Manifold-Aligned Counterfactual Explanations for Neural Networks

Asterios Tsiourvas
MIT

Wei Sun
IBM Research

Georgia Perakis
MIT

Abstract

We study the problem of finding optimal manifold-aligned counterfactual explanations for neural networks. Existing approaches that involve solving a complex mixed-integer optimization (MIP) problem frequently suffer from scalability issues, limiting their practical usefulness. Furthermore, the solutions are not guaranteed to follow the data manifold, resulting in unrealistic counterfactual explanations. To address these challenges, we first present a MIP formulation where we explicitly enforce manifold alignment by reformulating the highly nonlinear Local Outlier Factor (LOF) metric as mixed-integer constraints. To address the computational challenge, we leverage the geometry of a trained neural network and propose an efficient decomposition scheme that reduces the initial large, hard-to-solve optimization problem into a series of significantly smaller, easier-to-solve problems by constraining the search space to "live" polytopes, i.e., regions that contain at least one actual data point. Experiments on real-world datasets demonstrate the efficacy of our approach in producing both optimal and realistic counterfactual explanations, and computational traceability. Code available at `https://github.com/asterios-tsiourvas/relu_cfx`.

1 Introduction

In recent years, there has been a growing demand for interpretable AI [Doshi-Velez and Kim, 2017, Murdoch et al., 2019], where machine learning models can be understood by humans. Counterfactual explanations have been used to explain model predictions and provide actionable insights [Mothilal et al., 2020a, Guidotti, 2022]. Specifically, for a data point, a counterfactual explanation identifies the minimum change that will lead to a different outcome under a given predictive model. A desirable property of counterfactual explanations is being *realistic*. Take the example of a loan application where an applicant got rejected by a machine learning model, which takes into account various features, such as the applicant's income and loan amount. A recommendation of reducing the loan amount by 5% to gain loan approval is far more realistic to execute, hence a more desirable counterfactual explanation, compared to an alternative suggestion of doubling the income.

To measure the realism of counterfactual explanations, one of the most well-known metrics is Local Outlier Factor (LOF) [Breunig et al., 2000]. The LOF score for a data point measures the local deviation in the density of a given sample, i.e., a low LOF score signifies a stronger alignment of the resulting counterfactual explanation with the data distribution, while a high LOF score indicates that a data point is an outlier. The vast majority of the literature has utilized LOF as an evaluation metric [Guidotti, 2022, Dutta et al., 2022, Lucic et al., 2022]. To the best of our knowledge, no prior work has explicitly incorporated constraints that aimed at achieving a desirable LOF value of counterfactual explanations due to its nonlinearity and computational complexity.

Neural networks, especially with non-linear activations such as ReLU, have gained immense popularity due to their remarkable ability to model complex nonlinear relationships in data, making them a ubiquitous technology across applications [LeCun et al., 2015]. However, due to their complex structure, obtaining high-quality counterfactual explanations that are both optimal (in terms of the minimum distance from the given sample point) and realistic remains a challenging task. This is the key question that we are attempting to address in this paper. Our contributions are as follows:

- We show how to explicitly enforce manifold alignment constraints into the optimization problem

Proceedings of the 27[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238.

by reformulating the LOF metric as a set of mixed-integer constraints. This approach guarantees the adherence of the resulting optimal solution to the underlying data distribution, i.e., a more realistic counterfactual explanation. More specifically, we show that with ℓ_1 or ℓ_∞ norms as the distance measure, we obtain a set of mixed-integer *linear* constraints. Meanwhile, with ℓ_2 norm, it gives rise to a set of mixed-integer *quadratic* constraints. We also want to point out that in addition to neural networks, this result on reformulating LOF such that the manifold alignment constraints can then be incorporated into the counterfactual explanation problem is applicable to any type of machine learning model that can be expressed by mixed-integer constraints, such as logistic regression, decision trees, tree ensembles, and more.

- Even in the absence of the manifold alignment constraint, the initial MIP formulation that determines the optimal counterfactual explanation can easily become intractable due to the large number of binary decision variables required to model the complex neural network structure. Having the LOF constraint exacerbates the existing computational challenge. We propose an efficient decomposition scheme that utilizes the geometry of ReLU networks and reduces the initial large, hard-to-solve optimization problem into a series of significantly smaller and easier-to-solve problems. This is achieved by limiting the search space to live polytopes, i.e. polytopes of the input space generated by the network that contain at least one data point in the desired outcome class. We further enhance the proposed decomposition scheme by strategically selecting a subset of live polytopes as the search space. We show analytically that with our strategy, the probability of missing the live polytope that yields the optimal solution decreases exponentially as the subset size increases.

- We conduct experiments on multiple real-world datasets and demonstrate that our proposed formulation consistently produces more realistic and closer counterfactual explanations to the factual data compared to competing benchmarks. For larger and more complex neural networks, our proposed decomposition scheme achieves significant gains in computational tractability. Besides the speedup, experiments also reveal an added benefit of leveraging the live polytopes which implicitly encourages realistic counterfactual explanations even without explicitly enforcing the manifold alignment constraint.

2 Related Literature

In recent years, ReLU networks have gained significant attention because of their inherent piecewise linear structure which promotes analytical tractability [Montúfar et al., 2014, Lee et al., 2019]. This structure has been utilized for a variety of applications, such as robustness verification [Tjeng et al., 2018] and network compression [Serra et al., 2020]. Additionally, multiple researchers have focused on optimizing already trained ReLU networks for downstream tasks, utilizing both mixed-integer optimization [Fischetti and Jo, 2018, Anderson et al., 2020, Palma et al., 2021] and approximate methods [Katz et al., 2017, Xu et al., 2020, Perakis and Tsiourvas, 2022]. Recent studies have also studied ReLU networks' expressive power [Arora et al., 2018, Yarotsky, 2017] as well as their connection with other machine learning algorithms and settings [Lee and Jaakkola, 2020, Sun and Tsiourvas, 2023].

To generate counterfactual explanations from ReLU networks, people have mostly used MIP [Mohammadi et al., 2021] or SMT solvers [Karimi et al., 2020]. While both approaches offer optimality guarantees in terms of their proximity to the factual sample when compared to model-agnostic approaches [Wexler et al., 2019, Mothilal et al., 2020b], their extensive runtime severely hinders their practicality. For instance, [Karimi et al., 2020] showed that even for small ReLU networks (i.e., 1 hidden layer with 20 neurons), SMT solvers fail to scale effectively. While MIP-based optimization methods for counterfactual explanations have found their successes for simpler linear models [Ustun et al., 2019, Russell, 2019] or tree-based models [Kanamori et al., 2020, Carreira-Perpiñán and Hada, 2021], they are only limited to moderate-sized neural networks due to their computational challenge [Mohammadi et al., 2021]. To improve computation tractability, one of the techniques we employ in our algorithm is to restrict the search space to live polytopes. The concept of live regions was initially introduced in [Carreira-Perpiñán and Hada, 2023] for finding counterfactual explanations under random forests. We extend this concept to ReLU networks. Moreover, we propose an efficient heuristic with a provable guarantee that tackles the issue of increased computational cost when the number of live regions expands, a problem initially discussed in [Carreira-Perpiñán and Hada, 2023].

A crucial requirement for counterfactual explanations is realism [Verma et al., 2020]. Numerous research studies have employed various met-

rics to evaluate whether the resulting counterfactual explanation conforms to the data distribution, with the most well-known metric being LOF [Guidotti, 2022, Dutta et al., 2022, Lucic et al., 2022]. With the exception of [Kanamori et al., 2020], existing work merely utilizes LOF as an evaluation metric. [Kanamori et al., 2020] propose an MIP approach that utilizes a special case of LOF (nearest neighbor equal to 1) as a regularization term in the objective. While this approach encourages the generation of realistic counterfactual explanations, it does not explicitly guarantee manifold alignment. Moreover, their approach requires tuning the regularization hyperparameter and is only applicable to linear and tree-based models. In our work, we explicitly impose a constraint within the MIP that guarantees realistic counterfactual explanations. Furthermore, we consider the general case (number of nearest neighbors is k) and propose an efficient algorithm for highly nonlinear models such as neural networks.

3 Methodology

3.1 Problem Definition

Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space and let $\mathcal{D} = \{x_i \in \mathcal{X}\}_{i=1}^n$ be a dataset consisting of n data points. We let $f : \mathcal{X} \to [0, 1]$ denote a machine learning model that takes a d-dimensional sample as input, and outputs a probability between 0 and 1. The final decision is denoted by $\mathbb{1}[f(x) \geq 0.5]$, where $\mathbb{1}[\cdot]$ is the indicator function. We say that all $x \in \mathcal{D}$ for which $f(x) < 0.5$ belong to the *negative* class $\mathcal{D}_-$, while all $x \in \mathcal{X}$ for which $f(x) \geq 0.5$ we say that they belong to the *positive* class $\mathcal{D}_+$. As expected, $\mathcal{D}_- \cap \mathcal{D}_+ = \emptyset, \mathcal{D}_- \cup \mathcal{D}_+ = \mathcal{D}$, while we also define that $|\mathcal{D}_-| = n_-$ and $|\mathcal{D}_+| = n_+$, with $n = n_- + n_+$. Finally, we define that $[n] := \{1, \ldots, n\}$.

Definition 3.1. (Counterfactual Explanation) *Given a factual data sample $x_F \in \mathcal{X}$ such that $f(x_F) < 0.5$, its closest counterfactual with respect to $f(\cdot)$ in terms of the ℓ_p norm, is a point $x_{CF} \in \mathcal{X}$ that is the solution to the following optimization problem*

$$\begin{aligned} x_{CF} \coloneqq \arg\min_{x \in \mathcal{X}} \quad & ||x_F - x||_p \\ \text{s.t.} \quad & f(x) \geq 0.5. \end{aligned} \tag{1}$$

The complexity of the problem (1) depends heavily on the structure of $f(\cdot)$. For example, for the well-known norms where $p \in \{1, 2, \infty\}$, if $f(\cdot)$ is a linear model, problem (1) is either a linear or a quadratic mixed-integer optimization problem, that can be solved efficiently using commercial solvers [Ustun et al., 2019]. In contrast, if $f(\cdot)$ is a highly nonlinear model such as a neural network, problem (1) may become a nonlinear, non-convex optimization problem which is significantly harder to solve [Mohammadi et al., 2021]. In this work, we focus on the case where $f(\cdot)$ is a trained ReLU neural network.

3.2 ReLU Neural Network Architecture

We consider the densely connected architecture [Huang et al., 2017], wherein each neuron receives inputs from the neurons of the preceding layer. The final output layer consists of a single neuron that outputs the probability via a sigmoid function.

Formally, we define the network as a function $f : \mathcal{X} \to \mathbb{R}$. We denote the number of hidden layers as L and the number of neurons at layer i as n_i. We also denote the output of layer i as $x^i \in \mathbb{R}^{n_i}$. For notational convenience we define $x^0 := x$, $n_0 := d$, $n_{L+1} := 1$. The neurons are defined by the weight matrix $W^i \in \mathbb{R}^{n_i \times n_{i-1}}$ and the bias vector $b^i \in \mathbb{R}^{n_i}$. We define x^i as $x^i = \max\{W^i x^{i-1} + b^i, 0\}$, where $\max\{\cdot, 0\}$ is the ReLU activation function [Nair and Hinton, 2010]. Finally, the output of the network is defined as $f(x) = \sigma(W^{L+1}x^L + b^{L+1})$, where $\sigma(\cdot)$ is the sigmoid activation, i.e., $\sigma(x) = (1 + e^{-x})^{-1}$.

When $f(\cdot)$ is a trained ReLU network, the constraint $f(x) \geq 0.5$ of problem (1) can be expressed as a set of mixed-integer linear constraints [Fischetti and Jo, 2018]. Specifically, for layer i, the equality constraint $x^i = \max\{W^i x^{i-1} + b^i, 0\}$ is equivalent to $x^i \in \mathcal{C}(x^{i-1})$ where

$$\mathcal{C}(x^{i-1}) = \left\{ y \;\middle|\; \begin{array}{l} y \geq W^i x^{i-1} + b^i, \\ y \leq W^i x^{i-1} + b^i - l^i \odot (1 - z^i), \\ y \leq u^i \odot z^i, y \geq 0 \end{array} \right\}.$$

In the previous definition of $\mathcal{C}(x^{i-1})$, $z^i \in \{0, 1\}^{n_i}$ are binary variables with z^i_j being equal to 1 if neuron j of layer i is activated and 0 otherwise, $\odot$ denotes the element-wise multiplication, u^i is the upper bound of x^i and l^i is the lower bound. The upper and lower bounds u^i and l^i, are calculated sequentially by solving the following problems $u^i = \max_{l^{i-1} \leq x^{i-1} \leq u^{i-1}} \{W^i x^{i-1} + b^i\}$ and $l^i = \min_{l^{i-1} \leq x^{i-1} \leq u^{i-1}} \{W^i x^{i-1} + b^i\}$ [Liu et al., 2020]. For x^0, we obtain the upper and lower bounds from $\mathcal{X}$. Therefore, we can rewrite problem (1) as the following MIP

$$\begin{aligned} \min_{x^0 \in \mathcal{X}, x^1, \ldots, x^L, z^1, \ldots, z^L} \quad & ||x_F - x^0||_p \\ \text{s.t.} \quad & x^i \in \mathcal{C}(x^{i-1}), \ \forall i \in [L], \\ & W^{L+1}x^L + b^{L+1} \geq 0, \end{aligned} \tag{2}$$

where the last constraint comes from the requirement that $f(x^0) \geq 0.5 \implies \sigma(W^{L+1}x^L + b^{L+1}) \geq 0.5 \implies$

$W^{L+1}x^L + b^{L+1} \geq \sigma^{-1}(0.5) = 0$. The closest counterfactual x_{CF} in this case is the optimal x_0.

Remark 3.1. The reformulation in problem (2) requires $\sum_{i=1}^{L} n_i$ binary, and $d + \sum_{i=1}^{L} n_i$ continuous variables.

As a result, the MIP problem is prone to scalability issues, especially for large and complex networks which are widely used due to their expressive modeling power.

3.3 Enforcing Manifold Alignment

Given its modeling flexibility, it is highly plausible that a ReLU network produces counterfactuals that deviate significantly from the data manifold, leading to unrealistic explanations. A well-known metric in the literature that quantifies whether a sample follows the underlying data distribution is Local Outlier Factor [Breunig et al., 2000].

Definition 3.2. (Local Outlier Factor (LOF) [Breunig et al., 2000]) *For $x \in \mathcal{D}$, let $N_k(x)$ to be its $k-$Nearest Neighbors in $\mathcal{D}$. The $k-$reachability distance* rd_k *of x with respect to x' is defined by* $\mathrm{rd}_k(x, x') = \max\{\delta(x, x'), d_k(x')\}$*, where $d_k(x')$ is the distance δ between x' and its $k-$th nearest instance in $\mathcal{D}$. The $k-$local reachability density of x is defined by* $\mathrm{lrd}_k(x) = |N_k(x)|(\sum_{x' \in N_k(x)} \mathrm{rd}_k(x, x'))^{-1}$*. Then, the $k-$LOF of x on $\mathcal{D}$ is defined as*

$$LOF_{k,\mathcal{D}}(x) = \frac{1}{|N_k(x)|} \sum_{x' \in N_k(x)} \frac{\mathrm{lrd}_k(x')}{\mathrm{lrd}_k(x)}.$$

For the distance metric $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, we consider the ℓ_p norm with $p \in \{1, 2, \infty\}$. By convention, a value of $LOF_{k,\mathcal{D}}(x) \leq 1$ indicates that x is an inlier that is aligned with the data manifold, while $LOF_{k,\mathcal{D}}(x) > 1$ indicates that x is an outlier.

The vast majority of the literature [Guidotti, 2022, Dutta et al., 2022, Lucic et al., 2022] has used LOF as a *post-process* evaluation metric that measures whether the learned closest counterfactual follows the manifold. We take a *proactive* approach in this work – we explicitly incorporate this metric into a constraint which requires the resulting counterfactual explanation to be close to the data manifold by solving the following optimization problem

$$\begin{array}{ll} \min\limits_{x \in \mathcal{X}} & ||x_F - x||_p \\ \text{s.t.} & f(x) \geq 0.5, \\ & LOF_{k,\mathcal{D}}(x) \leq t, \end{array} \tag{3}$$

where t is user-defined threshold.

At first glance, this constraint in terms of LOF appears highly nonlinear, exacerbating the known computational challenge associated with optimizing ReLU networks and making problem (3) unsolvable. As we will show in Theorem 3.1, in fact, this constraint can be rewritten as a set of well-behaved mixed-integer optimization constraints.

Theorem 3.1. *The constraint $LOF_{k,\mathcal{D}}(x) \leq t$ for $x \in \mathcal{X}$, fixed k and $p \in \{1, \infty\}$ can be expressed as a set of mixed-integer linear constraints. If $p = 2$, it can be expressed as a set of mixed-integer quadratic constraints.*

The proof can be found in the Appendix. The key step is to show that the k-reachability distance rd_k which uses the maximum operator can be linearized via algebraic manipulations and the introduction of additional binary variables.

Remark 3.2. Theorem 3.1 can be applied to any machine learning model that can be expressed via a set of mixed-integer constraints. In other words, in addition to ReLU networks, the manifold alignment constraint can also be added to the counterfactual optimization model when the underlying models are logistic regression, decision trees, and tree ensembles, etc.

Corollary 3.1. To formulate the constraint $LOF_{k,\mathcal{D}}(x) \leq t$ we need in total $n + n \cdot k = n(k+1)$ new binary variables. For the special case of $k = 1$, $LOF_{1,\mathcal{D}}(x)$ requires the introduction of n new binary variables.

Despite the expressiveness and optimality guarantees, this MIP formulation does not scale well for large neural networks. Moreover, the integration of the LOF constraints further increases the computational cost. In what follows, we show how we can reduce the initial large, hard-to-solve MIP in (2) or (3) into a sequence of easier-to-solve optimization problems with much fewer decision variables by exploiting the geometry of ReLU networks.

4 An Efficient Decomposition Algorithm

4.1 Geometry of ReLU Networks

It is known that once an activation pattern on the hidden layers of a ReLU network is fixed (or equivalently when the binary variables z^i of problem (2) are known), the network reduces into a linear model [Huchette et al., 2023]. The feasible set of this linear model is a polyhedron that is a subset of the input space $\mathcal{X}$ [Serra et al., 2018, Lee et al., 2019]. Feasible sets, coming from all feasible ReLU activation patterns, partition $\mathcal{X}$ into a finite number of polyhedra such that $\mathcal{P}_j \cap \mathcal{P}_{j'} = \emptyset$, $\forall j \neq j'$, and $\cup_j \mathcal{P}_j = \mathcal{X}$. We present a toy example in Figure 1 to illustrate the partition scheme.

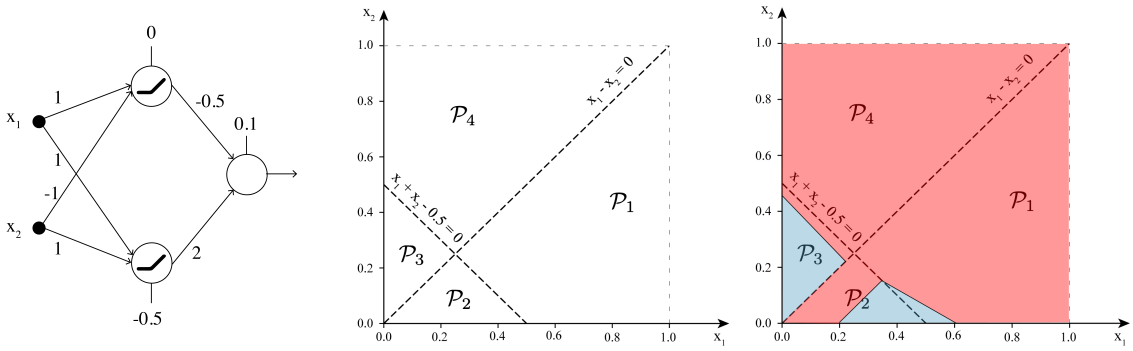

Figure 1: (Left) A one-layer ReLU neural network. (Middle) The partition of the input space $\mathcal{X}$ by the hidden layer of the ReLU network into 4 polytopes. (Right) The final partition of the input space $\mathcal{X}$ by the output neuron of the ReLU network, i.e. $\mathbb{1}[f(x) \geq 0.5]$, between positive and negative regions.

Example 1: Consider a setting with two features x_1, x_2, where $\mathcal{X} = [0,1]^2$ and a trained one-layer ReLU network, as depicted in Figure 1 (Left). We have that $x^1 = (\max\{x_1 - x_2, 0\}, \max\{x_1 + x_2 - 0.5, 0\})$. By enumerating all possible activation patterns for the hidden layer, we obtain four convex polyhedra, $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ and $\mathcal{P}_4$, that partition the input space $\mathcal{X}$ as shown in Figure 1 (Middle). The partitions are

- $\mathcal{P}_1 = \{(x_1, x_2) \in \mathcal{X} : x_1 - x_2 \geq 0, x_1 + x_2 - 0.5 \geq 0\}$,
- $\mathcal{P}_2 = \{(x_1, x_2) \in \mathcal{X} : x_1 - x_2 \geq 0, x_1 + x_2 - 0.5 < 0\}$,
- $\mathcal{P}_3 = \{(x_1, x_2) \in \mathcal{X} : x_1 - x_2 < 0, x_1 + x_2 - 0.5 < 0\}$,
- $\mathcal{P}_4 = \{(x_1, x_2) \in \mathcal{X} : x_1 - x_2 < 0, x_1 + x_2 - 0.5 \geq 0\}$.

Finally, in Figure 1 (Right) we observe the final partition of the input space $\mathcal{X}$ where all instances that belong to the blue polytopes are predicted to belong to the negative class and all instances that belong to the red polytopes are predicted to belong to the positive class. For a given polytope $\mathcal{P}_j$, the decision boundary is retrieved by solving the linear equation $f(x) = 0$ for $x \in \mathcal{P}_j$.

Given a trained ReLU network, a key observation is that problems such as (2) and (3) can be optimally solved by enumerating all the feasible polytopes $\mathcal{P}_j$ while solving an optimization problem over each $\mathcal{P}_j$. This is because obtaining a feasible polytope is equivalent to setting the binary variables z^i to its corresponding activation pattern. Specifically, problem (2) only requires solving a single LP (if $p \in \{1, \infty\}$) or a single CQP problem (if $p = 2$) for each feasible polytope $\mathcal{P}_j$. Similarly, for problem (3), a significantly smaller MIP needs to be solved, as the binary variables z^i are already fixed. For the network presented in Example 1, to solve problems (2) and (3) we need to solve 4 smaller, hence, easier-to-solve optimization problems and output as x_{CF} the solution that gives the lowest objective value out of the 4 problems.

This approach generalizes and requires solving N smaller easier-to-solve optimization problems, where N is the number of all feasible polytopes. Nevertheless, N can become very large, i.e., exponential with respect to the parameters of the network, up to $\sum_{(j_1,\ldots,j_L)\in J} \prod_{l=1}^{L} \binom{n_l}{j_l}$, where $J = \{(j_1, \ldots, j_L) \in \mathbb{Z}^L : 0 \leq j_L \leq \min\{n_0, n_1 - j_1, \ldots, n_{l-1} - j_{l-1}\}, \forall l = 1, \ldots, L\}$ [Serra et al., 2018], making this approach remain challenging for very large ReLU networks. In the following section, we will discuss ways we can circumvent this computational hurdle.

4.2 Searching over Live Polytopes

Our key idea is to approximate the solution of the initial MIP by solving a moderate number of smaller and easier-to-solve optimization problems by searching over live polytopes. We first provide a formal definition, followed by an example to illustrate the concept.

Definition 4.1. (Live Polytope) *Given a trained ReLU network f, a live polytope of f is a feasible polytope generated by f that contains at least one actual data point of $\mathcal{D}_+$.*

Example 2: We continue with the neural network described in Example 1. In Figure 2, the live polytopes are the partitions of the input space $\mathcal{X}$ that contain data points with red crosses (belong to the positive class). Based on this example, our algorithm would only solve 2 sub-problems, the ones that correspond to polytopes $\mathcal{P}_2$ and $\mathcal{P}_3$, since $\mathcal{P}_1$ and $\mathcal{P}_4$ do not contain positive data points. This reduces the complexity of the problem from solving 4 sub-problems to 2 problems instead.

When this method is applied to a general ReLU net-

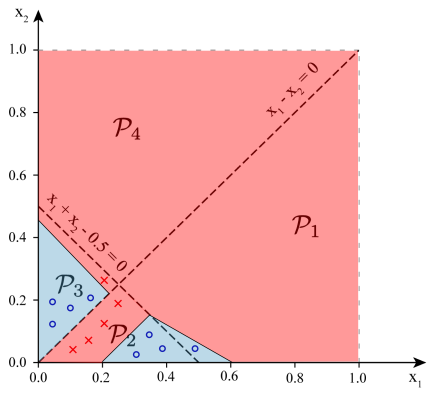

Figure 2: The final partition of the input space $\mathcal{X}$ for the neural network of the previous example. The red crosses correspond to positive data points and the blue circles to negative data points. $\mathcal{P}_2$ and $\mathcal{P}_3$ are live polytopes as they contain positive data points.

work, it reduces the complexity of solving N subproblems to solving at most n_+, as the number of live polytopes is upper-bounded by $n_+ < n$. Consequently, this method offers improved computational tractability compared to the initial MIP formulation.

As the size of the network f and the dataset $\mathcal{D}_+$ increases, the number of live polytopes may also increase, and as a result, the cost of the search will primarily be determined by the exhaustive exploration of all live polytopes. Meanwhile, it is expected that the nearest counterfactual explanation is more likely to be found within one of the closest live polytopes to x_F, rather than within one of the more distant polytopes.

Taking this insight into account, we propose a simple but efficient heuristic that only searches over a subset of the closest live polytopes. It takes a user-defined quantity m as an input, which specifies the maximum number of live polytopes to search over. Then, the algorithm calculates the distance between all points in $\mathcal{D}_+$ and x_F and retrieves the m live polytopes that contain a point with minimum distance to x_F. The method is described in detail in Algorithm 1. In Theorem 4.1, we analytically characterize the probability of missing the polytope that contains the optimal counterfactual explanation under this heuristic.

Theorem 4.1. The probability of not selecting the live polytope that leads to the closest counterfactual is of $\mathcal{O}(e^{-m})$, i.e. drops exponentially as m increases.

The proof is available in the Appendix.

Furthermore, assuming that the distance between a data point, denoted as $x \in \mathcal{D}_+$, and the nearest point to x_F belonging to the same live polytope as x, follows a known distribution, one can establish an upper

Algorithm 1 Heuristic Live Polytope Search

Input: Training set $\mathcal{D}_+$, ReLU network f, factual observation x_F, m number of live polytopes to search over.
Initialize $\tilde{x}_{CF} = None$, $dist_{CF} = +\infty$ and heap $\mathcal{F} = \{\}$.
For all $x \in \mathcal{D}_+$
    Calculate the distance d between x and x_F.
    Perform the feed-forward pass on x and retrieve activation pattern z.
    Insert the key-value pair (z, d) to $\mathcal{F}$.
    if $|\mathcal{F}| > m$
        Remove from $\mathcal{F}$ the key-pair value with the highest distance d.
    end if
end for
For all keys $z \in \mathcal{F}$
    Solve (2) or (3) with fixed activation pattern z and retrieve the solution $x, dist$.
    if $dist < dist_{CF}$
        $dist_{CF} \leftarrow dist$, $\tilde{x}_{CF} \leftarrow x$
    end if
end for
Return $\tilde{x}_{CF}$, $dist_{CF}$

threshold for the probability of not selecting the live polytope that results in the closest counterfactual. By solving for m one can obtain the value of m necessary to achieve a probability of error less than or equal to this threshold.

An interesting property of the proposed live polytope search is that, even when employed without the manifold-adhering constraints, it yields more realistic counterfactual explanations than the original MIP alone. Intuitively, when the input space $\mathcal{X}$ is $\mathbb{R}^d$, the data $\mathcal{D}$ usually reside in a manifold or subset of $\mathbb{R}^d$. As a result, to retrieve a realistic counterfactual explanation, one may need to estimate the distribution of the data and then solve problem (2) or (3) with respect to the estimated distribution. The live polytope search implicitly addresses this issue by performing a nonparametric probability density estimate. This estimate assigns positive mass to every live polytope and zero mass to all other polytopes. As demonstrated in the experimental results in the next section, this approach is capable of generating high-quality counterfactual explanations that align with the data manifold in many cases.

5 Experiments

We conduct experiments on three real-world datasets to validate the performance of our

methods compared to various benchmarks. All computational experiments were performed using Python 3.9 [Van Rossum and Drake, 2009], PyTorch 1.13 [Paszke et al., 2019], Gurobi 10.0 [Gurobi Optimization, LLC, 2023] and Scikit-learn [Buitinck et al., 2013]. All experiments were run on an internal cluster with a 2.20GHz Intel(R) Xeon(R) Gold 5120 CPU and 256 GB memory.

5.1 Setup

5.1.1 Datasets

We use the Adult income dataset ($d = 73$) [Dua and Graff, 2017] where the goal is to predict whether a person has an income of over $\$50,000$, the FICO dataset ($d = 34$) [Holter et al., 2018] to predict the chances of default, and lastly, the German credit dataset ($d = 27$) [Dua and Graff, 2017] to classify the credit of an individual as good or bad. For all three datasets, we perform one-hot encoding to incorporate categorical variables. We scale all continuous features using the min-max scaler to ensure that their domain falls within the range of $[0, 1]$. For the Adult dataset, we use the already existing train-test split, while for the remaining datasets we randomly split each dataset into into train (70%) and test (30%) instances.

5.1.2 Methods

For the experiments, we consider a 6-layer, densely connected ReLU network with either $50, 100$, or 200 neurons per hidden layer respectively as the underlying machine learning model that gives predictions. We train each network with a learning rate of 0.001 and a batch size of 128 for 20 epochs with early stopping where the patience parameter is set to 3.

Several benchmarks are considered, including the following model-agnostic methods: Minimum Observable (MO) [Wexler et al., 2019], that searches in the existing dataset for the closest sample that changes the prediction from negative to positive. Diverse Counterfactual Explanations (DiCE) [Mothilal et al., 2020b] aims to discover a diverse set of counterfactual explanations by solving an unconstrained optimization problem through gradient descent. In our experiments using DiCE, for each factual sample, we generate 10 diverse counterfactual explanations from which we report the one with the lowest distance. ReLAX [Chen et al., 2022] formulates the problem of generating counterfactual explanations as a sequential decision-making task and solving it via deep reinforcement learning. LORE [Guidotti et al., 2018] generates counterfactual explanations by deriving a set of counterfactual rules, suggesting the changes in the instance's features that lead to a different outcome. We also consider a gradient-based approach, PGD, [Verma et al., 2020] in which we use projected gradient descent to minimize the objective $\min_{x \in \mathcal{X}} \lambda(f(x) - 1)^2 + ||x_F - x||_2^2$. We run PGD for $10,000$ iterations (or until convergence), using the Adam optimizer [Kingma and Ba, 2015] with an initial learning rate of 0.001 and we test as λ all values ranging from 10 to 1000 with step size 10. We report the counterfactual explanation c_{CF} that achieves the lower distance $||x_F - x_{CF}||_2$.

We also consider the following model-specific benchmarks: GRACE [Le et al., 2020] which is a generative-based approach to explain neural network models' predictions. The MIP method for ReLU networks [Mohammadi et al., 2021] directly solves eq. (2). Lastly, our proposed methods, i.e., the live polytope search without the data-manifold adhering constraint (MIP-Live) with $m = 5$ presented in Section 4.2, and the live polytope search with the data-manifold adhering constraint (MIP-Live-DM) with $m = 5$, $k = 1$ and $t = 1$. For each MIP-based method, we set a time limit of 60 seconds per factual data point. We chose $m = 5$ in our experiments based on the sensitivity study where we compare the quality of the obtained solutions by varying the value of m. We observe that the performance of our methods plateaus for $m \geq 5$, thereby empirically verifying Theorem 4.1. Details of the sensitivity analysis can be found in the Appendix. We do not consider any SMT-based solver due to their known scalability issues (e.g., SMT solvers fail to solve ReLU networks with more than 20 neurons per hidden layer [Karimi et al., 2020]).

5.1.3 Evaluation Metrics

For evaluation, we randomly select 20 test instances that are assigned by the network to the negative class, and we seek their closest counterfactual explanations. To compare the resulting closest counterfactuals from each method, we report the average ℓ_2 distance from the factual data point (proximity), the percentage of the closest counterfactuals that are considered to be outliers by the LOF classifier (scikit-learn; default parameters), the average runtime per method, the optimality gap per MIP-based method and dataset, and, the number of features changed (sparsity). We include sparsity as a sparse counterfactual explanation promotes interpretability by reducing the complexity of the final suggestion. In our framework, sparsity can be imposed explicitly by incorporating the following constraints, $-M \cdot b_i \leq x_{F,i} - x_{CF,i} \leq M \cdot b_i, \sum_{i=1}^{d} b_i \leq s, b_i \in \{0, 1\}$, where s is a user-defined constant specifying the maximum number of allowed feature changes. More specifically, we re-run our methods with sparsity constraints with $s = 2$ to demonstrate that our meth-

Table 1: Average proximity (ℓ_2 distance), percentage of outliers, sparsity, and generation time for Adult, FICO, and German.

	50				100				200			
Adult	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time
MO	$0.50_{\pm 0.14}$	30%	$3.75_{\pm 0.77}$	0.46	$0.57_{\pm 0.23}$	20%	$4.05_{\pm 0.74}$	0.50	$0.61_{\pm 0.27}$	15%	$3.85_{\pm 0.91}$	0.54
PGD	$0.90_{\pm 0.26}$	60%	$4.50_{\pm 0.59}$	0.56	$0.83_{\pm 0.23}$	80%	$5.00_{\pm 0.32}$	0.18	$0.89_{\pm 0.20}$	75%	$5.00_{\pm 0.00}$	0.99
DiCE	$0.91_{\pm 0.35}$	70%	$5.00_{\pm 1.00}$	0.14	$0.85_{\pm 0.37}$	50%	$6.00_{\pm 1.22}$	0.15	$0.71_{\pm 0.30}$	65%	$6.30_{\pm 0.84}$	0.50
ReLAX	$0.49_{\pm 0.17}$	45%	$2.25_{\pm 0.43}$	60.0	$\mathbf{0.45}_{\pm 0.19}$	80%	$2.40_{\pm 0.49}$	60.0	$0.51_{\pm 0.25}$	75%	$2.30_{\pm 0.46}$	60.0
GRACE	$0.45_{\pm 0.21}$	60%	$\mathbf{2.00}_{\pm 0.00}$	0.26	$0.46_{\pm 0.19}$	65%	$\mathbf{2.00}_{\pm 0.00}$	0.25	$0.56_{\pm 0.30}$	65%	$\mathbf{2.00}_{\pm 0.00}$	0.26
LORE	$0.70_{\pm 0.28}$	65%	$2.75_{\pm 1.51}$	8.95	$0.59_{\pm 0.28}$	55%	$2.65_{\pm 1.06}$	12.29	$0.71_{\pm 0.27}$	80%	$2.50_{\pm 1.53}$	12.25
MIP	$1.90_{\pm 0.24}$	40%	$6.20_{\pm 0.87}$	60.0	-	-	-	60.0	-	-	-	60.0
MIP-Live-m=5	$\mathbf{0.36}_{\pm 0.18}$	35%	$4.45_{\pm 0.80}$	0.66	$\mathbf{0.45}_{\pm 0.22}$	25%	$2.35_{\pm 0.48}$	5.69	$\mathbf{0.48}_{\pm 0.12}$	20%	$4.10_{\pm 0.70}$	14.88
MIP-Live-m=5-s=2	$0.45_{\pm 0.25}$	30%	$\mathbf{2.00}_{\pm 0.00}$	0.85	$0.48_{\pm 0.25}$	25%	$\mathbf{2.00}_{\pm 0.00}$	5.97	$0.53_{\pm 0.28}$	25%	$\mathbf{2.00}_{\pm 0.00}$	19.58
MIP-Live-DM-m=5	$0.44_{\pm 0.22}$	**20%**	$4.30_{\pm 0.84}$	0.75	$0.51_{\pm 0.23}$	**10%**	$4.95_{\pm 0.86}$	6.00	$0.54_{\pm 0.26}$	**10%**	$4.45_{\pm 0.50}$	20.24
MIP-Live-DM-m=5-s=2	$0.50_{\pm 0.22}$	**20%**	$\mathbf{2.00}_{\pm 0.00}$	1.08	$0.54_{\pm 0.21}$	15%	$\mathbf{2.00}_{\pm 0.00}$	5.14	$0.56_{\pm 0.28}$	15%	$\mathbf{2.00}_{\pm 0.00}$	24.89

	50				100				200			
FICO	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time
MO	$0.58_{\pm 0.21}$	**0%**	$16.60_{\pm 1.80}$	0.33	$0.61_{\pm 0.27}$	**0%**	$16.60_{\pm 1.80}$	0.34	$0.59_{\pm 0.28}$	**0%**	$16.05_{\pm 1.83}$	0.36
PGD	$0.58_{\pm 0.22}$	10%	$20.10_{\pm 0.89}$	0.51	$\mathbf{0.58}_{\pm 0.15}$	10%	$20.20_{\pm 0.60}$	0.72	$0.60_{\pm 0.16}$	10%	$20.15_{\pm 0.79}$	0.58
DiCE	$0.95_{\pm 0.34}$	15%	$19.95_{\pm 1.77}$	0.86	$0.91_{\pm 0.29}$	10%	$20.10_{\pm 1.64}$	0.49	$0.98_{\pm 0.25}$	5%	$18.40_{\pm 2.20}$	0.52
ReLAX	$0.56_{\pm 0.24}$	10%	$2.50_{\pm 0.50}$	60.0	$0.60_{\pm 0.28}$	10%	$2.50_{\pm 0.50}$	60.0	$0.61_{\pm 0.26}$	10%	$2.50_{\pm 0.50}$	60.0
GRACE	$0.59_{\pm 0.28}$	15%	$2.45_{\pm 0.67}$	0.05	$0.61_{\pm 0.42}$	15%	$2.40_{\pm 0.66}$	0.04	$0.59_{\pm 0.24}$	5%	$2.50_{\pm 0.59}$	0.05
LORE	$0.70_{\pm 0.29}$	20%	$5.00_{\pm 1.67}$	6.03	$0.77_{\pm 0.37}$	35%	$4.95_{\pm}0.38$	8.01	$0.94_{\pm 0.42}$	40%	$4.90_{\pm 1.55}$	8.02
MIP	$2.48_{\pm 0.13}$	20%	$22.50_{\pm 0.50}$	60.0	-	-	-	60.0	-	-	-	60.0
MIP-Live-m=5	$\mathbf{0.52}_{\pm 0.14}$	**0%**	$16.50_{\pm 3.49}$	0.48	$\mathbf{0.58}_{\pm 0.27}$	**0%**	$18.45_{\pm 1.66}$	4.25	$\mathbf{0.56}_{\pm 0.11}$	**0%**	$18.60_{\pm 1.20}$	18.86
MIP-Live-m=5-s=2	$0.63_{\pm 0.17}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.62	$0.69_{\pm 0.27}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	9.76	$0.63_{\pm 0.26}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	24.11
MIP-Live-DM-m=5	$0.55_{\pm 0.12}$	**0%**	$17.90_{\pm 1.48}$	0.63	$0.62_{\pm 0.23}$	**0%**	$17.85_{\pm 1.06}$	8.10	$0.67_{\pm 0.21}$	**0%**	$19.35_{\pm 2.20}$	24.73
MIP-Live-DM-m=5-s=2	$0.67_{\pm 0.15}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.88	$0.74_{\pm 0.11}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	10.65	$0.73_{\pm 0.21}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	26.16

	50				100				200			
German	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time
MO	$1.35_{\pm 0.32}$	**0%**	$3.60_{\pm 0.49}$	0.12	$1.21_{\pm 0.50}$	**0%**	$3.35_{\pm 0.73}$	0.09	$1.18_{\pm 0.57}$	**0%**	$3.60_{\pm 0.80}$	0.08
PGD	$0.70_{\pm 0.23}$	5%	$2.65_{\pm 0.48}$	0.88	$1.14_{\pm 0.50}$	5%	$3.20_{\pm 0.93}$	0.33	$0.80_{\pm 0.38}$	5%	$2.85_{\pm 0.57}$	0.39
DiCE	$1.25_{\pm 0.20}$	5%	$5.80_{\pm 1.25}$	0.08	$1.23_{\pm 0.22}$	5%	$5.20_{\pm 1.57}$	0.08	$1.23_{\pm 0.37}$	5%	$5.65_{\pm 1.35}$	0.10
ReLAX	$0.62_{\pm 0.17}$	10%	$2.45_{\pm 0.67}$	26.91	$0.74_{\pm 0.30}$	5%	$2.20_{\pm 0.40}$	29.06	$0.73_{\pm 0.38}$	**0%**	$2.20_{\pm 0.60}$	30.32
GRACE	$0.66_{\pm 0.24}$	5%	$2.25_{\pm 0.62}$	0.07	$0.75_{\pm 0.25}$	**0%**	$2.85_{\pm 0.36}$	0.16	$0.71_{\pm 0.28}$	**0%**	$2.50_{\pm 0.67}$	0.14
LORE	$1.63_{\pm 0.21}$	**0%**	$3.10_{\pm 0.3}$	5.54	$1.45_{\pm 0.44}$	5%	$3.45_{\pm 1.24}$	7.41	$1.55_{\pm 0.40}$	10%	$3.40_{\pm 1.20}$	9.90
MIP	$2.44_{\pm 0.52}$	10%	$4.70_{\pm 0.78}$	60.0	-	-	-	60.0	-	-	-	60.0
MIP-Live-m=5	$\mathbf{0.59}_{\pm 0.18}$	**0%**	$3.45_{\pm 0.67}$	0.96	$\mathbf{0.71}_{\pm 0.31}$	**0%**	$3.45_{\pm 1.24}$	10.10	$\mathbf{0.70}_{\pm 0.29}$	**0%**	$2.80_{\pm 1.17}$	18.01
MIP-Live-m=5-s=2	$0.64_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.25	$0.79_{\pm 0.29}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	12.25	$0.80_{\pm 0.32}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	22.65
MIP-Live-DM-m=5	$0.62_{\pm 0.24}$	**0%**	$5.00_{\pm 1.10}$	1.12	$0.77_{\pm 0.28}$	**0%**	$5.10_{\pm 0.94}$	16.96	$0.78_{\pm 0.28}$	**0%**	$5.05_{\pm 1.83}$	24.90
MIP-Live-DM-m=5-s=2	$0.72_{\pm 0.27}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.81	$0.83_{\pm 0.35}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	17.81	$0.84_{\pm 0.35}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	29.32

ods can generate high-quality and sparse counterfactual explanations. Finally, we also report the validity, i.e. the ratio of the counterfactuals that actually have the desired class label to the total number of counterfactuals generated.

5.2 Results

In Table 1, we present the experimental results including the average ℓ_2 distance, the percentage of outliers, the sparsity measure, and the generation time achieved by every counterfactual explanation generation algorithm across datasets.

We observe that our MIP-based approaches achieve lower average ℓ_2 distances than the model-agnostic approaches. This is not surprising since the MIP-based approaches are constructed to generate optimal counterfactual explanations for ReLU networks. Among the MIP-based approaches, we observe that when the size of the ReLU network is moderate (50 neurons per hidden layer), the original MIP method is able to retrieve a counterfactual explanation. As the size of the ReLU network increases (100 and 200 neurons per hidden layer), the MIP method is unable to converge within the given time frame and produces sub-optimal solutions. On the other hand, MIP-Live and MIP-Live-DM, which incorporate the live polytope search, circumvent the scalability issue and are able to produce high-quality counterfactual explanations across datasets within the given time frame.

In terms of outliers, it is evident that MIP-Live-DM consistently outperforms the other methods. This outcome is expected, given that MIP-Live-DM explicitly incorporates the manifold alignment constraint. Meanwhile, MIP-Live also yields manifold-adherent counterfactual explanations in many cases, due to the implicit density estimation procedure discussed in Section 4.2.

For the sparsity measure, we observe that our methods with the sparsity constraints produce the counterfactual explanations with the lowest sparsity, while

also maintaining a low ℓ_2 distance. Among the remaining methods, we also observe that ReLAX and GRACE tend to produce sparse solutions, albeit with other metrics being worse off than ours.

Finally, it is worth mentioning the validity of the resulting counterfactual explanations and the optimality gap of the MIP-based methods. In our experiments, we observed that all benchmarks have a validity of 100% by design. In terms of the optimality gap, we observe that the gap achieved by the original MIP method is very high ($\geq 100\%$) even in the smallest ReLU network instance (50 neurons), while for larger networks it is practically infinite. In contrast, our methods, MIP-Live and MIP-Live-DM, that incorporate the live polytope search achieve a significantly lower optimality gap (up to 2%) and thus, always produce near-optimal solutions, highlighting the scalability of our approach.

6 Conclusion

Our work contributes to a growing body of research focused on generating counterfactual explanations from trained machine learning models to provide interpretability as well as actionable insights. We demonstrate that manifold alignment constraints based on the popular LOF metric can be directly incorporated into the optimization problem. This is achieved by reformulating the LOF metric into a set of mixed-integer constraints. This result can be applied to any machine learning model that can be expressed as a set of mixed-integer constraints. To circumvent the computational challenges of the resulting MIP problem, we propose an efficient decomposition scheme that leverages the geometry of ReLU networks and significantly reduces the search space into a moderately sized set of polytopes. Through experimental evaluation of real-world datasets, in addition to demonstrating computational tractability, we also validate the advantages of the proposed methods in terms of generating optimal and realistic counterfactual explanations.

References

[Anderson et al., 2020] Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., and Vielma, J. P. (2020). Strong mixed-integer programming formulations for trained neural networks. *Math. Program.*, 183(1):3–39.

[Arora et al., 2018] Arora, R., Basu, A., Mianjy, P., and Mukherjee, A. (2018). Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations, ICLR 2018*.

[Breunig et al., 2000] Breunig, M. M., Kriegel, H., Ng, R. T., and Sander, J. (2000). LOF: identifying density-based local outliers. In Chen, W., Naughton, J. F., and Bernstein, P. A., editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 93–104.

[Buitinck et al., 2013] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. *CoRR*, abs/1309.0238.

[Carreira-Perpiñán and Hada, 2021] Carreira-Perpiñán, M. Á. and Hada, S. S. (2021). Counterfactual explanations for oblique decision trees: Exact, efficient algorithms. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 6903–6911.

[Carreira-Perpiñán and Hada, 2023] Carreira-Perpiñán, M. Á. and Hada, S. S. (2023). Very fast, approximate counterfactual explanations for decision forests. In Williams, B., Chen, Y., and Neville, J., editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI*, pages 6935–6943.

[Chen et al., 2022] Chen, Z., Silvestri, F., Wang, J., Zhu, H., Ahn, H., and Tolomei, G. (2022). Relax: Reinforcement learning agent explainer for arbitrary predictive models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 252–261, New York, NY, USA. Association for Computing Machinery.

[Doshi-Velez and Kim, 2017] Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning.

[Dua and Graff, 2017] Dua, D. and Graff, C. (2017). UCI Machine Learning Repository.

[Dutta et al., 2022] Dutta, S., Long, J., Mishra, S., Tilli, C., and Magazzeni, D. (2022). Robust counterfactual explanations for tree-based ensembles. In *International Conference on Machine Learning*, pages 5742–5756. PMLR.

[Fischetti and Jo, 2018] Fischetti, M. and Jo, J. (2018). Deep neural networks and mixed integer linear optimization. *Constraints*, 23(3):296–309.

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

[Guidotti et al., 2018] Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. (2018). Local rule-based explanations of black box decision systems. *CoRR*, abs/1805.10820.

[Gurobi Optimization, LLC, 2023] Gurobi Optimization, LLC (2023). Gurobi Optimizer Reference Manual.

[Holter et al., 2018] Holter, S., Gomez, O., and Bertini, E. (2018). FICO Explainable Machine Learning Challenge.

[Huang et al., 2017] Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Huchette et al., 2023] Huchette, J., Muñoz, G., Serra, T., and Tsay, C. (2023). When deep learning meets polyhedral theory: A survey.

[Kanamori et al., 2020] Kanamori, K., Takagi, T., Kobayashi, K., and Arimura, H. (2020). Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *IJCAI*, pages 2855–2862.

[Karimi et al., 2020] Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. (2020). Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*, pages 895–905. PMLR.

[Katz et al., 2017] Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. (2017). Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*.

[Le et al., 2020] Le, T., Wang, S., and Lee, D. (2020). Grace: Generating concise and informative contrastive sample to explain neural network model's prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 238–248, New York, NY, USA. Association for Computing Machinery.

[LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *nature*, 521(7553):436–444.

[Lee et al., 2019] Lee, G.-H., Alvarez-Melis, D., and Jaakkola, T. S. (2019). Towards robust, locally linear deep networks. In *International Conference on Learning Representations*.

[Lee and Jaakkola, 2020] Lee, G.-H. and Jaakkola, T. S. (2020). Oblique decision trees from derivatives of relu networks. In *International Conference on Learning Representations*.

[Liu et al., 2020] Liu, X., Han, X., Zhang, N., and Liu, Q. (2020). Certified monotonic neural networks. *Advances in Neural Information Processing Systems*, 33:15427–15438.

[Lucic et al., 2022] Lucic, A., Oosterhuis, H., Haned, H., and de Rijke, M. (2022). FOCUS: flexible optimizable counterfactual explanations for tree ensembles. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, pages 5313–5322.

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

[Montúfar et al., 2014] Montúfar, G., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 2924–2932.

[Mothilal et al., 2020a] Mothilal, R. K., Sharma, A., and Tan, C. (2020a). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 607–617.

[Mothilal et al., 2020b] Mothilal, R. K., Sharma, A., and Tan, C. (2020b). Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 607–617.

[Murdoch et al., 2019] Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080.

[Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814.

[Palma et al., 2021] Palma, A. D., Behl, H. S., Bunel, R., Torr, P. H. S., and Kumar, M. P. (2021). Scaling the convex barrier with active sets. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

[Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., 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*, pages 8024–8035. Curran Associates, Inc.

[Perakis and Tsiourvas, 2022] Perakis, G. and Tsiourvas, A. (2022). Optimizing objective functions from trained relu neural networks via sampling. *CoRR*, abs/2205.14189.

[Russell, 2019] Russell, C. (2019). Efficient search for diverse coherent explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 20–28.

[Serra et al., 2020] Serra, T., Kumar, A., and Ramalingam, S. (2020). Lossless compression of deep neural networks. In *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 417–430. Springer.

[Serra et al., 2018] Serra, T., Tjandraatmadja, C., and Ramalingam, S. (2018). Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pages 4558–4566. PMLR.

[Sun and Tsiourvas, 2023] Sun, W. and Tsiourvas, A. (2023). Learning prescriptive relu networks. In *International Conference on Machine Learning, ICML*, pages 33044–33060. PMLR.

[Tjeng et al., 2018] Tjeng, V., Xiao, K. Y., and Tedrake, R. (2018). Evaluating robustness of neural networks with mixed integer programming. In *International Conference on Learning Representations*.

[Ustun et al., 2019] Ustun, B., Spangher, A., and Liu, Y. (2019). Actionable recourse in linear classification. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 10–19.

[Van Rossum and Drake, 2009] Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

[Verma et al., 2020] Verma, S., Boonsanong, V., Hoang, M., Hines, K. E., Dickerson, J. P., and Shah, C. (2020). Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*.

[Wexler et al., 2019] Wexler, J., Pushkarna, M., Bolukbasi, T., Wattenberg, M., Viégas, F., and Wilson, J. (2019). The what-if tool: Interactive probing of machine learning models. *IEEE transactions on visualization and computer graphics*, 26(1):56–65.

[Xu et al., 2020] Xu, J., Li, Z., Du, B., Zhang, M., and Liu, J. (2020). Reluplex made more practical: Leaky relu. In *2020 IEEE Symposium on Computers and communications (ISCC)*, pages 1–7. IEEE.

[Yarotsky, 2017] Yarotsky, D. (2017). Error bounds for approximations with deep relu networks. *Neural Networks*, 94:103–114.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes. Specifically, for each algorithm we report the average runtime and the sample size. Furthermore, for each MIP-based method, we report the number of continuous and binary variables.**
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**
   (b) Complete proofs of all theoretical results. **Yes**
   (c) Clear explanations of any assumptions. **Yes**

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses existing assets. **Yes**

   (b) The license information of the assets, if applicable. **Yes**

   (c) New assets either in the supplemental material or as a URL, if applicable. **Yes**

   (d) Information about consent from data providers/curators. **Not Applicable**

   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

   (a) The full text of instructions given to participants and screenshots. **Not Applicable**

   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**

   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

Manifold-Aligned Counterfactual Explanations for Neural Networks

Proof of Theorem 3.1

Theorem 3.1 *The constraint $LOF_{k,\mathcal{D}}(x) \leq t$ for $x \in \mathcal{X}$, fixed k and $p \in \{1, \infty\}$ can be expressed as a set of mixed-integer linear constraints. If $p = 2$, it can be expressed as a set of mixed-integer quadratic constraints.*

Proof. First, we can re-write

$$LOF_{k,\mathcal{D}}(x) \leq t \implies \frac{1}{|N_k(x)| \cdot \text{lrd}_k(x)} \sum_{x' \in N_k(x)} \text{lrd}_k(x') \leq t \implies \frac{1}{k \cdot \text{lrd}_k(x)} \sum_{x' \in N_k(x)} \text{lrd}_k(x') \leq t \implies$$
$$\implies \frac{1}{\text{lrd}_k(x)} \leq \frac{kt}{\sum_{x' \in N_k(x)} \text{lrd}_k(x')} \tag{1}$$

assuming that $|N_k(x)| = k$. Given that $\mathcal{D}$ is known, we can pre-compute for every $x \in \mathcal{D}$ its k-local reachability density $\text{lrd}_k(x)$, and the distance d_k with its k-th nearest instance in $\mathcal{D}$. For instance $x_i \in \mathcal{D}$, we denote for brevity the $\text{lrd}_k(x_i)$ as δ_i.

Case 1: $k = 1$

Given that $k = 1$, we introduce a set of binary variables $q_i, i = 1, \ldots, n$, that are equal to 1 only if x_i is the nearest neighbor of x. This can be modeled as the following set of constraints:

$$\sum_{i=1}^{n} q_i = 1, \tag{2}$$
$$\delta(x, x_j) \geq \delta(x, x_i) - M(1 - q_i), \ \forall i, j \in [n], \tag{3}$$
$$q_i \in \{0, 1\}, \ \forall i \in [n], \tag{4}$$

where $\delta(x, y) = ||x - y||_p$ for $p \in \{1, 2, \infty\}$ and M is a large value. Furthermore, we have that $\sum_{x' \in N_1(x)} \text{lrd}_1(x') = \sum_{i=1}^{n} q_i \frac{1}{\delta_i}$ and $\text{lrd}_1(x) = \frac{1}{\sum_{i=1}^{n} q_i \max\{\delta(x, x_i), d_1(x_i)\}}$. Therefore, the constraint becomes:

$$\sum_{i=1}^{n} q_i \max\{\delta(x, x_i), d_1(x_i)\} \leq t \cdot \sum_{i=1}^{n} q_i \frac{1}{\delta_i}, \tag{5}$$

that is equivalent to

$$\sum_{i=1}^{n} q_i \delta(x, x_i) \leq t \cdot \sum_{i=1}^{n} q_i \frac{1}{\delta_i}, \tag{6}$$
$$\sum_{i=1}^{n} q_i d_1(x_i) \leq t \cdot \sum_{i=1}^{n} q_i \frac{1}{\delta_i}. \tag{7}$$

The only thing left to linearize is the first constraint. Given that $\delta(x, x_i) \geq 0$, we introduce a new variable v_i and we linearize the constraint as follows:

$$\sum_{i=1}^{n} v_i \leq t \cdot \sum_{i=1}^{n} q_i \frac{1}{\delta_i}, \tag{8}$$

$$v_i \leq M q_i, \forall i \in [n], \tag{9}$$

$$v_i \geq 0, \forall i, i' \in [n], \tag{10}$$

$$v_i \leq \delta(x, x_i), \forall i, i' \in [n], \tag{11}$$

$$v_i \geq \delta(x, x_i) - M(1 - q_i), \forall i \in [n]. \tag{12}$$

Finally, the distance $\delta(x, x_i)$ can be incorporated using standard techniques depending on the value of p. For $p = 2$, which is the value of p that we use in the experiments, we obtain the following MIP.

$$\begin{aligned}
\min_{x^0 \in \mathcal{X}, x^1, \ldots, x^L, z^1, \ldots, z^L, v_1, \ldots, v_n, q_1, \ldots, q_n} \quad & ||x_F - x^0||_2 + M \cdot \textstyle\sum_{i=1}^{n} ||x_i - x^0||_2 \\
\text{s.t.} \quad & x^i \geq W^i x^{i-1} + b^i, \ \forall i \in [L], \\
& x^i \geq 0, \ \forall i \in [L], \\
& x^i \leq W^i x^{i-1} + b^i - l^i \odot (1 - z^i), \ \forall i \in [L], \\
& x^i \leq u^i \odot z^i, \ \forall i \in [L], \\
& W^{L+1} x^L + b^{L+1} \geq 0, \\
& v_i \leq M q_i, \forall i \in [n], \\
& v_i \geq 0, \forall i, i' \in [n], \\
& v_i \leq ||x_i - x^0||_2, \forall i, i' \in [n], \\
& v_i \geq ||x_i - x^0||_2 - M(1 - q_i), \forall i \in [n], \\
& ||x_j - x^0||_2 \geq ||x_i - x^0||_2 - M(1 - q_i), \ \forall i, j \in [n], \\
& \textstyle\sum_{i=1}^{n} q_i d_1(x_i) \leq t \cdot \sum_{i=1}^{n} q_i \frac{1}{\delta_i}, \\
& \textstyle\sum_{i=1}^{n} v_i \leq t \cdot \sum_{i=1}^{n} q_i \frac{1}{\delta_i}, \\
& \textstyle\sum_{i=1}^{n} q_i = 1, \\
& q_i \in \{0, 1\}, \ \forall i \in [n].
\end{aligned} \tag{13}$$

□

Case 2: General k.

We introduce the following set of binary variables, $q_{i,j}, i = 1, \ldots, n$ and $j = 1, \ldots, k$ that are equal to 1 when i is the j-th nearest neighbor of x. This is modeled by the following set of constraints:

$$\sum_{i=1}^{n} q_{i,j} = 1, \ \forall j \in [k], \tag{14}$$

$$\sum_{j=1}^{k} q_{i,j} \leq 1, \ \forall i \in [n], \tag{15}$$

$$\delta(x, x_l) \geq \delta(x, x_i) - M(1 - q_{i,j}) - M \sum_{j'=j-1}^{1} (1 - q_{l,j'}), \forall i, l \in [n], \forall j \in [k], \tag{16}$$

where M is a large value. We also define the set of binary variables $q_i, i = 1, \ldots, n$ that are equal to 1 if x_i is a nearest neighbor of x, i.e.,

$$q_i = \sum_{j=1}^{k} q_{i,k}, \ \forall i \in [n]. \tag{17}$$

We have that $\sum_{x' \in N_k(x)} \text{lrd}_k(x') = \sum_{i=1}^{n} q_i \cdot \frac{1}{\delta_i}$ and $\text{lrd}_k(x) = k \cdot \frac{1}{\sum_{i=1}^{n} q_i \max\{\delta(x, x_i), d_k(x_i)\}}$. Therefore, the constraint becomes:

$$\frac{1}{k} \cdot \sum_{i=1}^{n} q_i \max\{\delta(x, x_i), d_k(x_i)\} \leq kt \sum_{i=1}^{n} q_i \frac{1}{\delta_i}. \tag{18}$$

To linearize the maximum, we introduce the set of continuous variables $y_i, i = 1 \ldots, n$, and the set of binary variables $u_i, i = 1 \ldots, n$, for which we have that

$$y_i \geq \delta(x, x_i),\ \forall i \in [n], \tag{19}$$
$$y_i \geq d_k(x_i),\ \forall i \in [n], \tag{20}$$
$$y_i \leq \delta(x, x_i) + M u_i,\ \forall i \in [n], \tag{21}$$
$$y_i \leq d_k(x_i) + M(1 - u_i).\ \forall i \in [n]. \tag{22}$$

Furthermore, to linearize the product $q_i y_i$ we introduce the set of continuous variables $w_i, i = 1, \ldots, n$ such that

$$w_i \leq M q_i,\ \forall i \in [n], \tag{23}$$
$$w_i \leq y_i,\ \forall i \in [n], \tag{24}$$
$$w_i \geq 0.\ \forall i \in [n]. \tag{25}$$

Therefore, the constraint $\frac{1}{k} \cdot \sum_{i=1}^{n} q_i \max\{\delta(x, x_i), d_k(x_i)\} \leq kt \sum_{i=1}^{n} q_i \frac{1}{\delta_i}$ can be re-written as

$$\sum_{i=1}^{n} w_i \leq k^2 t \sum_{i=1}^{n} q_i \frac{1}{\delta_i}. \tag{26}$$

Finally, the distance $\delta(x, x_i)$ can be incorporated using standard techniques depending on the value of p. For $p = 2$, which is the value of p that we use in the experiments, we obtain the following MIP.

$$\begin{aligned}
\min_{\substack{x^0 \in \mathcal{X}, x^1, \ldots, x^L, z^1, \ldots, z^L, \\ q_{1,1}, \ldots, q_{n,k}, q_1, \ldots, q_n, \\ y_1, \ldots, y_n, w_1, \ldots, w_n}} \quad & ||x_F - x^0||_2 + M \cdot \textstyle\sum_{i=1}^{n} ||x_i - x^0||_2 \\
\text{s.t.} \quad & x^i \geq W^i x^{i-1} + b^i,\ \forall i \in [L], \\
& x^i \geq 0,\ \forall i \in [L], \\
& x^i \leq W^i x^{i-1} + b^i - l^i \odot (1 - z^i),\ \forall i \in [L], \\
& x^i \leq u^i \odot z^i,\ \forall i \in [L], \\
& W^{L+1} x^L + b^{L+1} \geq 0, \\
& \textstyle\sum_{i=1}^{n} q_{i,j} = 1,\ \forall j \in [k], \\
& \textstyle\sum_{j=1}^{k} q_{i,j} \leq 1,\ \forall i \in [n], \\
& ||x_l - x^0||_2 \geq ||x_i - x^0||_2 - M(1 - q_{i,j}) - M \textstyle\sum_{j'=j-1}^{1} (1 - q_{l,j'}), \forall i, l \in [n], \forall j \in [k], \\
& q_i = \textstyle\sum_{j=1}^{k} q_{i,k},\ \forall i \in [n], \\
& y_i \geq ||x_i - x^0||_2,\ \forall i \in [n], \\
& y_i \geq d_k(x_i),\ \forall i \in [n], \\
& y_i \leq ||x_i - x^0||_2 + M u_i,\ \forall i \in [n], \\
& y_i \leq d_k(x_i) + M(1 - u_i).\ \forall i \in [n], \\
& w_i \leq M q_i,\ \forall i \in [n], \\
& w_i \leq y_i,\ \forall i \in [n], \\
& w_i \geq 0.\ \forall i \in [n], \\
& \textstyle\sum_{i=1}^{n} w_i \leq k^2 t \sum_{i=1}^{n} q_i \frac{1}{\delta_i}, \\
& q_{i,j} \in \{0, 1\},\ \forall i \in [n], j \in [k], \\
& q_i \in \{0, 1\},\ \forall i \in [n],
\end{aligned} \tag{27}$$

□

Remark A. *In practice, the value M is an upper bound on the maximum possible distance $\delta(x, x_i)$. If we assume that $\mathcal{X} = [0, 1]^d$ (after normalization), $M = d$ for $p \in \{1, 2\}$ and $M = 1$ for $p = \infty$.*

Remark B. *For the general case of a natural k, $LOF_{k,\mathcal{D}}(x)$ requires the introduction of $n + n \cdot k$ new binary variables and $n + n = 2n$ new continuous variables.*

Remark C. *For the special of $k = 1$, $LOF_{1,\mathcal{D}}(x)$ requires the introduction of n new binary variables and n new continuous variables.*

Remark D. *In practice, for $p = 2$, the linearization of the distance is conducted automatically by Gurobi.*

Proof of Theorem 4.1

Theorem 4.1 *The probability of not selecting the live polytope that leads to the closest counterfactual is of $\mathcal{O}(e^{-m})$, i.e. drops exponentially as m increases.*

Proof. We search over $m < N$ live polytopes that contain the closest to x_F data points from $\mathcal{D}_+$. We denote that the closest point to x_F is x_1 that belongs to the live polytope $\mathcal{P}_1$, the second closest point to x_F is x_2 that belongs to the live polytope $\mathcal{P}_2$ and so on. From each live polytope, we select the closest point to x_F. Formally, we have

$$d(x_1, x_F) \leq d(x_2, x_F) \leq \cdots \leq d(x_m, x_F) \leq d(x_{m+1}, x_F) \leq \cdots \leq d(x_N, x_F). \tag{28}$$

After solving the optimization problem over live polytope P_i, the optimal counterfactual explanation is x_i'. We have that $d(x_i', x_F) = d(x_i, x_F) - X_i$ where X_i is a random variable that represents the distance between x_i and x_i'. We assume that $X_i \perp\!\!\!\perp X_j$ for every $i \neq j$. The probability of not selecting the live polytope which leads to the closest counterfactual explanation is:

$$\begin{aligned}
\mathbb{P}[\texttt{error}] &= \mathbb{P}[\min_{m+1\leq i\leq N} d(x_i', x_F) \leq \min_{1\leq i\leq m} d(x_i', x_F)] = \\
&= \mathbb{P}[\min_{m+1\leq i\leq N} \{d(x_i, x_F) - X_i\} \leq \min_{1\leq i\leq m} \{d(x_i, x_F) - X_i\}] = \\
&= \prod_{i=1}^{m} \mathbb{P}[\min_{m+1\leq j\leq N} \{d(x_j, x_F) - X_j\} \leq d(x_i, x_F) - X_i] = \\
&= \prod_{i=1}^{m} (1 - \mathbb{P}[\min_{m+1\leq j\leq N} \{d(x_j, x_F) - X_j\} \geq d(x_i, x_F) - X_i]) = \\
&= \prod_{i=1}^{m} (1 - \prod_{j=m+1}^{N} \mathbb{P}[d(x_j, x_F) - X_j \geq d(x_i, x_F) - X_i]) = \\
&= \prod_{i=1}^{m} (1 - \prod_{j=m+1}^{N} \mathbb{P}[X_j - X_i \leq d(x_j, x_F) - d(x_i, x_F)]) = \\
&= \prod_{i=1}^{m} (1 - \prod_{j=m+1}^{N} F_{j,i}(d(x_j, x_F) - d(x_i, x_F))) \leq \\
&\leq \prod_{i=1}^{m} (1 - F^{N-m}(d(x_{m+1}, x_F) - d(x_i, x_F))) \leq \\
&\leq (1 - F^{N-m}(d(x_{m+1}, x_F) - d(x_m, x_F))^m \leq \\
&\leq e^{-mF^{N-m}(d(x_{m+1}, x_F) - d(x_m, x_F))}
\end{aligned} \tag{29}$$

where $F_{j,i}(\cdot)$ is the cdf of of $X_j - X_i$ and $F(\cdot)$ is the cdf for which $F_{j,i}(d(x_{m+1}, x_F) - d(x_m, x_F))$ attains the minimum value among all $F_{i,j}$. In the last inequality, we used that $1 - x \leq e^{-x}$ for $x \geq 0$.

□

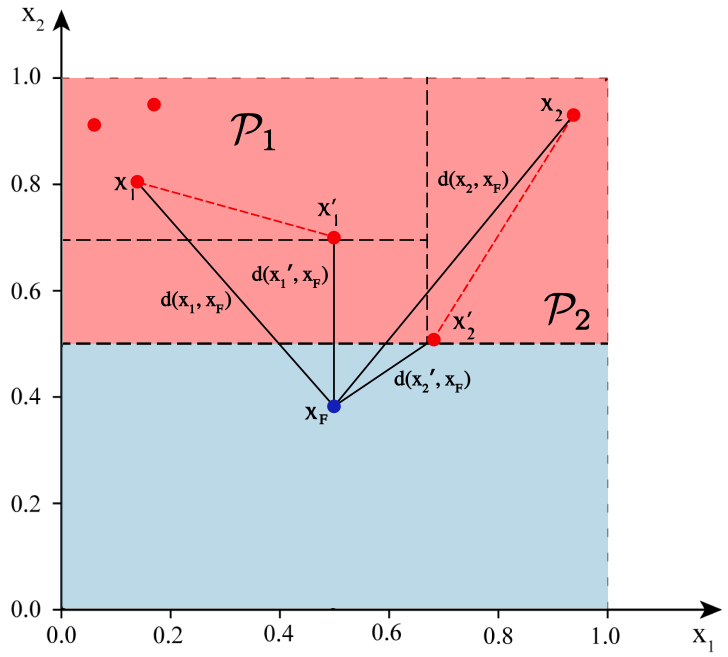

Figure 1: A case (for $m = 1$) of not selecting the live polytope that leads to the closest counterfactual explanation. $\mathcal{P}_1$ and $\mathcal{P}_2$ are the live polytopes under consideration. Initially $d(x_1, x_F) \leq d(x_2, x_F)$ and thus we search only over $\mathcal{P}_1$. However, we observe that the closest counterfactual explanation over $\mathcal{P}_2$, i.e. x'_2 is closer to x_F than the closest counterfactual explanation for $\mathcal{P}_1$, i.e. x'_1. In other words, $d(x'_1, x_F) \geq d(x'_2, x_F)$, and thus x'_2 is the most preferable counterfactual explanation.

Remark E. *It can be seen that* $\mathbb{P}[\texttt{error}]$ *drops exponentially with respect to* m*. For* $m = N$*, the upper bound (and thus,* $\mathbb{P}[\texttt{error}]$*) equals 0. This is because when* $m = N$*, we search through all live polytopes, and consequently, the probability of not selecting the live polytope that results in the closest counterfactual explanation is 0.*

Remark F. *If we additionally assume that the* X_i*'s follow a known distribution (for instance,* $X_i \sim \mathcal{U}[0, \Delta]$*, where* $\Delta > 0$ *represents an upper bound on the diameter of live polytopes, as discussed in [Serra et al., 2018]), it becomes possible to calculate the number of live polytopes required to achieve a probability of error lower than a predefined threshold* $\epsilon > 0$*. This can be accomplished by solving the following inequality for* m*:*

$$\prod_{i=1}^{m}(1 - \prod_{j=m+1}^{N} F_{i,j}(d(x_j, x_F) - d(x_i, x_F))) \leq \epsilon. \tag{30}$$

On the Sensitivity of the Number of Live Polytopes m

In this section, we study empirically the sensitivity of our proposed methods, by varying the value of m and comparing the quality of the obtained solutions across all datasets. We present the results in Table 1.

Table 1: Average proximity (ℓ_2 distance), percentage of outliers, sparsity, and generation time for Adult, FICO and German for different values of m.

	50				100				200			
`Adult`	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time
MIP-Live-m=2	$0.37_{\pm 0.17}$	35%	$4.30_{\pm 0.71}$	0.20	$\mathbf{0.45}_{\pm 0.22}$	30%	$2.15_{\pm 0.36}$	0.68	$0.53_{\pm 0.16}$	20%	$4.05_{\pm 0.74}$	1.50
MIP-Live-m=5	$\mathbf{0.36}_{\pm 0.18}$	35%	$4.45_{\pm 0.80}$	0.66	$\mathbf{0.45}_{\pm 0.22}$	25%	$2.35_{\pm 0.48}$	5.69	$\mathbf{0.48}_{\pm 0.12}$	20%	$4.10_{\pm 0.70}$	14.88
MIP-Live-m=10	$\mathbf{0.36}_{\pm 0.18}$	35%	$4.45_{\pm 0.80}$	0.84	$\mathbf{0.45}_{\pm 0.22}$	25%	$2.35_{\pm 0.48}$	8.03	$\mathbf{0.48}_{\pm 0.12}$	20%	$4.10_{\pm 0.70}$	25.85
MIP-Live-m=20	$\mathbf{0.36}_{\pm 0.18}$	35%	$4.45_{\pm 0.80}$	1.30	$\mathbf{0.45}_{\pm 0.22}$	25%	$2.35_{\pm 0.48}$	11.02	$\mathbf{0.48}_{\pm 0.12}$	20%	$4.10_{\pm 0.70}$	34.07
MIP-Live-m=2-s=2	$0.45_{\pm 0.25}$	30%	$\mathbf{2.00}_{\pm 0.00}$	0.31	$0.48_{\pm 0.25}$	25%	$\mathbf{2.00}_{\pm 0.00}$	1.01	$0.56_{\pm 0.26}$	25%	$\mathbf{2.00}_{\pm 0.00}$	2.35
MIP-Live-m=5-s=2	$0.45_{\pm 0.25}$	30%	$\mathbf{2.00}_{\pm 0.00}$	0.85	$0.48_{\pm 0.25}$	25%	$\mathbf{2.00}_{\pm 0.00}$	5.97	$0.53_{\pm 0.28}$	25%	$\mathbf{2.00}_{\pm 0.00}$	19.58
MIP-Live-m=10-s=2	$0.45_{\pm 0.25}$	30%	$\mathbf{2.00}_{\pm 0.00}$	1.28	$0.48_{\pm 0.25}$	25%	$\mathbf{2.00}_{\pm 0.00}$	10.68	$0.50_{\pm 0.23}$	25%	$\mathbf{2.00}_{\pm 0.00}$	30.78
MIP-Live-m=20-s=2	$0.45_{\pm 0.25}$	30%	$\mathbf{2.00}_{\pm 0.00}$	2.34	$0.48_{\pm 0.25}$	25%	$\mathbf{2.00}_{\pm 0.00}$	16.94	$0.50_{\pm 0.23}$	25%	$\mathbf{2.00}_{\pm 0.00}$	45.41
MIP-Live-DM-m=2	$0.44_{\pm 0.22}$	**20%**	$4.30_{\pm 0.84}$	0.16	$0.51_{\pm 0.23}$	**10%**	$4.95_{\pm 0.86}$	0.54	$0.59_{\pm 0.29}$	**10%**	$4.55_{\pm 0.67}$	1.24
MIP-Live-DM-m=5	$0.44_{\pm 0.22}$	**20%**	$4.30_{\pm 0.84}$	0.75	$0.51_{\pm 0.23}$	**10%**	$4.95_{\pm 0.86}$	6.00	$0.54_{\pm 0.26}$	**10%**	$4.45_{\pm 0.50}$	20.24
MIP-Live-DM-m=10	$0.44_{\pm 0.22}$	**20%**	$4.30_{\pm 0.84}$	0.88	$0.51_{\pm 0.23}$	**10%**	$4.95_{\pm 0.86}$	8.68	$0.54_{\pm 0.26}$	**10%**	$4.45_{\pm 0.50}$	32.49
MIP-Live-DM-m=20	$0.44_{\pm 0.22}$	**20%**	$4.30_{\pm 0.84}$	1.30	$0.51_{\pm 0.23}$	**10%**	$4.95_{\pm 0.86}$	11.46	$0.54_{\pm 0.26}$	**10%**	$4.45_{\pm 0.50}$	44.01
MIP-Live-DM-m=2-s=2	$0.54_{\pm 0.25}$	**20%**	$\mathbf{2.00}_{\pm 0.00}$	0.26	$0.54_{\pm 0.21}$	15%	$\mathbf{2.00}_{\pm 0.00}$	1.05	$0.56_{\pm 0.28}$	15%	$\mathbf{2.00}_{\pm 0.00}$	10.24
MIP-Live-DM-m=5-s=2	$0.50_{\pm 0.22}$	**20%**	$\mathbf{2.00}_{\pm 0.00}$	1.08	$0.54_{\pm 0.21}$	15%	$\mathbf{2.00}_{\pm 0.00}$	5.14	$0.56_{\pm 0.28}$	15%	$\mathbf{2.00}_{\pm 0.00}$	24.89
MIP-Live-DM-m=10-s=2	$0.50_{\pm 0.22}$	**20%**	$\mathbf{2.00}_{\pm 0.00}$	1.42	$0.54_{\pm 0.21}$	15%	$\mathbf{2.00}_{\pm 0.00}$	10.76	$0.56_{\pm 0.28}$	15%	$\mathbf{2.00}_{\pm 0.00}$	33.06
MIP-Live-DM-m=20-s=2	$0.50_{\pm 0.22}$	**20%**	$\mathbf{2.00}_{\pm 0.00}$	2.05	$0.54_{\pm 0.21}$	15%	$\mathbf{2.00}_{\pm 0.00}$	17.62	$0.56_{\pm 0.28}$	15%	$\mathbf{2.00}_{\pm 0.00}$	50.10

	50				100				200			
`FICO`	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time
MIP-Live-m=2	$0.54_{\pm 0.17}$	**0%**	$17.75_{\pm 2.43}$	0.26	$0.60_{\pm 0.28}$	**0%**	$17.95_{\pm 1.47}$	1.02	$0.61_{\pm 0.18}$	**0%**	$18.95_{\pm 1.32}$	5.81
MIP-Live-m=5	$0.52_{\pm 0.14}$	**0%**	$16.50_{\pm 3.49}$	0.48	$\mathbf{0.58}_{\pm 0.27}$	**0%**	$18.45_{\pm 1.66}$	4.25	$\mathbf{0.56}_{\pm 0.11}$	**0%**	$18.60_{\pm 1.20}$	18.86
MIP-Live-m=10	$\mathbf{0.51}_{\pm 0.13}$	**0%**	$16.45_{\pm 3.75}$	0.58	$\mathbf{0.58}_{\pm 0.27}$	**0%**	$18.45_{\pm 1.66}$	7.04	$\mathbf{0.56}_{\pm 0.11}$	**0%**	$18.60_{\pm 1.20}$	24.63
MIP-Live-m=20	$\mathbf{0.51}_{\pm 0.13}$	**0%**	$16.45_{\pm 3.75}$	0.93	$\mathbf{0.58}_{\pm 0.27}$	**0%**	$18.45_{\pm 1.66}$	12.91	$\mathbf{0.56}_{\pm 0.11}$	**0%**	$18.60_{\pm 1.20}$	30.00
MIP-Live-m=2-s=2	$0.67_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.20	$0.70_{\pm 0.25}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	5.86	$0.67_{\pm 0.20}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	14.14
MIP-Live-m=5-s=2	$0.63_{\pm 0.17}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.62	$0.69_{\pm 0.27}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	9.76	$0.63_{\pm 0.26}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	24.11
MIP-Live-m=10-s=2	$0.60_{\pm 0.18}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	2.32	$0.65_{\pm 0.13}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	10.72	$0.62_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	37.54
MIP-Live-m=20-s=2	$0.60_{\pm 0.18}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	3.47	$0.64_{\pm 0.11}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	23.99	$0.62_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	46.87
MIP-Live-DM-m=2	$0.58_{\pm 0.17}$	**0%**	$18.10_{\pm 1.58}$	0.35	$0.67_{\pm 0.38}$	**0%**	$18.10_{\pm 0.99}$	1.93	$0.70_{\pm 0.18}$	**0%**	$19.80_{\pm 0.93}$	7.18
MIP-Live-DM-m=5	$0.55_{\pm 0.12}$	**0%**	$17.90_{\pm 1.48}$	0.63	$0.62_{\pm 0.23}$	**0%**	$17.85_{\pm 1.06}$	8.10	$0.67_{\pm 0.21}$	**0%**	$19.35_{\pm 2.20}$	24.73
MIP-Live-DM-m=10	$0.54_{\pm 0.12}$	**0%**	$17.80_{\pm 1.99}$	0.72	$0.62_{\pm 0.23}$	**0%**	$17.85_{\pm 1.06}$	8.65	$0.67_{\pm 0.21}$	**0%**	$19.35_{\pm 2.20}$	34.32
MIP-Live-DM-m=20	$0.54_{\pm 0.12}$	**0%**	$17.80_{\pm 1.99}$	1.33	$0.62_{\pm 0.23}$	**0%**	$17.85_{\pm 1.06}$	15.26	$0.67_{\pm 0.21}$	**0%**	$19.35_{\pm 2.20}$	51.32
MIP-Live-DM-m=2-s=2	$0.71_{\pm 0.18}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.50	$0.74_{\pm 0.11}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	6.37	$0.75_{\pm 0.13}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	16.58
MIP-Live-DM-m=5-s=2	$0.67_{\pm 0.15}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.88	$0.74_{\pm 0.11}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	10.65	$0.73_{\pm 0.21}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	26.16
MIP-Live-DM-m=10-s=2	$0.65_{\pm 0.13}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	2.19	$0.66_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	14.91	$0.70_{\pm 0.18}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	42.38
MIP-Live-DM-m=20-s=2	$0.65_{\pm 0.13}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	3.37	$0.66_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	28.43	$0.70_{\pm 0.18}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	54.20

	50				100				200			
`German`	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time	ℓ_2	Outliers	Sparsity	Time
MIP-Live-m=2	$0.62_{\pm 0.22}$	**0%**	$3.95_{\pm 0.78}$	0.29	$0.75_{\pm 0.34}$	**0%**	$3.10_{\pm 0.83}$	1.57	$0.73_{\pm 0.31}$	**0%**	$2.70_{\pm 0.64}$	10.33
MIP-Live-m=5	$\mathbf{0.59}_{\pm 0.18}$	**0%**	$3.45_{\pm 0.67}$	0.96	$\mathbf{0.71}_{\pm 0.31}$	**0%**	$3.45_{\pm 1.24}$	10.10	$\mathbf{0.70}_{\pm 0.29}$	**0%**	$2.80_{\pm 1.17}$	18.01
MIP-Live-m=10	$\mathbf{0.59}_{\pm 0.18}$	**0%**	$3.45_{\pm 0.67}$	1.55	$\mathbf{0.71}_{\pm 0.31}$	**0%**	$3.45_{\pm 1.24}$	19.55	$\mathbf{0.70}_{\pm 0.29}$	**0%**	$2.80_{\pm 1.17}$	38.01
MIP-Live-m=20	$\mathbf{0.59}_{\pm 0.18}$	**0%**	$3.45_{\pm 0.67}$	2.86	$\mathbf{0.71}_{\pm 0.31}$	**0%**	$3.45_{\pm 1.24}$	24.11	$\mathbf{0.70}_{\pm 0.29}$	**0%**	$2.80_{\pm 1.17}$	42.26
MIP-Live-m=2-s=2	$0.67_{\pm 0.24}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	0.32	$0.82_{\pm 0.34}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	2.20	$0.85_{\pm 0.37}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	12.65
MIP-Live-m=5-s=2	$0.64_{\pm 0.22}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.25	$0.79_{\pm 0.29}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	12.25	$0.80_{\pm 0.32}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	22.65
MIP-Live-m=10-s=2	$0.63_{\pm 0.20}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.84	$0.78_{\pm 0.30}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	20.29	$0.79_{\pm 0.31}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	43.68
MIP-Live-m=20-s=2	$0.63_{\pm 0.20}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	3.00	$0.78_{\pm 0.30}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	26.43	$0.79_{\pm 0.31}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	50.97
MIP-Live-DM-m=2	$0.65_{\pm 0.26}$	**0%**	$5.90_{\pm 0.70}$	0.47	$0.80_{\pm 0.29}$	**0%**	$5.80_{\pm 1.47}$	1.94	$0.80_{\pm 0.29}$	**0%**	$5.35_{\pm 0.73}$	14.55
MIP-Live-DM-m=5	$0.62_{\pm 0.24}$	**0%**	$5.00_{\pm 1.10}$	1.12	$0.77_{\pm 0.28}$	**0%**	$5.10_{\pm 0.94}$	16.96	$0.78_{\pm 0.28}$	**0%**	$5.05_{\pm 1.83}$	24.90
MIP-Live-DM-m=10	$0.62_{\pm 0.24}$	**0%**	$5.00_{\pm 1.10}$	2.59	$0.77_{\pm 0.28}$	**0%**	$5.10_{\pm 0.94}$	27.48	$0.78_{\pm 0.28}$	**0%**	$5.05_{\pm 1.83}$	46.90
MIP-Live-DM-m=20	$0.62_{\pm 0.24}$	**0%**	$5.00_{\pm 1.10}$	3.26	$0.77_{\pm 0.28}$	**0%**	$5.10_{\pm 0.94}$	35.99	$0.78_{\pm 0.28}$	**0%**	$5.05_{\pm 1.83}$	49.66
MIP-Live-DM-m=2-s=2	$0.75_{\pm 0.30}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	0.48	$0.88_{\pm 0.39}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	3.10	$0.87_{\pm 0.38}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	17.29
MIP-Live-DM-m=5-s=2	$0.72_{\pm 0.27}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	1.81	$0.83_{\pm 0.35}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	17.81	$0.84_{\pm 0.35}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	29.32
MIP-Live-DM-m=10-s=2	$0.71_{\pm 0.28}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	3.10	$0.81_{\pm 0.34}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	30.88	$0.84_{\pm 0.35}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	50.70
MIP-Live-DM-m=20-s=2	$0.71_{\pm 0.28}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	4.03	$0.81_{\pm 0.34}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	41.58	$0.84_{\pm 0.35}$	**0%**	$\mathbf{2.00}_{\pm 0.00}$	59.56

We observe that across all datasets the performance with respect to proximity (ℓ_2 distance) of our methods plateaus for $m \geq 5$, thereby empirically verifying Theorem 4.1 and justifying our choice of using $m = 5$.

References

[Serra et al., 2018] Serra, T., Tjandraatmadja, C., and Ramalingam, S. (2018). Bounding and counting linear regions of deep neural networks. In *International Conference on Machine Learning*, pages 4558–4566. PMLR.