# Counterfactual Explanations and Model Multiplicity: a Relational Verification View

**Francesco Leofante**[1] , **Elena Botoeva**[2] , **Vineet Rajani**[2]

[1]Department of Computing, Imperial College London, United Kingdom
[2]School of Computing, University of Kent, United Kingdom
f.leofante@imperial.ac.uk,{e.botoeva, v.rajani}@kent.ac.uk

**Abstract**

We study the interplay between counterfactual explanations and model multiplicity in the context of neural network classifiers. We show that current explanation methods often produce counterfactuals whose validity is not preserved under model multiplicity. We then study the problem of generating counterfactuals that are guaranteed to be robust to model multiplicity, characterise its complexity and propose an approach to solve this problem using ideas from relational verification.

## 1 Introduction

*Counterfactual explanations* (CFXs) are often used to provide recourse recommendations to individuals that have been affected by the predictions of Machine Learning (ML) models (Karimi et al. 2023). Algorithms to generate CFXs for a given input $x$ to a model $\mathcal{M}$ typically try to find a minimally altered input $x'$ for which $\mathcal{M}$ gives a different output to that of $x$ (Wachter, Mittelstadt, and Russell 2018). These algorithms often operate under the assumption that there exist *only one model with the best accuracy* and that this is the model that will eventually be deployed and need explanations. While using accuracy as a criterion for model selection is common, recent studies have drawn attention to the fact that there often exist multiple models that achieve the same accuracy on a prediction task, but greatly differ in their internals (Marx, Calmon, and Ustun 2020; Black, Raghavan, and Barocas 2022). This phenomenon, often referred to as *model multiplicity*, has deep implications for CFXs. Consider the classic scenario of a loan application, where a bank employs an ML model to predict whether a loan should be granted or not. Assume an applicant with features *unemployed* status, *30* years of age and *low* credit rating is rejected by the bank's model. A CFX for this prediction might suggest to increase the credit rating to *medium* for the loan to be granted. However, under model multiplicity, there may exist another equally accurate model, which also rejects the original application, but for which increasing the credit rating to *medium* would still result in the loan being rejected. This problem is far from being mere theoretical speculation. In Table 1, we present a scenario where $n$ neural networks with same architecture are trained to the same accuracy on the German Credit dataset (Dua and Graff 2017), using different seeds. We then use two state-of-the-art algorithms to generate 50 CFXs for each model, and test the validity of each explanation on the remaining $n-1$ models. As we can observe, large fractions of the explanations cease to be valid under model multiplicity, ultimately raising concerns about the justifiability of these CFXs.

| | $n=2$ | $n=3$ | $n=4$ | $n=5$ |
|---|---|---|---|---|
| (Wachter, Mittelstadt, and Russell 2018) | 98% | 66.7% | 74.5% | 79.2% |
| (Looveren and Klaise 2021) | 50% | 34% | 25.5% | 19.6% |

Table 1: Amount of CFXs that are valid on $n$ different networks.

**Our contribution.** Though previous work has considered the interplay between CFXs and model multiplicity (Pawelczyk, Broelemann, and Kasneci 2020), formal methods for generating explanations satisfying this property are currently lacking. In line with recent calls for formal explanations of ML models (Darwiche 2020; Marques-Silva and Ignatiev 2022; Jiang et al. 2023), we here seek to derive new techniques to generate CFXs whose validity is guaranteed across an ensemble of neural network classifiers. We propose to tackle this problem using product constructions from *relational program verification* (Barthe, Crespo, and Kunz 2011). Relational verfication has mainly been studied in the context of verifying relations between programs/executions in the classical program verification setting, but recent work have extended its scope to the analysis of neural networks (Paulsen, Wang, and Wang 2020; Paulsen et al. 2020; Khedr and Shoukry 2023; Christakis et al. 2022). As an example, (Paulsen, Wang, and Wang 2020) consider two neural networks $\mathcal{M}$ and $\mathcal{M}'$ trained for the same task and assert relational properties such as "$\mathcal{M}$ and $\mathcal{M}'$ are expected to produce the same output when receiving the same input".

Inspired by these ideas, in this paper we propose the first study of robustness to model multiplicity as a relational property. In particular, we propose a novel product construction and use it to show that generating a CFX satisfying this property is NP-complete. Building on this result, we propose an approach based on Mixed-Integer Linear Programming (MILP) to compute CFXs and demonstrate its applicability on neural network classifiers trained on tabular data.

**Related work.** Several techniques have been proposed to compute CFXs for ML models – see, e.g. (Karimi et al. 2023) for a recent survey. Due to space constraints, we will focus only on approaches that generate CFXs with improved

robustness. Previous work has considered robustness against perturbations in the input to be explained (Sharma, Henderson, and Ghosh 2020; Dominguez-Olmedo, Karimi, and Schölkopf 2022), to adversarial perturbations applied to the CFX itself (Slack et al. 2021; Pawelczyk et al. 2022; Leofante and Lomuscio 2023) or to bounded changes in the ML model parameters (Upadhyay, Joshi, and Lakkaraju 2021; Dutta et al. 2022; Jiang et al. 2023). Our work shares some similarities with the latter class of approaches. However, we generalise the notion of robustness by relaxing the requirement that changes must be bounded as typically assumed in the literature. This allows us to capture a broader class of problems, including scenarios where models with different architectures are deployed.

Model multiplicity has been the subject of previous studies within ML (Marx, Calmon, and Ustun 2020; Black, Raghavan, and Barocas 2022). Most recently, (Pawelczyk, Broelemann, and Kasneci 2020) proved that CFXs that are on-manifold also exhibit increased levels of robustness under model multiplicity. In this paper we take a step further and devise procedures to generate CFXs that are formally guaranteed to be robust under model multiplicity. Our motivation for doing so is in line with recent works arguing that formal approaches are needed to develop trustworthy explainable AI (Ignatiev, Narodytska, and Marques-Silva 2019; Darwiche 2020; Ignatiev and Silva 2021; Marques-Silva and Ignatiev 2022; Leofante and Lomuscio 2023).

## 2 Background

**Feed-forward neural networks.** A feed-forward neural network (FFNN) is a directed acyclic graph whose nodes are arranged in subsequent layers $L_0, \ldots, L_k$ (Goodfellow, Bengio, and Courville 2016). $L_0$ is the *input* layer, $L_k$ is the *output* layer and each non-input layer $L_i$, $i \in \{1, \ldots, k\}$, is parametrised by a weight matrix $W_i$ and a bias vector $B_i$. An FFNN computes an output by propagating a given input through its layers as defined below.

**Definition 1.** *Given an input $x \in \mathbb{R}^m$ and an FFNN $\mathcal{M}$, let:*

- $L_0 = x$;
- $L_i = \sigma_i(W_i \cdot L_{i-1} + B_i)$ *for* $i \in \{1, \ldots, k-1\}$, *where $\sigma_i$ is an activation function applied element-wise.*

*The* output of $\mathcal{M}$ (on $x$) *is defined as* $L_k = W_k \cdot L_{k-1} + B_k$.

Here, w.l.o.g. we focus on FFNNs using ReLU activations trained to solve binary classification tasks, i.e., we assume that $L_k$ is the vector $(o_0, o_1)^\intercal$. In this setting, we characterise the classification outcome of an FFNN as follows.

**Definition 2.** *Given an input $x \in \mathbb{R}^m$, we say that an FFNN $\mathcal{M}$* classifies *$x$ as 0 if $o_0 > o_1$, 1 if $o_0 < o_1$ and* undefined *otherwise. We write $\mathcal{M}(x) = 0$ and $\mathcal{M}(x) = 1$ to denote the binary classification outcomes.*

Definition 2 distinguishes between three cases to remove the ambiguity in determining the classification outcome as implemented in modern deep learning libraries, which typically return the lowest index of the maximal value in $L_k$ (see, e.g. (Paszke et al. 2019)). This is crucial to ensure that CFXs achieve their goal reliably.

**Counterfactual explanations.** CFXs attempt to explain the outcome of an ML model by showing how an input could be changed to produce a different decision. CFXs are commonly computed as solutions to the constraint problem defined as follows (Mohammadi et al. 2021).

**Definition 3.** *Let $\mathcal{M}$ be a binary classifier, $x \in \mathbb{R}^m$ a factual input s.t. $\mathcal{M}(x) = c$, $d\colon \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$ a distance metric and $\delta \in \mathbb{R}^+$ a distance threshold. A* counterfactual explanation (CFX) for *$x$, $\delta$, $\mathcal{M}$ is any $x' \in \mathbb{R}^m$ such that*

$$\mathcal{M}(x') = 1 - c \quad \textit{and} \quad d(x, x') \leq \delta.$$

*Moreover, a CFX $x'$ is* optimal *if $d(x, x')$ is minimal.*

The formulation above ensures that the input $x'$ is close enough to $x$ under metric $d$ and makes the classification flip. In many cases, one is actually interested in computing an optimal CFX that represents a *minimal change* to the factual input, which in practice is used to provide a *recourse recommendation* to the user for the desired outcome to be achieved. The $\ell_1$ norm, also known as Manhattan distance, (Wachter, Mittelstadt, and Russell 2018; Mohammadi et al. 2021) is a common choice for $d$ and the one we assume in this paper. This metric enforces sparsity of changes, which is often used as a proxy for the effort a user will have to make to implement the recommended recourse (Wachter, Mittelstadt, and Russell 2018).

## 3 Robust CFXs under Model Multiplicity

In this section we formalise the problem of finding a CFX that is robust under model multiplicity. We begin by defining a property of a set of models called *consistency*, an invariance criterion under model multiplicity.

**Definition 4.** *A set $\boldsymbol{\mathcal{M}}$ of models is* consistent for an input *$x$ if all models in $\boldsymbol{\mathcal{M}}$ produce the same classification for $x$.*

We are interested in studying the decision problem concerned with the existence of a CFX that is robust across a set of models, the latter being formally defined as follows.

**Definition 5.** *Consider a factual input $x \in \mathbb{R}^m$, a distance threshold $\delta \in \mathbb{R}^+$ and a set $\boldsymbol{\mathcal{M}}$ of models consistent for $x$. An input $x' \in \mathbb{R}^m$ is a* robust counterfactual across *$\boldsymbol{\mathcal{M}}$ if:*

**(C1)** *$x'$ is a CFX for $x$, $\delta$ and for each $\mathcal{M} \in \boldsymbol{\mathcal{M}}$, and*

**(C2)** *$\boldsymbol{\mathcal{M}}$ is consistent for $x'$.*

Note that condition (C1) subsumes condition (C2) for binary classifiers that are the focus of this paper. Therefore, in what follows we will only be concerned with checking (C1).

Definition 5 does not make any assumptions about the type of models under consideration. The only requirement is that the models are compatible in the inputs they accept and the outputs they produce. A special case is when $\boldsymbol{\mathcal{M}}$ is a set of *homogeneous* FFNNs, that is, FFNNs with the same number, sizes and activations of layers.

The robustness property in Definition 5 is an instance of a *relational property*, relating the executions of two or more models. One of the key contributions of our work is a reduction of this relational property (the robustness property) for sets of homogeneous FFNNs into a unary property defined over the execution of a single FFNN. This step relies on the concept of *product construction*, presented below.

**Product construction.** Given a set $\mathcal{M} = \{\mathcal{M}_1, \ldots, \mathcal{M}_n\}$ of homogeneous FFNNs and a class $c \in \{0, 1\}$, we seek to construct a *product network* $\mathcal{P}_c$ such that for $x' \in \mathbb{R}^m$ and $\mathcal{P}_c(x') = (v, u^1, \ldots, u^n)^\intercal$, the following are equivalent:

**(P1)** $v = 0$ and $u^j > 0$ for all $j \in \{1, \ldots, n\}$

**(P2)** $\mathcal{M}(x') = 1 - c$ for each $\mathcal{M} \in \mathcal{M}$.

Intuitively, $v = 0$ ensures that $\mathcal{M}_j(x') \neq c$, for each $j$. Additionally, we use $u^1, \ldots, u^n$ to detect whether the classification outcome of some model is undefined for $x'$, thus ruling it out as a potential CFX. Namely, $u^j = 0$ would mean that $\mathcal{M}_j(x')$ is undefined. We now present how $\mathcal{P}_c$ is constructed (see Figure 1 for a pictorial representation).

Assume that each model $\mathcal{M}_j$ has $k$ layers with $W_i^j$ being the weight matrix and $B_i^j$ the bias vector in its $i$-th layer. Additionally, we use $(o_0^j, o_1^j)^\intercal$ to denote the output of model $\mathcal{M}_j$. We set $\mathcal{P}_c$ to be an FFNN with $k + 3$ layers, where:

- The input layer of $\mathcal{P}_c$ is of size $m$
- The layer $i \in \{1, \ldots, k-1\}$ of $\mathcal{P}_c$ uses ReLU activation function and is parameterised by the matrix $W_i$ and bias vector $B_i$ obtained as:

$$W_1 = \begin{bmatrix} W_1^1 \\ \vdots \\ W_1^n \end{bmatrix} \quad W_i = \begin{bmatrix} W_i^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & W_i^n \end{bmatrix} \quad B_i = \begin{bmatrix} B_i^1 \\ \vdots \\ B_i^n \end{bmatrix}$$

- Layer $k$ is of size $2n$ and uses identity activation function, layers $k+1$ and $k+2$ are of size $2n$ and use ReLU activation function, and final layer is of size $n + 1$.
- The weights $W_{k+1} \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}$ is the block diagonal matrix:

$$W_{k+1} = \begin{bmatrix} A & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & A \end{bmatrix} \text{ where } A = \begin{bmatrix} a_0 & a_1 \\ -a_0 & -a_1 \end{bmatrix},$$

  for $a_c = 1$, $a_{1-c} = -1$, while $B_{k+1}$ is a zero vector.
- The weights $W_{k+2} \in \mathbb{R}^{2n} \times \mathbb{R}^{2n}$ is the block diagonal matrix:

$$W_{k+2} = \begin{bmatrix} D & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & D \end{bmatrix} \text{ where } D = \begin{bmatrix} -1 & 0 \\ 1 & 1 \end{bmatrix},$$

  and $B_{k+2} = (1, 0, 1, 0, \ldots, 1, 0)^\intercal$.
- The weights $W_{k+3} \in \mathbb{R}^{n+1} \times \mathbb{R}^{2n}$ is the matrix:

$$W_{k+3} = \begin{bmatrix} -\frac{1}{n} & 0 & -\frac{1}{n} & 0 & \cdots & -\frac{1}{n} & 0 \\ 0 & 1 & 0 & & \cdots & & \\ 0 & 0 & 0 & 1 & 0 & \cdots & \\ & & & \cdots & & & \\ 0 & & & \cdots & & 0 & 1 \end{bmatrix},$$

  and $B_{k+3} = (1, 0, \cdots, 0)^\intercal$.

Next, we show that finding a robust CFX $x'$ across a set of models reduces to finding an input for which (P1) holds and vice versa.

**Lemma 1.** *Let $\mathcal{M}$ be a set of homogeneous FFNNs consistent for a factual input $x \in \mathbb{R}^m$, classified as $c \in \{0, 1\}$ by $\mathcal{M}$, and $x' \in \mathbb{R}^m$ such that $d(x, x') \leq \delta$. Then (P1) holds iff $x'$ is a robust counterfactual for $x$ across $\mathcal{M}$.*

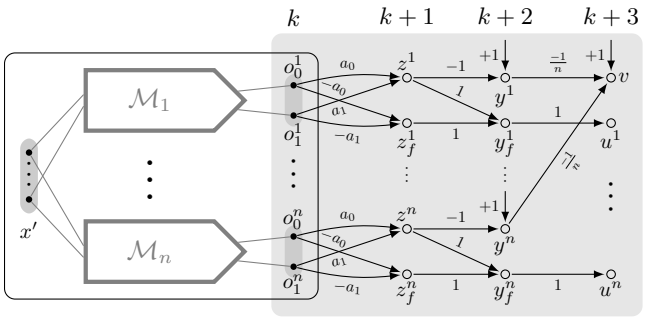


Figure 1: Product network $\mathcal{P}_c$.

*Proof.* Let $x \in \mathbb{R}^m$ be classified as $c \in \{0, 1\}$ by $\mathcal{M}$. Let $x' \in \mathbb{R}^m$ such that $d(x, x') \leq \delta$.

($\Rightarrow$) Since (P2) follows from (P1), we have that $\mathcal{M}(x') = 1 - c$ for each $\mathcal{M} \in \mathcal{M}$. In conjunction with $d(x, x') \leq \delta$, we obtain that $x'$ is a CFX for $x, \delta, \mathcal{M}$ for each $\mathcal{M} \in \mathcal{M}$. Hence, $x'$ is a robust counterfactual for $x$ across $\mathcal{M}$.

($\Leftarrow$) Let $x'$ be a robust counterfactual for $x$ across $\mathcal{M}$. Then $\mathcal{M}(x') = 1 - c$ for each $\mathcal{M} \in \mathcal{M}$. By construction of $\mathcal{P}_c$, from (P2) we obtain (P1). □

Lemma 1 is a strong result that enables us to generate robust explanations by using existing off-the-shelf approaches for verification of reachability properties for FFNNs. We can now prove our main result, which shows that determining the existence of a robust counterfactual is NP-complete.

**Theorem 1.** *The problem of existence of a robust explanation for sets of homogeneous FFNNs is NP-complete.*

*Proof.* The lower bound follows from the lower bound for the existence of a CFX for a single model, which in turn can be obtained by a reduction from the complement of the local robustness property for ReLU FFNNs (Katz et al. 2017).

As for the upper bound, let $\mathcal{M}$ be a set of models consistent for $x \in \mathbb{R}^m$, and let $c$ be the class of $x$ in $\mathcal{M}$. By Lemma 1 it follows that there exists a robust explanation $x'$ for $x$ across $\mathcal{M}$ iff $d(x, x') \leq \delta$ and $\mathcal{P}_c(x') = 0$. Since checking the latter is NP-complete and the product network $\mathcal{P}_c$ is linear in the size of $\mathcal{M}$, the result follows. □

Finally, we relax the requirement of homogeneity and extend the above upper bound to sets of arbitrary models representing piecewise linear functions. For such sets of models we can realise a product construction as a MILP, e.g., following the approach of (Akintunde et al. 2020). We can then reduce checking existence of a robust explanation to the MILP feasibility problem known to be NP-complete, which gives our last result.

**Theorem 2.** *The problem of existence of a robust explanation for sets of piecewise-linear models is NP-complete.*

## 4 Experimental Results

We have implemented a prototype for computing robust CFXs for ReLU-activated FFNNs[1]. The implementation

[1]https://github.com/fraleo/kr23_model_multiplicity

| | $n=2$ | | | $n=3$ | | | $n=4$ | | | $n=5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **no** | $\ell_1$ | **lof** | **no** | $\ell_1$ | **lof** | **no** | $\ell_1$ | **lof** | **no** | $\ell_1$ | **lof** |
| *credit* | 50/50 | 0.35 | 0.04 | 50/50 | 0.36 | 0.04 | 50/50 | 0.36 | 0.04 | 50/50 | 0.36 | 0.04 |
| *diabetes* | 50/50 | 0.90 | 0.72 | 50/50 | 1.03 | 0.76 | 50/50 | 1.07 | 0.76 | 50/50 | 1.09 | 0.8 |
| *no2* | 41/41 | 0.37 | 1.0 | 29/29 | 0.48 | 1.0 | 22/22 | 0.54 | 1.0 | 22/22 | 0.57 | 1.0 |

Table 2: Statistics for increasing values of $n$ and different datasets.

builds a product construction as a single program using the MILP encoding of a neural network as a building block (Mohammadi et al. 2021). This MILP computes an optimal $x'$ for which the classification outcomes for all models are required to flip, as per Definition 3.

We have conducted numerical experiments on commonly used tabular datasets: *credit* (Hofmann 2016), *diabetes* (Smith et al. 1988) and *no2* (Vanschoren et al. 2013). These datasets require solving binary classification tasks, for which we train sets of neural network classifiers using Keras (Chollet 2015). All models have two hidden layers, each with ten nodes and ReLU activations. Experiments were carried out on a standard PC running Ubuntu 22.04 with 16GB RAM and processor Intel(R) Core i5-4460 CPU @ 3.20GHz. Gurobi v9.5.1 was used to solve MILP (Gu, Rothberg, and Bixby 2020).

In the following we present two sets of experiments:

- we start by evaluating the practical applicability of our approach and show that product construction in MILP can effectively be used to generate *optimal robust CFXs* for increasingly larger sets of models across all three datasets;
- we then focus on the *credit* dataset and study the scalability of the MILP-based approach and report results obtained while generating robust CFXs for up to 50 models.

**Generating robust CFXs.** For this experiment, we train $n = \{2, 3, 4, 5\}$ neural networks for each dataset. To induce model multiplicity, we change the seed used for training thus obtaining sets of models that *(i)* share the same architecture, *(ii)* achieve similar accuracy, but *(iii)* differ in their internal parametrisation. The resulting accuracies, averaged over five models are: $0.94 \pm 1.4\text{e}{-7}$, $0.72 \pm 9.8\text{e}{-4}$ and $0.52 \pm 1.1\text{e}{-3}$ for *credit*, *diabetes* and *no2* respectively. We note that this is the same scenario as the one presented in Table 1, where state of the art algorithms failed to produce entirely robust CFXs.

Table 2 summarises the results. For each dataset and each set of models, we report **(no)** the number of optimal CFXs obtained out of the number of factual inputs considered; ($\ell_1$) the Manhattan distance between each CFX and the corresponding input; **(lof)** the local outlier factor score (Breunig et al. 2000), which measures the extent to which an instance lies within the data manifold (+1 for inliers, -1 otherwise). We note that for the *no2* dataset we were unable to find 50 inputs satisfying Definition 4, hence **(no)** includes the number of considered inputs. We average ($\ell_1$) and **(lof)** over the generated CFXs. Overall we can observe that our approach was always successful at generating robust CFXs for all the datasets and values of $n$ considered. We observe that generating robust CFXs results in a slight increase in cost ($\ell_1$ distance) wrt to the single model case[2]; however, this is in line with what reported in other works focusing on robustness to model changes (Dutta et al. 2022; Jiang et al. 2023). The **(lof)** score tends to be high in many cases, showing that our CFXs are well-within the data manifold and thus more plausible. However, for the *credit* dataset, the score drops considerably, suggesting that a better strategy to generate robust CFXs for this dataset may exist.

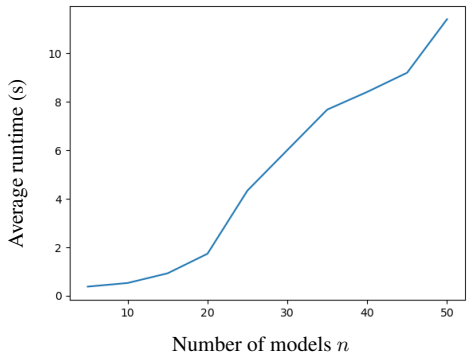


Figure 2: Computation time of a CFX robust across $n$ models.

**Assessing scalability.** Our earlier experiments demonstrated that our approach is able to generate robust CFXs for sets of moderate size without compromising the quality of the resulting CFXs. We now show that the approach can scale to larger sets. We focus on the *credit* dataset and train 50 models using the same training strategy adopted for previous experiments. We generate 30 CFXs for sets of size $n = \{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$; we report the runtime for each value of $n$ averaged over the generated CFXs in Figure 2. Leveraging on modern MILP solvers, our approach is able to handle large values of $n$ efficiently, taking approximately 12 seconds to generate a CFX for 50 models on a standard PC. Our results demonstrate the practical applicability of our approach, despite the worst case complexity discussed in the previous section.

## 5 Conclusions

Recent studies on model multiplicity have highlighted potential concerns regarding the justifiability of CFXs. In this paper we used relational verification as a formal framework to reason about robustness of CFXs under model multiplicity. We showed that generating CFXs that satisfy this property is NP-complete, and proposed a MILP-based approach to find robust CFXs for sets of up to 50 models.

This paper opens several avenues for future work. Firstly, while our experiments only considered neural networks with the same architecture, we plan to apply similar techniques to wider ranges of models. Additionally, we plan to investigate new algorithms to further improve the scalability of our approach and extend its applicability to models operating on high-dimensional data.

[2]Using PROTO (Looveren and Klaise 2021) on a single model, we obtain average $\ell_1$ distances of 0.38, 0.77 and 0.20 for *credit*, *diabetes* and *no2* respectively

## Acknowledgments

Leofante was funded by Imperial College London under the Imperial College Research Fellowship programme.

## References

Akintunde, M. E.; Botoeva, E.; Kouvaros, P.; and Lomuscio, A. 2020. Verifying strategic abilities of neural-symbolic multi-agent systems. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR 2020)*, 22–32.

Barthe, G.; Crespo, J. M.; and Kunz, C. 2011. Relational verification using product programs. In *Proceedings of the 17th International Symposium on Formal Methods (FM 2011)*, volume 6664 of *Lecture Notes in Computer Science*, 200–214. Springer.

Black, E.; Raghavan, M.; and Barocas, S. 2022. Model multiplicity: Opportunities, concerns, and solutions. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT 2022)*, 850–863. ACM.

Breunig, M. M.; Kriegel, H.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104. ACM.

Chollet, F. 2015. Keras. https://keras.io.

Christakis, M.; Eniser, H. F.; Hoffmann, J.; Singla, A.; and Wüstholz, V. 2022. Specifying and testing k-safety properties for machine-learning models. *CoRR* abs/2206.06054.

Darwiche, A. 2020. Three modern roles for logic in AI. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS 2020)*, 229–243. ACM.

Dominguez-Olmedo, R.; Karimi, A.; and Schölkopf, B. 2022. On the adversarial robustness of causal algorithmic recourse. In *Proceedings of the International Conference on Machine Learning (ICML 2022)*, volume 162 of *Proceedings of Machine Learning Research*, 5324–5342. PMLR.

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

Dutta, S.; Long, J.; Mishra, S.; Tilli, C.; and Magazzeni, D. 2022. Robust counterfactual explanations for tree-based ensembles. In *Proceedings of the International Conference on Machine Learning (ICML 2022)*, volume 162 of *Proceedings of Machine Learning Research*, 5742–5756. PMLR.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gu, Z.; Rothberg, E.; and Bixby, R. 2020. Gurobi optimizer reference manual. http://www.gurobi.com.

Hofmann, H. 2016. German credit risk credit classification. https://www.kaggle.com/uciml/german-credit.

Ignatiev, A., and Silva, J. P. M. 2021. Sat-based rigorous explanations for decision lists. In *Proceedings of the 24th International Conference on Theory and Applications of Satisfiability Testing (SAT 2021)*, volume 12831 of *Lecture Notes in Computer Science*, 251–269. Springer.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019. Abduction-based explanations for machine learning models. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, 1511–1519. AAAI Press.

Jiang, J.; Leofante, F.; Rago, A.; and Toni, F. 2023. Formalising the robustness of counterfactual explanations for neural networks. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI 2023)*. AAAI Press.

Karimi, A.; Barthe, G.; Schölkopf, B.; and Valera, I. 2023. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Comput. Surv.* 55(5):95:1–95:29.

Katz, G.; Barrett, C.; Dill, D.; Julian, K.; and Kochenderfer, M. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the 29th International Conference on Computer Aided Verification (CAV 2017)*, volume 10426 of *Lecture Notes in Computer Science*, 97–117. Springer.

Khedr, H., and Shoukry, Y. 2023. Certifair: A framework for certified global fairness of neural networks. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI 2023)*. AAAI Press.

Leofante, F., and Lomuscio, A. 2023. Towards robust contrastive explanations for human-neural multi-agent systems. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)*, 2343–2345. ACM.

Looveren, A. V., and Klaise, J. 2021. Interpretable counterfactual explanations guided by prototypes. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2021)*, volume 12976 of *Lecture Notes in Computer Science*, 650–665. Springer.

Marques-Silva, J., and Ignatiev, A. 2022. Delivering trustworthy AI through formal XAI. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, 12342–12350. AAAI Press.

Marx, C. T.; Calmon, F. P.; and Ustun, B. 2020. Predictive multiplicity in classification. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, 6765–6774. PMLR.

Mohammadi, K.; Karimi, A.; Barthe, G.; and Valera, I. 2021. Scaling guarantees for nearest counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES 2021 )*, 177–187. ACM.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, 8024–8035.

Paulsen, B.; Wang, J.; Wang, J.; and Wang, C. 2020. NEURODIFF: scalable differential verification of neural net-

works using fine-grained approximation. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE 2020)*, 784–796. IEEE.

Paulsen, B.; Wang, J.; and Wang, C. 2020. Reludiff: differential verification of deep neural networks. In *Proceedings of the 42nd International Conference on Software Engineering (ICSE 2020)*, 714–726. ACM.

Pawelczyk, M.; Datta, T.; van den Heuvel, J.; Kasneci, G.; and Lakkaraju, H. 2022. Algorithmic recourse in the face of noisy human responses. *CoRR* abs/2203.06768.

Pawelczyk, M.; Broelemann, K.; and Kasneci, G. 2020. On counterfactual explanations under predictive multiplicity. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI 2020)*, volume 124 of *Proceedings of Machine Learning Research*, 809–818. AUAI Press.

Sharma, S.; Henderson, J.; and Ghosh, J. 2020. CERTIFAI: A common framework to provide explanations and analyse the fairness and robustness of black-box models. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES 2020)*, 166–172.

Slack, D.; Hilgard, A.; Lakkaraju, H.; and Singh, S. 2021. Counterfactual explanations can be manipulated. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 62–75.

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

Upadhyay, S.; Joshi, S.; and Lakkaraju, H. 2021. Towards robust and reliable algorithmic recourse. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 16926–16937.

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. Openml: networked science in machine learning. *SIGKDD Explor.* 15(2):49–60.

Wachter, S.; Mittelstadt, B.; and Russell, C. 2018. Counterfactuals explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology* 31(2).