increases, S^v would require finding CEs which are valid for more models, and the number of CEs found would drop.

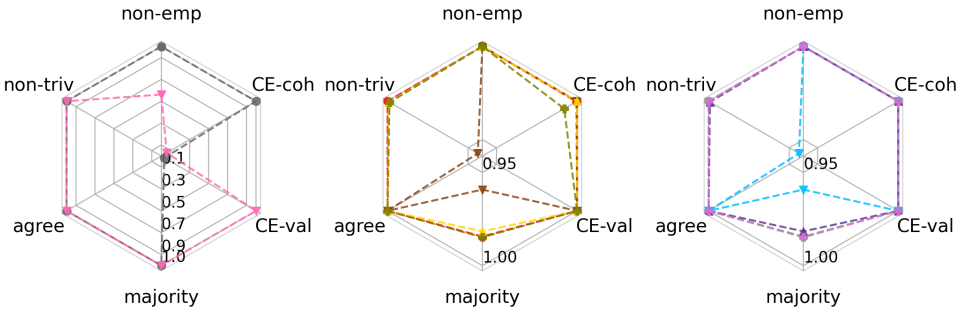
Argumentative ensembling, on the other hand, can always return non-empty ensembles while guaranteeing the validity of CEs. We observe that although theoretically d-preferred argumentative ensembling only satisfies



(a) heloc dataset, $|\mathcal{M}| = 10$



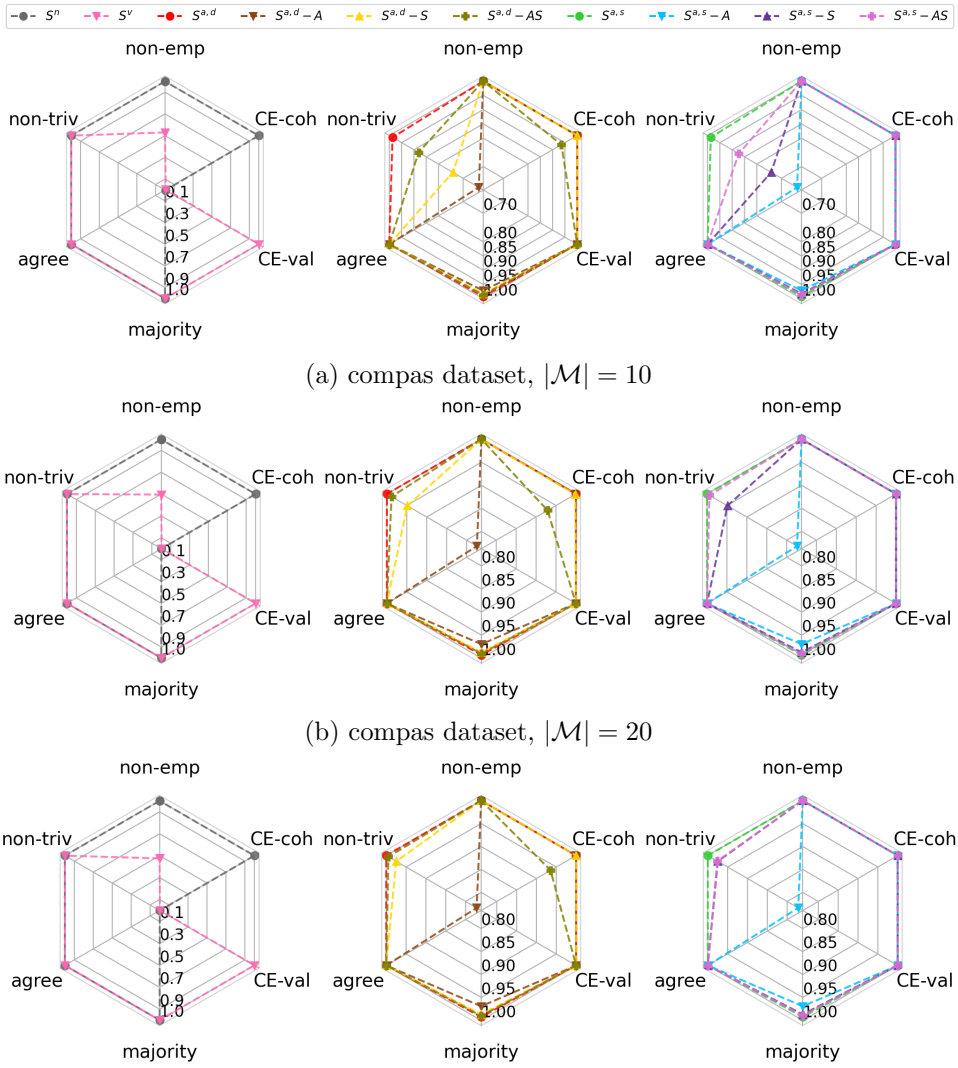
(b) heloc dataset, $|\mathcal{M}| = 20$



(c) heloc dataset, $|\mathcal{M}| = 30$

Figure 7.4: Satisfaction of desirable properties on heloc dataset. The three subplots show results respectively for the two baselines, d-preferred argumentative ensembling, and s-preferred argumentative ensembling.

model agreement and counterfactual validity, it practically always satisfies non-emptiness, and also achieves counterfactual coherence for over 95% test points. Contrastively, all $S^{a,s}$ instantiations always satisfies model these four properties. Both d- and s-preferred argumentative ensembling satisfy non-triviality for most of the time, and more frequently so with larger $|\mathcal{M}|$. They do not satisfy majority vote, although this has little effect in



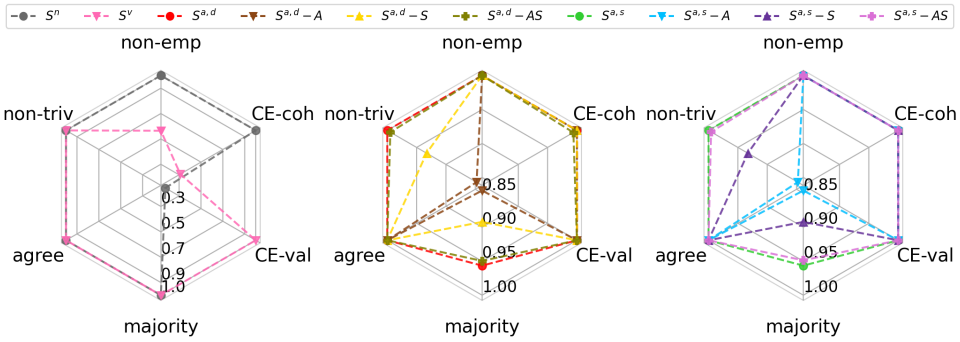
(a) compas dataset, $|\mathcal{M}| = 10$

(b) compas dataset, $|\mathcal{M}| = 20$

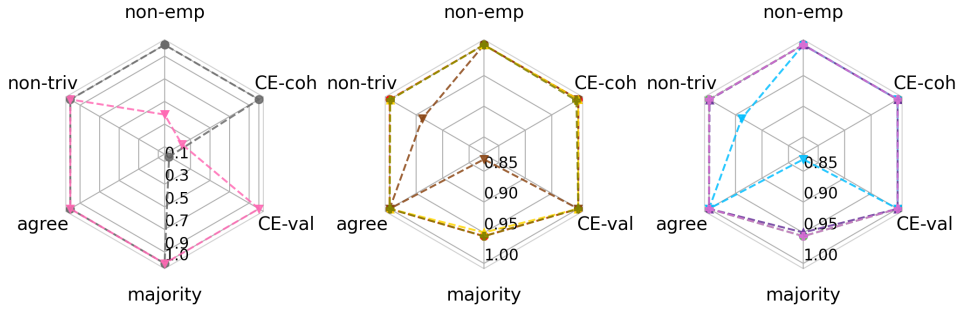
(c) compas dataset, $|\mathcal{M}| = 30$

Figure 7.5: Satisfaction of desirable properties on compas dataset. The three subplots show results respectively for the two baselines, d-preferred argumentative ensembling, and s-preferred argumentative ensembling.

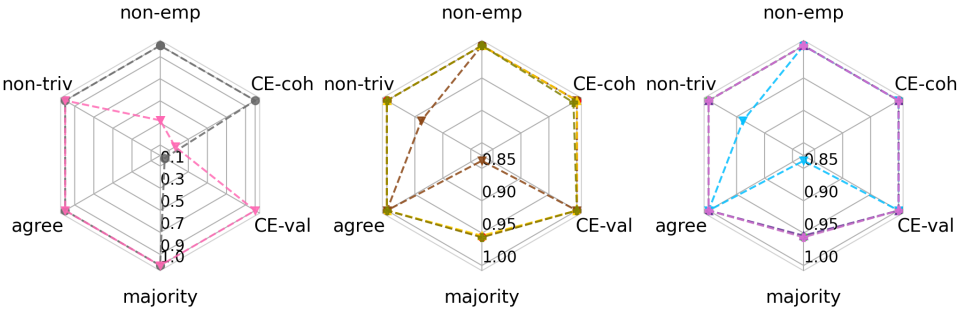
terms of task accuracy. The four specific instantiations with different model preferences of $S^{a,d}$ and $S^{a,s}$ perform respectively very similarly to each other. $S^{a,\sigma}-A$ tends to have the lowest non-triviality and majority vote satisfaction, because if there exists a single most accurate model, it will be included in the ensemble (Proposition 3), which may behave more differently than the



(a) credit dataset, $|\mathcal{M}| = 10$



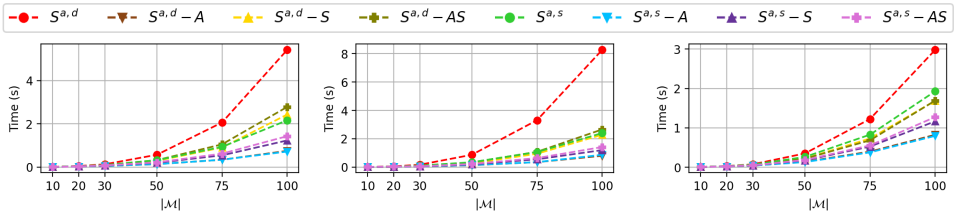
(b) credit dataset, $|\mathcal{M}| = 20$



(c) credit dataset, $|\mathcal{M}| = 30$

Figure 7.6: Satisfaction of desirable properties on credit dataset. The three subplots show results respectively for the two baselines, d-preferred argumentative ensembling, and s-preferred argumentative ensembling.

other models. This is most obvious for the credit dataset.



(a) heloc dataset (b) compas dataset (c) credit dataset

Figure 7.7: Average computation time of solving argumentative ensembling for each input.

7.4.4 Results on Scalability

We further experiment with $|\mathcal{M}| = 50, 75, 100$ to illustrate the computation time required to solve the BAF in argumentative ensembling, although these numbers of models are less likely to occur in real-life applications. The computation time results are reported in Figure 7.7. It can be seen that $S^{a,s}$ instantiations are faster to solve than their d-preferred counterparts since they are more restrictive and thus have smaller search spaces. When experimenting with less than 30 models, the computation times are close to 0 for three datasets. Time differences between datasets for the same $|\mathcal{M}|$, which are only governed by the number of conflicts in model predictions and CE validities, become more obvious as $|\mathcal{M}|$ grows. Note that another part of the computation time of argumentative ensembling not reported in Figure 7.7 is the time required to construct the BAF itself. This involves testing every CE on every model in $\mathcal{M}$, and is $O(|\mathcal{M}|^2)$ to the inference time of each model.

7.5 Summary, Limitations and Discussion

Summary. In this chapter, we have presented a formal study of the problem of providing recourse under MM. We defined several properties which are desirable in methods for solving this problem, highlighting deficiencies in extending conservatively the standard naive ensembling used for MM without recourse. We have then introduced argumentative ensembling, a novel

method for providing recourse under MM, which leverages computational argumentation to incorporate robustness guarantees and user preferences over models. We show, by means of a theoretical analysis, that argumentative ensembling hosts advantages over other methods, notably in non-emptiness of solutions and validity of CEs, notwithstanding its ability to handle user preferences. This is, however, achieved by sacrificing the satisfaction of the property of majority vote. Our *empirical* results, however, demonstrate that argumentative ensembling always finds valid CEs without compromising prediction accuracy, and shows the usefulness of specifying preferences over models.

Limitations. This work comes with some limitations. First, we assumed the availability of a valid CE for each model to instantiate the RAE problem. However, in practice, methods for computing CEs, especially the heuristic-based methods using gradient descent, may not always find valid CEs, limiting the range of CE generation methods to be used in our framework. It would be interesting to account for invalid CEs in the framework and leverage the power of argumentation to resolve inconsistencies and allow for more flexibility. Second, the desirable properties we proposed for RAE solutions are fundamental, binary properties. Additional metrics taking continuous values could potentially provide more fine-grained evaluations.

Discussion. Chapter 7 formalises the recourse under MM problem and proposes a solution based on computational argumentation. This opens up several interesting directions for future work. First, on the argumentation side, it would be interesting to examine whether considering attacks to or from *sets* of arguments, rather than single arguments, as in [105, 48, 41, 36], may help in MM. Further, extended AFs [97] and value-based AFs [14] may provide useful alternative ways to account for preferences. We would also like to exploit the explanatory potential of argumentation to support explainable ensembling, e.g., using sub-graphs as in [44, 150]. Moreover, in order to support experiments with a high number of models (beyond the 30 we considered), large-scale argumentation solvers would be highly desirable.

In terms of types of CEs used, future research could also investigate how different types of CEs (e.g., plausible or not plausible, robust or not robust)

interact with the RAE property satisfaction in our context. In this thesis, we used the NNCE method, which provides a balance between plausibility and proximity. On the contrary, employing a method optimised for high robustness against model changes, for instance, presents a critical trade-off. While it may simplify the argumentation framework by being valid across most models, it could also trivialise our approach of using validity patterns to obtain model subgroups. Another promising area is to extend the framework to accommodate multiple CEs for a single input (diversity) per model under MM, providing more flexible recourse options.

Chapter 8

Conclusion

In this thesis, we systematically studied several forms of CE robustness, characterising their problem settings and proposing new methods to address them. Finally, we conclude the thesis by revisiting the thesis contributions (Section 8.1) and discussing future work (Section 8.2).

8.1 Summary of Contribution

Referring to Section 1.2, we originally set out four objectives for this thesis, namely to **characterise**, **evaluate**, **guarantee**, and **integrate** various robustness scenarios.

For Objective 1, we formally describe in Chapter 4 the problem of CE robustness against a novel type of model changes, Δ-robustness. In Chapter 7, we provide the first characterisation of the CE robustness against the model multiplicity problem, also identifying the novel desirable properties which the solution to this new problem should satisfy.

For Objective 2, these same two chapters (4 and 7) also propose new methods to quantitatively evaluate robustness. In Chapter 4, we develop a method based on mixed integer programming optimisation for certifying whether Δ-robustness is satisfied by a CE. Chapter 7 proposes new evaluation metrics

for robustness against MM following the introduction of relevant desirable properties.

Objective 3, proposing new methods for addressing CE robustness with provable guarantees, is the core target of this thesis and has been achieved by every technical section on different robustness instantiations. The new CE generation methods introduced in Chapters 4 and 5 all give provable robustness against model changes. This is enabled by novel optimisation formulations via mixed integer programming. LAPACE in Chapter 6 provides a way to provide perfect robustness against input changes. This is achieved by the specific design of our new versatile generative model, L-GMVAE. Chapter 7 proposes a new ensembling method based on computational argumentation, which satisfies the core desirable properties which we set out for recourse under MM problem.

Objective 4 aims to further improve CEs' quality by incorporating the robustness considerations with existing desiderata. This is also addressed by Chapters 4 and 5, which combine robustness against model changes with validity, proximity, and plausibility, and Chapter 6, which addresses two forms of robustness (input changes and model changes) together with validity, proximity, plausibility, diversity, and actionability.

In summary, this thesis advances the study of robust CEs by contributing novel formalisations, evaluation frameworks, and algorithms. These contributions collectively enhance the reliability and trustworthiness of CEs when used for recourse.

8.2 Future Work

Finally, we outline avenues for future research that build upon the contributions of this thesis. The technical chapters each present opportunities for direct extension, aimed at further developing high-quality, robust CE methods for algorithmic recourse. These are discussed in detail at the end of each technical chapter (Sections 4.7, 5.5, 6.4, 7.5). Beyond this primary application, the framework for characterising CE properties developed herein

is valuable for informing the use of CEs for other purposes, such as data augmentation and model evaluation. We also discuss the important directions for CEs to be applied more broadly in real-world scenarios.

8.2.1 Counterfactual Explanations More Generally

The previous sections have discussed future research that directly extends the technical contributions of this thesis, largely focusing on generating robust CEs for algorithmic recourse. However, the utility of CEs can extend beyond this application. Because they reveal the minimum changes needed to flip a prediction, CEs serve as a fundamental tool for probing an ML model's local decision boundary in any data modality. This is also useful for a range of tasks like model debugging [1], outlier detection [134], and data augmentation [45].

An important insight for future work is that the desirable properties of CEs, such as plausibility, diversity, and robustness, are not universal but are highly dependent on the specific application. For example, for model debugging, the primary goal for the CEs is to be faithful to the classifier's true decision boundary. Therefore, plausibility and robustness might be less relevant. Diversity, however, would be valuable for finding multiple such points. For data augmentation, on the other hand, both plausibility and robustness are important to generate new data points that are both realistic and can improve downstream model performance. Existing works using CEs for such purposes often ignore the distinction between different types of CEs, as characterised by various properties [78]. Therefore, a promising research agenda could involve systematically characterising and optimising CE properties for these different use cases. The technical contributions in this thesis offer a rich methodology regarding CEs' robustness and the combination of multiple properties.

More generally, the idea of counterfactual reasoning can be valuable for ML model evaluations, going beyond the static accuracies on some test sets or benchmarks [107]. For example, we might have some high-level aspects aligning with our human knowledge or values, which we want to test the model behaviours against [126]. This is also a common objective

for the recent stream of AI alignment research [9]. Two distinct evaluation paradigms emerge. The first uses CE methods, like those in this thesis, to discover the minimal changes that alter a model's prediction; one can then assess if these discovered changes align with domain knowledge. A second paradigm involves testing a model's response to a pre-defined set of meaningful perturbations. For example, as demonstrated in recent work on large language models, inputs can be perturbed along human concepts like relevance and appropriateness to evaluate the model's sensitivity to certain aspects [69]. Further, information from these local explanations can be combined at the level of group inputs or a dataset to form global insights of a model. The desirable properties of CEs explored throughout this thesis can help inform these advanced evaluation pipelines.

8.2.2 Counterfactual Explanations In The Wild

A primary challenge for the research field of CEs is translating these explanations from academic contexts into reliable tools for real-world applications at scale [78]. For example, CEs, when used for recourse, can be a helpful product in critical sectors like healthcare and finance, and in high-stakes decision-making scenarios. To bridge this gap, CEs must be characterised more comprehensively, a goal to which the robustness properties studied in this thesis meaningfully contribute. A limitation of current research, including this thesis, is that the recourse task is instantiated only on a dataset and a trained ML classifier, possibly with some user inputs regarding actionability constraints. This setup may not capture complex real-world risks, such as those embedded in the data's causal structure. Therefore, a crucial next step is to systematically incorporate richer information sources, like structural causal models, to enable a deeper level of reasoning [32].

Furthermore, scalability remains a key barrier to the industrialisation of CEs. Many current approaches rely on per-instance optimisation, which may not be sufficiently fast for large-scale systems. The development of methods with amortised inference, such as our proposed LAPACE, is therefore valuable. These approaches shift the computational burden from a slow, on-demand generation to a one-off training phase, enabling the efficient

delivery of recourse. Beyond this, more global types of CEs, which typically find a counterfactual region for a group of inputs (such as [122, 15]), could offer richer information as explanations while being more scalable at inference time. Characterising the desirable properties in these global CE methods, specifically having controlled trade-offs between properties, remains a challenge.

While quantitative metrics are essential for technical validation, the ultimate measure of CEs' values is their utility to an end-user. Therefore, a critical avenue for future work will involve conducting rigorous user studies to understand the human-computer interaction aspects of recourse [79]. Existing research has found that reasoning with counterfactual simulation leads to stronger causal reasoning in human cognition[53]. In XAI, CEs help humans predict an ML model's future predictions while giving users trust and satisfaction [146]. Following methods that allow a more comprehensive characterisation of CE properties (such as those contributions in this thesis), further research is needed to explore how users perceive and act upon different types of CEs, particularly how they navigate the inherent trade-offs between properties. Moving beyond the static quantitative measures to evaluate the real-world impact on user trust, understanding, and successful outcomes is essential for the future of the CEs research field.

Finally, the principles of CEs can be extended to more complex data modalities, which remains a largely open research area. While significant initial works have been done on graphs [119, 151], time series [6, 145], planning scenarios [82, 54], images [55, 140], natural language [149, 69], many of them have focused primarily on achieving proximity and plausibility. Across these diverse domains, important areas for future work should focus on systematically translating the desirable properties defined for tabular data, such as diversity and robustness, possibly defining new desiderata, and adapting CE methods to address a more holistic set of properties.

Bibliography

[1] Abubakar Abid, Mert Yüksekgönül, and James Zou. Meaningfully debugging model mistakes using conceptual counterfactual explanations. In *International Conference on Machine Learning, ICML 2022*, volume 162, pages 66–88, 2022.

[2] AIHLEG. European commission, ethics guidelines for trustworthy ai. *COM*, 2019.

[3] Edgar Anderson. The species problem in iris. *Annals of the Missouri Botanical Garden*, 23(3):457–509, 1936.

[4] Surya Mattu Julia Angwin, Jeff Larson, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks, 2016.

[5] André Artelt, Valerie Vaquet, Riza Velioglu, Fabian Hinder, Johannes Brinkrolf, Malte Schilling, and Barbara Hammer. Evaluating robustness of counterfactual explanations. In *IEEE SSCI 2021*, pages 1–9, 2021.

[6] Emre Ates, Burak Aksar, Vitus J Leung, and Ayse K Coskun. Counterfactual explanations for multivariate time series. In *2021 international conference on applied artificial intelligence (ICAPAI)*, pages 1–8. IEEE, 2021.

[7] Katie Atkinson, Pietro Baroni, Massimiliano Giacomin, Anthony Hunter, Henry Prakken, Chris Reed, Guillermo Ricardo Simari, Matthias Thimm, and Serena Villata. Towards artificial argumentation. *AI Magazine*, 38(3):25–36, 2017.

[8] Maximilian Augustin, Valentyn Boreiko, Francesco Croce, and Matthias Hein. Diffusion visual counterfactual explanations. In *Advances in Neural Information Processing Systems 35, NeurIPS*, 2022.

[9] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[10] Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. In *Advances in Neural Information Processing Systems 34, NeurIPS*, pages 5644–5655, 2021.

[11] Pietro Baroni, Dov Gabbay, Massimilino Giacomin, and Leendert Van der Torre. Handbook of formal argumentation, 2018.

[12] Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996.

[13] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton university press, 2009.

[14] Trevor J. M. Bench-Capon. Value-based argumentation frameworks. In *NMR 2002*, pages 443–454, 2002.

[15] Tom Bewley, Salim I Amoukou, Saumitra Mishra, Daniele Magazzeni, and Manuela Veloso. Counterfactual metarules for local and global recourse. In *Proceedings of the 41st International Conference on Machine Learning*, pages 3707–3724, 2024.

[16] Emily Black, Klas Leino, and Matt Fredrikson. Selective ensembles for consistent predictions. In *ICLR 2022*, 2022.

[17] Emily Black, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. In *FAccT 2022*, pages 850–863, 2022.

[18] Emily Black, Zifan Wang, and Matt Fredrikson. Consistent counterfactuals for deep models. In *ICLR 2022*, 2022.

[19] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[20] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: identifying density-based local outliers. In *The 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104, 2000.

[21] Dieter Brughmans, Pieter Leyman, and David Martens. Nice: an algorithm for nearest instance counterfactual explanations. *Data mining and knowledge discovery*, pages 1–39, 2023.

[22] Ngoc Bui, Duy Nguyen, and Viet Anh Nguyen. Counterfactual plans under distributional ambiguity. In *The 10th International Conference on Learning Representations, ICLR*, 2022.

[23] Ngoc Bui, Duy Nguyen, Man-Chung Yue, and Viet Anh Nguyen. Coverage-validity-aware algorithmic recourse. *arXiv:2311.11349*, 2023.

[24] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. On the acceptability of arguments in bipolar argumentation frameworks. In *ECSQARU 2005*, pages 378–389, 2005.

[25] Lenart Celar and Ruth M. J. Byrne. How people reason with counterfactual and causal explanations for AI decisions in familiar and unfamiliar domains. *Memory & Cognition*, 51(7):1481–1496, 2023. ISSN 1532-5946.

[26] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

[27] Amanda Coston, Ashesh Rambachan, and Alexandra Chouldechova. Characterizing fairness over the set of good models under selective labels. In *ICML 2021*, volume 139, pages 2144–2155, 2021.

[28] Will Cukierski. Give me some credit, 2011. URL `https://kaggle.com/competitions/GiveMeSomeCredit`.

[29] Ariel Tadeu da Silva and João Roberto Bertini. Using the k-associated optimal graph to provide counterfactual explanations. In *FUZZ-IEEE 2022*, pages 1–8, 2022.

[30] Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *JMLR*, 23(1):10237–10297, 2022.

[31] Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *16th International Conference on Parallel Problem Solving from Nature*, pages 448–469, 2020.

[32] Laura N Deighan and Ruth MJ Byrne. On the different content of counterfactual and causal explanations. *Thinking & Reasoning*, pages 1–37, 2025.

[33] Eoin Delaney, Derek Greene, and Mark T. Keane. Instance-based counterfactual explanations for time series classification. In *The 29th International conference on case-based reasoning*, volume 12877 of *Lecture Notes in Computer Science*, pages 32–47, 2021.

[34] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems 31, NeurIPS*, pages 590–601, 2018.

[35] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[36] Yannis Dimopoulos, Wolfgang Dvorák, Matthias König, Anna Rapberger, Markus Ulbricht, and Stefan Woltran. Sets attacking sets in abstract argumentation. In *NMR 2023*, pages 22–31, 2023.

[37] Ricardo Dominguez-Olmedo, Amir-Hossein Karimi, and Bernhard Schölkopf. On the adversarial robustness of causal algorithmic recourse. In *Proceedings of the 39th International Conference on Machine Learning, ICML 2022*, volume 162 of *PMLR*, pages 5324–5342, 2022.

[38] Dheeru Dua, Casey Graff, et al. Uci machine learning repository, 2017.

[39] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

[40] Sanghamitra Dutta, Jason Long, Saumitra Mishra, Cecilia Tilli, and Daniele Magazzeni. Robust counterfactual explanations for tree-based ensembles. In *Proceedings of the 39th International Conference on Machine Learning, ICML*, volume 162 of *PMLR*, pages 5742–5756, 2022.

[41] Wolfgang Dvorák, Matthias König, Markus Ulbricht, and Stefan Woltran. Rediscovering argumentation principles utilizing collective attacks. In *KR 2022*, pages 122–131, 2022.

[42] Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Omer F. Rana, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, and Rajiv Ranjan. Explainable AI (XAI): core ideas, techniques, and solutions. *ACM Comput. Surv.*, 55(9):194:1–194:33, 2023.

[43] European Parliament and the Council of the European Union. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence and amending certain Union legislative acts (Artificial Intelligence Act). Official Journal of the European Union, L 2024/1689, 2024.

[44] Xiuyi Fan and Francesca Toni. On computing explanations in abstract argumentation. In *ECAI 2014*, pages 1005–1006, 2014.

[45] Andrea Ferrario and Michele Loi. The robustness of counterfactual explanations over time. *IEEE Access*, 10:82736–82750, 2022.

[46] FICO. Explainable machine learning challenge, 2018.

[47] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[48] Giorgos Flouris and Antonis Bikakis. A comprehensive study of argumentation frameworks with sets of attacking arguments. *Int. J. Approx. Reason.*, 109:55–86, 2019.

[49] João Fonseca, Andrew Bell, Carlo Abrate, Francesco Bonchi, and Julia Stoyanovich. Setting the right expectations: Algorithmic recourse over time. In *EAAMO 2023*, pages 29:1–29:11, 2023.

[50] Alexandre Forel, Axel Parmentier, and Thibaut Vidal. Robust counterfactual explanations for random forests. *arXiv:2205.14116*, 2022.

[51] Prakhar Ganesh, Afaf Taik, and Golnoosh Farnadi. The curious case of arbitrariness in machine learning. *arXiv preprint arXiv:2501.14959*, 2025.

[52] GDPR. European commission, regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation) (text with eea relevance). *OJ L 119*, 2016.

[53] Tobias Gerstenberg, Matthew F Peterson, Noah D Goodman, David A Lagnado, and Joshua B Tenenbaum. Eye-tracking causality. *Psychological science*, 28(12):1731–1744, 2017.

[54] Nicola Gigante, Francesco Leofante, and Andrea Micheli. Counterfactual scenarios for automated planning. In *The 22nd International Conference on Principles of Knowledge Representation and Reasoning, KR 2025*, 2025.

[55] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[56] Riccardo Guidotti. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.

[57] Hangzhi Guo, Feiran Jia, Jinghui Chen, Anna Cinzia Squicciarini, and Amulya Yadav. Rocoursenet: Robust training of a prediction aware recourse model. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM*, pages 619–628, 2023.

[58] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023.

[59] Victor Guyomard, Françoise Fessant, Thomas Guyet, Tassadit Bouadi, and Alexandre Termier. Generating robust counterfactual explanations. In *ECML PKDD*, 2023.

[60] Suryabhan Singh Hada and Miguel Á. Carreira-Perpiñán. Exploring counterfactual explanations for classification and regression trees. In *ECML PKDD 2021 Workshops*, volume 1524, pages 489–504, 2021.

[61] Faisal Hamman, Erfaun Noorani, Saumitra Mishra, Daniele Magazzeni, and Sanghamitra Dutta. Robust counterfactual explanations for neural networks with probabilistic guarantees. In *Proceedings of the 40th International Conference on Machine Learning, ICML*, volume 202 of *PMLR*, pages 12351–12367, 2023.

[62] Leif Hancox-Li. Robustness in machine learning explanations: does it matter? In *FAT* 20*, pages 640–647, 2020.

[63] Hsiang Hsu and Flávio P. Calmon. Rashomon capacity: A metric for predictive multiplicity in classification. In *NeurIPS 2023*, pages 28988–29000, 2022.

[64] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Formalising the robustness of counterfactual explanations for neural networks. In *AAAI 2023*, pages 14901–14909, 2023.

[65] Junqi Jiang, Jianglin Lan, Francesco Leofante, Antonio Rago, and Francesca Toni. Provably robust and plausible counterfactual explanations for neural networks via robust optimisation. In *Proceedings of the 15th Asian Conference on Machine Learning, ACML*, volume 222 of *PMLR*, pages 582–597, 11–14 Nov 2024.

[66] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Recourse under model multiplicity via argumentative ensembling. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2024*, pages 954–963, 2024.

[67] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Robust counterfactual explanations in machine learning: A survey. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024*, pages 8086–8094, 2024.

[68] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Interval abstractions for robust counterfactual explanations. *Artif. Intell.*, 336:104218, 2024.

[69] Junqi Jiang, Tom Bewley, Saumitra Mishra, Freddy Lécué, and Manuela Veloso. Interpreting language reward models via contrastive explanations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025.

[70] Junqi Jiang, Luca Marzari, Aaryan Purohit, and Francesco Leofante. Robustx: Robust counterfactual explanations made easy. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025*, 2025.

[71] Junqi Jiang, Antonio Rago, Francesco Leofante, and Francesca Toni. Argumentative ensembling for robust recourse under model multiplicity. *arXiv preprint arXiv:2506.20260*, 2025.

[72] Junqi Jiang, Tom Bewley, Salim I Amoukou, Francesco Leofante, Antonio Rago, Saumitra Mishra, and Francesca Toni. Representation consistency for accurate and coherent llm answer aggregation. *Advances in Neural Information Processing Systems*, 38:51560–51591, 2026.

[73] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Synthesising counterfactual explanations via label-conditional gaussian mixture variational autoencoders. In *International Conference on Learning Representations*, volume 2026, pages 82767–82782, 2026.

[74] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*, 2019.

[75] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Yuichi Ike. Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2022*, volume 151 of *Proceedings of Machine Learning Research*, pages 1846–1870, 2022.

[76] Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Yuichi Ike. Counterfactual explanation with missing values. *arXiv:2304.14606*, 2023.

[77] Amir-Hossein Karimi, Bodo Julius von Kügelgen, Bernhard Schölkopf, and Isabel Valera. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. In *Advances in Neural Information Processing Systems 33, NeurIPS*, 2020.

[78] Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Comput. Surv.*, 55(5): 95:1–95:29, 2023.

[79] Mark T. Keane, Eoin M. Kenny, Eoin Delaney, and Barry Smyth. If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual XAI techniques. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021*, pages 4466–4474, 2021.

[80] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[81] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27, NIPS 2014*, pages 3581–3589, 2014.

[82] Paul Kobialka, Lina Gerlach, Francesco Leofante, Erika Abrahám, Silvia Lizeth Tapia Tarifa, and Einar Broch Johnsen. Counterfactual strategies for markov decision processes. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025*, 2025.

[83] Satyapriya Krishna, Jiaqi Ma, and Himabindu Lakkaraju. Towards bridging the gaps between the right to explanation and the right to be forgotten. In *Proceedings of the 40th International Conference on Machine Learning, ICML*, volume 202 of *PMLR*, pages 17808–17826, 2023.

[84] Yann LeCun. The mnist database of handwritten digits, 1998.

[85] Francesco Leofante and Alessio Lomuscio. Robust explanations for human-neural multi-agent systems with formal verification. In *EUMAS 2023*, volume 14282, pages 244–262, 2023.

[86] Francesco Leofante and Nico Potyka. Promoting counterfactual robustness through diversity. In *The 38th AAAI Conference on Artificial Intelligence, AAAI*, 2024.

[87] Francesco Leofante, Elena Botoeva, and Vineet Rajani. Counterfactual explanations and model multiplicity: a relational verification view. In *KR 2023*, pages 763–768, 2023.

[88] Francesco Leofante, Hamed Ayoobi, Adam Dejl, Gabriel Freedman, Deniz Gorur, Junqi Jiang, Guilherme Paulino-Passos, Antonio Rago, Anna Rapberger, Fabrizio Russo, Xiang Yin, Dekai Zhang, and Francesca Toni. Contestable AI Needs Computational Argumentation. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning*, pages 888–896, 8 2024. doi: 10.24963/kr.2024/83.

[89] Dan Ley, Saumitra Mishra, and Daniele Magazzeni. Global counterfactual explanations: Investigations, implementations and improvements. *CoRR*, abs/2204.06917, 2022.

[90] Alessio Lomuscio and Lalit Maganti. An approach to reachability analysis for feed-forward relu neural networks. *arXiv preprint arXiv:1706.07351*, 2017.

[91] Donato Maragno, Jannis Kurtz, Tabea E. Röber, Rob Goedhart, S. Ilker Birbil, and Dick den Hertog. Finding regions of counterfactual explanations via robust optimization. *arXiv:2301.11113*, 2023.

[92] João Marques-Silva and Alexey Ignatiev. Delivering trustworthy AI through formal XAI. In *The 36th AAAI Conference on Artificial Intelligence*, pages 12342–12350, 2022.

[93] Charles T. Marx, Flávio P. Calmon, and Berk Ustun. Predictive multiplicity in classification. In *ICML 2020*, pages 6765–6774, 2020.

[94] Luca Marzari, Francesco Leofante, Ferdinando Cicalese, and Alessandro Farinelli. Rigorous probabilistic guarantees for robust counterfactual explanations. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI)*, 2024.

[95] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019.

[96] Rami Mochaourab, Sugandh Sinha, Stanley Greenstein, and Panagiotis Papapetrou. Robust explanations for private support vector machines. *arXiv:2102.03785*, 2021.

[97] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. *Artif. Intell.*, 173(9-10):901–934, 2009.

[98] Kiarash Mohammadi, Amir-Hossein Karimi, Gilles Barthe, and Isabel Valera. Scaling guarantees for nearest counterfactual explanations. In *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society*, pages 177–187, 2021.

[99] Ramaravind Kommiya Mothilal, Amit Sharma, and Chenhao Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *FAT* 2020: ACM Conference on Fairness, Accountability, and Transparency*, pages 607–617, 2020.

[100] Almir Mutapcic and Stephen Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 24(3):381–406, 2009.

[101] Seung-Hyup Na and Seong-Whan Lee. Counterfactual explanation through latent adjustment in disentangled space of diffusion model. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[102] Ming Pok Ng, Junqi Jiang, Gabriel Freedman, Antonio Rago, and Francesca Toni. Marge: Meshing argumentative evidence from multiple large language models for justifiable claim verification. *arXiv preprint arXiv:2508.02584*, 2025.

[103] Duy Nguyen, Ngoc Bui, and Viet Anh Nguyen. Distributionally robust recourse action. In *The 11th International Conference on Learning Representations, ICLR*, 2023.

[104] Tuan-Duy H. Nguyen, Ngoc Bui, Duy Nguyen, Man-Chung Yue, and Viet Anh Nguyen. Robust bayesian recourse. In *UAI 2022*, volume 180, pages 1498–1508, 2022.

[105] Søren Holbech Nielsen and Simon Parsons. A generalization of dung's abstract framework for argumentation: Arguing with sets of attacking arguments. In *ArgMAS 2006*, pages 54–73, 2006.

[106] R Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

[107] Jiazhen Pan, Bailiang Jian, Paul Hager, Yundi Zhang, Che Liu, Friedrike Jungmann, Hongwei Bran Li, Chenyu You, Junde Wu, Jiayuan Zhu, et al. Beyond benchmarks: Dynamic, automatic and systematic red-teaming agents for trustworthy medical language models. *arXiv preprint arXiv:2508.00923*, 2025.

[108] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *WWW '20: The Web Conference 2020*, pages 3126–3132, 2020.

[109] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. On counterfactual explanations under predictive multiplicity. In *UAI 2020*, pages 809–818, 2020.

[110] Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. CARLA: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms. In *NeurIPS Datasets and Benchmarks track*, 2021.

[111] Martin Pawelczyk, Chirag Agarwal, Shalmali Joshi, Sohini Upadhyay, and Himabindu Lakkaraju. Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In *AISTATS*, 2022.

[112] Martin Pawelczyk, Teresa Datta, Johannes van den Heuvel, Gjergji Kasneci, and Himabindu Lakkaraju. Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In *The 11th International Conference on Learning Representations, ICLR*, 2023.

[113] Martin Pawelczyk, Tobias Leemann, Asia Biega, and Gjergji Kasneci. On the trade-off between actionable explanations and the right to be forgotten. In *ICLR 2023*, 2023.

[114] Judea Pearl. *Causality*. Cambridge university press, 2009.

[115] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

[116] Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: Model-based counterfactual data augmentation. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022*, 2022.

[117] Rafael Poyiadzi, Kacper Sokol, Raúl Santos-Rodríguez, Tijl De Bie, and Peter A. Flach. FACE: feasible and actionable counterfactual explanations. In *Conf. on AI, Ethics, and Society, AIES*, 2020.

[118] Pavithra Prabhakar and Zahra Rahimi Afzal. Abstraction based output range analysis for neural networks. In *Advances in Neural Information Processing Systems 32, NeurIPS*, pages 15762–15772, 2019.

[119] Mario Alfonso Prado-Romero, Bardh Prenkaj, Giovanni Stilo, and Fosca Giannotti. A survey on graph counterfactual explanations: Definitions, methods, evaluation, and research challenges. *ACM Comput. Surv.*, 56(7):171:1–171:37, 2024.

[120] Natraj Raman, Daniele Magazzeni, and Sameena Shah. Bayesian hierarchical models for counterfactual estimation. In *AISTATS 2023*, volume 206, pages 1115–1128, 2023.

[121] Kaivalya Rawal and Himabindu Lakkaraju. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. In *Advances in Neural Information Processing Systems 33: NeurIPS 2020*, 2020.

[122] Kaivalya Rawal, Ece Kamar, and Himabindu Lakkaraju. Algorithmic recourse in the wild: Understanding the impact of data and model shifts. *arXiv preprint arXiv:2012.11788*, 2020.

[123] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD 2016*, pages 1135–1144, 2016.

[124] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.

[125] Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *FAccT 2022*, pages 1827–1858, 2022.

[126] Zahra Shams Khoozani, Aznul Qalid Md Sabri, Woo Chaw Seng, Manjeevan Seera, and Kah Yee Eg. Navigating the landscape of

concept-supported xai: Challenges, innovations, and future directions. *Multimedia Tools and Applications*, 83(25):67147–67197, 2024.

[127] Shubham Sharma, Jette Henderson, and Joydeep Ghosh. CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In *AIES 2020*, pages 166–172, 2020.

[128] Shubham Sharma, Alan H. Gee, Jette Henderson, and Joydeep Ghosh. FASTER-CE: fast, sparse, transparent, and robust counterfactual explanations. *arXiv:2210.06578*, 2022.

[129] Rui Shu. Gaussian mixture vae, 2016. URL `https://ruishu.io/2016/12/25/gmvae/`. Blog post, December 25, 2016.

[130] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, and Sameer Singh. Counterfactual explanations can be manipulated. In *NeurIPS 2021*, pages 62–75, 2021.

[131] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28: NIPS 2015*, pages 3483–3491, 2015.

[132] Mihaela C. Stoian and Eleonora Giunchiglia. Beyond the convexity assumption: Realistic tabular data generation under quantifier-free real linear constraints. In *The Thirteenth International Conference on Learning Representations, ICLR 2025*, 2025.

[133] Mihaela CÄ Stoian, Eleonora Giunchiglia, and Thomas Lukasiewicz. A survey on tabular data generation: Utility, alignment, fidelity, privacy, and beyond. *arXiv preprint arXiv:2503.05954*, 2025.

[134] Maria Stoica, Francesco Leofante, and Alessio Lomuscio. Out-of-distribution detection using counterfactual distance. *arXiv preprint arXiv:2508.10148*, 2025.

[135] Devin Thomas, Junqi Jiang, Avinash Kori, Aaron Russo, Steffen Winkler, Stuart Sale, Joseph McMillan, Francesco Belardinelli, and

Antonio Rago. Explainable reinforcement learning for formula one race strategy. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 1090–1097, 2025.

[136] Jamie Todd, Junqi Jiang, Aaron Russo, Steffen Winkler, Stuart Sale, Joseph McMillan, and Antonio Rago. Explainable time series prediction of tyre energy in formula one race strategy. In *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, pages 1082–1089, 2025.

[137] Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *KDD 2017*, pages 465–474, 2017.

[138] Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. Towards robust and reliable algorithmic recourse. In *Advances in Neural Information Processing Systems 34, NeurIPS*, pages 16926–16937, 2021.

[139] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *FAT* 2019: ACM Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019.

[140] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665, 2021.

[141] Marco Virgolin and Saverio Fracaros. On the robustness of sparse counterfactual explanations to adverse perturbations. *Artif. Intell.*, 316:103840, 2023.

[142] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.*, 31:841, 2017.

[143] Ming Wang, Daling Wang, Wenfang Wu, Shi Feng, and Yifei Zhang. T-COL: generating counterfactual explanations for general user preferences on variable machine learning systems. *arXiv:2309.16146*, 2023.

[144] Yongjie Wang, Hangwei Qian, Yongjie Liu, Wei Guo, and Chunyan Miao. Flexible and robust counterfactual explanations with minimal satisfiable perturbations. In *CIKM 2023*, pages 2596–2605, 2023.

[145] Zhendong Wang, Ioanna Miliou, Isak Samsten, and Panagiotis Papapetrou. Counterfactual explanations for time series forecasting. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 1391–1396. IEEE, 2023.

[146] Greta Warren, Mark T. Keane, and Ruth M. J. Byrne. Features of explainability: How users understand counterfactual and causal explanations for categorical and continuous features in XAI. In *Proceedings of the Workshop on Cognitive Aspects of Knowledge Representation co-located with the 31st international join conference on artificial intelligence*, volume 3251 of *CEUR Workshop Proceedings*, 2022.

[147] Jamelle Watson-Daniels, David C. Parkes, and Berk Ustun. Predictive multiplicity in probabilistic classification. In *AAAI 2023*, pages 10306–10314, 2023.

[148] Patryk Wielopolski, Oleksii Furman, Jerzy Stefanowski, and Maciej Zieba. Probabilistically plausible counterfactual explanations with normalizing flows. In *27th European Conference on Artificial Intelligence, ECAI 2024*, volume 392, pages 954–961, 2024.

[149] Linyi Yang, Eoin M. Kenny, Tin Lok James Ng, Yi Yang, Barry Smyth, and Ruihai Dong. Generating plausible counterfactual explanations for deep transformers in financial text classification. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020*, pages 6150–6160, 2020.

[150] Zhiwei Zeng, Chunyan Miao, Cyril Leung, Zhiqi Shen, and Jing Jih Chin. Computing argumentative explanations in bipolar argumentation frameworks. In *AAAI 2019*, pages 10079–10080, 2019.

[151] Xuehao Zhai, Junqi Jiang, Adam Dejl, Antonio Rago, Fangce Guo, Francesca Toni, and Aruna Sivakumar. Heterogeneous graph neural networks with post-hoc explanations for multi-modal and explainable land use inference. *Inf. Fusion*, 120:103057, 2025.

[152] Songming Zhang, Xiaofeng Chen, Shiping Wen, and Zhongshan Li. Density-based reliable and robust explainer for counterfactual explanation. *Expert Syst. Appl.*, 226:120214, 2023.

Appendix A

Appendices for Chapter 4

A.1 Computation Time of RNCE and the Impact of `robustInit`

Since querying for the nearest neighbour from a k-d tree is in logarithm time complexity wrt the tree size, the computation time bottleneck of RNCE is the total time for the more complex Δ-robustness tests. Therefore, the computation time of RNCE mainly depends on two factors, the time required for each Δ-robustness test, and the number of Δ-robustness tests performed in total. For the former factor, each Δ-robustness test is a MILP program whose problem size, affected by the number of attributes in the dataset and the architecture of the classifier in our setting, determines its computation time. The latter factor, however, is directly controlled by the algorithm parameter, `robustInit`. When `robustInit=F`, the number of Δ-robustness tests is the product of the number of inputs for which CEs are generated and the average number of times querying for the next nearest neighbour (Alg.3, lines 4-6) before reaching a Δ-robust CE, which is further affected by the value of δ constructing Δ. When `robustInit=T`, the number of Δ-robustness tests is upper-bounded by the dataset size.

We empirically compare the computation time when `robustInit=T` with `robustInit=F`. Specifically, since the line search controlled by the parameter

`optimal` (Alg. 4, lines 7-10) is independent of the impact of `robustInit`, we report results for `optimal=F`, i.e., RNCE-`FF` and RNCE-`TF`, under two different settings, in Tables A.1 and A.2.

configuration	adult		compas		gmc		heloc	
	NN	LR	NN	LR	NN	LR	NN	LR
RNCE-`FF`	7.69	0.47	11.47	4.34	4.33	0.41	55.83	0.49
RNCE-`TF` incl. time for obtaining $\mathcal{T}$	59.55	3.84	20.21	1.77	268.35	43.20	44.23	3.17
RNCE-`TF` excl. time for obtaining $\mathcal{T}$	0.002	0.002	0.001	0.001	0.008	0.007	0.001	0.002

Table A.1: RNCE computation time (seconds) of generating robust CEs for 100 randomly selected inputs using neural networks (NN) or logistic regressions (LR), when configured with different `robustInit` settings.

Configuration	**adult** (15500)		**compas** (200)		**gmc** (700)		**heloc** (2000)	
	NN	LR	NN	LR	NN	LR	NN	LR
RNCE-`FF`	1306	59.55	27.16	5.63	42.44	1.22	1052	9.79
RNCE-`TF` incl. time for obtaining $\mathcal{T}$	59.75	4.15	20.21	1.77	268.42	43.21	44.24	3.19
RNCE-`TF` excl. time for obtaining $\mathcal{T}$	0.21	0.31	0.001	0.001	0.08	0.02	0.01	0.02

Table A.2: RNCE computation time (seconds) of generating robust CEs for all data points in the training dataset that are classified to class 0 using neural networks (NN) or logistic regressions (LR), when configured with different `robustInit` settings. The numbers following the dataset names represent the approximate number of inputs for each dataset.

The results empirically support our analysis. The computation times of both RNCE-`TF` and RNCE-`FF` for neural networks are much higher than those for logistic regressions due to the fact that neural networks are more complex in terms of model architecture. The computation time of RNCE-`FF` tends to be lower than RNCE-`TF` when the number of inputs is smaller because fewer Δ-robustness tests are required. When `robustInit=F`, the time is almost proportional to the number of inputs. When `robustInit=T`, the computation times are almost identical regardless of the number of inputs because building the k-d tree $\mathcal{T}$ is more time-consuming than querying for CEs. Our RNCE-`TF` excl. time for obtaining $\mathcal{T}$ results show that, when `robustInit=T`, after obtaining $\mathcal{T}$, the time needed for querying CEs is almost negligible, since for every input, the first query will be the closest Δ-robust CE.

A.2 Full Experiment Results

	vr	**vΔ_{val}**	**vΔ_{inc}**	ℓ_1	**lof**	**vr**	**vΔ_{val}**	**vΔ_{inc}**	ℓ_1	**lof**
	adult					**compas**				
GCE	.223	0	0	.003	.036	.011	0	0	.007	.697
PROTO	.178	.020	0	.002	.089	.172	.055	0	.062	.192
MCE	.201	0	0	.001	.063	.126	0	0	.006	.047
NNCE	.139	.024	.024	.004	.114	.123	.024	0	.003	.057
ROAR	0	0	.065	.208	4.187	0	0	.033	.033	.574
RBR	.092	0	0	.002	.051	.008	.024	0	.003	.170
ST-CE	.014	.071	.020	.008	.087	.003	.128	0	.003	.057
	target $\delta_{val} = 0.02$					target $\delta_{val} = 0.02$				
GCE-R	0	0	0	.007	.079	.041	.040	0	.006	.664
PROTO-R	0	.203	0	.010	.119	.037	.037	0	.008	.299
MCE-R	0	0	0	.001	.099	.002	0	0	.003	.161
RNCE-FF	0	0	.020	.005	.021	0	0	0	.003	.062
RNCE-FT	0	0	0	.003	.045	0	0	0	.003	.060
	target $\delta_{inc} = 0.061$					target $\delta_{inc} = 0.079$				
GCE-R	0	0	0	.008	.130	.040	.040	.051	.005	.565
PROTO-R	0	0	.091	.053	.356	.037	.037	.058	.009	.244
MCE-R	0	0	0	.005	.054	0	0	0	.004	.235
RNCE-FF	0	0	0	.006	.168	0	0	0	.004	.086
RNCE-FT	0	0	0	.011	.085	0	0	0	.004	.086
	gmc					**heloc**				
GCE	.003	0	0	.005	.064	.042	0	0	.002	.045
PROTO	.012	.020	0	.004	.057	.052	0	0	.002	.036
MCE	.032	0	0	.003	.103	.053	0	0	.001	.034
NNCE	.027	.068	.020	.003	.031	.053	0	0	.003	.011
ROAR	.015	.025	.025	.014	9.69	0	0	0	.010	.164
RBR	0	.117	0	.004	.105	.070	0	0	.003	.022
ST-CE	0	.087	.058	.005	.068	0	.152	0	.004	.016
	target $\delta_{val} = 0.02$					target $\delta_{val} = 0.02$				
GCE-R	0	0	0	.006	.019	0	0	0	.003	.043
PROTO-R	0	0	0	.004	.019	0	0	.073	.002	.061
MCE-R	0	0	0	.003	.103	.002	0	0	.001	.043
RNCE-FF	0	0	.087	.004	.033	0	0	0	.002	.019
RNCE-FT	0	0	0	.004	.157	0	0	0	.003	.019
	target $\delta_{inc} = 0.091$					target $\delta_{inc} = 0.04$				
GCE-R	0	0	0	.006	.021	0	0	0	.010	.114
PROTO-R	0	0	0	.011	.014	0	0	0	.003	.025
MCE-R	0	0	0	.003	.100	0	0	0	.002	.055
RNCE-FF	0	0	0	.007	.036	0	0	0	.003	.022
RNCE-FT	0	0	0	.007	.036	0	0	0	.003	.022

Table A.3: Standard deviation results of the quantitative evaluation of the compared CE generation methods on neural networks.

Standard deviation results accompanying Table 4.3 are presented in Table A.3.

	vr↑	**v**Δ_{val}↑	**v**Δ_{inc}↑	ℓ_1↓	**lof**↓	**vr**↑	**v**Δ_{val}↑	**v**Δ_{inc}↑	ℓ_1↓	**lof**↓
	adult					**compas**				
NNCE	97.9%	45%	42%	.078	1.99	73.6%	1%	49%	.034	1.32
ROAR	100%	100%	100%	.265	1.69	100%	100%	96.8%	.220	2.59
	target $\delta_{val} = 0.04$					target $\delta_{val} = 0.08$				
RNCE-FF	100%	100%	70%	.085	2.21	100%	100%	100%	.045	1.22
RNCE-FT	100%	100%	16%	.060	1.71	100%	100%	100%	.043	1.30
	target $\delta_{inc} = 0.063$					target $\delta_{inc} = 0.01$				
RNCE-FF	100%	100%	100%	.087	2.18	86.1%	1%	100%	.035	1.30
RNCE-FT	100%	100%	100%	.064	1.74	78.1%	0%	100%	.033	1.34
	gmc					**heloc**				
NNCE	92.7%	57%	51%	.035	1.21	77.5%	5%	49%	.057	1.06
ROAR	100%	100%	100%	.119	2.76	100%	100%	100%	.089	1.42
	target $\delta_{val} = 0.06$					target $\delta_{val} = 0.07$				
RNCE-FF	100%	100%	92%	.043	1.23	100%	100%	100%	.070	1.06
RNCE-FT	100%	100%	35%	.034	1.23	100%	100%	0%	.066	1.05
	target $\delta_{inc} = 0.079$					target $\delta_{inc} = 0.019$				
RNCE-FF	100%	100%	100%	.043	1.17	98.8%	10%	100%	.061	1.06
RNCE-FT	100%	100%	100%	.035	1.23	96.7%	0%	100%	.054	1.06

Table A.4: Quantitative evaluation of the compared CE generation methods on logistic regression models. The evaluation metrics are the same as in Table 4.3.

	vr	**v**Δ_{val}	**v**Δ_{inc}	ℓ_1	**lof**	**vr**	**v**Δ_{val}	**v**Δ_{inc}	ℓ_1	**lof**
	adult					**compas**				
NNCE	.035	.130	.144	.017	.188	.025	.020	.037	.003	.068
ROAR	0	0	0	.028	.020	0	0	0	.002	.151
	target $\delta_{val} = 0.04$					target $\delta_{val} = 0.08$				
RNCE-FF	0	0	.122	.016	.251	0	0	0	.003	.032
RNCE-FT	0	0	.092	.010	.070	0	0	0	.002	.060
	target $\delta_{inc} = 0.063$					target $\delta_{inc} = 0.01$				
RNCE-FF	0	0	0	.015	.273	.015	.020	0	.003	.068
RNCE-FT	0	0	0	.009	.104	.002	0	0	.003	.093
	gmc					**heloc**				
NNCE	.028	.129	.159	.003	.021	.023	.045	.102	.003	.013
ROAR	0	0	0	.010	.225	0	0	0	.003	.038
	target $\delta_{val} = 0.06$					target $\delta_{val} = 0.07$				
RNCE-FF	0	0	.112	.004	.055	0	0	0	.003	.021
RNCE-FT	0	0	.055	.004	.048	0	0	0	.004	.019
	target $\delta_{inc} = 0.079$					target $\delta_{inc} = 0.019$				
RNCE-FF	0	0	0	.003	.039	.010	.008	0	.003	.016
RNCE-FT	0	0	0	.004	.053	.012	0	0	.004	.015

Table A.5: Standard deviation results of the quantitative evaluation of the compared CE generation methods on logistic regression models.

For linear regression models, we find δ_{inc} and δ_{val} using the same strategy for

neural networks as stated in Section 4.6.2. We quantitatively compare the CEs found by NNCE, ROAR, and two configurations of our algorithm RNCE using the same evaluation metrics introduced in Section 4.6.4. Tables A.4 and A.5 report the mean and standard deviation results for logistic regression classifiers.

Slightly different from the neural network results, the δ_{inc} values, found by incrementally retraining on 10% of $\mathcal{D}_2$, can be smaller than the δ_{val}, and sometimes insufficient to induce 100% empirical robustness (indicated by vr). Comparing RNCE with baselines, similar to the results for neural networks in Section 4.6.4, RNCE produces more robust CEs than the NNCE method, and is less costly and more plausible than ROAR. The impact of changing `optimal` to `True` is more obvious in this set of experiments, with ℓ_1 costs decreasing to a greater extent and plausibility remaining comparable to RNCE-`FF`.

Appendix B

Appendices for Chapter 6

B.1 L-GMVAE in the MNIST Example

For illustration purposes, the L-GMVAE for the MNIST examples in Figures 6.2 and 6.3 is trained on the original MNIST dataset, instead of using any classifier's predictions. As mentioned in Section 6.2.1, practically the L-GMVAE loss function is a weighted combination of the two KL terms and the reconstruction term. For this particular model, the weights are 1:1:1 (unweighted). This model is trained with input size of 28x28 images, label dimension of 10, 3 latent clusters for each label for the latent Gaussian model, a learning rate of 1e-3 for the Adam optimiser and a batch size of 1024, with early stopping based on the validation loss.

Our network architecture is implemented as follows. The encoder network has two components. One takes in the input image and its one-hot-encoded label, produces a probability vector matching the cluster dimension through a 3-layer MLP component with ReLU activation and 512 hidden neurons, plus a softmax function. This is the $q(c \mid x, y)$ component in Equation 6.2 and is useful for predicting the assigned cluster for an input once the L-GMVAE is trained. Another component, $q(z \mid x, c, y)$ in Equation 6.2, takes in the input images, label, and cluster information, to produce the mean and logvar for the corresponding latent variable z, through a similar

3-layer MLP with two output heads. At inference time, this is useful for obtaining the latent encoding of an input.

The decoder network also has two components. One is responsible for the Gaussian mixture prior (Equation 6.1c), which simply takes in cluster information and outputs the mean and logvar of the latent variable via a linear layer. The second part implements the reconstruction functionality, mapping a latent vector back to the input space through a 3-layer MLP with ReLU activation and 512 hidden neurons.

More implementation details can be found in the accompanying code repository.



Figure B.1: Reconstructed cluster centroids for the L-GMVAE model on MNIST dataset used in the examples.

Figure B.1 visualises the reconstructed cluster centroids for each digit class. We observe that this L-GMVAE model recognises different prototypical writing styles of each digit.

B.2 L-GMVAE in the CE Evaluation

The architecture of the L-GMVAE models for Section 6.3 is identical to the one described in Appendix B.1, apart from different input and label sizes.

The input sizes are decided by the dataset, and the label size matches the binary classification task. Each label is associated with five clusters. The training details are summarised in Table B.1. The other details have been covered in the main text of the paper.

	Input Size	**Latent Size**	**Batch Size**	**Learning Rate**	**Loss Weights**
heloc-RF	23	18	1024	1e-3	0.1, 0.1, 1
heloc-NN	23	15	1024	1e-3	0.1, 0.1, 1
wine-RF	11	8	64	1e-3	0.1, 0.05, 1
wine-NN	11	8	16	3e-4	0.5, 0.3, 1
adult-RF	13	10	1024	1e-3	0.4, 0.2, 1
adult-NN	13	10	1024	1e-3	0.4, 0.2, 1
compas-RF	7	5	512	5e-4	0.5, 0.3, 1
compas-NN	7	6	512	5e-4	0.5, 0.3, 1

Table B.1: L-GMVAE training details. The loss weights are the weighting terms for KL(z), KL(z), and reconstruction.