# Probabilistically Robust Counterfactual Explanations under Model Changes

Luca Marzari[a,*], Francesco Leofante[b], Ferdinando Cicalese[a], Alessandro Farinelli[a]

[a]*Department of Computer Science, University of Verona, Str. le Grazie, 15, Verona, 37134, Italy*
[b]*Department of Computing, Imperial College London, 180 Queen's Gate, London, SW7 2AZ, United Kingdom*

---

**Abstract**

We study the problem of generating robust counterfactual explanations for deep learning models subject to model changes. We focus on *plausible model changes* altering model parameters and propose a novel framework to reason about the robustness property in this setting. To motivate our solution, we begin by showing for the first time that computing the robustness of counterfactuals with respect to model changes is NP-hard. As this (practically) rules out the existence of scalable algorithms for exactly computing robustness, we propose a novel probabilistic approach which is able to provide tight estimates of robustness with strong guarantees while preserving scalability. Remarkably, and differently from existing solutions targeting plausible model changes, our approach does not impose requirements on the network to be analysed, thus enabling robustness analysis on a wider range of architectures, including state-of-the-art tabular transformers. A thorough experimental analysis on four binary classification datasets reveals that our method improves the state of the art in generating robust explanations, outperforming existing methods.

*Keywords:* Explainable AI, Counterfactual Explanations, Algorithmic Recourse, Robustness of Explanations

*Corresponding author.
*Email addresses:* luca.marzari@univr.it (Luca Marzari), f.leofante@imperial.ac.uk (Francesco Leofante), ferdinando.cicalese@univr.it (Ferdinando Cicalese), alessandro.farinelli@univr.it (Alessandro Farinelli)

## 1. Introduction

Deep Neural Networks (DNNs) have emerged as a groundbreaking technology revolutionizing several fields ranging from autonomous navigation [1, 2] to image classification [3] and robotics for medical applications [4]. However, despite remarkable successes, their vulnerability to adversarial attacks [5, 6], i.e., imperceptible modifications to input data that can lead to wrong and potentially catastrophic decisions when deployed, has raised crucial safety concerns. Consequently, understanding and explaining the decisions of black-box deep learning models has become a dominant goal in AI research. In this paper, we focus on counterfactual explanations (CFX), a popular class of explanation methods that aim to demystify the decision-making of a DNN by showing how an input needs to be changed to yield a different, typically more desirable, decision (see [7, 8] for recent surveys).

To understand what makes CFXs useful, consider the widely used example of a loan application, where a mortgage applicant represented by an input $x$ with features *unemployed* status, 25 years of age, and *low* credit rating applies for a loan and is rejected by the bank's AI. A CFX for this decision could be a slightly modified input, where increasing credit rating to *medium* would result in the loan being granted. Ideally, a counterfactual explanation should be as close as possible to the original input to ensure that the changes it suggests are feasible. The approach of [9], showed how this requirement can be mathematically achieved by generating a counterfactual as close as possible to the decision boundary of a DNN. However, producing explanations in this way raises critical concerns about their reliability (see [10] for a survey). For example, as illustrated in Figure 1, fine-tuning the model with additional data can significantly alter its decision boundary, potentially invalidating previously generated counterfactuals. This sensitivity to the model changes for the counterfactuals poses critical questions about the reliability of explanations and long-term usability in dynamic settings.

In particular, recent work has highlighted issues related to the robustness of CFXs against *Plausible Model Changes* (PMC) [11, 12], showing that the validity of CFXs is likely to be compromised when bounded perturbations are applied to the parameters of a DNN, e.g., as a result of fine-tuning [11, 13, 14, 12, 15]. Consider the loan example: if retraining occurs while the applicant is working toward improving their credit rating, without robustness, their

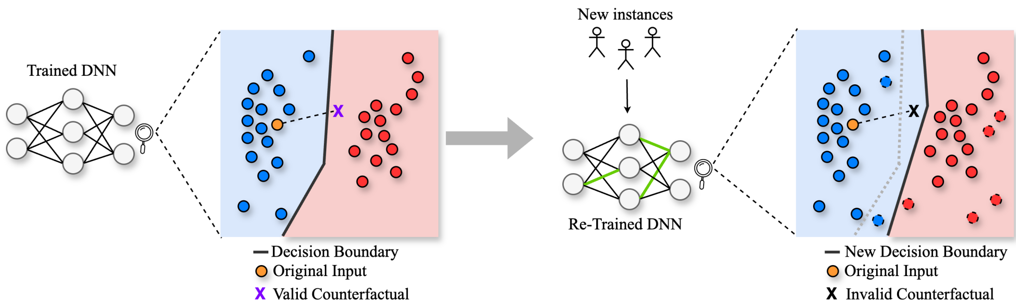


Figure 1: Vignette illustrating the problem of robustness under model changes. A counterfactual explanation is initially generated for a trained model (left). Then, the model is updated to include new data (right). This step might induce slight changes in the decision boundary of the model, ultimately invalidating the counterfactual explanation generated in the first step.

modified case may still result in a rejected application, leaving the bank liable due to their conflicting statements.

In this paper, we focus on this troubling phenomenon and advance the state of the art in CFX robustness research in several directions. More specifically, we start by studying the computational complexity of exactly determining whether a CFX is robust to PMC in § 3. Our result formally shows for the first time that this is an NP-hard problem, thus providing new insights into algorithmic developments in this area.

As our hardness results rule out the existence of practical algorithms to compute the CFX robustness in an exact fashion, we argue that probabilistic approaches are needed to obtain answers on the CFX robustness under model changes. Notably, the work by Hamman et al. [15], proposes a probabilistic approach to compute the robustness of CFX under *Naturally-Occurring Model Changes* (NOMC).[1] Even though both PMC and NOMC notions are commonly used in the literature, very little is known about their potential interplay, and whether robustness to NOMC subsumes robustness to PMC is still unresolved. In § 4, we report a complete study of the two notions and formally prove that these two notions capture profoundly different scenarios.

[1]In this work, we primarily use the term "model changes", following the notation used in recent surveys on the topic [10]. An alternative term "model shifts", with similar meaning, has also been used in related literature, as in [16]. The two terms will be used interchangeably throughout.

As a result, we demonstrate that robustness guarantees given for NOMC do not directly extend to PMC. Having settled this, in § 5, we present an extended overview of our $\texttt{AP}\Delta\texttt{S}$ , a novel sampling-based certification algorithm that allows us to determine a provable probabilistic bound on the maximum shift a CFX can tolerate under PMC. Unlike existing solutions for robustness under PMC, our approach comes with significantly reduced computational requirements and does not make any assumption on the underlying DNN, thus making it applicable to a wider range of architectures, including state-of-the-art transformer architectures.

To confirm this aspect, in § 6, we present a thorough experimental evaluation analysing the performance of $\texttt{AP}\Delta\texttt{S}$ , providing a comprehensive comparison of the proposed approach against several state-of-the-art methodologies for CFX robustness and different ablation studies. Crucially, we show that our approach outperforms existing methods on several metrics from the CFX literature, including validity, proximity, and plausibility.

The paper is structured as follows. In § 2, we cover the related work, and in § 3, we introduce background notions on computing robust CFXs under model changes. § 3 presents our complexity analysis and offers complete proof of NP-hardness for both PMC and NOMC. Motivated by this result, in § 4, we study existing approaches to generate probabilistically robust CFXs and analyze their interplay. Then, in § 5, we introduce our method to generate robust CFXs, $\texttt{AP}\Delta\texttt{S}$ , and evaluate it extensively in § 6. The core contributions of this work can be summarised as follows:

- We prove, for the first time, that determining whether a CFX is robust to model changes in a deep neural network is an NP-complete problem, for both existing notions of NOMC and PMC. This finding highlights the need for further research into probabilistic methods to address this problem effectively.

- We analyse existing approaches to generate probabilistic guarantees for CFXs under NOMC and demonstrate that these guarantees do not extend to PMC.

- We present $\texttt{AP}\Delta\texttt{S}$ , a scalable procedure that is able to generate provably robust CFXs. This approach introduces an iterative algorithm to generate probabilistically robust CFXs, which are demonstrated to have superior performance against four robust baselines.

- To confirm the scalability and effectiveness of our solution, we employ `AP∆S` to certify the robustness of CFXs for state-of-the-art transformer architectures [17] employed in tabular data classification. To the best of our knowledge, we are the first to consider models of this size within the robust CFX literature [10].

This paper builds upon our previous work[16] with significant extensions. Specifically, § 3 non trivially extends the corresponding section in [16] and offers a full hardness proof for the problem of deciding robustness of counterfactual explanations under PMC. As a corollary of this result, we are also able to show the hardness with respect to NOMC, thus providing a rigorous characterization of the complexity of verifying CFX robustness under existing notions of model changes. § 4 is also extended with a thorough experimental evaluation, complementing our theoretical findings of [16] and showing that PMC and NOMC capture very different robustness requirements in practice. Our experimental analysis in § 6 is also extended considerably. In particular, we present a novel analysis of the impact that the main hyper-parameters of `AP∆S` can have on the quality of CFXs it generates. Moreover, we demonstrate the scalability of our approach by presenting new results obtained on large-scale tabular transformers. To the best of our knowledge, this is the first time a method for robust CFXs has been shown to scale to state-of-the-art transformer models. These results complement our previous analysis and demonstrate the versatility of `AP∆S` , as well as its effectiveness in solving robustness issues in state-of-the-art machine learning models.

## 2. Related Work

Various methods for generating CFXs for DNNs have been proposed. The seminal work of [9] framed the task of generating CFXs as a gradient-based optimization problem and proposed a loss that promotes CFX *validity* (i.e., the CFX successfully changes the classification outcome of the network) and *proximity* (i.e., the CFX is as close as possible to the original input for some distance metric). In addition to these metrics, other important properties have been highlighted as crucial for the practical applicability of CFXs. Prominent examples include *plausiblity* (i.e., the CFX must lie on the data manifold) [18, 19] and *actionability* (i.e., the changes suggested by the CFX must be achievable by the user in practice) [20]. Differently from these works, here we focus on the robustness property of CFXs.

Several forms of CFX robustness have been studied in the literature [10]. Robustness to input changes is the focus of, e.g. [21, 22, 23, 24, 25], where solutions are devised to ensure that explanation algorithms return similar CFXs for similar inputs. In another line of work, [26, 27, 28, 29] considered the problem of generating adversarially robust CFXs that preserve validity under imperfect (or noisy) execution. Robustness to model multiplicity is instead considered in, e.g. [19, 30, 31], where CFXs that preserve validity across sets of models are sought. However, the study of these forms of robustness is outside the scope of this paper as our focus is on model changes. Robustness to model changes has been studied in, e.g. [32, 13, 14, 12, 15, 33]. Of these, the approaches of [11] and [33] are the most closely related to our work. The former presents an approach to generate robust CFXs under PMS using techniques from continuous optimization, which is able to guarantee robustness in the average-case scenario. The latter instead solves the same problem using abstraction techniques and discrete optimization tools, obtaining robustness guarantees that hold under worst-case conditions. Given their relevance, both approaches will be considered for an extensive experimental comparison in § 6.

## 3. Background

***Neural networks and classification tasks.*** Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space of a *classifier* $\mathcal{M}_\theta : \mathcal{X} \rightarrow [0, 1]$ mapping an input $x \in \mathcal{X}$ to an output probability between 0 and 1. We consider classifiers implemented by feed-forward DNNs parameterized by a *(parameter) vector* $\theta \in \Theta \subseteq \mathbb{R}^k$. Given two parameter vectors $\theta, \theta' \in \Theta$, we refer to the corresponding classifiers $\mathcal{M}_\theta$ and $\mathcal{M}_{\theta'}$ as *instantiations* of the same parametric classifier $\mathcal{M}_\Theta$. We assume concrete valuations of $\theta$ are learned from a set of labeled inputs as customary in supervised learning settings [34]. Once $\theta$ has been learned, the classifier can be used for inference. Without any loss of generality, we focus on binary classification tasks, i.e., the classification decision produced by $\mathcal{M}_\theta$ for an unlabeled input $x$ is 1 if $\mathcal{M}_\theta(x) \geq 0.5$, and 0 otherwise.

***Counterfactual explanations.*** Existing methods in the literature define CFXs as follows.

**Definition 1.** *Consider an input $x \in \mathcal{X}$ and a classifier $\mathcal{M}_\theta$ s.t. $\mathcal{M}_\theta(x) <$ 0.5. Given a distance metric $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$, a* (valid) counterfactual explanation *is any $x'$ such that:*

$$x' = \underset{\hat{x} \in \mathcal{X} : \mathcal{M}_\theta(\hat{x}) \geq 0.5}{\arg\min} \; d(x, \hat{x})$$

Intuitively, given an input $x$ for which the classifier produces a negative outcome, a counterfactual explanation is a new input $x'$ which is similar to $x$, e.g., in terms of some specified distance between features values, and for which the classifier predicts a different outcome. Common choices for $d$ include the $\ell_1$ and $\ell_\infty$ norms [9], which will also be used in this work.

***Robustness to model changes***. Among several notions of robustness, recent work has placed emphasis on generating CFXs that remain valid under (slight) changes in the classifier they were generated for. While existing approaches rely on a diverse range of techniques to solve this problem, they all share a common understanding of what constitutes a model shift, which we present next.

**Definition 2** (Jiang et al. [12])**.** *Let $\mathcal{M}_\theta$ and $\mathcal{M}_{\theta'}$ be two instantiations of a parametric classifier $\mathcal{M}_\Theta$. For $0 \leq p \leq \infty$, the* p-distance between $\mathcal{M}_\theta$ and $\mathcal{M}_{\theta'}$ *is defined as $d_p(\mathcal{M}_\theta, \mathcal{M}_{\theta'}) = \|\theta - \theta'\|_p$.*

**Definition 3** (Jiang et al. [12])**.** *A* model shift *(w.r.t. a fixed p-distance) is a function $S$ mapping a classifier $\mathcal{M}_\theta$ into another classifier $\mathcal{M}_{\theta'} = S(\mathcal{M}_\theta)$ such that:*

- *$\mathcal{M}_\theta$ and $\mathcal{M}_{\theta'}$ are instantiations of the same $\mathcal{M}_\Theta$;*
- *$d_p(\mathcal{M}_\theta, \mathcal{M}_{\theta'}) > 0$.*

Informally, a model shift captures changes in the parameters of a DNN, but does not affect its architecture. Based on this definition, we can formalize the robustness property for a CFX as follows.

**Definition 4.** *Consider an input $x \in \mathcal{X}$ and a classifier $\mathcal{M}_\theta$ s.t. $\mathcal{M}_\theta(x) < 0.5$. Let $x'$ be a counterfactual explanation computed for $x$ s.t. $\mathcal{M}_\theta(x') \geq 0.5$. Given a set of model changes $\Delta$, we say that the counterfactual $x'$ is* $\Delta$-robust *if $S(\mathcal{M}_\theta)(x') \geq 0.5$ for all $S \in \Delta$.*

The definition of a model shift can be specialized to better characterize how $\theta$ is allowed to change under $S$. In the following, we report two most commonly studied notions of model changes: *Naturally-Occurring Model Changes* and *Plausible Model Changes*.

**Definition 5** ( Hamman et al. [15] (NOMC))**.** *Consider a classifier $\mathcal{M}_\theta$. A set of model changes $\Delta$ is said to be* naturally occurring *if for a (randomly) chosen model change $S$ from $\Delta$ and $\mathcal{M}_{\theta'} = S(\mathcal{M}_\theta)$ being the new classifier obtained after applying $S$ to $\mathcal{M}_\theta$ the following hold:*

- $\mathbb{E}[\mathcal{M}_{\theta'}(x)] = \mathcal{M}_\theta(x)$*; where the expectation is over the randomness of $\mathcal{M}_{\theta'}$ given a fixed value of $x$;*

- *Var*$[\mathcal{M}_{\theta'}(x)] = \nu_x$*, where $\nu_x$ represents the maximum variance of the prediction of $\mathcal{M}_{\theta'}(x)$, and whenever $x$ lies on the data manifold $\mathcal{X}$, $\nu_x$ is upper bounded by a small constant $\nu$;*

- *If $\mathcal{M}_\theta$ is Lipschitz continuous for some $\gamma_1$, then $\mathcal{M}_{\theta'}(x)$ is also Lipschitz continuous for some $\gamma_2$.*

Broadly speaking, a naturally-occurring model shift allows the application of arbitrary changes to $\theta$ as long as the resulting model remains part of a class of models that are expected to have the same behaviour. This is in contrast with the notion of plausible model shift [11, 12], which requires changes to be bounded.

**Definition 6** (Jiang et al. [12] (PMC))**.** *Consider a classifier $\mathcal{M}_\theta$ and a new classifier $\mathcal{M}_{\theta'} = S(\mathcal{M}_\theta)$ obtained after applying a model shift $S$ to $\mathcal{M}_\theta$. Given some $\delta \in \mathbb{R}_{>0}$ and $0 \leq p \leq \infty$, $S$ is said to be* plausible *(w.r.t. the choice of parameters $\delta$ and $p$) if $d_p(\mathcal{M}_\theta, S(\mathcal{M}_\theta)) \leq \delta$.*

Therefore, for any choice of parameters $p, \delta$, and any instantiation $\mathcal{M}_\theta$ of a parametric classifier $\mathcal{M}_\Theta$ we define the set of PMC $\Delta_p$ obtained by considering all changes $S$ that satisfy Definition 6, i.e. $\Delta = \{S \mid d_p(\mathcal{M}_\theta, S(\mathcal{M}_\theta)) \leq \delta\}$.

In the following, we will refer to any instantiation $\mathcal{M}_{\theta'}$ of $\mathcal{M}_\Theta$ which is obtainable by applying a model change in $\Delta$ to $\mathcal{M}_\theta$ as a realisation of $\Delta$. Moreover, whenever it is not explicitly specified, we will tacitly assume that the underlying distance $d_p(\cdot,\cdot)$ is the $\infty$-norm.

Jiang et al. [12] proposed to reason about robustness under PMC using an Interval Neural Network (INN) [35] as an intermediate representation.

**Definition 7.** *An interval neural network $\mathcal{I}$ is a neural network where on each edge $e$ is associated with an interval $I_e = [a_e, b_e]$. A realisation of the INN $\mathcal{I}$ is a neural network having the same topology of $\mathcal{I}$ and such that the weight $w_e$ on edge $e$ satisfies $w_e \in I_e$,, i.e., it is taken from the interval associated to the same arc in $\mathcal{I}$.*

Jiang et al. [12] exploits the fact that the interval weights of an INN allow to represent an over-approximation of all the possible models obtainable under a set of PMC $\Delta$, thus providing a compact representation of the problem. Similarly, in our work, we use the INN representation to model the PMC concept. However, instead of analysing the robustness of counterfactual explanations through the entire INN, we focus directly on reasoning regarding the potential realisations within the set $\Delta$. This results in several computational improvements as we will discuss in section 5.

In this section, we study the computational complexity of deciding whether a given counterfactual explanation is robust in the presence of model shifts. Our aim here is to better understand the computational challenges arising from this problem and to use these results to guide the development of novel, more efficient certification procedures. Without loss of generality, we first focus on PMC and show the NP-hardness of verifying CFX robustness with respect to this definition of model changes. Later, we show that the set of PMC used by our reduction also constitutes a set of NOMC, which implies that CFX robustness is, in general, also hard to verify with respect to NOMC.

Deciding whether for a given $\mathcal{M}_\theta$ a CFX $x'$ is robust with respect to a set of PMC $\Delta$ requires to check if, for at least one model shift in $\Delta$, there exists a realisation $\mathcal{M}_{\theta'}$ which classifies CFX $x'$ differently from $\mathcal{M}_\theta$, i.e., $\mathcal{M}_{\theta'}(x) < 0.5 \leq \mathcal{M}_\theta$. This question is encoded in the following problem.

DISTINCT-REALISATIONS PROBLEM (DRP)

**Input**: an instantiation $\mathcal{M}_{\theta_1}$ of a parametric classifier $\mathcal{M}_\Theta$, an input $x$ such that $\mathcal{M}_{\theta_1}(x) \geq 0.5$, and a set $\Delta$ of PMC.

**Output**: yes $\iff$ there exists an instantiation $\mathcal{M}_{\theta_2}$ of $\mathcal{M}_\Theta$ which is a realisation of $\Delta$ and such that $\mathcal{M}_{\theta_2}(x) < 0.5$

To prove the hardness of the problem, we show a reduction from a simple variant of 3-SAT, which we refer to as 3-NAF-SAT.

3-NOTALLFALSE-SAT (3-NAF-SAT)

**Input**: a 3-CNF $\phi$ such that the assignment of all false values is not satisfying, i.e., $\phi(false, false, \ldots, false) = false$.

**Output**: yes $\iff$ there exists an assignment $\mathbf{a}$ such that $\phi(\mathbf{a}) = true$.

The $NP$-completeness of 3-NAF-SAT immediately follows from the NP-completeness of 3-SAT. We provide a proof of this fact for the sake of self-containment of the paper.

**Theorem 1.** 3-NAF-SAT *is* $NP$*-complete.*

*Proof.* We show a reduction from 3-SAT. Let $\psi$ be a 3-CNF formula over $n$ variables $x_1, \dots, x_n$. Consider the 3-CNF formula $\phi$ over $n+1$ variables defined by $\phi(x_1, \dots, x_n, x_{n+1}) = \psi(x_1, \dots, x_n) \wedge (x_{n+1} \vee x_{n+1} \vee x_{n+1})$. Clearly for the assignment $\mathbf{a}$ such that $a_i = false$ for each $i = 1, \dots, n+1$ we have $\phi(\mathbf{a}) = false$, hence $\phi$ is a proper instance of 3-NAF-SAT, which is obtainable in polynomial time from the instance $\psi$ of 3-SAT. Moreover, $\mathbf{a} = (a_1, \dots, a_n, a_{n+1})$ is a satisfying assignment for $\phi$ if and only if $a_{n+1} = true$ and $\psi(a_1, \dots, a_n) = true$, i.e., if and only if $a_1, \dots, a_n$ is satisfying for $\psi$. $\square$

**Theorem 2.** *Deciding* DRP *is NP-complete.*

*Proof.* The inclusion of DRP in NP is trivial. A certificate is a $\mathcal{M}_{\theta_2}$ which is of the same size as $\mathcal{M}_{\theta_1}$, hence polynomial in the input size. The verification of such a certificate, consists of a forward propagations of $x$ through $\mathcal{M}_{\theta_2}$ in order to check that $\mathcal{M}_{\theta_2}(x) < 0.5$. This is clearly doable in time polynomial in the size of the classifier, i.e., polynomial in the input.

For the hardness of DRP we show a reduction from 3-NAF-SAT. In particular, we show that there is a $\delta \in (0, 1]$ such that, given a 3-CNF formula $\phi$, not satisfied by the all-false assignment, we can construct an INN $\mathcal{I}$ whose edge intervals are all of the width $2\delta$ and an input $x$ such that

1. for $\mathcal{M}_{\theta_1}$ being the DNN with the same topology of $\mathcal{I}$ and such that for each edge $e$ the weight $w_e$ is taken as the central point of the interval assigned to $e$ in $\mathcal{I}$, we have $\mathcal{M}_{\theta_1}(x) \geq 0.5$;
2. $\phi$ is satisfiable if and only if there exists another DNN $\mathcal{M}_{\theta_2}$ which is also a realisation of $\mathcal{I}$ and such that $\mathcal{M}_{\theta_2}(x) < 0.5$.

Note that we are using the INN $\mathcal{I}$ to represent both the parametric classifier $\mathcal{M}_\Theta$ and the set of PMC $\Delta$, consisting of all the possible $DNN$ being a realisation of $\mathcal{I}$.

We start by analysing several gadgets that will be used as building blocks of $\mathcal{I}$. These gadgets are shown in Figures 2, 3, 4, 5.

Lemmas 3-8 provide the key properties of such gadgets which will be used in the reduction. The parameter $\delta$ is a number in $(0, 1)$ whose value will be fixed by the analysis.

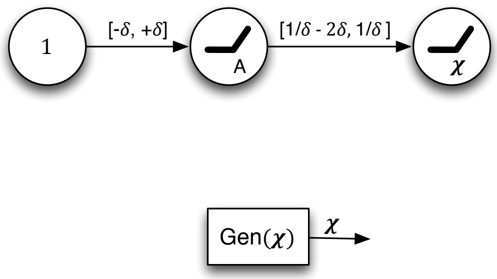


Figure 2: Generating-gadget. The input to this gadget is the constant 1 represented by the leftmost node. The output is the value $\chi$ computed in the rightmost node, that depends on the weights chosen in the intervals on the two edges.

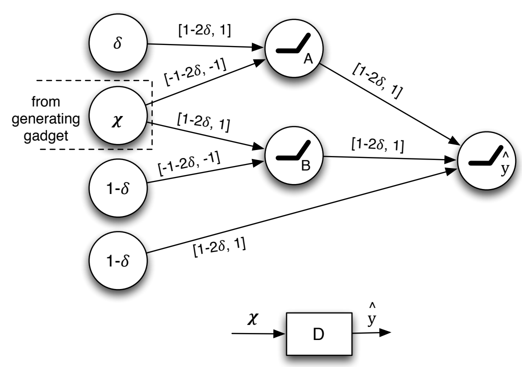


Figure 3: Discretizer-gadget. The only non-constant input is the value computed in the node labelled $\chi$. The output is the value computed in the node labelled $\hat{y}$.

**Lemma 3** (Generating-gadget)**.** *The value $\chi$ computed in the leftmost node of the Generating-gadget in Fig. 2, satisfies $\chi \in [0, 1]$.*

*Proof.* The value computed by the first node satisfies $A \in [0, \delta]$. Hence, since $\chi = \max\{0, A \cdot w\}$ with $w \in [\frac{1}{\delta} - 2\delta, \frac{1}{\delta}]$ we have $\chi \in [0, 1]$. $\square$

**Lemma 4** (Discretizer-gadget)**.** *Consider the* Discretizer*-gadget in figure 3 with $\chi$ being the the leftmost node of a Generating-gadget, i.e., the corresponding value satisfies $\chi \in [0, 1]$. Then, for the value $\hat{y}$ the following holds:*

1. *if $\hat{y} > 1 - \delta$ then $\chi \in [0, \delta] \cup [1 - \delta, 1]$;*

2. *if* $\chi \in \{0,1\}$ *then there are possible choices of the weights yielding* $\hat{y} = 1$*;*
3. *if* $\hat{y} \neq 1$ *then* $\chi \notin \{0,1\}$.
4. $\hat{y} \in [0,1]$.

*Proof.* The claims are a direct consequence of the following observations (refer to Fig. 3 for the notation):

(a) if $\chi \in (\delta, 1-\delta)$ the $A = 0$ and $B = 0$, hence $\hat{y} \in [(1-\delta)(1-2\delta), 1-\delta]$.

(b) if $0 < \chi \leq \delta$ then $B = 0$ and $A \in [\max\{0, \delta(1-2\delta) - \delta(1+2\delta)\}, \delta - \chi] \subseteq [0, \delta)$. Hence

$$\hat{y} \in [(1-\delta)(1-2\delta), (1-\delta) + (\delta - \chi)] \subseteq [(1-\delta)(1-2\delta), 1).$$

(c) if $(1-\delta) \leq \chi < 1$ then $A = 0$ and $B \in [\max\{0, (1-\delta)(1-2\delta) - (1-\delta)(1+2\delta)\}, \chi - (1-\delta)] \subseteq [0, \delta)$. Hence,

$$\hat{y} \in [(1-\delta)(1-2\delta), (1-\delta) + \chi - (1-\delta)] \subseteq [(1-\delta)(1-2\delta), 1).$$

(d) if $\chi = 0$ then $B = 0$ and $A \in [\delta(1-2\delta), \delta]$, hence

$$\hat{y} \in [\delta(1-2\delta)^2 + (1-\delta)(1-2\delta), 1].$$

In particular, for the realisations of $\mathcal{I}$ where the weights on the topmost edges and on the bottommost edge are chosen to be 1 we have $\hat{y} = 1$.

(e) if $\chi = 1$ then $A = 0$ and $B \in [\max\{0, (1-2\delta) - (1+2\delta)(1-\delta), \delta] = [0, \delta]$, hence

$$\hat{y} \in [(1-\delta)(1-2\delta), (1-\delta) + \delta] = [(1-\delta)(1-2\delta), 1].$$

In particular, for the realisations of $\mathcal{I}$ where all the weights on the edges are chosen to be the maximum possible value, we have $\hat{y} = 1$.

Item 1 in the statement follows directly from (a). Item 2 in the statement follows from (d) and (e). Item 3 in the statement follows from (b) and (c). Finally, Item 4 follows from the (a)-(e). $\square$

**Lemma 5** (Negation-gadget)**.** *With reference to the* Negation-*gadget in Fig.4, for any* $0 < \delta < \frac{\sqrt{6}}{2} - 1$, *and* $\chi \in [0, \delta] \cup [1-\delta, 1]$, *the following holds:*

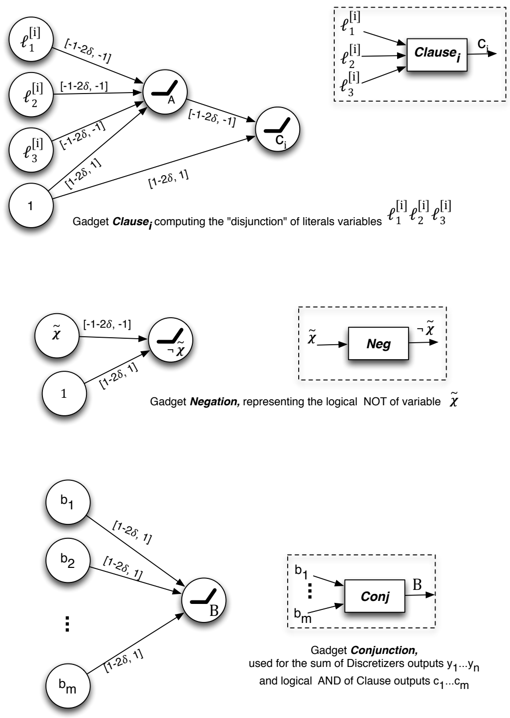


Figure 4: LOGICALPORTS

1. $\neg\chi \in [0, \delta] \iff \chi \in [1-\delta, 1]$*;*
2. $\neg\chi \in [1 - 3\delta - 2\delta^2, 1] \iff \chi \in [0, \delta]$.
3. *if* $\chi \in \{0, 1\}$ *then there is a choice of the weights of the Negation-gadget such that* $\neg\chi = 1 - \chi$*. In other words, if the input value is binary, then there is a choice of the weights such that the Negation-gadget computes the boolean NOT of the input* $\chi$*.*

*Proof.* We have $\neg\chi = \max\{0, w_2 - |w_1|\chi\}$ where $w_1 \in [-1-2\delta]$ and $w_2 \in [1-2\delta, 1]$. Therefore

(i) if $\chi \in [0, \delta]$ it follows that

$$\neg\chi \in [(1-2\delta) - \delta(1+2\delta), 1-0] = [1 - 3\delta - 2\delta^2, 1];$$

(ii) if $\chi \in [1-\delta, 1]$ then

$$\neg\chi \in [\max\{(1-2\delta) - (1+2\delta), 0\}, 1 - (1-\delta)] = [0, \delta].$$

Moreover, because of the hypothesis $0 < \delta < \frac{\sqrt{6}}{2} - 1$, we have that $1 - 3\delta - 2\delta^2 > \delta$ and

$$[1 - 3\delta - 2\delta^2, 1] \cap [0, \delta] = \emptyset,$$

which implies that the implications hold also in the opposite direction.

For the third claim of the lemma, it is enough to consider the weight in the associated interval for each edge whose absolute value is equal to 1. □

**Lemma 6** (Clause-gadget)**.** *For the* Clause-*gadget in Fig.4, the following holds: Let* $0 < \delta \leq \frac{1}{12}$*, and for each* $t = 1, 2, 3$*, let* $\ell_t^{[i]} \in [0, \delta] \cup [1 - 3\delta - 2\delta^2, 1]$[2] *We have that,*

1. $c_i \in [0, 5\delta + 6\delta^2]$ *if and only if for all* $t = 1, 2, 3$, $\ell_t^{[i]} \in [0, \delta]$*;*
2. $c_i \in [1 - 5\delta - 8\delta^2 - 4\delta^3, 1]$ *if and only if there is* $t \in \{1, 2, 3\}$ *such that* $\ell_t^{[i]} \in [1 - 3\delta - 2\delta^2, 1]$.
3. *if for each* $t = 1, 2, 3$, $\ell_t^{[i]} \in \{0, 1\}$ *then there is a choice of the weights of the Clause-gadget such that* $c_i \in \{0, 1\}$ *and* $c_i = 0$ *if and only if* $\ell_1^{[i]} = \ell_2^{[i]} = \ell_3^{[i]} = 0$*. In other words, if the input values are binary, then there is a choice of the weights such that the Clause-gadget computes the boolean OR of the inputs* $\ell_t^{[i]}$.

*Proof.* Consider a realisation of the Clause-gadget. Let us denote by $w_t^L$ the weight taken from the interval on the edge connecting $\ell_t^{[i]}$ to $A$. Moreover, let $w_1$ denote the weight taken from the interval on the edge connecting the fixed value node 1 to $A$. Let $w_2$ be the weight taken from the interval on the edge connecting the fixed value node 1 to the output node of the gadget. Finally, let $w_A$ be the weight taken from the interval associated with the edge connecting the node $A$ to the output node of the gadget.

(i) If for all $t = 1, 2, 3$, it holds that $\ell_t^{[i]} \in [0, \delta]$ then, using $A = \max\{0, w_1 - \sum_{t=1}^{3} |w_t^L| \ell_t^{[i]}\}$, we have that $A \in [\max\{0, 1 - 2\delta - 3\delta(1 + 2\delta)\}, 1] \subseteq [1 - 5\delta - 6\delta^2, 1]$. Since $c_i = \max\{0, w_2 - |w_A| \cdot A\}$, we have

$$c_i \in [\max\{0, (1 - 2\delta - (1 + 2\delta)\}, 1 - (1 - 5\delta - 6\delta^2)] \subseteq [0, 5\delta - 6\delta^2].$$

This shows the sufficiency of the condition in the first item of the statement.

[2] Note that this corresponds to the case when $\ell_t^{[i]}$ is either the output $\neg\chi$ of a Negation-gadget or the output $\chi$ of a Generating-gadget such that $\hat{y} > 1 - \delta$

(ii) Assume there exists $\hat{t} \in \{1, 2, 3\}$ such that $\ell_{\hat{t}}^{[i]} \in [1 - 3\delta - 2\delta^2, 1]$. Then

$$A = \max\{0, w_1 - |w_{\hat{t}}^L|\ell_t^{[i]} + \sum_{t \neq \hat{t}} |w_t^L|\ell_t^{[i]}\}.$$

It follows that

$$A \geq \max\{0, (1 - 2\delta) - (1 + 2\delta) \cdot 1 - 2(1 + 2\delta) \cdot 1\} = 0,$$

and

$$A \leq \max\{0, 1-1\cdot(1-3\delta-2\delta^2)-2\cdot(1)\cdot(0)\} = \max\{0, 1-(1-3\delta-2\delta^2)\} = 3\delta+2\delta^2.$$

Hence $A \in [0, 3\delta + 2\delta^2]$. Therefore,

$$c_i \in [\max\{0, (1-2\delta)-(1+2\delta)(3\delta+2\delta^2)\}, \max\{0, 1-(1)\cdot 0\}] = [1-5\delta-8\delta^2-4\delta^3, 1].$$

This shows the sufficiency of the condition in the second item of the statement.

Finally, because of the assumption $\delta \leq \frac{1}{12}$ we have that $5\delta - 6\delta^2 < 1 - 5\delta - 8\delta^2 - 4\delta^3$ hence

$$[0, 5\delta - 6\delta^2] \cap [1 - 5\delta - 8\delta^2 - 4\delta^3, 1] = \emptyset,$$

which implies that the conditions in both items of the statement are also necessary.

For the third claim of the lemma, it is enough to consider the weight in the associated interval for each edge whose absolute value is equal to 1. □

**Lemma 7** (Conjunction-gadget)**.** *Consider the* Conjunction-*gadget in Fig.4.*

*Let $\tilde{n}$ denote the output of a Conjunction-gadget whose inputs are the values $\hat{y}_1, \ldots, \hat{y}_n$ output by the $n$ Discretizer-gadgets, with input $\chi_1, \ldots, \chi_n$, respectively, such that $\chi_j \in [0, 1]$.*

*Let $\tilde{c}$ denote the output of a Conjunction-gadget whose inputs are values $c_1, \ldots, c_m$. Assume also that for each $i = 1, \ldots, m$, $c_i$ is the output of a Clause-gadget whose inputs are either the output $\chi_j$ of a Generating-gadget or the output of a Negation-gadget whose input is the output of a Generating-gadget.*

1. *If* $\tilde{n} > n - \delta$ *then for each* $i = 1, \ldots, m$ *it holds that* $\hat{y}_i \in (1 - \delta, 1]$, *and* $\chi_i \in [0, \delta] \cup [1 - \delta, 1]$.
2. *If* $\tilde{c} > m - (5\delta + 8\delta^2 + 4\delta^3)$ *then for each* $i = 1, \ldots, m$ *it holds that* $c_i \in (1 - (5\delta + 8\delta^2 + 4\delta^3), 1]$.

*Proof.* The first claim follows from Lemma 4. In particular, by Lemma 4, the condition on the values $\chi_j$ implies $\hat{y}_i \in [0, 1]$. Moreover, by $\tilde{n} > n - \delta$, it follows that for each $i$ we have $\hat{y}_i > 1 - \delta$. Again, by Lemma 4, this implies that $\chi_i \in [0, \delta] \cup [1 - \delta, 1]$.

For the second claim, we first observe that the hypotheses on the Clause-gadget whose outputs are the values $c_1, \ldots c_m$, imply that the input to such gadgets satisfies the hypotheses of Lemma 6. Therefore, for each $i = 1, \ldots, m$, it holds that $c_i \in [0, 5\delta + 6\delta^2] \cup [1 - 5\delta - 8\delta^2 - 4\delta^3, 1]$. It follows that, if $\tilde{c} > m - 5\delta - 8\delta^2 - 4\delta^3$ for each $i = 1, \ldots, m$, it holds that $c_i > 1 - 5\delta - 8\delta^2 - 4\delta^3$. □

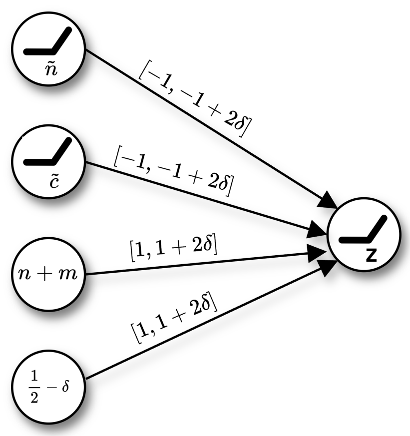


Figure 5: End-gadget

**Lemma 8** (End-gadget)**.** *Consider the* End-*gadget in Fig. 5). For any choice of edge weights, it holds that if* $z < 1/2$ *then*

- $\tilde{n} > n - \delta$*;*
- $\tilde{c} > m - (5\delta + 8\delta^2 + 4\delta^3)$.

*Proof.* We have that

$$z \geq \max\{0, n + m + \frac{1}{2} - \delta - \tilde{n} - \tilde{c}\}.$$

We show that if one of the inequalities in the statement is violated, then $z \geq 1/2$.

Suppose that $\tilde{n} \leq n - \delta$. Then, since $\tilde{c} \leq m$, it follows that $z \geq n + m + 1/2 - \delta - n + \delta - m = 1/2$.

Suppose now that $\tilde{c} \leq m - (5\delta + 8\delta^2 + 4\delta^3) \leq m - \delta = m - \delta$. Then, since $\tilde{n} \leq n$, it follows that $z \geq n + m + 1/2 - \delta - n - m + \delta = 1/2$. □

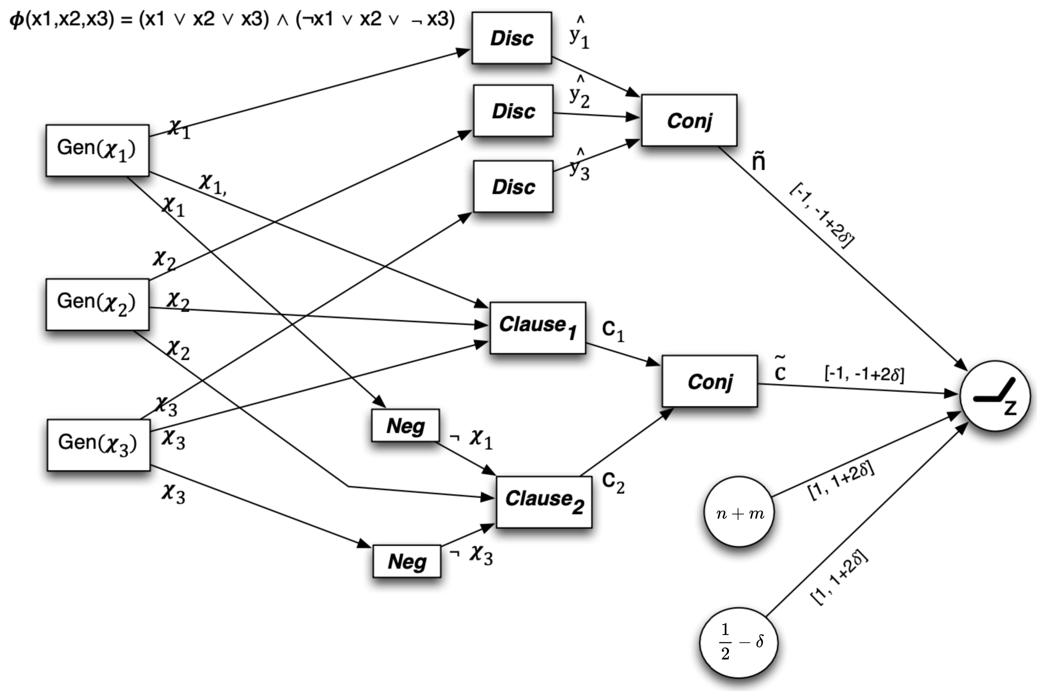


Figure 6: A complete example of the reduction on a simple formula, with $n = 3$ variables and $m = 2$ clauses. All the interval weights not explicitly given are $[1 - 2\delta, 1]$

**The reduction $\mathcal{R} : \phi \mapsto I^\phi = (\mathcal{M}^\phi_{\theta_1}, x^\phi, \Delta^\phi)$.** Fix a 3-CNF $\phi(x_1, \ldots, x_n)$, such that $\phi(\mathbf{a}) = false$, for the assignment $\mathbf{a} = (false, \ldots, false)$. Fix a positive rational number $\delta \leq \frac{1}{12}$. Consider the INN $\mathcal{I} = \mathcal{I}^{\phi}$ built as follows (refer to Fig. 6 for an example of this construction):

1. For each variable $x_i$ add to the network a copy of the *Generating*-gadget (and refer to it as $Gen(\chi_i)$) and a copy of the *Discretizer*-gadget (and refer to it as $Disc_i$).
2. For each $i = 1, \ldots, n$, connect $Gen(\chi_i)$ to $Disc_i$ by identifying the output node $\chi_i$ of $Gen(\chi_i)$ with the non-constant input node $\chi$ of $Disc_i$. Refer to the output node/value of $Disc_i$ as $\hat{y}_i$ (see also Fig. 3).
3. For each clause $C_j = (\lambda_1^{(j)} \vee \lambda_2^{(j)} \vee \lambda_3^{(j)})$ $(j = 1, \ldots, m)$ of $\phi$ add a *Clause*-gadget, henceforth referred to as $Clause_j$. For each $t = 1, 2, 3$,
    - if $\lambda_t^{(j)}$ corresponds to the positive variable $x_i$ then create a connection so that the input of $Clause_j$ that is labelled $\ell_t^{(j)}$ is the output $\chi_i$ of the generating $Gen(\chi_i)$ associated to $x_i$.
    - if $\lambda_t^{(j)}$ corresponds to the negated variable $\neg x_i$ then make a connection so that the input of $Clause_j$ that is labelled $\ell_t^{(j)}$ is the output of a negation gadget, and the input of such negation gadget is the output $\chi_i$ of the Generating-gadget $Gen(\chi_i)$.
4. Add a conjunction gadget such that its inputs are the outputs $\hat{y}_i$ $(i = 1, \ldots, n)$ of the *Discretizer*-gadgets. Let $\tilde{n}$ denote the output of such conjunction gadget.
5. Add a conjunction gadget such that its inputs are the outputs $c_j$ $(i = 1, \ldots, m)$ of the *Clause*-gadgets. Let $\tilde{c}$ denote the output of such conjunction gadget.
6. Finally, add an *End*-gadget (Fig. 5) and connect it to the rest of the network by making the output $\tilde{n}, \tilde{c}$ of the above conjunction gadgets (defined in items 4, 5) coincide with the *End*-gadget inputs marked with $\tilde{n}$ and $\tilde{c}$, respectively.

The above construction defines the topology of the DNN, representing the parametric classifier $\mathcal{M}_{\Theta}$. The classifier $\mathcal{M}_{\theta_1}^{\phi}$ is chosen to be the realisation of $\mathcal{I}^{\phi}$ obtained by setting the weight on each edge $e$ to the middle point of the interval associated to $e$. Such a classifier takes as input $x$ a vector whose components are

- the value in the leftmost node of each Generating-gadget. In the input $x^{\phi}$ defined for our reduction these values are set to 1 as in Fig.7 ;

- the values in the first (top) and the two last (bottom) nodes in each Discretizer-gadget. In the input $x^\phi$ defined for our reduction these values are set to $\delta, (1-\delta)$, and $(1-\delta)$, respectively as in Fig.3;

- the values in the lowest node of each Clause-gadget. In the input $x^\phi$ defined for our reduction these values are set to 1 as in Fig.4;

- the values in the lowest node of each Negation-gadget. In the input $x^\phi$ defined for our reduction these values are set to 1 as in Fig.4;

- the values in two bottom nodes of the End-gadget. In the input $x^\phi$ defined for our reduction these values are set to $n+m$ and $\frac{1}{2}-\delta$, respectively, as in Fig.5.

Finally $\Delta^\phi$ is defined as the set of realisations of $\mathcal{I}$.

It is easy to see that by fixing the value $\delta$ so that it can be encoded by number of bits polynomial in the size of $\phi$, **the instance $I^\phi$ can be constructed from $\phi$ in polynomial time**, since each gadget has a constant size, the number of gadgets is polynomial in the size of the formula, and the input vector $x$ can be described by a number of bits polynomial in the size of $\delta$ and the size of $\mathcal{I}^\phi$.

We first prove a lemma that characterises realisations of $\mathcal{I}$ such that the output of each $\chi_i$ is binary.

**Lemma 9.** *The following two claims characterise the realisations of $\mathcal{I}$ such that for each $i = 1, \ldots, n$, it holds that $\chi_i \in \{0, 1\}$.*

1. *Fix a truth assignment $\mathbf{a}$ such that $\phi(\mathbf{a}) = false$. For any realisation $\mathcal{M}_\theta$ of $\mathcal{I}$ such that for each $i = 1, \ldots, n$ it holds that $\chi_i = 0$ if $a_i = false$ and $\chi_i = 1$ if $a_i = true$ it holds that $\mathcal{M}_\theta(x) \geq \frac{1}{2}$.*
2. *Fix a truth assignment $\mathbf{a}$ such that $\phi(\mathbf{a}) = true$. Then, there exists a realisation $\mathcal{M}_\theta$ of $\mathcal{I}$ such that for each $i = 1, \ldots, n$ it holds that $\chi_i = 0$ if $a_i = false$ and $\chi_i = 1$ if $a_i = true$, and $\mathcal{M}_\theta(x) < \frac{1}{2}$.*

*Proof.* We show the two claims separately.

1. For the first claim, we observe that
   (a) $\tilde{n} \leq n$.

(b) Since $\phi(\mathbf{a}) = false$, there exists $i \in [m]$ such that the assignment $\mathbf{a}$ makes all the literals in the $i$th clause to be false. We also have that the values $\ell_t^{[i]}$s, input to the clause gadget encoding the $i$th clause, will satisfy $\ell_t^{[i]} \in [0, \delta]$—in particular, we have $\ell_t^{[i]} = \chi_j = 0$ if the literal corresponds to some variable $x_j = false$; and, if the the literal correspond to the negation of some variable $x_j = true$, hence $\ell_t^{[i]} = \neg\chi_j$ with $\chi_j = 1$ and by Lemma 5 $\neg\chi_j \in [0, \delta]$.

Therefore, by Lemma 6, we have $c_i \in [0, 5\delta + 6\delta^2]$. It follows that $\tilde{c} < m - \delta$ and

$$z \geq \max\{0, \frac{1}{2} - \delta + n + m - \tilde{n} - \tilde{c}\} \geq \frac{1}{2} - \delta + n + m - n - m + \delta = \frac{1}{2}$$

2. For the second claim, consider the realisation obtained by setting the weights as follows:
   - in the $i$-th generating gadget (the one associated to $x_i$) the weights are chosen in order to have output $\chi = 1$ if $a_i = true$ and $\chi = 0$ if $a_i = false$;
   - in all the other gadgets, the weights are set to the value $w$ such that $|w| = 1$.

   Because of the correspondence $a_i = true \to \chi_i = 1$ and $a_i = false \to \chi_i = 0$, by Lemma 4, we have $\hat{y}_i = 1$ for each $i = 1, \ldots, n$. Hence $\tilde{n} = n$. Because of the choice of the weights being all of the absolute value one, it is also easy to see that, interpreting $true$ as 1 and $false$ as 0, for each $i = 1, \ldots, m$ and $t = 1, 2, 3$, we have an exact correspondence between the truth value assigned by $\mathbf{a}$ to the $t$th literal of the $i$th clause and the value $\ell_t^{[i]}$. Hence, by the assumption that $\phi(\mathbf{a}) = true$, we also have that $c_i = 1$ for each $i = 1, \ldots, m$. It follows that $\tilde{c} = m$.
   Therefore,

$$z = \max\{0, \frac{1}{2} - \delta + n + m - \tilde{n} - \tilde{c}\} = \frac{1}{2} - \delta < \frac{1}{2}$$

□

Let $\mathcal{M}_{\theta_1}$ be the realisation of $\mathcal{I}$ obtained by setting the weight on each edge $e$ to the middle point of the interval associated to $e$. Let $\mathbf{a}$ be the assignment for $\phi$ such that $a_i = false$ for each $i = 1, \ldots, n$. Therefore, $\mathcal{M}_{\theta_1}$

coincides with the realisation of $\mathcal{I}$ such that for each $i = 1, \ldots, n$ it holds that $\chi_i = 0$ and the hypotheses of Lemma 9 are satisfied Hence, by Lemma 9, it holds that $\mathcal{M}_\theta(x) \geq \frac{1}{2}$. We have shown the following.

**Lemma 10.** *For each instance $\phi$ of* 3-NAF-SAT*, the reduction $\mathcal{R}$ produces in polynomial time a proper instance $(\mathcal{M}_{\theta_1}, \mathcal{I}, x)$ of* DRP*.*

In order to complete the proof of the Theorem, the following remains to be shown.

**Lemma 11.** *The formula $\phi$ is satisfiable if and only if there is a realisation $\mathcal{M}_{\theta_2}$ of $\mathcal{I}$, such that $\mathcal{M}_{\theta_2}(x) < \frac{1}{2}$.*

*Proof.* The sufficiency of the condition directly follows from the second claim of Lemma 9, which shows that: *If there exists an assignment* $\mathbf{a}$ *such that $\phi(\mathbf{a}) = true$ then there is a realisation $\mathcal{M}_{\theta_2}$ of $\mathcal{I}$, such that $\mathcal{M}_{\theta_2}(x) < \frac{1}{2}$.*

Let us now focus on the other direction. Assume that there is a realisation $\mathcal{M}_{\theta_2}$ such that $\mathcal{M}_{\theta_2}(x) < \frac{1}{2}$. By Lemma 8, it follows that for the realisation $\mathcal{M}_{\theta_2}(x)$ it holds that $\tilde{n} > n - \delta$ and $\tilde{c} > m - (5\delta + 8\delta^2 + 4\delta^3)$.

Then, by Lemma 7 it follows that

1. for each $i = 1, \ldots, n$ it holds that $\hat{y}_i \in (1-\delta, 1]$, and $\chi_i \in [0, \delta] \cup [1-\delta, 1]$;
2. for each $i = 1, \ldots, m$ it holds that $c_i \in (1 - (5\delta + 8\delta^2 + 4\delta^3), 1]$.

These two conditions together with $\delta < 1/12$ imply, by Lemmas 4, 5 and 6, that for each $i = 1, \ldots, m$, there is $t \in \{1, 2, 3\}$ such that one of the following holds

1. $\ell_t^{[i]} \in [1 - 3\delta - 2\delta^2, 1]$ and $\ell_t^{[i]}$ coincides with some output $\neg\chi_j$ of a negation-gadget, and the input value satisfies $\chi_j \in [0, \delta]$; moreover by construction, then literal $\lambda_t^{[i]}$ is $\neg x_j$;
2. $\ell_t^{[i]} \in [1 - \delta, 1]$ and $\ell_t^{[i]}$ coincides with some output $\chi_j$ of a generating-gadget, whence by construction, then literal $\lambda_t^{[i]}$ is $x_j$

Let $\mathbf{a} = (a_1, \ldots, a_n)$ be the truth assignment defined by

$$a_i = \begin{cases} true & if\ \chi_j \geq 1 - \delta \\ false & if\ \chi_j \leq \delta. \end{cases}$$

Then, for each clause $i = 1, \ldots, m$ at least one literal is set to $true$, and $\phi(\mathbf{a}) = true$. $\square$

The proof is complete. $\square$

From Theorem 2, it follows that deciding whether a CFX $x'$ is not robust to a set of PMC $\Delta$ is NP-complete. We now show that the above reduction can be used to prove the NP-completeness of deciding robustness with respect to a set $\Delta$ of NOMC models. In particular, we have the following:

**Theorem 12** (Hardness of DRP for NOMC)**.** *Given classifier $\mathcal{M}_{\theta_1}$, an input $x$ and a set $\Delta$ of NOMC, deciding whether $\exists\, \mathcal{M}_{\theta_2} \in \Delta$ s.t $\mathbb{E}[\mathcal{M}_{\theta_2}(x)] < \frac{1}{2} \leq \mathcal{M}_{\theta_1}(x)$ is NP-complete.*

*Proof.* The proof of the inclusion in NP is analogous to the one of DRP for PMC models.

For the hardness, we use again a reduction from 3-NAF-SAT: given a 3-CNF $\phi$ that is not satisfied by the all-false assignment, build an interval neural network exactly like in Theorem 2 but for one difference consisting in the interval of weights on the first edge of the Generating-gadget, which are now set to $[0, 2\delta]$ as in Fig. 7.

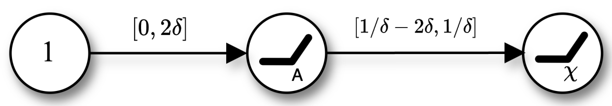


Figure 7: *Generating-gadget* used in this proof.

We denote by $\mathcal{I}_{NOMC}$ this interval neural network. obtained from this reduction starting from a 3-CNF $\phi$. Note that the new generating-gadget can also produce any value in $[0, 1]$. More generally, we have the following important remark.

**Remark 1.** *Lemmas 3-8 also hold for the interval neural network $\mathcal{I}_{NOMC}$.*

We define $\mathcal{M}_{\theta_1}$ to be the realisation of $\mathcal{I}_{NOMC}$ obtained by setting the weight on each edge $e$ to the middle point of the interval associated with $e$. The input value $x$ is defined as in the proof of Theorem 2. We also let $\Delta$ to be the set of realisation of $\mathcal{I}_{NOMC}$.

From the above remark and the proof of Theorem 2, it follows that the 3-CNF $\phi$ has a satisfying assignment if and only if there exists a realisation $\mathcal{M}_{\theta_2}$ in $\Delta$ such that $\mathcal{M}_{\theta_2}(x) < 0.5$.

Then, in order to complete the proof, we only need to show that $\mathcal{M}_{\theta_1}(x) \geq \frac{1}{2}$ and $\Delta$ properly defines a set of NOMC for $\mathcal{M}_{\theta_1}$ (Def. 5), which we recall here for readability purposes:

1. $\mathbb{E}[\mathcal{M}_\theta(x)] = \mathcal{M}_{\theta_1}(x)$; where the expectation is over the randomness[3] of $\mathcal{M}_\theta$;
2. $\mathrm{Var}[\mathcal{M}_\theta(x)] = \nu_x$, where $\nu_x$ represents the maximum variance of the prediction of $\mathcal{M}_\theta(x)$, and whenever $x$ lies on the data manifold $\mathcal{X}$, $\nu_x$ is upper bounded by a small constant $\nu$;
3. If $\mathcal{M}_{\theta_1}$ is Lipschitz continuous for some $\gamma_1$, then $\mathcal{M}_\theta$ is also Lipschitz continuous for some $\gamma_2$.

For a (random or fixed) realisation $\mathcal{M}_\theta$ of $\mathcal{I}_{NOMC}$ and a node $\nu$, let us denote by $\nu_\theta$ the value computed in the node $\nu$ by $\mathcal{M}_\theta$ on input $x$. In accordance with the analysis in Theorem 2, we assume $\delta = 1/12$.

To show that $\mathcal{M}_{\theta_1}(x) \geq \frac{1}{2}$ and that $\Delta$ satisfies property 1 for being a set of NOMC, we prepare the following.

**Lemma 13.** *Let $\mathcal{M}_\theta$ be a random realisation of $\mathcal{I}_{NOMC}$. It holds that*

1. *for the output node $\chi$ of each Generating-gadget, we have $\mathbb{E}[\chi_\theta] = \chi_{\theta_1} = \frac{1}{2} - \delta^2$.*
2. *for the the output node $\neg\chi$ of each Negation-gadget we have $\mathbb{E}[\neg\chi] = \neg\chi_{\theta_1} = \frac{1}{2} - \frac{3}{2}\delta + \delta^2 + \delta^3$.*
3. *for the the output node $\hat{y}$ of each Discretizing-gadget we have $\mathbb{E}[\hat{y}_\theta] = \hat{y}_{\theta_1} = 1 - \delta$*
4. *for the output node $c$ of each Clause-gadget we have $\mathbb{E}[c_\theta] = c_{\theta_1} = 1 - \delta$*
5. *for the output node $\tilde{c}$ of the Conjunction-gadget collecting the outputs of the Clause-gadgets we have $\mathbb{E}[\tilde{c}_\theta] = \tilde{c}_{\theta_1} = m(1-\delta)^2$*
6. *for the output node $\tilde{n}$ of the Conjunction-gadget collecting the outputs of the Discretizing-gadgets we have $\mathbb{E}[\tilde{n}_\theta] = \tilde{n}_{\theta_1} = n(1-\delta)^2$*
7. *for the output node $z$ of the End-gadget, we have $\mathbb{E}[z_\theta] = z_{\theta_1} \geq \frac{1}{2}$.*

*Proof.*

[3] In our case this is a *random realisation of* $\mathcal{I}_{NOMC}$, i.e., a realisation of $\mathcal{I}_{NOMC}$obtained by independently choosing the weight on each edge sampling uniformly at random from the interval associated to that edge.

1. Let $w_1, w_2$ denote the weights on the edges of the Generating-gadget, respectively, in order from left to right. Because of the independence of the choices of the weights of the random realisation $\mathcal{M}_\theta$, we have

$$\mathbb{E}[\chi_\theta] = \mathbb{E}[w_2]\mathbb{E}[A_\theta] = \mathbb{E}[w_2]\mathbb{E}[w_1] = \left(\frac{1}{\delta} - \delta\right)\delta = \frac{1}{2} - \delta^2.$$

   It is immediate to verify that this value is equal to $\chi_{\theta_1}$.
2. Let $w_1, w_2$ denote the weights on the top edge and the bottom edge, respectively, of the Negation-gadget. Using 1. the independence in the choice of the weights, and the fact that with $\delta = \frac{1}{12}$ the argument of the ReLU is always non-negative, we have that

$$\mathbb{E}[\neg\chi_\theta] = \mathbb{E}[\chi_\theta] \cdot \mathbb{E}[w_1] + \mathbb{E}[w_2].$$

   The claim then follows by the fact that the expected values of the weights are given by the middle point of the intervals from which they are respectively taken.
3. Item 1. and the first claim of Lemma 4, together with $\delta = 1/12$ imply that both for a random realisation and for the realisation $\mathcal{M}_{\theta_1}$, the (expected) values computed in nodes $A$ and $B$ of the discretizing-gadget are both 0. Hence, we have $\mathbb{E}[\hat{y}] = \mathbb{E}[w_\theta]$, where $w$ denotes the weight on the lowest edge of the gadget. By noticing that this expected value is equal to the middle point of the interval, we have the desired result.
4. Because of 1. and 2. we have that the expected value (as well as the value computed by $\mathcal{M}_{\theta_1}$ on $x$) of the input nodes $\ell_t^{[i]}$ of each clause-gadgets are from the set $\{\frac{1}{2} - \delta^2, \frac{1}{2} - \frac{3}{2}\delta + \delta^2 + \delta^3\}$ It follows that the argument of the ReLU function computed in the node $A$ is negative. Hence, we have $\mathbb{E}[c] = \mathbb{E}[w_\theta]$, where $w$ denotes the weight on the lowest edge of the gadget. Again, noticing that this expected value is equal to the middle point of the interval gives the desired result.
5. For a realisation $\mathcal{M}_\theta$ let $w_{\theta,i}$ denote the weight on the $i$th edge (counting from top to bottom) of the conjunction-gadget collecting the outputs of the Clause-gadgets, and $c_{\theta,i}$ the output value of the $i$th clause gadget, as computed by $\mathcal{M}_\theta$ on input $x$. Then, we have

$$\mathbb{E}[\tilde{c}_\theta] = \sum_{i=1}^{m} \mathbb{E}[c_{\theta,i}] \cdot \mathbb{E}[w_{\theta,i}].$$

The results follow from 4. and the fact that the expected value of a uniformly sampled weight is equal to the middle point of the interval from which it is taken.

6. The proof of this point is analogous to the proof of 5.
7. For a realisation $\mathcal{M}_\theta$ let $w_{\theta,i}$ denote the weight on the $i$th edge (counting from top to bottom) of the End-gadget. We start by observing that from the results of the previous points, it follows that the argument of the RELU function in node $z$ is always non-negative. Hence, we have

$$\mathbb{E}[z_\theta] = \mathbb{E}[\tilde{n}_\theta]\cdot\mathbb{E}[w_{\theta,1}]+\mathbb{E}[\tilde{c}_\theta]\cdot\mathbb{E}[w_{\theta,2}]+(n+m)\cdot\mathbb{E}[w_{\theta,3}]+\left(\frac{1}{2}-\delta\right)\cdot\mathbb{E}[w_{\theta,4}]$$

Then, the equality $\mathbb{E}[z_\theta] = z_{\theta_1}$ in the claim follows again from the fact that the expected values of the weights of a random realisation are equal to the middle point of the interval, i.e., the value of the weight on the edge in the realisation $\mathcal{M}_{\theta_1}$. The inequality in the claim follows from Lemma 8 since, with $\delta = 1/12$, it holds that $n(1-\delta)^2 < n - \delta$.

□

Claim 7 of the lemma directly implies that the first property of an NOMC is satisfied by $\Delta$, i.e., $\mathbb{E}[\mathcal{M}_\theta(x)] = \mathcal{M}_{\theta_1}(x)$. The same claim also proves that $\mathcal{M}_{\theta_1}(x) \geq \frac{1}{2}$.

For the second property, namely $\mathrm{Var}[\mathcal{I}(x)] = \nu_x$, we use the uniform continuity of the function computed by the realisations of $\mathcal{I}_{NOMC}$, which is a direct consequence of being linear combinations of RELU functions which are Lipschitz continuous functions, hence uniform continuous. By the Extreme Value Theorem (see, e.g., [36, Thm. 4.16]) any realisation of $\mathcal{I}_{NOMC}$ will be bounded and achieve its minimum and maximum on the compact domain, and thus the variance will be indeed bounded.

Finally, the last property follows from the fact that the linear composition of Lipschitz continuous operations (ReLU) is also Lipschitz continuous, which is indeed the case of any realisation of $\mathcal{I}_{NOMC}$. □

Summarizing, we have shown the following.

**Corollary 14.** *Given a model $\mathcal{M}_\theta$, a CFX $x'$ and a set of either NOMC or PMC model changes $\Delta$, the problem of verifying the $\Delta$-robustness of $x'$ is NP-complete.*

These hardness results motivate the introduction of novel approximate solutions to estimate the robustness of a counterfactual under a set of PMC $\Delta$.

## 4. Probabilistic Guarantees for Existing Notions of Model Changes

As we have established in the previous section, exact methods for computing robustness under model changes are bound to lack scalability. This motivates the design of approximate and/or probabilistic approaches to solve the problem. Previous work by Hamman et al. [15] presented an approach to obtain counterfactual explanations that are probabilistically robust under NOMC. A natural question that arises then is whether guarantees obtained for NOMC also transfer to the PMC setting. As we show for the first time below, this is not the case in general.

**Lemma 15.** *Naturally-Occurring model changes may not be Plausible, and vice-versa.*

*Proof.* Consider the DNN $\mathcal{M}_\theta$ depicted in Fig. 8 (a) with two input nodes, one hidden layer with two ReLU nodes[4] and one single output. The parameters $\theta = [w_1, \ldots, w_6]$ are the weights on the edges listed top-bottom and left-right.


Figure 8: (a) The model $\mathcal{M}_\theta$ used as an example to prove the lemma. (b) An interval neural network representing the realisations that can be obtained from $\mathcal{M}_\theta$ considering a set of PMC $\Delta_\delta$ with $\delta = 0.3$.

Propagating an input vector $x = [x_1, x_2]^T$ through $\mathcal{M}_\theta$, we obtain $\mathcal{M}_\theta(x) = y = w_5 \cdot \max\{0, w_1 \cdot x_1 + x_2 \cdot w_3\} + w_6 \cdot \max\{0, w_2 \cdot x_1 + x_2 \cdot w_4\}$. Now assume

[4] In this proof, we consider a DNN with only ReLU activation functions. However, we notice that it is possible to have a similar counterexample even with other activations, e.g., Tanh, Sigmoid.

an input vector $x = [0.9, 0.9]^T$ and weights $w_1 = 1, w_2 = 0, w_3 = 0, w_4 = 0.6, w_5 = 1, w_6 = -1$. The corresponding output generated by the DNN is $\mathcal{M}_\theta(x) = 0.46$. A counterfactual for $x$ could be given as a new input vector $x' = [1, 0.8]^T$, for which we obtain $\mathcal{M}_\theta(x') = 0.52 > 0.5$. Now, following Definition 6, we consider a set of plausible model changes obtained for $\delta = 0.3$. This can be captured by defining on each weight $w_i$ the corresponding interval in $[w_i - \delta, w_i + \delta]$ depicted in Fig. 8 (b) that represents the set of all the possible models obtained from $\mathcal{M}_\theta$, replacing each $w_i$ with a weight in the interval $[w_i - \delta, w_i + \delta]$. We then have that the expected result of a model $\mathcal{M}_{\theta'}$ sampled uniformly from such a set satisfies:

$$\begin{aligned}
\mathbb{E}[\mathcal{M}_{\theta'}(x')] = \; & \mathbb{E}[w_5] \cdot \mathbb{E}[\mathrm{ReLU}(x_1 \cdot w_1 + x_2 \cdot w_3)] + \\
& \mathbb{E}[w_6] \cdot \mathbb{E}[\mathrm{ReLU}(x_1 \cdot w_2 + x_2 \cdot w_4)] \\
= \; & \mathbb{E}[[0.7, 1.3]] \cdot \mathbb{E}[\max\{0, x_1 \cdot [0.7, 1.3] + \\
& x_2 \cdot [-0.3, 0.3]\}] + \mathbb{E}[[-1.7, -0.3]] \cdot \\
& \mathbb{E}[\max\{0, x_1 \cdot [-0.3, 0.3] + x_2 \cdot [0.3, 0.9]\}] \\
& > 0.52 \neq \mathcal{M}_\theta(x')
\end{aligned}$$

Definition 5 states that a model change is naturally occurring if $\mathbb{E}[\mathcal{M}_{\theta'}(x)] = \mathcal{M}_\theta(x)$. This implies that $\Delta$ contains models that cannot be characterized as naturally occurring model changes. Vice versa, the existence of Naturally-Occurring model changes not being plausible is implicit in the definition, and for the sake of completeness, we provide an example network in Fig. 9.

Consider a DNN having a single input value $x$ and a single parameter $\theta$ and computing the function $\mathcal{M}_\theta(x) = \mathrm{ReLU}(0.5 - \mathrm{ReLU}(x - \theta))$

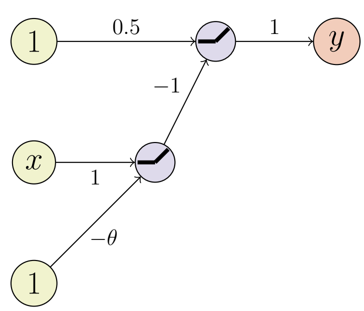


Figure 9: The DNN considered in this proof

Table 1: Empirical evaluation across model perturbations of increasing magnitude $\delta$ and different sample sizes $n$.

| | ***Credit*** | | | | ***Spam*** | | | | ***News*** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $n = 1000$ | | $n = 10000$ | | $n = 1000$ | | $n = 10000$ | | $n = 1000$ | | $n = 10000$ | |
| | Avg diff. | Rej. (%) | Avg diff. | Rej. (%) | Avg diff. | Rej. (%) | Avg diff. | Rej. (%) | Avg diff. | Rej. (%) | Avg diff. | Rej. (%) |
| $\delta = 0.05$ | 0.008 | 90 | 0.022 | 90 | 0.018 | 50 | 0.017 | 70 | 0.034 | 70 | 0.033 | 80 |
| $\delta = 0.1$ | 0.017 | 100 | 0.047 | 100 | 0.034 | 100 | 0.035 | 100 | 0.064 | 80 | 0.063 | 100 |
| $\delta = 0.2$ | 0.046 | 100 | 0.086 | 100 | 0.0748 | 90 | 0.064 | 100 | 0.127 | 90 | 0.141 | 100 |
| $\delta = 0.3$ | 0.110 | 100 | 0.140 | 90 | 0.121 | 100 | 0.087 | 100 | 0.207 | 90 | 0.173 | 100 |

Fix a data set $\mathcal{X}$ and let $\theta = \max_{x \in \mathcal{X}} x$. Let us consider the set of model changes $\Sigma = \{S_\tau \mid \tau \in \mathbb{R}_+\}$ defined by $S_\tau(\mathcal{M}_\theta) = \mathcal{M}_{\theta+\tau}$. Clearly for any $\tau \geq 0$, we have

$$\mathcal{M}_{\tau+\theta}(x) = \mathcal{M}_\theta(x) = 0.5,$$

for any $x \in \mathcal{X}$. This trivially implies that $\Sigma$ is a set of naturally occurring model changes (all changes considered have exactly the same value in all points in $\mathcal{X}$).

The claim now follows by observing that there is no finite $\delta$ such that the corresponding set of plausible model changes $\Delta = \{S \mid d_p(\mathcal{M}_\theta, S(\mathcal{M}_\theta)) \leq \delta\}$ contains $\Sigma$.

□

Lemma 15 shows the existence of witnesses proving that Definition 5 (NOMC) and Definition 6 (PMC) may capture very different model changes in general. To complement this observation, we also ran experiments to determine how often these definitions disagree empirically. In particular, we considered three binary classification datasets commonly used in Explainable AI:

- the *credit* dataset [37], which is used to predict the credit risk of a person (good or bad) based on a set of attributes describing their credit history;
- the *spambase* dataset [38] is used to predict whether an email is to be considered spam or not based on selected attributes of the email;
- the *online news popularity* dataset [39], referred to as *news* in the following, is used to predict the popularity of online articles.

We trained a neural network classifier with two hidden layers (20 and 10 neurons, respectively) for each dataset and used a Nearest-Neighbor Counterfactual Explainer [40] to generate counterfactual explanations for 10 different inputs. After generating a counterfactual, we produce $n$ different perturbations $\mathcal{M}_{\theta'}$ of the original neural network $\mathcal{M}_{\theta}$ for $n \in \{1000, 10000\}$ under *plausible* model change with $\Delta \in \{0.05, 0.1, 0.2, 0.3\}$. We then considered two measures:

- average difference in output between $\mathcal{M}_{\theta}$ and $\mathcal{M}_{\theta'}$, for each of the $n$ model $\mathcal{M}_{\theta'}$ and across all CFXs;
- for each counterfactual, we perform a one-sided t-test to check whether the average prediction generated by $n$ models $\mathcal{M}_{\theta'}$ equals the original prediction of $\mathcal{M}_{\theta}$. We report the percentage of CFXs for which the null hypothesis was rejected (p-value used 0.05).

Table 1 reports our results. We observe that the requirement that the expected output of perturbed models remains equal to the original prediction is often violated. These results complement the result of Lemma 15, confirming that the two notions indeed capture two different settings in general. In particular, our results show that (probabilistic) methods devised for NOMC may fail to guarantee robustness under PMC, thus motivating the development of dedicated approaches for probabilistic guarantees under PMC. Indeed, having clarified the relationship between the two notions of model changes, in the following, we focus on certification approaches for robustness under PMC, presenting a novel approximate solution with probabilistic guarantees.

## 5. Robustness under PMC with Probabilistic Guarantees

Jiang et al. [12, 33] proposed to use INNs to enable a compact representation of a superset of the models that can be obtained by a perturbation of the starting model under a set $\Delta$. By exploiting an exact reachable set computation method, e.g., based on MILP [41], the authors could determine whether or not a CFX is robust under the chosen $\Delta$ via a single forward propagation of the CFX. However, in view of the NP-hardness of the problem discussed in the § 3 and the typical non-linear nature of the classifiers, it presents some computational limitations.

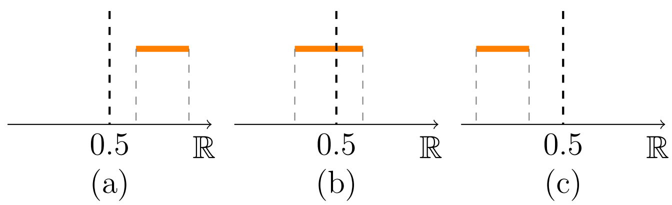


Figure 10: Visual representation of the possible output reachable set for an interval abstraction for a binary classification model. (a) For a given $\Delta$, we classify an input as 1 (robust) if the output range for that input is always greater 0.5. Otherwise, the input is classified as 0, i.e., not robust (b),(c).

In general, interval neural networks map inputs to intervals representing an over-approximation of all possible outcomes that can be produced by any shifted model $\mathcal{M}_{\theta'}$ obtained under $\Delta$. Given this property, if the output reachable set is completely disjoint from the decision threshold 0.5, then one can assert – in a sound and complete fashion – whether or not a given CFX is robust (Fig. 10 (a,c)). On the other hand, if we run into a situation such as the one depicted in Fig. 10 (b), one cannot assert robustness with certainty. In this scenario, Jiang et al. [12] propose to classify the CFX as not robust, which preserves the soundness of their result. Nonetheless, this might lead to discarding a CFX even when the actual probability that after retraining, we incur in plausible model changes for which the CFX is not robust is extremely low. As we will show in § 6, this worst-case notion of robustness affects the CFXs generated by [12], which may end up being unnecessarily expensive (in terms of proximity) and having low plausibility. Additionally, computing the exact output reachable set of an interval abstraction may be costly (e.g., MILP is known to be NP-hard). This is expected: Theorems 2 and 3 show that there is no polynomial time algorithm able to return an exact estimate of the fraction of plausible changes for which the CFX is robust (hence a fortiori deciding whether it is $\Delta$-robust), unless P=NP. In the following, we propose a novel certification approach that aims to alleviate this problem.

### *5.1. A Provable Probabilistic Approach*

One possible idea to avoid exact reachable set computation to determine the robustness of a CFX under PMC is to use naive interval propagation. Given an input CFX, we propagate this input through the network, keeping track of all the possible activation values that can be obtained under $\Delta$ until the output layer is reached. However, the non-linear and non-convex nature of DNNs may result in a significant overestimation of the actual reachable set, thus resulting in a spurious decision of non-robustness. In such cases, a CFX may end up being labeled as non-robust even though the CFX is actually robust. Additionally, even with exact methods, a CFX may be discarded even though the fraction of plausible model changes in $\Delta$ for which the CFX is not robust is negligible.

To avoid these problems, we propose an approximate certification approach based on Monte-Carlo sampling that draws sample realisations directly from $\Delta$ to obtain an underestimation of the space of possible classifications under PMC. The idea of using a sample-based approach stems from the fact that the $\Delta$ set, representing all the plausible model changes, abstracts an infinite number of models to test. As testing this infinite number of models may be impossible in practice, efficient sampling-based solutions hold great promise. In detail, given a CFX $x'$ we can compute an underestimation of the output reachable set under $\Delta$ by sampling $n$ random realisations $\mathcal{M}_{\theta_1}, \ldots, \mathcal{M}_{\theta_n}$ from $\Delta$, and compute the output reachable set by taking, respectively, the $\min_i \mathcal{M}_{\theta_i}(x')$ and the $\max_i \mathcal{M}_{\theta_i}(x')$ for $i \in \{1, \ldots, n\}$.

This approach is very effective and allows us to obtain an estimate of the output reachable set without using an exact solver. Nonetheless, the number $n$ of realisation to sample in order to achieve a good reachable set estimation remains unclear, as well as what kind of guarantees one could obtain from this approach. To answer these questions, we leverage previous results on the *statistical prediction of tolerance limits* [42]. Indeed, we observe that for each realisation $\mathcal{M}_{\theta_i}$ sampled from $\Delta$, the resulting output of the DNN $\mathcal{M}_{\theta_i}(x')$ can be interpreted as an instantiation of a random variable $X$ whose tolerance interval we are trying to estimate. Following this observation, we can derive a probabilistic bound on the correctness of the solution returned from $n$ samples, using the following lemma based on [42]:

**Lemma 16.** *Fix an integer $n > 0$ and an approximation parameter $R \in (0, 1)$. Given a sample of $n$ models $\mathcal{M}_{\theta_1}, \ldots \mathcal{M}_{\theta_n}$ from the (continuous) set of possible realisations $\Delta$, the probability that for at least a fraction $R$ of the models in a further possibly infinite sequence of samples $\mathcal{M}^{(2)}_{\theta_1}, \ldots \mathcal{M}^{(2)}_{\theta_m}$ from $\Delta$ we have*

$$\min_i \mathcal{M}^{(2)}_{\theta_i}(x) \geq \min_i \mathcal{M}_{\theta_i}(x) \tag{1}$$

$$\textit{(respectively } \max_i \mathcal{M}^{(2)}_{\theta_i}(x) \leq \max_i \mathcal{M}_{\theta_i}(x)\textit{)}$$

*is given by* $\alpha = n \cdot \int_R^1 x^{n-1}\, dx = 1 - R^n$.

Informally, Lemma 16 allows us to derive the minimum number $n$ of realisations that it is enough to sample and check in order to guarantee that with probability $\alpha$ at least a fraction $R$ of the models in $\Delta$ satisfy the robustness property. More precisely, from these $n$ realisations, we can obtain an underestimation of the reachable set of any realisation in $\Delta$ that is guarantee to be correct with confidence $\alpha$ for at least a fraction $R$ of a possibly infinite further sample of realisations from $\Delta$. In practice, if we set, e.g. $\alpha = 0.999$ and $R = 0.995$, we can derive $n$ as $n = \log_R(1-\alpha) = 1378$. After having selected 1378 random realisations from $\Delta$, if the lower bound of the underestimated reachable set computed as $\min_i \mathcal{M}_{\theta_i}(x')$ is greater than 0.5, then with probability $\alpha = 0.999$, $R$ is a lower bound on the fraction of plausible model changes in $\Delta$ for which $x'$ is robust. In other words, Lemma 16 allows us to assert with a confidence $\alpha$ that $x'$ is not $\Delta$-robust for at most a fraction $(1-R) = 0.05$ of models from $\Delta$.

### *5.2. The* `AP`$\Delta$`S` *Algorithm*

Using the result of Lemma 16, we now present our approximation method `AP`$\Delta$`S` to generate probabilistic robustness guarantees. The procedure, shown in Algorithm 1, receives as input a model $\mathcal{M}_\theta$, a CFX $x'$ for which robustness guarantees are sought, and the two confidence parameters $\alpha, R$. The algorithm then searches for the largest $\delta_{max}$ such that, with probability $\alpha$, the CFX $x'$ is robust for at least a fraction $R$ of the set of plausible model changes $\Delta = \{S \mid d_p(\mathcal{M}_\theta, S(\mathcal{M}_\theta)) \leq \delta_{max}\}$.

The algorithm starts by computing the size $n$ of a sample of realisations that is sufficient to guarantee the condition in Lemma 16 (line 3). `AP`$\Delta$`S` then initializes a small $\delta_{init}$ and checks if $x'$ is at least robust to a small model shift. To this end, it employs `realisations`$(\mathcal{M}_\theta, x', \delta, n)$ which samples $n$ realisations, pertubating each model parameter by at most a factor $\delta$ and checks if for each of these realisation $\mathcal{M}_{\theta_i}(x') \geq 0.5$, thus computing a robustness rate. If not all these realisations result in a robust outcome, thus achieving a final rate not equal to 1, the algorithm discards the CFX $x'$ as non-robust (lines 6-8). Otherwise, it combines an exponential search (lines 9-12) and a binary search (lines 13-24) to find $\delta_{\max}$. At each step of this search, the procedure checks whether for each of the $n$ realisations from $\Delta = \{S \mid d_p(\mathcal{M}_\theta, S(\mathcal{M}_\theta)) \leq \delta_{max}\}$ the condition $\mathcal{M}_{\theta_i}(x') \geq 0.5$ is verified.

**Algorithm 1** Approximate Plausible $\Delta$-Shift ($\texttt{AP}\Delta\texttt{S}$)

**Input:** Model $\mathcal{M}_\theta$, set of PMC $\Delta$, CFX $x'$, $\alpha$, $R$, $\delta_{init}$
**Output:** $\delta_{max}$

$n \leftarrow \log_R(1-\alpha)$ ▷ number of samples
rate $\leftarrow$ $\texttt{realisations}(\mathcal{M}_\theta, x', \delta_{init}, n)$
**if** rate $\neq 1$ **then**
    **return** 0 ▷ not robust for $\delta_{init}$
**end if**

$\delta \leftarrow \delta_{init}$
**while** rate $= 1$ **do**
    $\delta \leftarrow 2\delta$
    rate $\leftarrow$ $\texttt{realisations}(\mathcal{M}_\theta, x', \delta, n)$
**end while**
▷ we exit from the while because we have found at least one model in the realisations with an output $< 0.5$, and we have $[\delta/2,\ \delta)$ to search for a $\delta_{max}$.
$\delta_{max} \leftarrow \delta/2$
**while** True **do**
    **if** $|\delta - \delta_{max}| \leq \delta_{init}$ **then**
        **return** $\delta_{max}$
    **end if**
    $\delta_{new} \leftarrow (\delta_{max} + \delta)/2$
    rate $\leftarrow$ $\texttt{realisations}(\mathcal{M}_\theta, x', \delta_{new}, n)$
    **if** rate $= 1$ **then**
        $\delta_{max} \leftarrow \delta_{new}$
    **else**
        $\delta \leftarrow \delta_{new}$
    **end if**
**end while**

**Proposition 17.** *Fix $\delta_{init} > 0$. Given a model $\mathcal{M}_\theta$ and a CFX $x'$, let $\delta^*$ be the (exact) maximum magnitude of model changes such that $x'$ is robust with respect to the set of PMC $\Delta_{\delta^*} = \{S \mid d_p(\mathcal{M}_\theta, S(\mathcal{M}_\theta)) \leq \delta^*\}$. Then, with probability $\alpha$, $\texttt{AP}\Delta\texttt{S}$ returns a $\delta_{\max} \geq \delta^* - \delta_{init}$ such that the CFX $x'$ is robust for at least a fraction $R$ of the set of PMC $\Delta_{\delta_{\max}}$. Moreover, the computation of $\delta_{max}$ is polynomial.*

*Proof sketch.* The $\delta_{\max}$ returned by the algorithm is obtained by iteratively increasing $\delta$, sampling $n$ models from the corresponding $\Delta_\delta$ and verifying that

$\mathcal{M}_{\theta_i}(x) \geq 0.5$ for each model $\mathcal{M}_{\theta_i}$ sampled. By definition, $\delta^*$ is the actual value we are trying to estimate. When the algorithm stops, with values $\delta_{\max}$ and $\delta$ such that $\delta - \delta_{\max} \leq \delta_{init}$, we have that for $n$ models in $\Delta_{\delta_{\max}}$ the CFX $x'$ is robust and for at least one model in $\Delta_\delta$ the $x'$ is not robust. Since for each model in $\Delta_{\delta*}$ the CFX is robust, it must hold that $\delta > \delta^*$, hence $\delta_{\max} \geq \delta^* - \delta_{init}$.

Once the exponential search ends, by exploiting Lemma 16, we can state that with probability $\alpha$, the CFX $x'$ is robust for at least $R$ of any infinite further realisations from $\Delta_{\delta_{max}}$. The time complexity of the algorithm corresponds to $n \cdot m$ forward propagations, with $n$ being the sample size and $m = \log \frac{\delta_{\max}}{\delta_{init}}$ being the number of iterations of the exponential search, which is polynomial in the input size of the problem. □

## 6. Experimental Analysis

Section 5 laid the theoretical foundations of a novel sampling-based method that allows the obtaining of provable probabilistic guarantees on the robustness of CFXs. In this section, we evaluate our approach by considering five experiments:

- In § 6.1 we show how to instantiate AP$\Delta$S in practice using a synthetic example. Specifically, we first demonstrate the interplay of parameters $n$, $\alpha$, and $R$ used to obtain a probabilistic guarantee. Then, using the maximum $\delta_{max}$ discovered by AP$\Delta$S , we precisely characterize the subsets $\hat{\Delta}$ of the set of PMC $\Delta_{\delta_{max}}$ for which the given CFX $x'$ cannot be proved to be robust. In our experiments at most a fraction $(1 - R)$ of $\Delta_{\delta_{max}}$ is in $\hat{\Delta}$, so complementing empirically our theoretical results.

- In § 6.2 we compare our certification approach with the one proposed in [12]. In particular, we focus on the difference between the worst-case guarantees offered by their approach and compare them with the average-case guarantees of AP$\Delta$S in terms of maximum changes that can be certified. These experiments confirm our intuition that worst-case guarantees might be too conservative in practice, leading to a larger number of CFXs being discarded.

- In § 6.3, we consider the problem of generating robust CFXs and compare with two state-of-the-art approaches for robustness under PMC, [12] and [11]. We show that our approach produces CFXs that are less expensive (in terms of $\ell_1$ distance from the original input) and more plausible, without sacrificing robustness.

- In § 6.4, we perform an in-depth analysis of the impact that the two main hyper-parameters of `AP∆S` , $\alpha$ and $R$, have on the quality of generated CFXs. We show that higher values of there parameters typically lead to tighter estimates that result in improved robustness. These results also align with existing literature on CFXs in revealing that improved robustness appears to be correlated with higher plausibility and cost.
- Finally, in § 6.5 we analyse the scalability of `AP∆S` . We consider tabular transformer architectures, such as *TabNet* [17] and show that `AP∆S` scales well even when employed in recent architectures employed at the state of the art and containing hundreds of thousands of parameters, thus confirming the wide applicability of our method.

An implementation of `AP∆S` is integrated in the *RobustX* library [43] available at `https://github.com/RobustCounterfactualX/RobustX.git`. Additional material is available at `https://github.com/lmarza/APAS`.

### 6.1. `AP∆S` in Action

This experiment is designed to demonstrate how the three main parameters of `AP∆S` , i.e., $n, \alpha$, and $R$, can be used to obtain probabilistic robustness guarantees. To this end, we focus on the synthetic example depicted in Fig. 11. Weights for the original network $\mathcal{M}_\theta$, as well as the input used for testing robustness, are generated randomly.

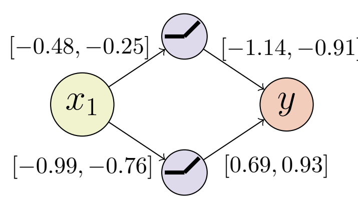


Figure 11: The interval neural network used for exact enumeration.

Considering a random input $x = -2.57$, we use `AP∆S` to estimate a $\delta_{max}$ for which we seek the guarantee that for at least $R = 90\%$ of the

**Algorithm 2** *Exact CFX $\Delta$-Robustness*

**Input:** An INN $\mathcal{N}$ and a CFX $x'$ and an maximum $\epsilon$-precision for the splitting phase
**Output:** set of INNs for which $x'$ is robust.
robust_INNs $\leftarrow \emptyset$
non-robust_INNs $\leftarrow \emptyset$
unknown $\leftarrow$ `Push`($\mathcal{N}$)
**while** (unknown $\neq \emptyset$) or ($\epsilon$-precision not reached) **do**
    $\mathcal{I} \leftarrow$ `GetINNToVerify`(unknown)
    $\mathcal{R}_{\mathcal{I}} \leftarrow$ `ComputeReachableSet`($\mathcal{I}$, $x'$)
    **if** `lower`($\mathcal{R}_{\mathcal{I}}$) $\geq 0.5$ **then**
        robust_INNs $\leftarrow$ `Push`($I$)
        unknown $\leftarrow$ `Pop`($\mathcal{I}$)
    **else if** `upper`($\mathcal{R}_{\mathcal{I}}$) $< 0.5$ **then**
        non-robust_INNs $\leftarrow$ `Push`($\mathcal{I}$)
        unknown $\leftarrow$ `Pop`($\mathcal{I}$)
    **else**
        $\mathcal{I}', \mathcal{I}'' \leftarrow$ `ChooseIntervalToSplit`($\mathcal{I}$)
        unknown $\leftarrow$ `Push`($\mathcal{I}', \mathcal{I}'$)
    **end if**
**end while**
**return** robust_INNs

plausible model changes induced by such $\delta_{max}$ the CFX $x'$ is robust. Following Proposition 17, we set a confidence level $\alpha > 1 - 10^{-40}$ (i.e., with certainty, in practice), which yields $n = 100k$ realisations. For this setting, `AP`$\Delta$`S` identifies a $\delta_{max} = 0.115$.

To validate this result, we define a procedure to exactly characterize, following the intuitions of [44, 45], the models within $\Delta_{\delta_{max}}$ for which the robustness property does not hold. The interval abstraction proposed by [12] can be used to exactly compute the portion of the model changes from $\Delta$ for which a CFX $x'$ is not robust. In fact, it is possible to build an interval neural network using the $\delta_{max}$ value identified by `AP`$\Delta$`S` , setting each weight $w_i$ in $\theta$ to $[w_i - \delta_{max}, w_i + \delta_{max}]$. Then, recursively splitting each interval weight of the network in half allows to identify portions of $\Delta$ that are not robust. Employing the following strategy (reported in Algorithm 2), after $s = 7$ splits, we obtain that for $\sim 92\%$ of sub-interval networks, the CFX is robust. The remaining 8% produced an *unknown* answer (i.e., the situation

Table 2: Comparison on the robustness of CFXs using five state-of-the-art methods and AP$\Delta$S proposed in this work.

| | ***Diabetes*** | | | | ***no2*** | | | | ***SBA*** | | | | ***Credit*** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **VM1** $\delta = 0.11$ | **VM2** $\delta_e = 0.27$ | $\ell_1$ | **lof** | **VM1** $\delta = 0.02$ | **VM2** $\delta_e = 0.07$ | $\ell_1$ | **lof** | **VM1** $\delta = 0.11$ | **VM2** $\delta = 0.25$ | $\ell_1$ | **lof** | **VM1** $\delta = 0.05$ | **VM2** $\delta_e = 1.28$ | $\ell_1$ | **lof** |
| Wacht-R | 100% | 100% | 0.122 | 1.00 | 100% | 100% | 0.084 | 1.00 | 92% | 92% | 0.023 | -0.78 | - | - | - | - |
| Proto-R | 100% | 96% | 0.104 | 1.00 | 100% | 100% | 0.069 | 1.00 | 90% | 88% | 0.011 | -0.02 | 32% | 30% | 0.300 | -1.00 |
| MILP-R | 100% | 100% | 0.212 | -0.48 | 100% | 100% | 0.059 | 1.00 | 100% | 100% | 0.018 | -0.88 | 100% | 100% | 0.031 | 1.00 |
| ROAR | 82% | 14% | 0.078 | 0.95 | 88% | 34% | 0.074 | 1.00 | 82% | 78% | 0.031 | -0.80 | 62% | 60% | 0.047 | 1.00 |
| AP$\Delta$S | 100% | 100% | 0.072 | 1.00 | 100% | 100% | 0.042 | 1.00 | 100% | 100% | 0.009 | 0.44 | 100% | 94% | 0.028 | 1.00 |

depicted in Fig. 10(b)) that would require further splits, corresponding to only ten nodes to explore in the next iteration. In the worst case, even considering all the remaining ten nodes left to explore as non-robust, we would have a maximum percentage of non-robustness still lower than the desired upper bound $(1-R) = 10\%$, confirming that the guarantees produced by AP$\Delta$S indeed hold in practice.

### *6.2. Worst-case vs Average-case Guarantees*

This set of experiments aims to compare the probabilistic guarantees offered by AP$\Delta$S with the worst-case guarantees offered by [12]. What we aim to show here is that adopting an average-case certification perspective may be more practical in some circumstances, as worst-case guarantees may be unnecessarily conservative. Our approach aims to obtain a $\delta_{max}$ for which the CFX is robust with confidence $\alpha$ for at least a fraction $R$ of model changes in $\Delta$. This is in stark contrast with the worst-case reasoning of [12], where even a single realisation of $\Delta$ for which the CFX is not robust results in the corresponding $\delta$ being discarded.

To show why such strict guarantees may not be needed, we use an analogous experimental setup and the training process of [12], which considers four datasets: *Diabetes* (continuous) [46], *Credit* (heterogeneous) [37], *no2* (continuous) [47] and *Small Business Administration* (SBA) (continuous features) [48]. In detail, for the training procedure of the classifier, we randomly shuffle each dataset and split it into two halves, denoted $\mathcal{D}_1$ and $\mathcal{D}_2$. First, we use $\mathcal{D}_1$ to train a base neural network; then we use both $\mathcal{D}_1$ and $\mathcal{D}_2$ to train a shifted model. We then generate 50 robust CFXs for the base network using the MILP-R and the same $\delta$ values as in [12] for a fair comparison.

**Algorithm 3** Provable Plausible $\Delta$-Shift

**Input:** Model $\mathcal{M}_\theta$, CFX $x'$, $\alpha$, $R$
**Output:** $\delta_{max}$
$\delta_{init} \leftarrow 0.0001$
rate $\leftarrow$ `MILP`$(\mathcal{M}_\theta, x', \delta_{init})$
**if** rate $\neq 1$ **then**
    **return** 0 $\triangleright$ no robustness
**end if**
$\delta \leftarrow \delta_{init}$
**while** rate $= 1$ **do**
    $\delta \leftarrow 2\delta$
    rate $\leftarrow$ `MILP`$(\mathcal{M}_\theta, x', \delta)$
**end while**
$\delta_{max} \leftarrow \delta/2$
**while** True **do**
    **if** $|\delta - \delta_{max}| \leq \delta_{init}$ **then**
        **return** $\delta_{max}$
    **end if**
    $\delta_{new} \leftarrow (\delta_{max} + \delta)/2$
    rate $\leftarrow$ `MILP`$(\mathcal{M}_\theta, x', \delta_{new})$
    **if** rate $= 1$ **then**
        $\delta_{max} \leftarrow \delta_{new}$
    **else**
        $\delta \leftarrow \delta_{new}$
    **end if**
**end while**

Specifically, we use $\delta = 0.11$ for *Diabetes*, $\delta = 0.02$ for *no2*, $\delta = 0.11$ for *SBA* and $\delta = 0.05$ for *Credit*. Subsequently, we evaluate the resulting CFXs by looking at two metrics: *(i)* **VM1**, the percentage of CFXs that are valid on the base neural network and *(ii)* **VM2**, the percentage of CFXs that remain valid for the shifted neural network trained using both $\mathcal{D}_1$ and $\mathcal{D}_2$. Table 2 reports the results we obtained for this experiment.

As previously observed by Jiang et al. [12], the training procedure used to generate shifted models may result in changes that exceed the $\delta$ used to generate provably robust CFXs. Indeed, after inspecting the networks obtained, we noted that the maximum empirical difference observed after retraining (denoted as $\delta_e$) is well above the $\delta$ values used during CFX generation. In

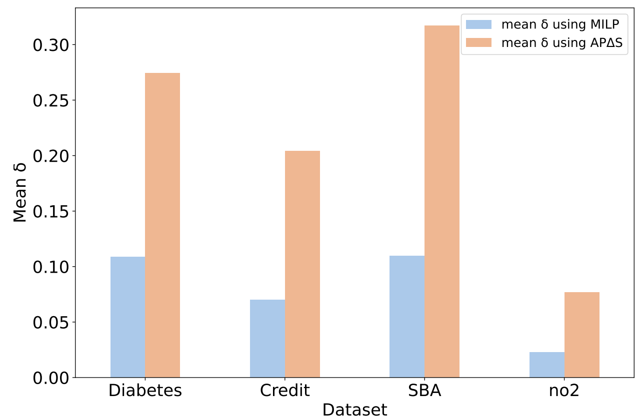


Figure 12: Average robust $\delta$ obtained using MILP-based certification and `APΔS` .

particular, we recorded $\delta_e = 0.27$ for *Diabetes*, $\delta_e = 0.07$ for *no2*, $\delta = 0.25$ for *SBA* and $\delta_e = 1.28$ for *Credit*. Given the magnitude of these changes, the robustness of the CFXs generated by MILP-R cannot be guaranteed in practice. However, the results show a rather intriguing picture: the **VM2** metric appears to be unaffected by retraining, and all CFXs remain valid on the respective final models.

These results suggest that certification approaches based on worst-case reasoning may be too strict in practical scenarios. To further understand the implications of worst-case vs average-case reasoning, we adapted Algorithm 1 to use the certification procedure of Jiang et al., i.e., a MILP solver instead of a sampled-based approach, and compute the maximum provable $\delta^*$ for which the previously generated CFXs are robust (Algorithm 3).

Fig. 12 shows a comparison between the average maximum provable $\delta$ obtained by this procedure and `APΔS` . As we can observe, our average-case guarantees allow to obtain $\delta$ values that are much higher, exceeding the MILP-certified in all instances. This is expected, given the results discussed in Proposition 17. However, what remains unclear is how these differences may affect the cost and plausibility of CFXs when certification procedures are embedded in procedures to generate CFXs.

### 6.3. Generating Robust CFXs using APΔS

The results discussed in the previous section have important implications on algorithms for the generation of robust CFXs. Recent works, e.g. [32, 12,

**Algorithm 4** Generation of Robust CFXs

1: **Input:** Model $\mathcal{M}$, input $x$ such that $\mathcal{M}(x) = c$, set of plausible model changes $\Delta$, maximum iteration number $\tau$
2: **Output:** $\Delta$-robust CFX $x'$
3: $t \leftarrow 0$ ▷ iteration number
4: **while** $t < \tau$ **do**
5:     $x' \leftarrow$ `ComputeCFX`$(x, \mathcal{M})$
6:     rate $\leftarrow$ `AP`$\Delta$`S` $(\mathcal{M}, x', \Delta)$
7:     **if** $rate = 1$ **then**
8:         **return** $x'$ ▷ $x'$ is approx. $\Delta$-robust
9:     **else**
10:         increase allowed distance of next CFX
11:         increase iteration number $t$
12:     **end if**
13: **end while**
14: **return** no robust CFX can be found

15], have proposed iterative procedures that generate provably robust CFXs by alternating two phases. First, a CFX is generated solving (variations of) Definition 1; then, a robustness certification procedure is invoked on the CFX. If the CFX is robust, then it is returned to the user; otherwise, the search continues, allowing for CFXs of increasing distance to be found. Clearly, the certification step has the potential to affect the CFXs computed in several ways. A robustness test that is too conservative may discard potentially good explanations and keep relaxing the distance constraint until the CFX is deemed robust. Ultimately, this may result in CFXs that exhibit poor proximity and plausibility.

To test this hypothesis, we adapt the CFX generation algorithm of [12] and replace their $\Delta$-robustness test with the one performed by `AP`$\Delta$`S` . The complete procedure is shown in Algorithm 4. In detail, after some initialization steps, we compute the first CFX using `ComputeCFX(`$x, \mathcal{M}$`)` (line 5), which employs the solution proposed in [12] and presented above. Given a CFX $x'$ and a plausible model shift $\Delta$, at line 6, we employ `AP`$\Delta$`S` setting $\alpha = 0.999$ and $R = 0.995$, thus obtaining 1378 realisations to perform in the robustness test. If the CFX $x'$ returned by our approximation results robust for all these realisations, then we return it to the user. Otherwise, we increased the allowed distance for the next CFX generation and the iteration number $t$ (lines 10-11).

We then compare the resulting procedure with the four generation algorithm studied in [12]: Wacht-R, Proto-R, MILP-R, and finally, ROAR [11]. Notably, ROAR is specifically designed to generate robust CFXs under plausible model changes using average-case certification. Using the same datasets and training procedures of § 6.2, we generate 50 CFXs for each dataset. We evaluate CFXs based on their proximity, measured by the $\ell_1$ distance, and plausibility, measured by the local outlier factor (**lof**) which determines if an instance is within the data manifold by quantifying the local data density [49] (+1 for inliers, −1 otherwise). We average $\ell_1$ and **lof** over the generated CFXs. We also report **VM1** and **VM2** for completeness. The results obtained, which we report in Table 2, confirm our hypothesis. Indeed, `AP`$\Delta$`S` produces the best results across all datasets, always generating CFXs with high plausibility and better proximity. Notably, `AP`$\Delta$`S` outperforms ROAR as well, producing CFXs that retain a higher degree of validity after retraining.

### *6.4. Impact of hyper-parameters on validity, plausibility and cost*

The previous set of experiments demonstrated that `AP`$\Delta$`S` is able to outperform existing approaches and generate CFXs that are robust, but also plausible and less expensive than other robust approaches. What remains unclear is the role that the main hyper-parameters of our algorithm, $\alpha$ and $R$, might play in obtaining these results. We therefore conducted additional experiments to evaluate the interplay between the tightness of the probabilistic guarantees offered by `AP`$\Delta$`S` and the quality of resulting explanations. In particular, focusing on the same datasets used in previous experiments, we started by checking the influence that $\alpha$ and $R$ have on the validity of CFXs after retraining. We generated 50 CFXs for each dataset using an instantiation of Algorithm 4 that uses MILP encodings to generate candidate CFXs as done in [12]. Figures 13-16 report the results obtained for the *Diabetes* and *SBA* datasets. Additional results for the two remaining datasets are reported in the appendix for this first set of experiments.

Our intuition is that lower values for $\alpha$ and $R$ should result in coarser robustness guarantees (i.e., larger $\delta$ values) and, thus, lower validity rates. As we can observe, our intuition is confirmed across all datasets, further clarifying the nature of the probabilistic guarantees that `AP`$\Delta$`S` can offer. Next, we investigate the impact that $\alpha$ and $R$ have on the plausibility and cost of CFXs generated by `AP`$\Delta$`S` . As per our previous experiments, we

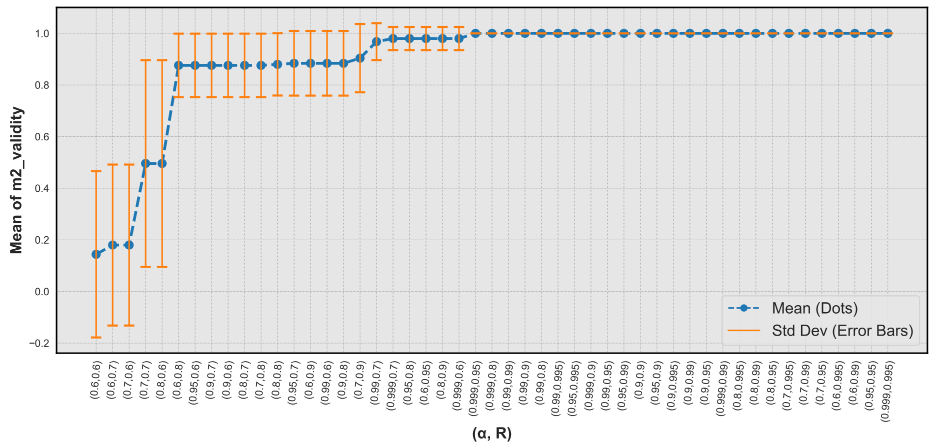


Figure 13: Mean validity after retraining visualised for increasing $\alpha$, $R$ values using the *Diabetes* dataset.

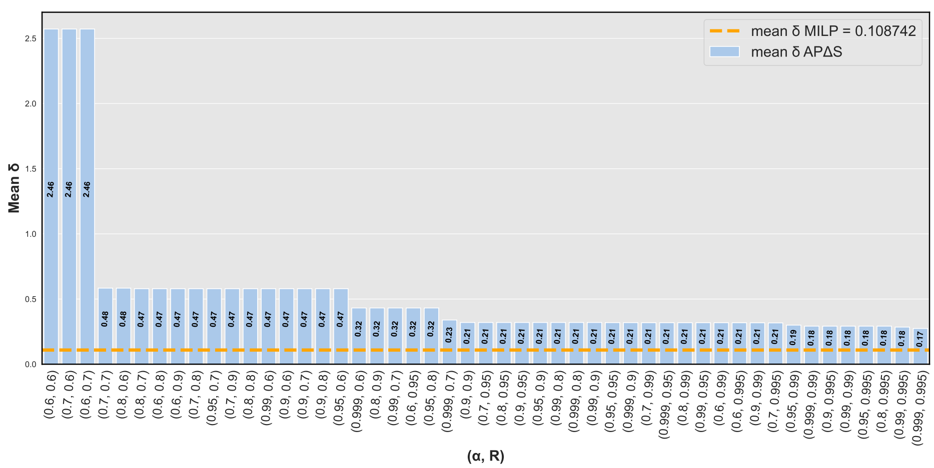


Figure 14: Mean certifiable $\delta$ obtained for increasing $\alpha$, $R$ values using the *Diabetes* dataset.

measure plausibility using the *LOF* score, and we use $\ell_0$, $\ell_1$ and $\ell_\infty$ norms to measure the proximity of CFXs. For conciseness, we only report results for the *Diabetes* dataset in Figure 17 below and delegate additional results to the appendix. To improve the readability of our results, we decided to separate CFXs that achieved 100% validity after retraining for the rest. Overall we can observe a clear trend, whereby increasing $\alpha$, $R$ results in CFXs that are further away from the decision boundary and thus more plausible. These

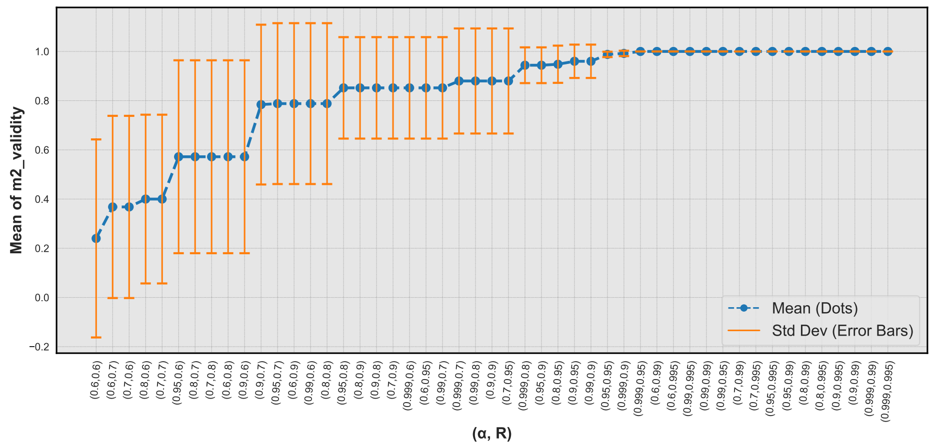


Figure 15: Mean validity after retraining visualised for increasing $\alpha$, $R$ values using the *SBA* dataset.

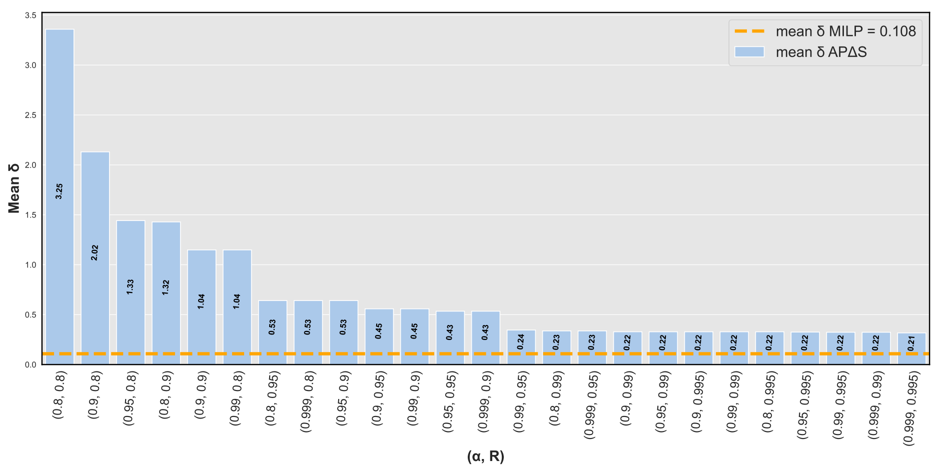


Figure 16: Mean certifiable $\delta$ obtained for increasing $\alpha$, $R$ values using the *SBA* dataset.

results are in line with observations made in other works on robustness to model changes, where it has been suggested that increasing cost improves the robustness and plausibility of CFXs [11, 29, 10].

### 6.5. Scalability analysis of APΔS

In this section we demonstrate that APΔS is able to scale to state-of-the-art architectures used for tabular data, thus providing further empirical evidence of the practical viability of our approach. More specifically, we


Figure 17: Mean *LOF*, $\ell_0, \ell_1, \ell_\infty$ metrics for increasing $\alpha$, $R$ values using the *Diabetes* dataset. CFXs with 100% validity after retraining are shown on the right, while the remaining CFXs are shown on the left.

focus on *TabNet* [17], a tabular transformer recently introduced that leverages sparse attention and sequential feature selection to learn interpretable feature representations. At its core, *TabNet* processes data in a series of decision steps, with each step using a learned attention mask to select a subset of features, which allows the model to focus on the most relevant attributes at each stage. This sparse attention mechanism makes *TabNet* computationally efficient and helps to improve interpretability in tabular datasets. From our perspective, this architecture is interesting as it comprises an attention mechanism, with encoder-decoder components typical of other recent transformer architectures and a consequent significant number of parameters to test the scalability of `APΔS` . To the best of our knowledge, this is the first time that CFXs with robustness guarantees are generated for such a complex architecture with tens of thousands of parameters.

Before considering the robustness property, we analyse the accuracy in the training and testing phases of *TabNet*. To this end, we employ a supervised training approach, splitting the datasets employed in the previous evaluation, namely *Diabetes*, *No2*, *SBA* and *Credit*, into training and testing datasets, and we first compare the accuracy obtained using this architecture with standard MLPs employed in § 6.2. To ensure statistical significance of our results, we consider, for each dataset tested, the mean of the accuracies obtained using ten random initializations of the transformer architecture.As highlighted in the first two columns of Tab. 3, with *TabNet*, we have an increased number of parameters in the model but similar or even higher accuracy with respect to the classical MLP, confirming the potential of this novel architecture in selecting important features to get more precise final accuracy in the prediction.

As our results show a similar level of accuracy between MLP and *TabNet*, we move on to how to generate robust CFXs for this transformer architecture. Given the significantly higher number of parameters in *TabNet*, we replace the MILP-based procedure used in Section 6.3, Algorithm 4 with a Nearest Neighbors Counterfactual Explainer (NNCFX) [40] to ensure scalability of our generation procedure. More specifically, line 5 in Algorithm 4 (reported in the appendix) now implements the following strategy. Given an input $x$ for which a robust CFX is sought, we identify the nearest data point belonging to the dataset for which *TabNet* produces a different classification outcome. Our implementation uses k-d trees to improve the efficiency of this nearest-

Table 3: Scalability experiments of `APΔS` .

**Diabetes**

| | **# Parameters** | **Mean Accuracy** | **Mean $\delta_{max}$** | **Mean Comp. Time** |
|---|---|---|---|---|
| *MLP* | 81 | 79% | 0.32 | 0.01s |
| *TabNet* | 30992 | 82% | 0.48 | 5.21s |

**no2**

| | **# Parameters** | **Mean Accuracy** | **Mean $\delta_{max}$** | **Mean Comp. Time** |
|---|---|---|---|---|
| *MLP* | 145 | 64% | 0.11 | 0.008s |
| *TabNet* | 30676 | 68% | 0.35 | 9.2s |

**SBA**

| | **# Parameters** | **Mean Accuracy** | **Mean $\delta_{max}$** | **Mean Comp. Time** |
|---|---|---|---|---|
| *MLP* | 199 | 99% | 0.53 | 0.02s |
| *TabNet* | 30992 | 100% | 0.16 | 10.3s |

**Credit**

| | **# Parameters** | **Mean Accuracy** | **Mean $\delta_{max}$** | **Mean Comp. Time** |
|---|---|---|---|---|
| *MLP* | 371 | 74% | 0.34 | 0.01s |
| *TabNet* | 40946 | 74% | 1.8 | 11.3s |

neighbor search.[5]

For the following experiments, we consider the same datasets used in Section 6.3 and the same 50 original inputs employed in those experiments. The last two columns of Table 3 report the results we obtained when generating CFX with robustness guarantees for *TabNet*. As we can observe, `APΔS` is still able to compute robust CFXs within tens of seconds, even when employed in transformer-based architecture with $\sim 400x$ times parameters, showing a linear growth time computation. Similar runtimes are observed across all four datasets, thus confirming the high scalability of our approach. To further confirm this aspect, we ran an in-depth scalability study by training a

[5] We perform a further experiment reported in the Appendix B to understand the difference between the two CFX-generation approaches, namely MILP and NNCFX.

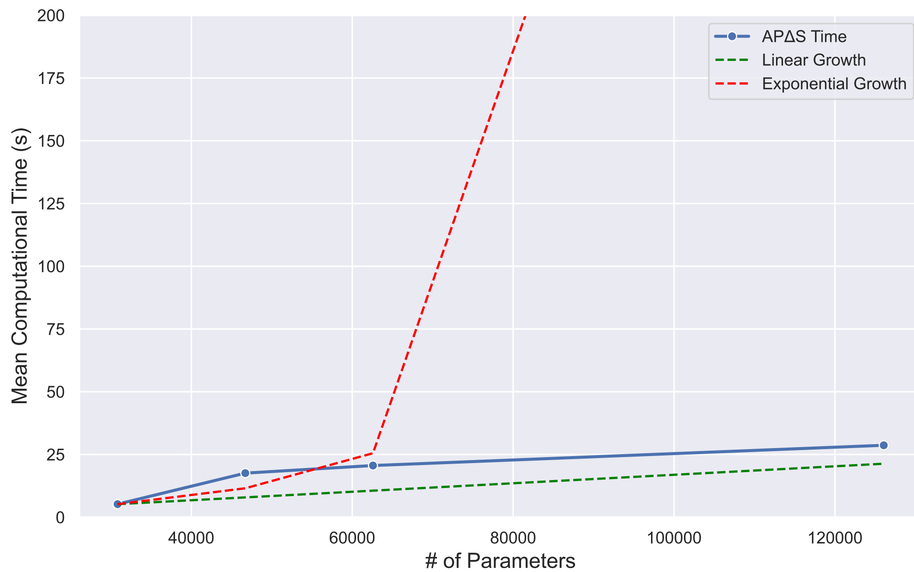


Figure 18: Mean computation time of `APΔS` applied to *TabNet* architectures with increasing number of parameters.

set of 4 *TabNet* models with an increasing number of parameters. Using the diabetes dataset, we trained models containing $[30834, 40946, 62578, 126066]$ respectively and generated 50 robust CFXs for each. We stored the runtimes for each robust CFX and report the mean computation time for each model in Figure 18. As we can observe, the runtime increase follows a linear trend, thus highlighting the effectiveness and applicability of our proposed solution even when targetting complex architectures.

## 7. Conclusions

We studied the problem of generating robust CFXs with respect to plausible model changes. We proved for the first time that certifying the robustness of CFX with respect to this notion of robustness is an NP-hard problem, and also extended this result to show that the same complexity results apply to naturally-occurring model changes. These results motivate the quest for new scalable algorithms to certify robustness under plausible model changes. To this end, we investigated existing methods to generate robust CFXs with probabilistic guarantees and showed that these approaches may not be directly applicable to our setting. We then introduced `APΔS` , a novel scalable approach for probabilistic robustness certification, and used it to generate robust CFXs under plausible model changes. We carried out an extensive experimental analysis, demonstrating the advantages of `AP`$\Delta$`S` and outperforming SOTA methods on a range of metrics, including validity, plausibility, and cost. Crucially, we also applied our method to certify CFXs' robustness for tabular transformers containing thousands of parameters. To the best of our knowledge, we are the first to consider models of this size within the robust CFX literature [10], further demonstrating the scalability and wide applicability of our approach. We see these outcomes as important contributions towards complementing existing formal approaches for Explainable AI and making them applicable in practice.

## Acknowledgements

Leofante was supported by an Imperial College Research Fellowship. Any views or opinions expressed herein are solely those of the authors listed.

## References

[1] L. Tai, G. Paolo, M. Liu, Virtual-to-real drl: Continuous control of mobile robots for mapless navigation, in: IROS, 2017.

[2] L. Marzari, D. Corsi, E. Marchesini, A. Farinelli, Curriculum learning for safe mapless navigation, in: Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, 2022, pp. 766–769.

[3] K. O'Shea, R. Nash, An introduction to convolutional neural networks, arXiv preprint arXiv:1511.08458 (2015).

[4] D. Corsi, L. Marzari, A. Pore, A. Farinelli, A. Casals, P. Fiorini, D. Dall'Alba, Constrained reinforcement learning and formal verification for safe colonoscopy navigation, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2023.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).

[6] G. Amir, D. Corsi, R. Yerushalmi, L. Marzari, D. Harel, A. Farinelli, G. Katz, Verifying learning-based robotic navigation systems, in: 29th International Conference, TACAS 2023, Springer, 2023, pp. 607–627.

[7] I. Stepin, J. M. Alonso, A. Catalá, M. Pereira-Fariña, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, IEEE Access 9 (2021) 11974–12001.

[8] A. Karimi, G. Barthe, B. Schölkopf, I. Valera, A survey of algorithmic recourse: Contrastive explanations and consequential recommendations, ACM Comput. Surv. 55 (2023) 95:1–95:29.

[9] S. Wachter, B. D. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, CoRR abs/1711.00399 (2017). URL: http://arxiv.org/abs/1711.00399. arXiv:1711.00399.

[10] J. Jiang, F. Leofante, A. Rago, F. Toni, Robust counterfactual explanations in machine learning: A survey, in: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024, ijcai.org, 2024, pp. 8086–8094. URL: https://www.ijcai.org/proceedings/2024/894.

[11] S. Upadhyay, S. Joshi, H. Lakkaraju, Towards robust and reliable algorithmic recourse, in: Advances in Neural Information Processing Systems 34 (NeurIPS21), 2021, pp. 16926–16937.

[12] J. Jiang, F. Leofante, A. Rago, F. Toni, Formalising the robustness of counterfactual explanations for neural networks, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI23), 2023, pp. 14901–14909.

[13] E. Black, Z. Wang, M. Fredrikson, Consistent counterfactuals for deep models, in: Proceedings of the 10th International Conference on Learning Representations (ICLR22), OpenReview.net, 2022.

[14] T.-D. H. Nguyen, N. Bui, D. Nguyen, M.-C. Yue, V. A. Nguyen, Robust Bayesian recourse, in: Proceedings of the 38hth Conference on Uncertainty in AI (UAI22), 2022, pp. 1498–1508.

[15] F. Hamman, E. Noorani, S. Mishra, D. Magazzeni, S. Dutta, Robust counterfactual explanations for neural networks with probabilistic guarantees, in: Proceedings of the International Conference on Machine Learning (ICML23), 2023, pp. 12351–12367.

[16] L. Marzari, F. Leofante, F. Cicalese, A. Farinelli, Rigorous probabilistic guarantees for robust counterfactual explanations, in: ECAI 2024, IOS Press, 2024, pp. 1059–1066.

[17] S. Ö. Arik, T. Pfister, Tabnet: Attentive interpretable tabular learning, in: Proceedings of the AAAI conference on artificial intelligence, volume 35, 2021, pp. 6679–6687.

[18] R. Poyiadzi, K. Sokol, R. Santos-Rodríguez, T. D. Bie, P. A. Flach, FACE: feasible and actionable counterfactual explanations, in: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES20), 2020, pp. 344–350.

[19] M. Pawelczyk, K. Broelemann, G. Kasneci, Learning model-agnostic counterfactual explanations for tabular data, in: Proceedings of the Web Conference (WWW20), ACM / IW3C2, 2020, pp. 3126–3132.

[20] B. Ustun, A. Spangher, Y. Liu, Actionable recourse in linear classification, in: Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*19), 2019, pp. 10–19. URL: `https://doi.org/10.1145/3287560.3287566`. doi:`10.1145/3287560.3287566`.

[21] A. Artelt, V. Vaquet, R. Velioglu, F. Hinder, J. Brinkrolf, M. Schilling, B. Hammer, Evaluating robustness of counterfactual explanations, in: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2021, pp. 01–09.

[22] D. Slack, A. Hilgard, H. Lakkaraju, S. Singh, Counterfactual explanations can be manipulated, in: Advances in Neural Information Processing Systems 34 (NeurIPS21), 2021, pp. 62–75. URL: `https://proceedings.neurips.cc/paper/2021/hash/009c434cab57de48a31f6b669e7ba266-Abstract.html`.

[23] R. Dominguez-Olmedo, A. Karimi, B. Schölkopf, On the adversarial robustness of causal algorithmic recourse, in: Proceedings of the International Conference on Machine Learning (ICML22), 2022, pp. 5324–5342.

[24] S. Zhang, X. Chen, S. Wen, Z. Li, Density-based reliable and robust explainer for counterfactual explanation, Expert Syst. Appl. 226 (2023) 120214.

[25] F. Leofante, N. Potyka, Promoting counterfactual robustness through diversity, in: Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, February 20-27, 2024, Vancouver, Canada, AAAI Press, 2024, pp. 21322–21330. URL: https://doi.org/10.1609/aaai.v38i19.30127. doi:10.1609/AAAI.V38I19.30127.

[26] V. Guyomard, F. Fessant, T. Guyet, T. Bouadi, A. Termier, Generating robust counterfactual explanations, in: Machine Learning and Knowledge Discovery in Databases: Research Track - European Conference, ECML PKDD 2023, volume 14171 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 394–409.

[27] M. Virgolin, S. Fracaros, On the robustness of sparse counterfactual expl. to adverse perturbations, Artif. Intell. 316 (2023). URL: https://doi.org/10.1016/j.artint.2022.103840. doi:10.1016/j.artint.2022.103840.

[28] F. Leofante, A. Lomuscio, Towards robust contrastive explanations for human-neural multi-agent systems, in: N. Agmon, B. An, A. Ricci, W. Yeoh (Eds.), Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS23), ACM, 2023, pp. 2343–2345.

[29] M. Pawelczyk, T. Datta, J. van den Heuvel, G. Kasneci, H. Lakkaraju, Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse, in: Proceedings of the 11th International Conference on Learning Representations (ICLR23), OpenReview.net, 2023.

[30] F. Leofante, E. Botoeva, V. Rajani, Counterfactual explanations and model multiplicity: a relational verification view, in: Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning (KR23), 2023, pp. 763–768.

[31] J. Jiang, A. Rago, F. Leofante, F. Toni, Recourse under model multiplicity via argumentative ensembling, in: Proceedings of the 2024 International Conference on Autonomous Agents and Multiagent Systems (AAMAS24), 2024.

[32] S. Dutta, J. Long, S. Mishra, C. Tilli, D. Magazzeni, Robust counterfactual explanations for tree-based ensembles, in: Proceedings of the International Conference on Machine Learning (ICML22), 2022, pp. 5742–5756.

[33] J. Jiang, F. Leofante, A. Rago, F. Toni, Interval abstractions for robust counterfactual explanations, Artif. Intell. 336 (2024) 104218. URL: https://doi.org/10.1016/j.artint.2024.104218. doi:10.1016/J.ARTINT.2024.104218.

[34] I. J. Goodfellow, Y. Bengio, A. C. Courville, Deep Learning, Adaptive computation and machine learning, MIT Press, 2016. URL: http://www.deeplearningbook.org/.

[35] P. Prabhakar, Z. R. Afzal, Abstraction based output range analysis for neural networks, in: Advances in Neural Information Processing Systems 32 (NeurIPS19), 2019, pp. 15762–15772.

[36] W. Rudin, et al., Principles of mathematical analysis, volume 3, McGraw-hill New York, 1964.

[37] D. Dua, C. Graff, UCI machine learning repository, http://archive.ics.uci.edu/ml, 2017. Accessed: 2022-08-30.

[38] M. Hopkins, E. Reeber, G. Forman, J. Suermondt, Spambase, UCI Machine Learning Repository, 1999. DOI: https://doi.org/10.24432/C53G6X.

[39] K. Fernandes, P. Vinagre, P. Cortez, P. Sernadela, Online News Popularity, UCI Machine Learning Repository, 2015. DOI: https://doi.org/10.24432/C5NS3V.

[40] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Mining and Knowledge Discovery (2022) 1–55.

[41] V. Tjeng, K. Xiao, R. Tedrake, Evaluating robustness of neural networks with mixed integer programming, arXiv preprint arXiv:1711.07356 (2017).

[42] S. S. Wilks, Statistical prediction with special reference to the problem of tolerance limits, The annals of mathematical statistics 13 (1942) 400–409.

[43] J. Jiang, L. Marzari, A. Purohit, F. Leofante, Robustx: Robust counterfactual explanations made easy, arXiv preprint arXiv:2502.13751 (2025).

[44] L. Marzari, D. Corsi, F. Cicalese, A. Farinelli, The #DNN-Verification Problem: Counting Unsafe Inputs for Deep Neural Networks, in: International Joint Conference on Artificial Intelligence (IJCAI), 2023, pp. 217–224.

[45] L. Marzari, D. Corsi, E. Marchesini, A. Farinelli, F. Cicalese, Enumerating safe regions in deep neural networks with provable probabilistic guarantees, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 21387–21394.

[46] J. W. Smith, J. E. Everhart, W. Dickson, W. C. Knowler, R. S. Johannes, Using the adap learning algorithm to forecast the onset of diabetes mellitus, in: Proceedings of the annual symposium on computer application in medical care, American Medical Informatics Association, 1988, p. 261.

[47] J. Vanschoren, J. N. van Rijn, B. Bischl, L. Torgo, Openml: networked science in machine learning, SIGKDD Explor. 15 (2013) 49–60. URL: https://doi.org/10.1145/2641190.2641198. doi:10.1145/2641190.2641198.

[48] M. Li, A. Mickel, S. Taylor, "should this loan be approved or denied?": A large dataset with class assignment guidelines, Journal of Statistics Education 26 (2018) 55–66.

[49] M. M. Breunig, H. Kriegel, R. T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, ACM, 2000, pp. 93–104.

# Appendix A. Additional results on *Credit* and *No2* datasets for § 6.4 experiments.

*Credit*

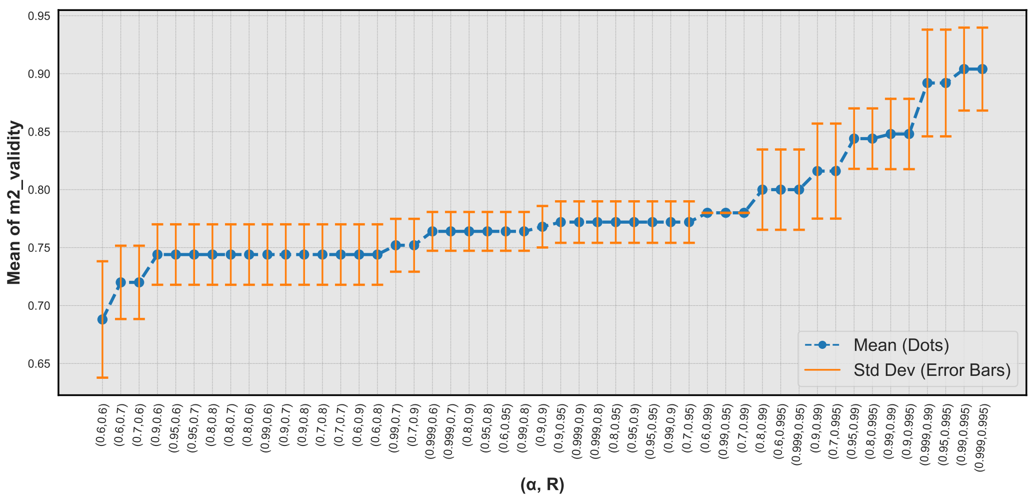


Figure A.19: Mean validity after retraining visualised for increasing $\alpha$, $R$ values using the *Credit* dataset.

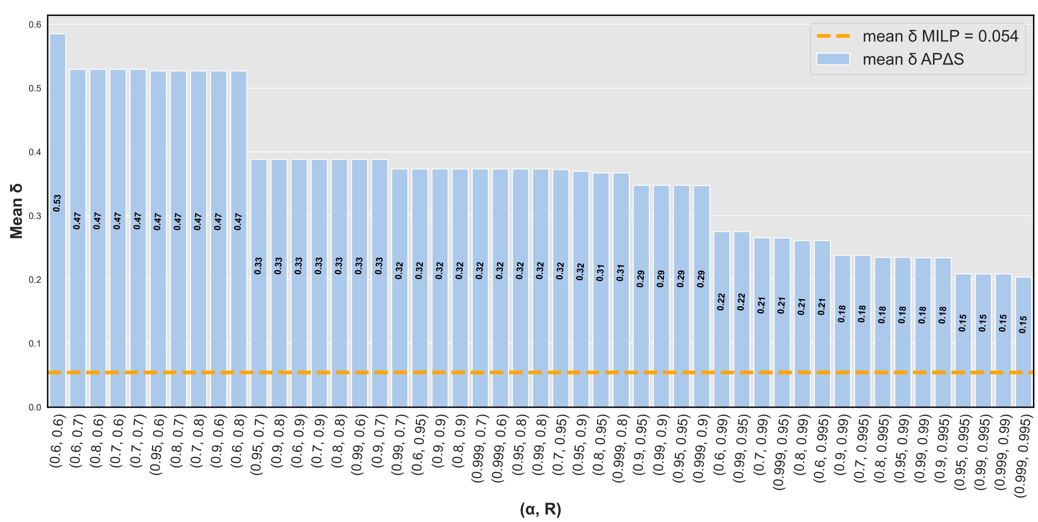


Figure A.20: Mean certifiable $\delta$ obtained for increasing $\alpha$, $R$ values using the *Credit* dataset.

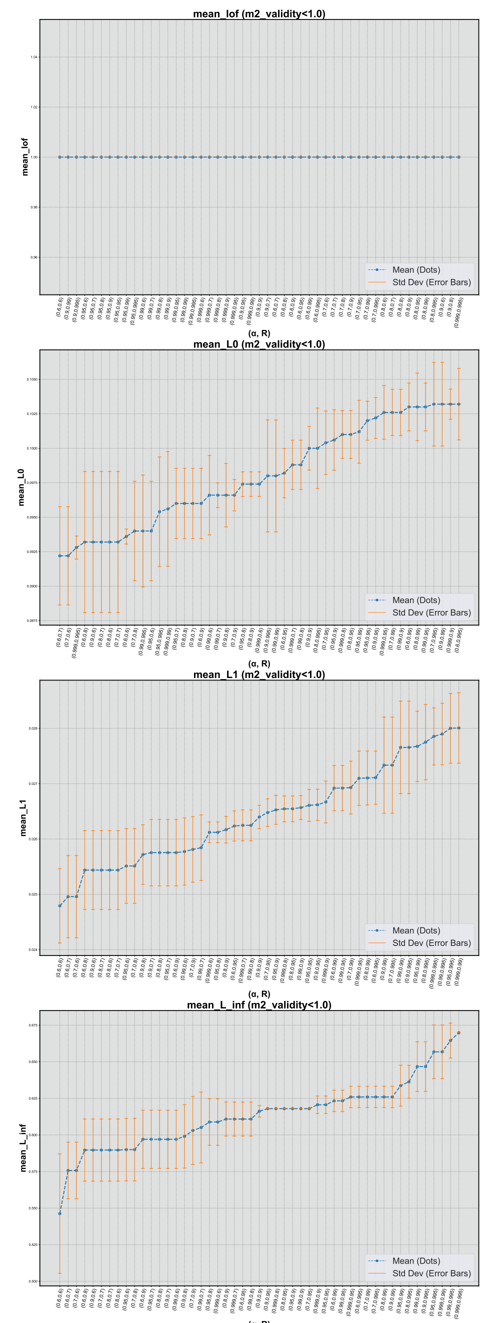


Figure A.21: Mean *LOF*, $\ell_0, \ell_1, \ell_\infty$ metrics for increasing $\alpha$, $R$ values using the *Credit* dataset. CFXs never reach 100% validity after retraining on this dataset.

1233 *No2*

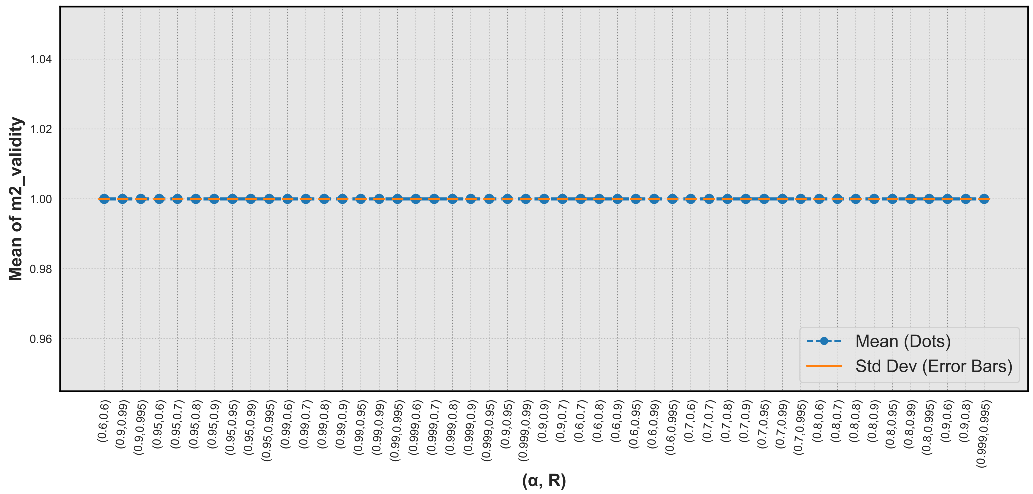


Figure A.22: Mean validity after retraining visualised for increasing $\alpha$, $R$ values using the *No2* dataset.

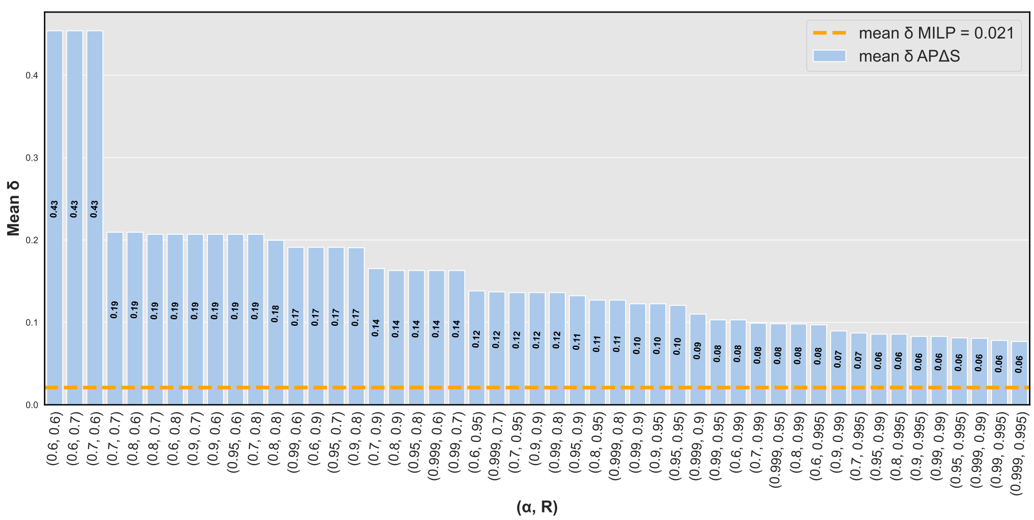


Figure A.23: Mean certifiable $\delta$ obtained for increasing $\alpha$, $R$ values using the *No2* dataset.

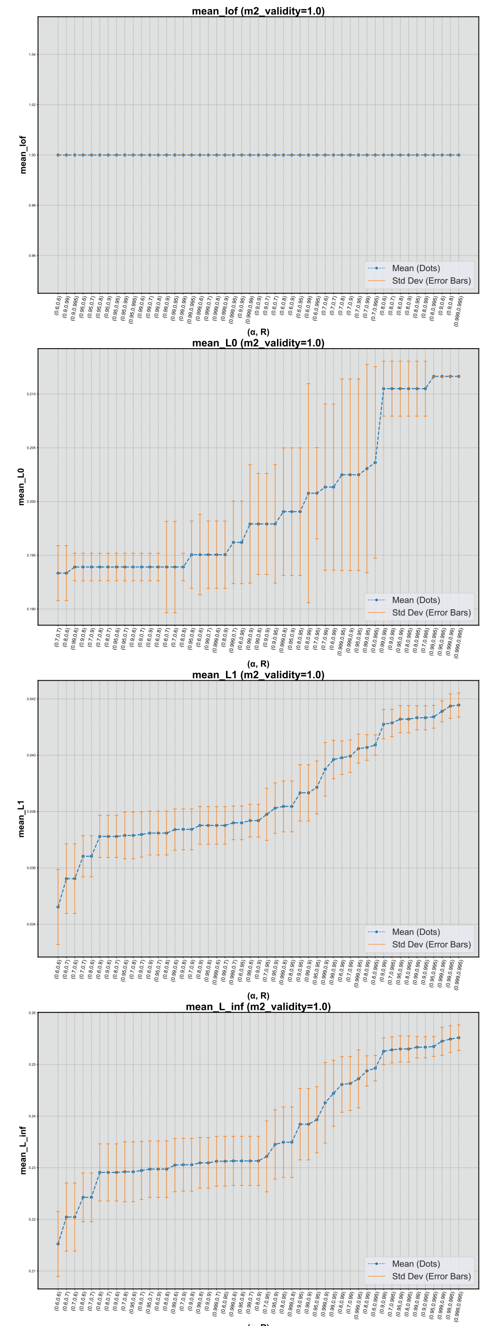


Figure A.24: Mean $LOF$, $\ell_0, \ell_1, \ell_\infty$ metrics for increasing $\alpha$, $R$ values using the *Credit* dataset. CFXs always obtain 100% validity after retraining on this dataset.

## Appendix B. Comparison of two CFXs generation approaches of § 6.5 experiments.

As stated in § 6.5 in the main paper, given the significantly higher number of parameters in *TabNet*, we replace the MILP-based procedure used in Section 6.3, Algorithm 4 with a Nearest Neighbors Counterfactual Explainer (NNCFX) [40] to ensure scalability of our generation procedure. More specifically, line 5 in Algorithm 4 now implements the following strategy.

**Algorithm 5** *Nearest Neighbors Counterfactual Explanation*

**Input:** Dataset $\mathbf{d}$, a k-d tree built from dataset features, $\mathbf{x}$ set of original inputs, $\mathbf{y}$ set of original outcomes<br>
**Output:** **x'** set of nearest counterfactual explanation.<br>
**x'** $\leftarrow \emptyset$<br>
**for** $i$ in $len(\mathbf{x})$ **do**<br>
    $x \leftarrow \mathbf{x}[i]$ ▷ original input<br>
    $y \leftarrow \mathbf{y}[i]$ ▷ original output<br>
    $y' \leftarrow 1 - y$ ▷ desired outcome<br>
    $idx, distance \leftarrow$ `k-d tree.query`$(x, len(features))$ ▷ already sorted per distance<br>
    **if** $\mathbf{d}[idx]['outcome'] == y'$ **then**<br>
        **x'** $\leftarrow \mathbf{d}[idx]$<br>
    **else**<br>
        **x'** $\leftarrow None$<br>
    **end if**<br>
**end for**<br>
**return x'**

This function iteratively searches for a neighboring data point with the opposite outcome by evaluating distances between features and selecting the nearest as possible. Clearly, this approach and the one of [12] can produce different explanations. We perform a further experiment to understand the difference between the two CFX-generation approaches. Hence, we consider the same datasets used in Section 6.3 and the same 50 original inputs employed in those experiments. In the NNCFX approach, once a valid counterfactual is found, the mean feature-wise distance between the identified counterfactual and the MILP-generated counterfactual is calculated. This distance serves as a measure of similarity between the counterfactuals identified by the TabNet-based approach and those obtained via MILP in an MLP setting. Our results are reported in Fig. B.25. On the x-axis, we report the index of CFX, while on the y-axis, the mean and standard deviation distance between our CFX and the one generated with MILP in each dataset.

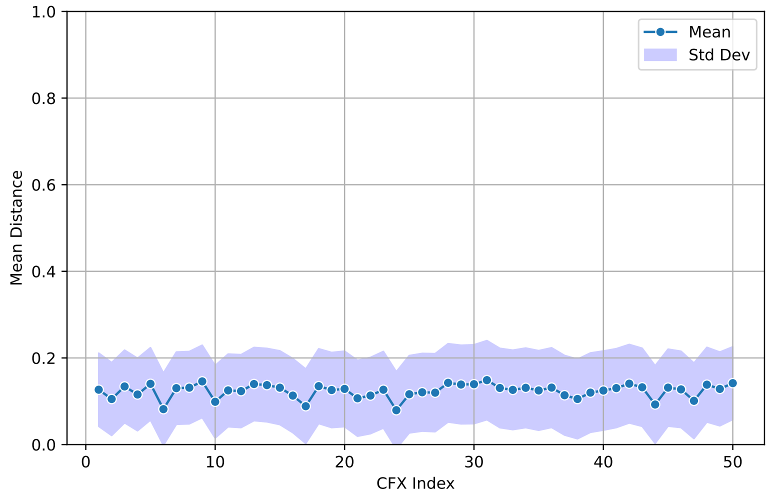


Figure B.25: Mean and standard deviation distance between CFXs generated with NNCFX and MILP in *Diabetes, Credit, SBA, no2* datasets.

As we can notice, since the values in the datasets are typically normalized in a range $[0, 1]$, the CFXs generated with the two approaches are consistently close. In fact, there is a mean feature distance between the CFX generated with NNCFX and MILP of $\sim 0.12$ for the 50 inputs selected. This result shows the correctness and efficiency of the NNCFX generation approach in the transformers-based setting.