

---

# Model Reconstruction Using Counterfactual Explanations: A Perspective From Polytope Theory

---

**Pasan Dissanayake**  
University of Maryland  
College Park, MD  
pasand@umd.edu

**Sanghamitra Dutta**  
University of Maryland  
College Park, MD  
sanghamd@umd.edu

## Abstract

Counterfactual explanations provide ways of achieving a favorable model outcome with minimum input perturbation. However, counterfactual explanations can also be leveraged to reconstruct the model by strategically training a surrogate model to give similar predictions as the original (target) model. In this work, we analyze how model reconstruction using counterfactuals can be improved by further leveraging the fact that the counterfactuals also lie quite close to the decision boundary. Our main contribution is to derive novel theoretical relationships between the error in model reconstruction and the number of counterfactual queries required using polytope theory. Our theoretical analysis leads us to propose a strategy for model reconstruction that we call Counterfactual Clamping Attack (CCA) which trains a surrogate model using a unique loss function that treats counterfactuals differently than ordinary instances. Our approach also alleviates the related problem of *decision boundary shift* that arises in existing model reconstruction approaches when counterfactuals are treated as ordinary instances. Experimental results demonstrate that our strategy improves fidelity between the target and surrogate model predictions on several datasets.

## 1 Introduction

Counterfactual explanations (also called *counterfactuals*) have emerged as a burgeoning area of research [Wachter et al., 2017, Guidotti, 2022, Barocas et al., 2020, Verma et al., 2022, Karimi et al., 2022] for providing guidance on how to obtain a more favorable outcome from a machine learning model, e.g., increase your income by 10K to qualify for the loan. Interestingly, counterfactuals can also reveal information about the underlying model, posing a nuanced interplay between model privacy and explainability [Aïvodji et al., 2020, Wang et al., 2022]. Our work provides the first theoretical analysis between the error in model reconstruction using counterfactuals and the number of counterfactuals queried for, through the lens of polytope theory.

Model reconstruction using counterfactuals can have serious implications in Machine Learning as a Service (MLaaS) platforms that allow users to query a model for a specified cost [Gong et al., 2020]. An adversary may be able to “steal” the model by querying for counterfactuals and training a surrogate model to provide similar predictions as the target model, a practice also referred to as *model extraction*. On the other hand, model reconstruction could also be beneficial for *preserving applicant privacy*, e.g., if an applicant wishes to evaluate their chances of acceptance from crowdsourced information before formally sharing their own application information with an institution, often due to resource constraints or having a limited number of attempts to apply (e.g., applying for credit cards reduces the credit score [Capital One, 2024]). Our goal is to formalize *how faithfully can one reconstruct an underlying model given a set of counterfactual queries*.

An existing approach for model reconstruction is to treat counterfactuals as ordinary labeled points and use them for training a surrogate model [Aïvodji et al., 2020]. While this may work for a well-balanced counterfactual queries from the two classes lying roughly equidistant to the decision boundary, it is not the same for unbalanced datasets. The surrogate decision boundary might not always overlap with that of the target model (see Fig. 1), a problem also referred to as a *decision boundary shift*. This is due to the learning process where the boundary is kept far from the training examples (margin) for better generalization [Shokri et al., 2021].

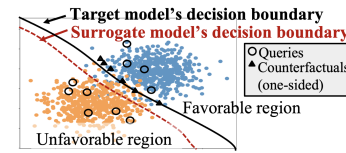


Figure 1: Decision boundary shift when counterfactuals are treated as ordinary labeled points.

The decision boundary shift is aggravated when the system provides only *one-sided counterfactuals*, i.e., counterfactuals only for queries with unfavorable predictions. If one were allowed to query for two-sided counterfactuals, the decision boundary shift may be tackled by querying for the counterfactual of the counterfactual [Wang et al., 2022]. However, such strategies cannot be applied when only one-sided counterfactuals are available, which is more common and also a more challenging use case for model reconstruction, e.g., counterfactuals are only available for the rejected applicants to get accepted for a loan but not the other way.

In this work, we analyze how model reconstruction using counterfactuals can be improved by leveraging the fact that the counterfactuals are quite close to the decision boundary. We provide novel theoretical analysis for model reconstruction using polytope theory, addressing an important knowledge gap in the existing literature. We demonstrate reconstruction strategies that alleviate the decision-boundary-shift issue for one-sided counterfactuals. In contrast to existing strategies Aïvodji et al. [2020] and Wang et al. [2022] which require the system to provide counterfactuals for queries from both sides of the decision boundary, we are able to reconstruct using only one-sided counterfactuals, a problem that we also demonstrate to be theoretically more challenging than the two-sided case (see Corollary 3.8). In summary, our contributions can be listed as follows:

**Fundamental guarantees on model reconstruction using counterfactuals:** We derive novel theoretical relationships between the error in model reconstruction and the number of counterfactual queries (query complexity) under three settings: (i) Convex decision boundaries and closest counterfactuals (Theorem 3.2): We rely on convex polytope approximations to derive an *exact* relationship between expected model reconstruction error and query complexity; (ii) ReLU networks and closest counterfactuals (Theorem 3.6): Relaxing the convexity assumption, we provide *probabilistic* guarantees on the success of model reconstruction as a function of number of counterfactual queries; and (iii) Beyond closest counterfactuals: We provide approximate guarantees for a broader class of models, including ReLU networks and locally-Lipschitz continuous models (Theorem 3.10).

**Model reconstruction strategy with unique loss function:** We devise a reconstruction strategy – that we call Counterfactual Clamping Attack (CCA) – that exploits only the fact that the counterfactuals lie reasonably close to the decision boundary, but need not be exactly the closest.

**Empirical validation:** We conduct experiments on both synthetic datasets, as well as, four real-world datasets, namely, Adult Income [Becker and Kohavi, 1996], COMPAS [Angwin et al., 2016], DCCC [Yeh, 2016], and HELOC [FICO, 2018]. Our strategy outperforms the existing baseline [Aïvodji et al., 2020] over all these datasets (Section 4) using one-sided counterfactuals, i.e., counterfactuals only for queries from the unfavorable side of the decision boundary. We also include additional experiments to observe the effects of model architecture, Lipschitzness, and other types of counterfactual generation methods as well as ablation studies with other loss functions. A python-based implementation is available at: <https://github.com/pasandissanayake/model-reconstruction-using-counterfactuals>.

**Related Works:** A plethora of counterfactual-generating mechanisms has been suggested in existing literature [Guidotti, 2022, Barocas et al., 2020, Verma et al., 2022, Karimi et al., 2022, 2020, Mothilal et al., 2020, Dhurandhar et al., 2018, Deuch and Frost, 2019, Mishra et al., 2021]. Related works that focus on leaking information about the dataset from counterfactual explanations include membership inference attacks [Pawelczyk et al., 2023] and explanation-linkage attacks [Goethals et al., 2023]. Shokri et al. [2021] examine membership inference from other types of explanations, e.g., feature-based. Model reconstruction (without counterfactuals) has been the topic of a wide array of studies (see surveys Gong et al. [2020] and Oliynyk et al. [2023]). Various mechanisms such as equation solving [Tramèr et al., 2016] and active learning have been considered [Pal et al., 2020].

Model inversion [Gong et al., 2021, Struppek et al., 2022, Zhao et al., 2021] is another form of extracting information about a black box model, under limited access to the model aspects. In contrast to model extraction where the goal is to replicate the model itself, in model inversion an adversary tries to extract the representative attributes of a certain class with respect to the target model. In this regard, Zhao et al. [2021] focuses on exploiting explanations for image classifiers such as saliency maps to improve model inversion attacks. Struppek et al. [2022] proposes various methods based on Generative Adversarial Networks to make model inversion attacks robust (for instance, to distributional shifts) in the domain of image classification.

Milli et al. [2019] looks into model reconstruction using other types of explanations, e.g., gradients. Yadav et al. [2023] explore algorithmic auditing using counterfactual explanations, focusing on linear classifiers and decision trees. Using counterfactual explanations for model reconstruction has received limited attention, with the notable exceptions of Aïvodji et al. [2020] and Wang et al. [2022]. Aïvodji et al. [2020] suggest using counterfactuals as ordinary labeled examples while training the surrogate model, leading to decision boundary shifts, particularly for unbalanced query datasets (one-sided counterfactuals). Wang et al. [2022] introduces a strategy of mitigating this issue by further querying for the counterfactual of the counterfactual. However, both these methods require the system to provide counterfactuals for queries from both sides of the decision boundary. Nevertheless, a user with a favorable decision may not usually require a counterfactual explanation, and hence a system providing one-sided counterfactuals might be more common, wherein lies our significance. While model reconstruction (without counterfactuals) has received interest from a theoretical perspective [Tramèr et al., 2016, Papernot et al., 2017, Milli et al., 2019], model reconstruction involving counterfactual explanations lack such a theoretical understanding. Our work theoretically analyzes model reconstruction using polytope theory and proposes novel strategies thereof, also addressing the decision-boundary shift issue.

## 2 Preliminaries

**Notations:** We consider binary classification models  $m$  that take an input value  $\mathbf{x} \in \mathbb{R}^d$  and output a probability between 0 and 1. The final predicted class is denoted by  $\lfloor m(\mathbf{x}) \rfloor \in \{0, 1\}$  which is obtained by thresholding the output probability at 0.5 as follows:  $\lfloor m(\mathbf{x}) \rfloor = \mathbb{1}[m(\mathbf{x}) \geq 0.5]$  where  $\mathbb{1}[\cdot]$  denotes the indicator function. Throughout the paper, we denote the output probability by  $m(\mathbf{x})$  and the corresponding thresholded output by  $\lfloor m(\mathbf{x}) \rfloor$ . Consequently, the decision boundary of the model  $m$  is the  $(d - 1)$ -dimensional hypersurface (generalization of surface in higher dimensions; see Definition 2.5) in the input space, given by  $\partial\mathbb{M} = \{\mathbf{x} : m(\mathbf{x}) = 0.5\}$ . We call the region where  $\lfloor m(\mathbf{x}) \rfloor = 1$  as the *favorable region* and the region where  $\lfloor m(\mathbf{x}) \rfloor = 0$  as the *unfavorable region*. We always state the convexity/concavity of the decision boundary with respect to the favorable region (i.e., the decision boundary is convex if the set  $\mathbb{M} = \{\mathbf{x} \in \mathbb{R}^d : \lfloor m(\mathbf{x}) \rfloor = 1\}$  is convex). We assume that upon knowing the range of values for each feature, the  $d$ -dimensional input space can be normalized so that the inputs lie within the set  $[0, 1]^d$  (the  $d$ -dimensional unit hypercube), as is common in literature [Liu et al., 2020, Tramèr et al., 2016, Hamman et al., 2023, Black et al., 2022]. We let  $g_m$  denote the counterfactual generating mechanism corresponding to the model  $m$ .

**Definition 2.1** (Counterfactual Generating Mechanism). Given a cost function  $c : [0, 1]^d \times [0, 1]^d \rightarrow \mathbb{R}_0^+$  for measuring the quality of a counterfactual, and a model  $m$ , the corresponding counterfactual generating mechanism is the mapping  $g_m : [0, 1]^d \rightarrow [0, 1]^d$  specified as follows:  $g_m(\mathbf{x}) = \arg \min_{\mathbf{w} \in [0, 1]^d} c(\mathbf{x}, \mathbf{w})$ , such that  $\lfloor m(\mathbf{x}) \rfloor \neq \lfloor m(\mathbf{w}) \rfloor$ .

The cost  $c(\mathbf{x}, \mathbf{w})$  is selected based on specific desirable criteria, e.g.,  $c(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|_p$ , with  $\|\cdot\|_p$  denoting the  $L_p$ -norm. Specifically,  $p = 2$  leads to the following definition of the *closest counterfactual* [Wachter et al., 2017, Laugel et al., 2017, Mothilal et al., 2020].

**Definition 2.2** (Closest Counterfactual). When  $c(\mathbf{x}, \mathbf{w}) \equiv \|\mathbf{x} - \mathbf{w}\|_2$ , the resulting counterfactual generated using  $g_m$  as per Definition 2.1 is called the closest counterfactual.

Given a model  $m$  and a counterfactual generating method  $g_m$ , we define the inverse counterfactual region  $\mathbb{G}$  for a subset  $\mathbb{H} \subseteq [0, 1]^d$  to be the region whose counterfactuals under  $g_m$  fall in  $\mathbb{H}$ .

**Definition 2.3** (Inverse Counterfactual Region). The inverse counterfactual region  $\mathbb{G}_{m, g_m}$  of  $\mathbb{H} \subseteq [0, 1]^d$  is the region defined as:  $\mathbb{G}_{m, g_m}(\mathbb{H}) = \{\mathbf{x} \in [0, 1]^d : g_m(\mathbf{x}) \in \mathbb{H}\}$ .

**Problem Setting:** Our problem setting involves a target model  $m$  which is pre-trained and assumed to be hosted on a MLaaS platform (see Fig. 2). Any user can query it with a set of input instances  $\mathbb{D} \subseteq [0, 1]^d$  (also called counterfactual queries) and will be provided with the set of predictions, i.e.,  $\{ \lfloor m(\mathbf{x}) \rfloor : \mathbf{x} \in \mathbb{D} \}$ , and a set of *one-sided* counterfactuals for the instances whose predicted class is 0, i.e.,  $\{ g_m(\mathbf{x}) : \mathbf{x} \in \mathbb{D}, \lfloor m(\mathbf{x}) \rfloor = 0 \}$ . Note that, by the definition of a counterfactual,  $\lfloor m(g_m(\mathbf{x})) \rfloor = 1$  for all  $\mathbf{x}$  with  $\lfloor m(\mathbf{x}) \rfloor = 0$ . The **goal** of the user is to train a surrogate model to achieve a certain level of performance with as few queries as possible. In this work, we use *fidelity* as our performance metric for model reconstruction<sup>1</sup>.

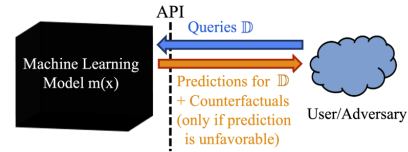


Figure 2: Problem setting

The **goal** of the user is to train a surrogate model to achieve a certain level of performance with as few queries as possible. In this work, we use *fidelity* as our performance metric for model reconstruction<sup>1</sup>.

**Definition 2.4** (Fidelity [Aïvodji et al., 2020]). With respect to a given target model  $m$  and a reference dataset  $\mathbb{D}_{\text{ref}} \subseteq [0, 1]^d$ , the fidelity of a surrogate model  $\tilde{m}$  is given by

$$\text{Fid}_{m, \mathbb{D}_{\text{ref}}}(\tilde{m}) = \frac{1}{|\mathbb{D}_{\text{ref}}|} \sum_{\mathbf{x} \in \mathbb{D}_{\text{ref}}} \mathbb{1} [\lfloor m(\mathbf{x}) \rfloor = \lfloor \tilde{m}(\mathbf{x}) \rfloor].$$

**Background on Geometry of Decision Boundaries:** Our theoretical analysis employs arguments based on the geometry of the involved models’ decision boundaries. We assume the decision boundaries are hypersurfaces. A hypersurface is a generalization of a surface into higher dimensions, e.g., a line or a curve in a 2-dimensional space, a surface in a 3-dimensional space, etc.

**Definition 2.5** (Hypersurface, Lee [2009]). A hypersurface is a  $(d - 1)$ -dimensional sub-manifold embedded in  $\mathbb{R}^d$ , which can also be denoted by a single implicit equation  $S(\mathbf{x}) = 0$  where  $\mathbf{x} \in \mathbb{R}^d$ .

We focus on the properties of hypersurfaces which are “touching” each other, as defined next.

**Definition 2.6** (Touching Hypersurfaces). Let  $S(\mathbf{x}) = 0$  and  $T(\mathbf{x}) = 0$  denote two differentiable hypersurfaces in  $\mathbb{R}^d$ .  $S(\mathbf{x}) = 0$  and  $T(\mathbf{x}) = 0$  are said to be touching each other at the point  $\mathbf{w}$  if and only if  $S(\mathbf{w}) = T(\mathbf{w}) = 0$ , and there exists a non-empty neighborhood  $\mathcal{B}_{\mathbf{w}}$  around  $\mathbf{w}$ , such that  $\forall \mathbf{x} \in \mathcal{B}_{\mathbf{w}}$  with  $S(\mathbf{x}) = 0$  and  $\mathbf{x} \neq \mathbf{w}$ , only one of  $T(\mathbf{x}) > 0$  or  $T(\mathbf{x}) < 0$  holds. (i.e., within  $\mathcal{B}_{\mathbf{w}}$ ,  $S(\mathbf{x}) = 0$  and  $T(\mathbf{x}) = 0$  lie on the same side of each other).

Next, we show that touching hypersurfaces share a common tangent hyperplane at their point of contact. This result is instrumental in exploiting the closest counterfactuals in model reconstruction (proof in Appendix A.1).

**Lemma 2.7.** Let  $S(\mathbf{x}) = 0$  and  $T(\mathbf{x}) = 0$  denote two differentiable hypersurfaces in  $\mathbb{R}^d$ , touching each other at point  $\mathbf{w}$ . Then,  $S(\mathbf{x}) = 0$  and  $T(\mathbf{x}) = 0$  have a common tangent hyperplane at  $\mathbf{w}$ .

### 3 Main Results

We provide a set of theoretical guarantees on query size of model reconstruction that progressively build on top of each other, starting from stronger guarantees in a somewhat restricted setting towards more approximate guarantees in broadly applicable settings. We discuss important intuitions that can be inferred from the results. Finally, we propose a model reconstruction strategy that is executable in practice, which is empirically evaluated in Section 4.

#### 3.1 Convex decision boundaries and closest counterfactuals

We start out with demonstrating how the closest counterfactuals provide a linear approximation of *any* decision boundary, around the corresponding counterfactual. Prior work [Yadav et al., 2023] shows that for linear models, the line joining a query instance  $\mathbf{x}$  and the closest counterfactual  $\mathbf{w} (= g_m(\mathbf{x}))$  is perpendicular to the linear decision boundary. We generalize this observation to any differentiable decision boundary, not necessarily linear.

**Lemma 3.1.** Let  $S$  denote the decision boundary of a classifier and  $\mathbf{x} \in [0, 1]^d$  be any point that is not on  $S$ . Then, the line joining  $\mathbf{x}$  and its closest counterfactual  $\mathbf{w}$  is perpendicular to  $S$  at  $\mathbf{w}$ .

<sup>1</sup>Performance can be evaluated using accuracy or fidelity [Jagielski et al., 2020]. Accuracy is a measure of how well the surrogate model can predict the true labels over the data manifold of interest. While attacks based on both measures have been proposed in literature, fidelity-based attacks have been deemed more useful as a first step in designing future attacks [Jagielski et al., 2020, Papernot et al., 2017].

The proof follows by showing that the  $d$ -dimensional ball with radius  $\|\mathbf{x} - \mathbf{w}\|_2$  touches (as in Definition 2.6)  $\mathcal{S}$  at  $\mathbf{w}$ , and invoking Lemma 2.7. For details see Appendix A.1. As a direct consequence of Lemma 3.1, a user may query the system and calculate tangent hyperplanes of the decision boundary drawn at the closest counterfactuals. This leads to a linear approximation of the decision boundary at the closest counterfactuals (see Fig. 3).

If the decision boundary is convex, such an approximation will provide a set of supporting hyperplanes. *The intersection of these supporting hyperplanes will provide a circumscribing convex polytope approximation of the decision boundary.* We show that the average fidelity of such an approximation, evaluated over uniformly distributed input instances, tends to 1 when the number of queries is large.

**Theorem 3.2.** *Let  $m$  be the target binary classifier whose decision boundary is convex (i.e., the set  $\{\mathbf{x} \in [0, 1]^d : \lfloor m(\mathbf{x}) \rfloor = 1\}$  is convex) and has a continuous second derivative. Denote by  $\tilde{M}_n$ , the convex polytope approximation of  $m$  constructed with  $n$  supporting hyperplanes obtained through i.i.d. counterfactual queries. Assume that the fidelity is evaluated with respect to  $\mathbb{D}_{\text{ref}}$  which is uniformly distributed over  $[0, 1]^d$ . Then, when  $n \rightarrow \infty$  the expected fidelity of  $\tilde{M}_n$  with respect to  $m$  is given by*

$$\mathbb{E} [\text{Fid}_{m, \mathbb{D}_{\text{ref}}}(\tilde{M}_n)] = 1 - \epsilon \quad (1)$$

where  $\epsilon \sim \mathcal{O}\left(n^{-\frac{2}{d+1}}\right)$  and the expectation is over both  $\tilde{M}_n$  and  $\mathbb{D}_{\text{ref}}$ .

Theorem 3.2 provides an exact relationship between the expected fidelity and number of queries. The proof utilizes a result from random polytope theory [Böröczky Jr and Reitzner, 2004] which provides a complexity bound on volume-approximating smooth convex sets by convex polytopes. The proof involves observing that the volume of the overlapping decision regions of  $m$  and  $\tilde{M}_n$  (for example, regions A and C in Fig. 3) translates to the expected fidelity when evaluated under a uniformly distributed  $\mathbb{D}_{\text{ref}}$ . Appendix A.2 provides the detailed steps.

**Remark 3.3 (Relaxing the Convexity Assumption).** This strategy of linear approximations can also be extended to a concave decision boundary since the closest counterfactual will always lead to a tangent hyperplane irrespective of convexity. Now the rejected region can be seen as the intersection of these half-spaces (Lemma 3.1 does not assume convexity). However, it is worth noting that approximating a concave decision boundary is, in general, more difficult than approximating a convex region. To obtain equally-spaced out tangent hyperplanes on the decision boundary, a concave region will require a much denser set of query points (see Fig. 4) due to the inverse effect of length contraction discussed in Aleksandrov [1967, Chapter III Lemma 2]. *Deriving similar theoretical guarantees for a decision boundary which is neither convex nor concave is much more challenging as the decision regions can no longer be approximated as intersections of half-spaces.* The assumption of convex decision boundaries may only be satisfied under limited scenarios such as input-convex neural networks [Amos et al., 2017]. However, we observe that this limitation can be circumvented in case of ReLU networks to arrive at a probabilistic guarantee as discussed next.

### 3.2 ReLU networks and closest counterfactuals

Rectified Linear Units (ReLU) are one of the most used activation functions in neural networks [Zeiler et al., 2013, Maas et al., 2013, Ronneberger et al., 2015, He et al., 2016]. A deep neural network that uses ReLU activations can be represented as a Continuous Piece-Wise Linear (CPWL) function [Chen et al., 2022, Hanin and Rolnick, 2019]. A CPWL function comprises of a union of linear functions over a partition of the domain. Definition 3.4 below provides a precise characterization.

**Definition 3.4** (Continuous Piece-Wise Linear (CPWL) Function [Chen et al., 2022]). A function  $\ell : \mathbb{R}^d \rightarrow \mathbb{R}$  is said to be continuous piece-wise linear if and only if

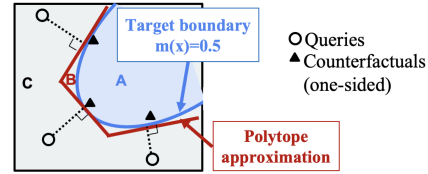


Figure 3: Polytope approximation of a convex decision boundary using the closest counterfactuals.

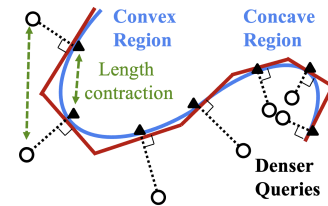


Figure 4: Approximating a concave region needs denser queries w.r.t. a convex region.

1. There exists a finite set of closed subsets of  $\mathbb{R}^d$ , denoted as  $\{\mathbb{U}_i\}_{i=1,2,\dots,q}$  such that  $\bigcup_{i=1}^q \mathbb{U}_i = \mathbb{R}^n$
2.  $\ell(\mathbf{x})$  is affine over each  $\mathbb{U}_i$  i.e., over each  $\mathbb{U}_i$ ,  $\ell(\mathbf{x}) = \mathbf{a}_i^T \mathbf{x} + b_i$  with  $\mathbf{a}_i \in \mathbb{R}^d$ ,  $b_i \in \mathbb{R}$ .

This definition can be readily applied to the models of our interest, of which the domain is the unit hypercube  $[0, 1]^d$ . A neural network with ReLU activations can be used as a classifier by appending a Sigmoid activation  $\sigma(z) = \frac{1}{1+e^{-z}}$  to the final output. We denote such a classifier by  $m(\mathbf{x}) = \sigma(\ell(\mathbf{x}))$  where  $\ell(\mathbf{x})$  is CPWL. It has been observed that the number of linear pieces  $q$  of a trained ReLU network is generally way below the theoretically allowed maximum [Hanin and Rolnick, 2019].

We first show that the decision boundaries of such CPWL functions are collections of polytopes (not necessarily convex). The proof of Lemma 3.5 is deferred to Appendix A.3.

**Lemma 3.5.** *Let  $m(\mathbf{x}) = \sigma(\ell(\mathbf{x}))$  be a ReLU classifier, where  $\ell(\mathbf{x})$  is CPWL and  $\sigma(\cdot)$  is the Sigmoid function. Then, the decision boundary  $\partial\mathbb{M} = \{\mathbf{x} \in [0, 1]^d : m(\mathbf{x}) = 0.5\}$  is a collection of (possibly non-convex) polytopes in  $[0, 1]^d$ , when considered along with the boundaries of the unit hypercube.*

Next we will analyze the probability of successful model reconstruction using counterfactuals. Consider a uniform grid  $\mathcal{N}_\epsilon$  over the unit hypercube  $[0, 1]^d$ , where each cell is a smaller hypercube with side length  $\epsilon$  (see Fig. 5). For this analysis, we make an assumption: *If a cell contains a part of the decision boundary, then that part is completely linear (affine) within that small cell*<sup>2</sup>.

Now, as the decision boundary becomes linear for each small cell that it passes through, having just one closest counterfactual in each such cell is sufficient to get the decision boundary in that cell (recall Lemma 3.1). We formalize this intuition in Theorem 3.6 to obtain a probabilistic guarantee on the success of model reconstruction. A proof is presented in Appendix A.4.

**Theorem 3.6.** *Let  $m$  be a target binary classifier with ReLU activations. Let  $k(\epsilon)$  be the number of cells through which the decision boundary passes. Define  $\{\mathbb{H}_i\}_{i=1,\dots,k(\epsilon)}$  to be the collection of affine pieces of the decision boundary within each decision boundary cell where each  $\mathbb{H}_i$  is an open set. Let  $v_i(\epsilon) = V(\mathbb{G}_{m,g_m}(\mathbb{H}_i))$  where  $V(\cdot)$  is the  $d$ -dimensional volume (i.e., the Lebesgue measure) and  $\mathbb{G}_{m,g_m}(\cdot)$  is the inverse counterfactual region w.r.t.  $m$  and the closest counterfactual generator  $g_m$ . Then the probability of successful reconstruction with counterfactual queries distributed uniformly over  $[0, 1]^d$  is lower-bounded as*

$$\mathbb{P}[\text{Reconstruction}] \geq 1 - k(\epsilon)(1 - v^*(\epsilon))^n \quad (2)$$

where  $v^*(\epsilon) = \min_{i=1,\dots,k(\epsilon)} v_i(\epsilon)$  and  $n$  is the number of queries.

**Remark 3.7.** Here  $k(\epsilon)$  and  $v^*(\epsilon)$  depend only on the nature of the model being reconstructed and are independent of the number of queries  $n$ . The value of  $k(\epsilon)$  roughly grows with the surface area of the decision boundary (e.g., length when input is 2D), showing that models with more convoluted decision boundaries might need more queries for reconstruction. Generally,  $k(\epsilon)$  lies within the interval  $\frac{A(\partial\mathbb{M})}{\sqrt{2}\epsilon^{d-1}} \leq k(\epsilon) \leq \frac{1}{\epsilon^d}$  where  $A(\cdot)$  denotes the surface area in  $d$ -dimensional space. The lower bound is due to the fact that the area of any slice of the unit hypercube being at-most  $\sqrt{2}$  [Ball, 1986]. Upper bound is reached when the decision boundary traverses through all the cells in the grid which is less likely in practice. When the model complexity increases, we get a larger  $k(\epsilon)$  as well as a smaller  $v^*(\epsilon)$ , requiring a higher number of queries to achieve similar probabilities of success.

**Corollary 3.8 (Linear Models).** *For linear models with one-sided counterfactuals,  $\mathbb{P}[\text{Reconstruction}] = 1 - (1 - v)^n$  where  $v$  is the volume of the unfavorable region. However, with two-sided counterfactuals,  $\mathbb{P}[\text{Reconstruction}] = 1$  with just one single query.*

<sup>2</sup>This is violated only for the cells containing parts of the edges of the decision boundary. However, we may assume that  $\epsilon$  is small enough so that the total number of such cells is negligible compared to the total cells containing the decision boundary. Generally,  $\epsilon$  can be made sufficiently small such that for most of the cells, if a cell contains a part of the decision boundary then that part is affine within the cell.

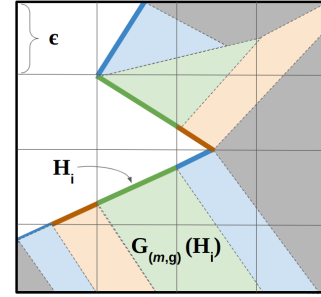


Figure 5:  $\mathcal{N}_\epsilon$  grid and inverse counterfactual regions. Thick solid lines indicate the decision boundary pieces ( $\mathbb{H}_i$ 's). White color depicts the accepted region. Pale-colored are the inverse counterfactual regions of the  $\mathbb{H}_i$ 's with the matching color. In this case  $k(\epsilon) = 7$  and  $v^*(\epsilon)$  is the area of lower amber region.



This result mathematically demonstrates that allowing counterfactuals from both accepted and rejected regions (as in Aivodji et al. [2020], Wang et al. [2022]) is easier for model reconstruction, when compared to the one-sided case. It effectively increases each  $v_i(\epsilon)$  (volume of the inverse counterfactual region). As everything else remains unaffected, for a given  $n$ ,  $\mathbb{P}[\text{Reconstruction}]$  is higher when counterfactuals from both regions are available. For a linear model, this translates to a guaranteed reconstruction with a single query since  $v = 1$ .

However, we note that all of the aforementioned analysis relies on the closest counterfactual which can be challenging to generate. Practical counterfactual generating mechanisms usually provide counterfactuals that are reasonably close but may not be exactly the closest. This motivates us to now propose a more general strategy assuming local Lipschitz continuity of the models involved.

### 3.3 Beyond closest counterfactuals

Lipschitz continuity, a property that is often encountered in related works [Bartlett et al., 2017, Gouk et al., 2021, Pauli et al., 2021, Hamman et al., 2023, 2024, Liu et al., 2020, Marques-Silva et al., 2021], demands the model output does not change too fast. Usually, a smaller Lipschitz constant is indicative of a higher generalizability of a model [Gouk et al., 2021, Pauli et al., 2021].

**Definition 3.9** (Local Lipschitz Continuity). A model  $m$  is said to be locally Lipschitz continuous if for every  $\mathbf{x}_1 \in [0, 1]^d$  there exists a neighborhood  $\mathbb{B}_{\mathbf{x}_1} \subseteq [0, 1]^d$  around  $\mathbf{x}_1$  such that for all  $\mathbf{x}_2 \in \mathbb{B}_{\mathbf{x}_1}$ ,  $|m(\mathbf{x}_1) - m(\mathbf{x}_2)| \leq \gamma \|\mathbf{x}_1 - \mathbf{x}_2\|_2$  for some  $\gamma \in \mathbb{R}_0^+$ .

For analyzing model reconstruction under local-Lipschitz assumptions, we consider the difference of the model output probabilities (before thresholding) as a measure of similarity between the target and surrogate models because it would force the decision boundaries to overlap. The proposed strategy is motivated from the following observation: the difference of two models' output probabilities corresponding to a given input instance  $\mathbf{x}$  can be bounded by having another point with matching outputs in the affinity of the instance  $\mathbf{x}$ . This observation is formally stated in Theorem 3.10. See Appendix A.5 for a proof.

**Theorem 3.10.** Let the target  $m$  and surrogate  $\tilde{m}$  be ReLU classifiers such that  $m(\mathbf{w}) = \tilde{m}(\mathbf{w})$  for every counterfactual  $\mathbf{w}$ . For any point  $\mathbf{x}$  that lies in a decision boundary cell,  $|\tilde{m}(\mathbf{x}) - m(\mathbf{x})| \leq \sqrt{d}(\gamma_m + \gamma_{\tilde{m}})\epsilon$  holds with probability  $p \geq 1 - k(\epsilon)(1 - v^*(\epsilon))^n$ .

Note that within each decision boundary cell, models are affine and hence locally Lipschitz for some  $\gamma_m, \gamma_{\tilde{m}} \in \mathbb{R}_0^+$ . Local Lipschitz property assures that the approximation is quite close ( $\gamma_m, \gamma_{\tilde{m}}$  are small) except over a few small ill-behaved regions of the decision boundary. This result can be extended to any locally Lipschitz pair of models as stated in following corollary.

**Corollary 3.11.** Suppose the target  $m$  and surrogate  $\tilde{m}$  are locally Lipschitz (not necessarily ReLU) such that  $m(\mathbf{w}) = \tilde{m}(\mathbf{w})$  for every counterfactual  $\mathbf{w}$ . Assume the counterfactuals are well-spaced out and forms a  $\delta$ -cover over the decision boundary. Then  $|\tilde{m}(\mathbf{x}) - m(\mathbf{x})| \leq (\gamma_m + \gamma_{\tilde{m}})\delta$ , over the target decision boundary.

Theorem 3.10 provides the motivation for a novel model reconstruction strategy. Let  $\mathbf{w}$  be a counterfactual. Recall that  $\partial\mathbb{M}$  denotes the decision boundary of  $m$ . As implied by the theorem, for any  $\mathbf{x} \in \partial\mathbb{M}$ , the deviation of the surrogate model output from the target model output can be bounded above by  $\sqrt{d}(\gamma_m + \gamma_{\tilde{m}})\epsilon$  given that all the counterfactuals satisfy  $m(\mathbf{w}) = \tilde{m}(\mathbf{w})$ . Knowing that  $m(\mathbf{w}) = 0.5$ , we may design a loss function which **clamps**  $\tilde{m}(\mathbf{w})$  to be 0.5. Consequently, with a sufficient number of well-spaced counterfactuals to cover  $\partial\mathbb{M}$ , we may achieve arbitrarily small  $|\tilde{m}(\mathbf{x}) - m(\mathbf{x})|$  at the decision boundary of  $m$  (Fig. 6).

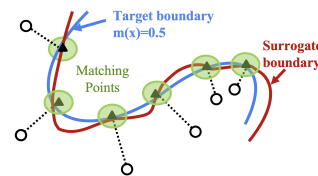


Figure 6: Rationale for Counterfactual Clamping Strategy.

However, we note that simply forcing  $\tilde{m}(\mathbf{w})$  to be exactly equal to 0.5 is quite unstable since in practice the target model's output  $m(\mathbf{w})$  is known to be close to 0.5, while being greater but may not be exactly equal. Thus, we propose a unique loss function for training the surrogate neural networks that does not penalize the counterfactuals that are already inside the favorable region of the surrogate model. For  $0 < k \leq 1$ , we define the Counterfactual Clamping loss function as

$$L_k(\tilde{m}(\mathbf{x}), y_{\mathbf{x}}) = \mathbb{1}[y_{\mathbf{x}} = 0.5, \tilde{m}(\mathbf{x}) \leq k] \{L(\tilde{m}(\mathbf{x}), k) - h(k)\} + \mathbb{1}[y_{\mathbf{x}} \neq 0.5] L(\tilde{m}(\mathbf{x}), y_{\mathbf{x}}). \quad (3)$$

Here,  $y_x$  denotes the label assigned to the input instance  $x$  by the target model, received from the API.  $L(\hat{y}, y)$  is the binary cross-entropy loss and  $h(\cdot)$  denotes the binary entropy function. We assume that the counterfactuals are distinguishable from the ordinary instances, and assign them a label of  $y_x = 0.5$ . The first term accounts for the counterfactuals, where they are assigned a non-zero loss if the surrogate model's prediction is below  $k$ . This term ensures  $\tilde{m}(w) \gtrsim k$  for the counterfactuals  $w$  while not penalizing the ones that lie farther inside the favorable region. The second term is the ordinary binary cross-entropy loss, which becomes non-zero only for ordinary query instances. Note that substituting  $k = 1$  in  $L_k(\tilde{m}(x), y_x)$  yields the ordinary binary cross entropy loss. Algorithm 1 summarizes the proposed strategy.

It is noteworthy that this approach is different from the broad area of soft-label learning Nguyen et al. [2011a,b] in two major aspects: (i) the labels in our problem do not smoothly span the interval  $[0, 1]$  – instead they are either 0, 1 or 0.5; (ii) labels of counterfactuals do not indicate a class probability; even though a label of 0.5 is assigned, there can be counterfactuals that are well within the surrogate decision boundary which should not be penalized. Nonetheless, we also perform ablation studies where we compare the performance of our proposed loss function with another potential loss which simply forces  $\tilde{m}(w)$  to be exactly 0.5, inspired from soft-label learning (see Appendix B.2.10 for results).

---

#### Algorithm 1 Counterfactual Clamping

---

**Require:** Attack dataset  $\mathbb{D}_{\text{attack}}$ ,  $k$  ( $k \in (0, 1]$ , usually 0.5), API for querying  
**Ensure:** Trained surrogate model  $\tilde{m}$

- 1: Initialize  $\mathbb{A} = \{\}$
- 2: **for**  $x \in \mathbb{D}_{\text{attack}}$  **do**
- 3:   Query API with  $x$  to get  $y_x$  ( $y_x \in \{0, 1\}$ )
- 4:    $\mathbb{A} \leftarrow \mathbb{A} \cup \{(x, y_x)\}$
- 5:   **if**  $y_x = 0$  **then**
- 6:     Query API for counterfactual  $w$  of  $x$
- 7:      $\mathbb{A} \leftarrow \mathbb{A} \cup \{(w, 0.5)\}$  {Assign  $w$  a label of 0.5}
- 8:   **end if**
- 9: **end for**
- 10: Train  $\tilde{m}$  on  $\mathbb{A}$  with  $L_k(\tilde{m}(x), y_x)$  as the loss
- 11: **return**  $\tilde{m}$

---

Counterfactual Clamping overcomes two challenges beset in existing works; (i) the problem of decision boundary shift (particularly with one-sided counterfactuals) present in the method suggested by Aïvodji et al. [2020] and (ii) the need for counterfactuals from both sides of the decision boundary in the methods of Aïvodji et al. [2020] and Wang et al. [2022].

## 4 Experiments

We carry out a number of experiments to study the performance of our proposed strategy Counterfactual Clamping. We include some results here and provide further details in Appendix B.

All the classifiers are neural networks unless specified otherwise and their decision boundaries are *not necessarily convex*. The performance of our strategy is compared with the existing attack presented in Aïvodji et al. [2020] that we refer to as “Baseline”, for the case of one-sided counterfactuals. As the initial counterfactual generating method, we use an implementation of the Minimum Cost Counterfactuals (MCCF) by Wachter et al. [2017].

**Performance metrics:** Fidelity is used for evaluating the agreement between the target and surrogate models. It is evaluated over both uniformly generated instances (denoted by  $\mathbb{D}_{\text{uni}}$ ) and test data instances from the data manifold (denoted by  $\mathbb{D}_{\text{test}}$ ) as the reference dataset  $\mathbb{D}_{\text{ref}}$ .

A summary of the experiments is provided below with additional details in Appendix B.

**(i) Visualizing the attack using synthetic data:** First, the effect of the proposed loss function in mitigating the decision boundary shift is observed over a 2-D synthetic dataset. Fig. 7 presents the results. In the figure, it is clearly visible that the Baseline model is affected by a decision boundary shift. In contrast, the CCA model's decision boundary closely approximates the target decision boundary. See Appendix B.2.1 for more details.

**(ii) Comparing performance over four real-world dataset:** We use four publicly available real-world datasets namely, Adult Income, COMPAS, DCCC, and HELOC (see Appendix B.1) for our experiments. Table 1 provides some of the results over four real-world datasets. We refer to Appendix B.2.2 (specifically Fig. 11) for additional results. In all cases, we observe that CCA has either better or similar fidelity as compared to Baseline.



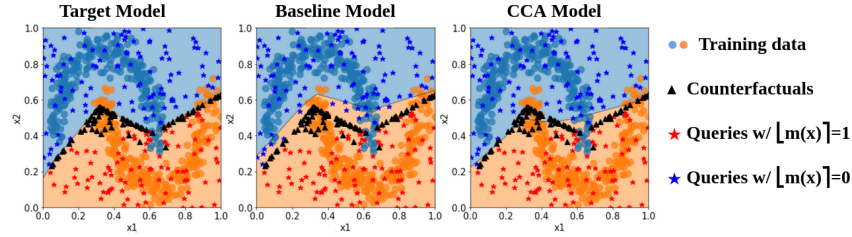


Figure 7: A 2-D demonstration of the proposed strategy. Orange and blue shades denote the favorable and unfavorable decision regions of each model. Circles denote the target model’s training data.

Table 1: Average fidelity achieved with 400 queries on the real-world datasets over an ensemble of size 100. Target model has hidden layers with neurons (20,10). Model 0 is similar to the target model in architecture. Model 1 has hidden layers with neurons (20, 10, 5).

| Dataset   | Architecture known (model 0) |        |                           |        | Architecture unknown (model 1) |        |                           |         |
|-----------|------------------------------|--------|---------------------------|--------|--------------------------------|--------|---------------------------|---------|
|           | $\mathbb{D}_{\text{test}}$   |        | $\mathbb{D}_{\text{uni}}$ |        | $\mathbb{D}_{\text{test}}$     |        | $\mathbb{D}_{\text{uni}}$ |         |
|           | Base.                        | CCA    | Base.                     | CCA    | Base.                          | CCA    | Base.                     | CCA     |
| Adult In. | 91±3.2                       | 94±3.2 | 84±3.2                    | 91±3.2 | 91±4.5                         | 94±3.2 | 84±3.2                    | 90±3.2  |
| COMPAS    | 92±3.2                       | 96±2.0 | 94±1.7                    | 96±2.0 | 91±8.9                         | 96±3.2 | 94±2.0                    | 94±8.9  |
| DCCC      | 89±8.9                       | 99±0.9 | 95±2.2                    | 96±1.4 | 90±7.7                         | 97±4.5 | 95±2.2                    | 95±11.8 |
| HELOC     | 91±4.7                       | 96±2.2 | 92±2.8                    | 94±2.4 | 90±7.4                         | 95±5.5 | 91±3.3                    | 93±3.2  |

**(iii) Studying effects of Lipschitz constants:** We study the connection between the target model’s Lipschitz constant and the CCA performance. Target model’s Lipschitz constant is controlled by changing the  $L_2$ –regularization coefficient, while keeping the surrogate models fixed. Results are presented in Fig. 15. Target models with a smaller Lipschitz constant are easier to extract. More details are provided in Appendix B.2.4.

**(iv) Studying different model architectures:** We also consider different surrogate model architectures spanning models that are more complex than the target model to much simpler ones. Results show that when sufficiently close to the target model in complexity, the surrogate architecture plays a little role on the performance. See Appendix B.2.5 for details. Furthermore, two situations are considered where the target model is not a neural network in Fig. 16 and Appendix B.2.8. In both scenarios, CCA surpasses the baseline.

**(v) Studying other counterfactual generating methods:** Effects of counterfactuals being sparse, actionable, realistic, and robust are observed. Sparse counterfactuals are generated by using  $L_1$ –norm as the cost function. Actionable counterfactuals are generated using DiCE [Mothilal et al., 2020] by defining a set of immutable features. Realistic counterfactuals (that lie on the data manifold) are generated by retrieving the 1-Nearest-Neighbor from the accepted side for a given query, as well as using the autoencoder-based method C-CHVAE [Pawelczyk et al., 2020]. Additionally, we generate robust counterfactuals using ROAR [Upadhyay et al., 2021]. We evaluate the attack performance on the HELOC dataset (Table 2). Moreover we observe the distribution of the counterfactuals generated using each method w.r.t. the target model’s decision boundary using histograms (Fig. 8). Additional details are given in Appendix B.2.6.

**(vi) Comparison with DualCFX:** DualCFX proposed by Wang et al. [2022] is a strategy that utilizes the counterfactual of the counterfactuals to mitigate the decision boundary shift. We compare CCA with DualCFX in Table 6, Appendix B.2.7.

**(vii) Studying alternate loss functions:** We explore using binary cross-entropy loss function directly with labels 0, 1 and 0.5, in place of the proposed loss. However, experiments indicate that this scheme performs poorly when compared with the CCA loss (see Fig. 18 and Appendix B.2.10).

**(viii) Validating Theorem 3.2:** Empirical verification of the theorem is done through synthetic experiments, where the model has a spherical decision boundary since they are known to be more difficult for polytope approximation [Arya et al., 2012]. Fig. 20 presents a log-log plot comparing the theoretical and empirical query complexities for several dimensionality values  $d$ . The empirical approximation error decays faster than  $n^{-2/(d-1)}$  as predicted by the theorem (see Appendix B.3).

Table 2: Fidelity achieved with different counterfactual generating methods on HELOC dataset. Target model has hidden layers with neurons (20, 30, 10). Surrogate model architecture is (10, 20).

| CF method                        | Fidelity over $\mathbb{D}_{\text{test}}$ |     |       |     | Fidelity over $\mathbb{D}_{\text{uni}}$ |     |       |     |
|----------------------------------|------------------------------------------|-----|-------|-----|-----------------------------------------|-----|-------|-----|
|                                  | n=100                                    |     | n=200 |     | n=100                                   |     | n=200 |     |
|                                  | Base.                                    | CCA | Base. | CCA | Base.                                   | CCA | Base. | CCA |
| MCCF L2-norm                     | 91                                       | 95  | 93    | 96  | 91                                      | 93  | 93    | 95  |
| MCCF L1-norm                     | 93                                       | 95  | 94    | 96  | 89                                      | 92  | 91    | 95  |
| DiCE Actionable                  | 93                                       | 94  | 95    | 95  | 90                                      | 91  | 93    | 94  |
| 1-Nearest-Neighbor               | 93                                       | 95  | 94    | 96  | 93                                      | 93  | 94    | 95  |
| ROAR [Upadhyay et al., 2021]     | 91                                       | 92  | 93    | 95  | 87                                      | 85  | 92    | 92  |
| C-CHVAE [Pawelczyk et al., 2020] | 77                                       | 80  | 78    | 82  | 90                                      | 89  | 85    | 78  |

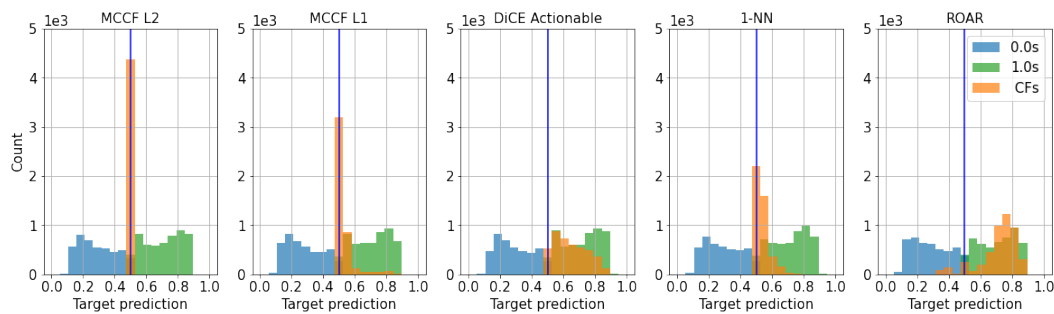


Figure 8: Histograms of the target model’s predictions on different types of input instances. Counterfactual generating methods except MCCF with  $L_2$  norm often generate counterfactuals that are farther inside the favorable region, hence having a target model prediction much greater than 0.5. We count all the query results across all the target models in the ensembles used to compute the average fidelities corresponding to each counterfactual generating method.

## 5 Conclusion

Our work provides novel insights that bridge explainability and privacy through a set of theoretical guarantees on model reconstruction using counterfactuals. We also propose a practical model reconstruction strategy based on the analysis. Experiments demonstrate a significant improvement in fidelity compared to the baseline method proposed in Aivodji et al. [2020] for the case of one-sided counterfactuals. Results show that the attack performs well across different model types and counterfactual generating methods. Our findings also highlight an interesting connection between Lipschitz constant and vulnerability to model reconstruction.

**Limitations and Future Work:** Even though Theorem 3.6 provides important insights about the role of query size in model reconstruction, it lacks an exact characterization of  $k(\epsilon)$  and  $v_i(\epsilon)$ . Moreover, local Lipschitz continuity might not be satisfied in some machine learning models such as decision trees. Any improvements along these lines can be avenues for future work. Utilizing techniques in active learning in conjunction with counterfactuals is another problem of interest. Extending the results of this work for multi-class classification scenarios can also be explored. The relationship between Lipschitz constant and vulnerability to model reconstruction may have implications for future work on generalization, robustness, etc. Another direction of future work is to design defenses for model reconstruction attacks, leveraging and building on strategies for robust counterfactuals [Dutta et al., 2022, Hamman et al., 2024, Upadhyay et al., 2021, Black et al., 2022, Jiang et al., 2023, Pawelczyk et al., 2020].

**Broader Impact:** We demonstrate that one-sided counterfactuals can be used for perfecting model reconstruction. While this can be beneficial in some cases, it also exposes a potential vulnerability in MLaaS platforms. Given the importance of counterfactuals in explaining model predictions, we hope our work will inspire countermeasures and defense strategies, paving the way toward secure and trustworthy machine learning systems.

## References

- U. Aïvodji, A. Bolot, and S. Gambs. Model extraction from counterfactual explanations. *arXiv preprint arXiv:2009.01884*, 2020.
- A. D. Aleksandrov. *A. D. Alexandrov: Selected Works Part II: Intrinsic Geometry of Convex Surfaces*. American Mathematical Society, 1967.
- B. Amos, L. Xu, and J. Z. Kolter. Input convex neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 146–155. PMLR, July 2017.
- J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias. ProPublica, 2016. URL <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing>.
- S. Arya, G. D. Da Fonseca, and D. M. Mount. Polytope approximation and the Mahler volume. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 29–42. Society for Industrial and Applied Mathematics, Jan. 2012.
- K. Ball. Cube slicing in  $\mathbb{R}^n$ . *Proceedings of the American Mathematical Society*, 97(3):465–473, 1986.
- S. Barocas, A. D. Selbst, and M. Raghavan. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 80–89, 2020.
- P. L. Bartlett, D. J. Foster, and M. J. Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- B. Becker and R. Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: <https://doi.org/10.24432/C5XW20>.
- E. Black, Z. Wang, and M. Fredrikson. Consistent counterfactuals for deep models. In *International Conference on Learning Representations*, 2022.
- K. Böröczky Jr and M. Reitzner. Approximation of smooth convex bodies by random circumscribed polytopes. *The Annals of Applied Probability*, 14(1):239–273, 2004.
- Capital One, Feb. 2024. URL <https://www.capitalone.com/learn-grow/money-management/does-getting-denied-for-credit-card-hurt-score/>.
- K.-L. Chen, H. Garudadri, and B. D. Rao. Improved bounds on neural complexity for representing piecewise linear functions. *Advances in Neural Information Processing Systems*, 35:7167–7180, 2022.
- D. Deutch and N. Frost. Constraints-based explanations of classifications. In *IEEE 35th International Conference on Data Engineering*, pages 530–541, 2019.
- A. Dhurandhar, P. Y. Chen, R. Luss, C. C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- S. Dutta, J. Long, S. Mishra, C. Tilli, and D. Magazzeni. Robust counterfactual explanations for tree-based ensembles. In *International Conference on Machine Learning*, pages 5742–5756. PMLR, 2022.
- FICO. Explainable machine learning challenge, 2018. URL <https://community.fico.com/s/explainable-machine-learning-challenge>.
- S. Goethals, K. Sörensen, and D. Martens. The privacy issue of counterfactual explanations: Explanation linkage attacks. *ACM Transactions on Intelligent Systems and Technology*, 14(5):1–24, 2023.
- X. Gong, Q. Wang, Y. Chen, W. Yang, and X. Jiang. Model extraction attacks and defenses on cloud-based machine learning models. *IEEE Communications Magazine*, 58(12):83–89, 2020.

- X. Gong, Y. Chen, W. Yang, G. Mei, and Q. Wang. Inversenet: Augmenting model extraction attacks with training data inversion. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 2439–2447, 2021.
- H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110:393–416, 2021.
- R. Guidotti. Counterfactual explanations and how to find them: Literature review and benchmarking. *Data Mining and Knowledge Discovery*, pages 1–55, 2022.
- F. Hamman, E. Noorani, S. Mishra, D. Magazzeni, and S. Dutta. Robust counterfactual explanations for neural networks with probabilistic guarantees. In *40th International Conference on Machine Learning*, 2023.
- F. Hamman, E. Noorani, S. Mishra, D. Magazzeni, and S. Dutta. Robust algorithmic recourse under model multiplicity with probabilistic guarantees. *IEEE Journal on Selected Areas in Information Theory*, 5:357–368, 2024. doi: 10.1109/JSAIT.2024.3401407.
- B. Hanin and D. Rolnick. Complexity of linear regions in deep networks. In *International Conference on Machine Learning*, pages 2596–2604. PMLR, 2019.
- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot. High accuracy and high fidelity extraction of neural networks. In *Proceedings of the 29th USENIX Conference on Security Symposium*, pages 1345–1362, 2020.
- J. Jiang, F. Leofante, A. Rago, and F. Toni. Formalising the robustness of counterfactual explanations for neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14901–14909, 2023.
- A.-H. Karimi, G. Barthe, B. Balle, and I. Valera. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics*, pages 895–905, 2020.
- A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5):1–29, 2022.
- T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443*, 2017.
- J. M. Lee. *Manifolds and Differential Geometry*. Graduate Studies in Mathematics. American Mathematical Society, 2009.
- X. Liu, X. Han, N. Zhang, and Q. Liu. Certified monotonic neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 15427–15438, 2020.
- A. L. Maas, A. Y. Hannun, A. Y. Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning*, volume 30, page 3, 2013.
- J. Marques-Silva, T. Gerspacher, M. C. Cooper, A. Ignatiev, and N. Narodytska. Explanations for monotonic classifiers. In *38th International Conference on Machine Learning*, 2021.
- S. Milli, L. Schmidt, A. D. Dragan, and M. Hardt. Model reconstruction from model explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 1–9, 2019.
- S. Mishra, S. Dutta, J. Long, and D. Magazzeni. A Survey on the Robustness of Feature Importance and Counterfactual Explanations. *arXiv e-prints*, arXiv:2111.00358, 2021.
- R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.

- Q. Nguyen, H. Valizadegan, and M. Hauskrecht. Learning classification with auxiliary probabilistic information. *IEEE International Conference on Data Mining*, 2011:477–486, 2011a.
- Q. Nguyen, H. Valizadegan, A. Seybert, and M. Hauskrecht. Sample-efficient learning with auxiliary class-label information. *AMIA Annual Symposium Proceedings*, 2011:1004–1012, 2011b.
- D. Oliynyk, R. Mayer, and A. Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55(14s), 2023.
- S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 865–872, 2020.
- N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017.
- P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer. Training robust neural networks using Lipschitz bounds. *IEEE Control Systems Letters*, 6:121–126, 2021.
- M. Pawelczyk, K. Broelemann, and G. Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of the web conference 2020*, pages 3126–3132, 2020.
- M. Pawelczyk, H. Lakkaraju, and S. Neel. On the privacy risks of algorithmic recourse. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206, pages 9680–9696, 2023.
- O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.
- R. Shokri, M. Strobel, and Y. Zick. On the privacy risks of model explanations. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 231–241, 2021.
- L. Struppek, D. Hintersdorf, A. D. A. Correia, A. Adler, and K. Kersting. Plug & play attacks: Towards robust and flexible model inversion attacks. In *International Conference on Machine Learning*, pages 20522–20545. PMLR, 2022.
- F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction APIs. In *25th USENIX Security Symposium*, pages 601–618, 2016.
- S. Upadhyay, S. Joshi, and H. Lakkaraju. Towards robust and reliable algorithmic recourse. *Advances in Neural Information Processing Systems*, 34:16926–16937, 2021.
- S. Verma, V. Boonsanong, M. Hoang, K. E. Hines, J. P. Dickerson, and C. Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2022.
- S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law and Technology*, 31:841, 2017.
- Y. Wang, H. Qian, and C. Miao. DualCF: Efficient model extraction attack from counterfactual explanations. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1318–1329, 2022.
- C. Yadav, M. Moshkovitz, and K. Chaudhuri. Xaudit : A theoretical look at auditing with explanations. *arXiv:2206.04740*, 2023.
- I.-C. Yeh. Default of Credit Card Clients. UCI Machine Learning Repository, 2016. DOI: <https://doi.org/10.24432/C55S3H>.
- M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton. On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521, 2013.
- X. Zhao, W. Zhang, X. Xiao, and B. Lim. Exploiting explanations for model inversion attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 682–692, 2021.

## A Proof of Theoretical Results

### A.1 Proof of Lemma 2.7 and Lemma 3.1

**Lemma 2.7.** Let  $S(x) = 0$  and  $T(x) = 0$  denote two differentiable hypersurfaces in  $\mathbb{R}^d$ , touching each other at point  $w$ . Then,  $S(x) = 0$  and  $T(x) = 0$  have a common tangent hyperplane at  $w$ .

*Proof.* From Definition 2.6, there exists a non-empty neighborhood  $\mathcal{B}_w$  around  $w$ , such that  $\forall x \in \mathcal{B}_w$  with  $S(x) = 0$  and  $x \neq w$ , only one of  $T(x) > 0$  or  $T(x) < 0$  holds. Let  $x = (x_1, x_2, \dots, x_d)$  and  $x_{[p]}$  denote  $x$  without  $x_p$  for  $1 \leq p \leq d$ . Then, within the neighborhood  $\mathcal{B}_w$ , we may re-parameterize  $S(x) = 0$  as  $x_p = S(x_{[p]})$ . Note that a similar re-parameterization denoted by  $x_p = T(x_{[p]})$  can be applied to  $T(x) = 0$  as well. Let  $\mathcal{A}_w = \{x_{[p]} : x \in \mathcal{B}_w \setminus \{w\}\}$ . From Definition 2.6, all  $x \in \mathcal{B}_w \setminus \{w\}$  satisfy only one of  $T(x) < 0$  or  $T(x) > 0$ , and hence without loss of generality the re-parameterization of  $T(x) = 0$  can be such that  $S(x_{[p]}) < T(x_{[p]})$  holds for all  $x_{[p]} \in \mathcal{A}_w$ . Now, define  $F(x_{[p]}) \equiv T(x_{[p]}) - S(x_{[p]})$ . Observe that  $F(x_{[p]})$  has a minimum at  $w$  and hence,  $\nabla_{x_{[p]}} F(w_{[p]}) = 0$ . Consequently,  $\nabla_{x_{[p]}} T(w_{[p]}) = \nabla_{x_{[p]}} S(w_{[p]})$ , which implies that the tangent hyperplanes to both hypersurfaces have the same gradient at  $w$ . Proof concludes by observing that since both tangent hyperplanes go through  $w$ , the two hypersurfaces should share a common tangent hyperplane at  $w$ .  $\square$

**Lemma 3.1.** Let  $S$  denote the decision boundary of a classifier and  $x \in [0, 1]^d$  be any point that is not on  $S$ . Then, the line joining  $x$  and its closest counterfactual  $w$  is perpendicular to  $S$  at  $w$ .

*Proof.* The proof utilizes the following lemma.

**Lemma A.1.** Consider the  $d$ -dimensional ball  $\mathcal{C}_x$  centered at  $x$ , with  $w$  lying on its boundary (hence  $\mathcal{C}_x$  intersects  $S$  at  $w$ ). Then,  $S$  lies completely outside  $\mathcal{C}_x$ .

The proof of Lemma A.1 follows from the following contradiction. Assume a part of  $S$  lies within  $\mathcal{C}_x$ . Then, points on the intersection of  $S$  and the interior of  $\mathcal{C}_x$  are closer to  $x$  than  $w$ . Hence,  $w$  can no longer be the closest point to  $x$ , on  $S$ .

From Lemma A.1,  $\mathcal{C}_x$  is touching the curve  $S$  at  $w$ , and hence, they share the same tangent hyperplane at  $w$  by Lemma 2.7. Now, observing that the line joining  $w$  and  $x$ , being a radius of  $\mathcal{C}_x$ , is the normal to the ball at  $w$  concludes the proof (see Fig. 9).  $\square$

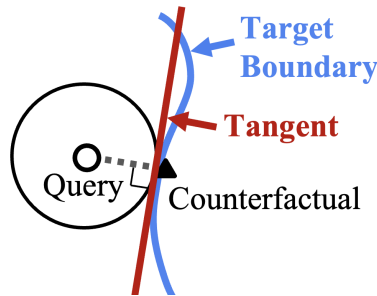


Figure 9: Line joining the query and its closest counterfactual is perpendicular to the decision boundary at the counterfactual. See Lemma 3.1 for details.

We present the following corollary as an additional observation resulting from Lemma A.1.

**Corollary A.2.** Following Lemma A.1, it can be seen that all the points in the  $d$ -dimensional ball with  $x$  as the center and  $w$  on boundary lies on the same side of  $S$  as  $x$ .



## A.2 Proof of Theorem 3.2

**Theorem 3.2.** *Let  $m$  be the target binary classifier whose decision boundary is convex (i.e., the set  $\{\mathbf{x} \in [0, 1]^d : \lfloor m(\mathbf{x}) \rfloor = 1\}$  is convex) and has a continuous second derivative. Denote by  $\tilde{M}_n$ , the convex polytope approximation of  $m$  constructed with  $n$  supporting hyperplanes obtained through i.i.d. counterfactual queries. Assume that the fidelity is evaluated with respect to  $\mathbb{D}_{\text{ref}}$  which is uniformly distributed over  $[0, 1]^d$ . Then, when  $n \rightarrow \infty$  the expected fidelity of  $\tilde{M}_n$  with respect to  $m$  is given by*

$$\mathbb{E} \left[ \text{Fid}_{m, \mathbb{D}_{\text{ref}}}(\tilde{M}_n) \right] = 1 - \epsilon \quad (1)$$

where  $\epsilon \sim \mathcal{O} \left( n^{-\frac{2}{d-1}} \right)$  and the expectation is over both  $\tilde{M}_n$  and  $\mathbb{D}_{\text{ref}}$ .

*Proof.* We first have a look at Böröczky Jr and Reitzner [2004, Theorem 1 (restated as Theorem A.3 below)] from the polytope theory. Let  $\mathbb{M}$  be a compact convex set with a second-order differentiable boundary denoted by  $\partial\mathbb{M}$ . Let  $\mathbf{a}_1, \dots, \mathbf{a}_n$  be  $n$  randomly chosen points on  $\partial\mathbb{M}$ , distributed independently and identically according to a given density  $d_{\partial\mathbb{M}}$ . Denote by  $H_+(\mathbf{a}_i)$  the supporting hyperplane of  $\partial\mathbb{M}$  at  $\mathbf{a}_i$ . Assume  $C$  to be a large enough hypercube which contains  $\mathbb{M}$  in its interior.

Now, define

$$\tilde{\mathbb{M}}_n = \bigcap_{i=1}^n H_+(\mathbf{a}_i) \cap C \quad (4)$$

which is the polytope created by the intersection of all the supporting hyperplanes. The theorem characterizes the expected difference of the volumes of  $\mathbb{M}$  and  $\tilde{\mathbb{M}}_n$ .

**Theorem A.3** (Random Polytope Approximation, [Böröczky Jr and Reitzner, 2004]). *For a convex compact set  $\mathbb{M}$  with second-order differentiable  $\partial\mathbb{M}$  and non-zero continuous density  $d_{\partial\mathbb{M}}$ ,*

$$\mathbb{E} \left[ V(\tilde{\mathbb{M}}_n) - V(\mathbb{M}) \right] = \tau(\partial\mathbb{M}, d) n^{-\frac{2}{d-1}} + o \left( n^{-\frac{2}{d-1}} \right) \quad (5)$$

as  $n \rightarrow \infty$ , where  $V(\cdot)$  denotes the volume (i.e., the Lebesgue measure), and  $\tau(\partial\mathbb{M}, d)$  is a constant that depends only on the boundary  $\partial\mathbb{M}$  and the dimensionality  $d$  of the space.

Let  $\mathbf{x}_i, i = 1, \dots, n$  be  $n$  i.i.d queries from the  $\lfloor m(\mathbf{x}) \rfloor = 0$  region of the target model. Then, their corresponding counterfactuals  $g_m(\mathbf{x}_i)$  are also i.i.d. Furthermore, they lie on the decision boundary of  $m$ . Hence, we may arrive at the following result.

**Corollary A.4.** *Let  $\mathbb{M} = \{\mathbf{x} \in [0, 1]^d : \lfloor m(\mathbf{x}) \rfloor = 1\}$  and  $\tilde{\mathbb{M}}_n = \{\mathbf{x} \in [0, 1]^d : \lfloor \tilde{M}_n(\mathbf{x}) \rfloor = 1\}$ . Then, by Theorem A.3,*

$$\mathbb{E} \left[ V(\tilde{\mathbb{M}}_n) - V(\mathbb{M}) \right] \sim \mathcal{O} \left( n^{-\frac{2}{d-1}} \right) \quad (6)$$

when  $n \rightarrow \infty$ . Note that  $\mathbb{M} \subseteq \tilde{\mathbb{M}}_n$  and hence, the left-hand side is always non-negative.

From Definition 2.4, we may write

$$\begin{aligned} & \mathbb{E} \left[ \text{Fid}_{m, \mathbb{D}_{\text{ref}}}(\tilde{M}_n) \right] \\ &= \mathbb{E} \left[ \frac{1}{|\mathbb{D}_{\text{ref}}|} \sum_{\mathbf{x} \in \mathbb{D}_{\text{ref}}} \mathbb{E} \left[ \mathbb{1} \left[ \lfloor m(\mathbf{x}) \rfloor = \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \right] \middle| \mathbb{D}_{\text{ref}} \right] \right] \end{aligned} \quad (7)$$

$$= \frac{1}{|\mathbb{D}_{\text{ref}}|} \mathbb{E} \left[ \sum_{\mathbf{x} \in \mathbb{D}_{\text{ref}}} \mathbb{P} \left[ \lfloor m(\mathbf{x}) \rfloor = \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \middle| \mathbf{x} \right] \right] \quad (\because \text{query size is fixed}) \quad (8)$$

$$= \mathbb{P} \left[ \lfloor m(\mathbf{x}) \rfloor = \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \right] \quad (\because \mathbf{x}'\text{s are i.i.d.}) \quad (9)$$

$$= \int_{\mathcal{M}_n} \mathbb{P} \left[ \lfloor m(\mathbf{x}) \rfloor = \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \middle| \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right] \mathbb{P} \left[ \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right] d\tilde{m}_n \quad (10)$$

where  $\mathcal{M}_n$  is the set of all possible  $\tilde{m}_n$ 's.

Now, by noting that

$$\mathbb{P} \left[ \lfloor m(\mathbf{x}) \rfloor = \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \mid \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right] = 1 - \mathbb{P} \left[ \lfloor m(\mathbf{x}) \rfloor \neq \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \mid \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right], \quad (11)$$

we may obtain

$$\mathbb{E} \left[ \text{Fid}_{m, \mathbb{D}_{\text{ref}}}(\tilde{M}_n) \right] = 1 - \int_{\mathcal{M}_n} \mathbb{P} \left[ \lfloor m(\mathbf{x}) \rfloor \neq \lfloor \tilde{M}_n(\mathbf{x}) \rfloor \mid \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right] \times \mathbb{P} \left[ \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right] d\tilde{m}_n \quad (12)$$

$$= 1 - \int_{\mathcal{M}_n} \underbrace{\frac{V(\tilde{\mathbb{M}}_n) - V(\mathbb{M})}{\text{Total volume}}}_{=1 \text{ for unit hypercube}} \mathbb{P} \left[ \tilde{M}_n(\mathbf{x}) = \tilde{m}_n(\mathbf{x}) \right] d\tilde{m}_n \quad (\because \mathbf{x}'\text{s are uniformly distributed}) \quad (13)$$

$$= 1 - \mathbb{E} \left[ V(\tilde{\mathbb{M}}_n) - V(\mathbb{M}) \right]. \quad (14)$$

The above result, in conjunction with Corollary A.4, concludes the proof.  $\square$

### A.3 Proof of Lemma 3.5

**Lemma 3.5.** *Let  $m(\mathbf{x}) = \sigma(\ell(\mathbf{x}))$  be a ReLU classifier, where  $\ell(\mathbf{x})$  is CPWL and  $\sigma(\cdot)$  is the Sigmoid function. Then, the decision boundary  $\partial\mathbb{M} = \{\mathbf{x} \in [0, 1]^d : m(\mathbf{x}) = 0.5\}$  is a collection of (possibly non-convex) polytopes in  $[0, 1]^d$ , when considered along with the boundaries of the unit hypercube.*

*Proof.* Consider the  $i^{\text{th}}$  piece  $m_i(\mathbf{x})$  of the classifier defined over  $\mathbb{U}_i$ . A part of the decision boundary exists within  $\mathbb{U}_i$  only if  $\exists \mathbf{x} \in \mathbb{U}_i$  such that  $m_i(\mathbf{x}) = 0.5$ . When it is the case, at the decision boundary,

$$m(\mathbf{x}) = 0.5 \quad (15)$$

$$\iff \frac{1}{1 + e^{-\ell_i(\mathbf{x})}} = 0.5 \quad (16)$$

$$\iff e^{-\ell_i(\mathbf{x})} = 1 \quad (17)$$

$$\iff \ell_i(\mathbf{x}) = 0 \quad (18)$$

$$\iff \mathbf{a}_i^T \mathbf{x} + b_i = 0 \quad (19)$$

which represents a hyperplane restricted to  $\mathbb{U}_i$ . Moreover, the continuity of the  $\ell(\mathbf{x})$  demands the decision boundary to be continuous across the boundaries of  $\mathbb{U}_i$ 's. This fact can be proved as follows:

Note that within each region  $\mathbb{U}_i$ , exactly one of the following three conditions holds:

- (a)  $\forall \mathbf{x} \in \mathbb{U}_i, \ell_i(\mathbf{x}) > 0 \rightarrow \mathbb{U}_i$  does not contain a part of the decision boundary
- (b)  $\forall \mathbf{x} \in \mathbb{U}_i, \ell_i(\mathbf{x}) < 0 \rightarrow \mathbb{U}_i$  does not contain a part of the decision boundary
- (c)  $\exists \mathbf{x} \in \mathbb{U}_i, \ell_i(\mathbf{x}) = 0 \rightarrow \mathbb{U}_i$  contains a part of the decision boundary

In case when (c) holds for some region  $\mathbb{U}_i$ , the decision boundary within  $\mathbb{U}_i$  is affine and it extends from one point to another on the region boundary. Now let  $\mathbb{U}_s$  and  $\mathbb{U}_t, s, t \in \{1, \dots, q\}, s \neq t$  be two adjacent regions sharing a boundary. Assume that  $\mathbb{U}_s$  contains a portion of the decision boundary, which intersects with a part of the shared boundary between  $\mathbb{U}_s$  and  $\mathbb{U}_t$  (note that  $\mathbb{U}_i$ 's are closed and hence they include their boundaries). Let  $\mathbf{x}_0$  be a point in the intersection of the decision boundary within  $\mathbb{U}_s$  and the shared region boundary. Now, continuity of  $\ell(\mathbf{x})$  at  $\mathbf{x}_0$  requires  $\ell_t(\mathbf{x}_0) = \ell_s(\mathbf{x}_0) = 0$ . Hence, condition (c) holds for  $\mathbb{U}_t$ . Moreover, this holds for all the points in the said intersection. Therefore, if such a shared boundary exists between  $\mathbb{U}_s$  and  $\mathbb{U}_t$ , then the decision boundary continues to  $\mathbb{U}_t$ . Applying the argument to all  $\mathbb{U}_s - \mathbb{U}_t$  pairs show that each segment of the decision boundary either closes upon itself or ends at a boundary of the unit hypercube. Hence, when taken along with the boundaries of the unit hypercube, the decision boundary is a collection of polytopes.  $\square$

#### A.4 Proof of Theorem 3.6

**Theorem 3.6.** Let  $m$  be a target binary classifier with ReLU activations. Let  $k(\epsilon)$  be the number of cells through which the decision boundary passes. Define  $\{\mathbb{H}_i\}_{i=1,\dots,k(\epsilon)}$  to be the collection of affine pieces of the decision boundary within each decision boundary cell where each  $\mathbb{H}_i$  is an open set. Let  $v_i(\epsilon) = V(\mathbb{G}_{m,g_m}(\mathbb{H}_i))$  where  $V(\cdot)$  is the  $d$ -dimensional volume (i.e., the Lebesgue measure) and  $\mathbb{G}_{m,g_m}(\cdot)$  is the inverse counterfactual region w.r.t.  $m$  and the closest counterfactual generator  $g_m$ . Then the probability of successful reconstruction with counterfactual queries distributed uniformly over  $[0, 1]^d$  is lower-bounded as

$$\mathbb{P}[\text{Reconstruction}] \geq 1 - k(\epsilon)(1 - v^*(\epsilon))^n \quad (2)$$

where  $v^*(\epsilon) = \min_{i=1,\dots,k(\epsilon)} v_i(\epsilon)$  and  $n$  is the number of queries.

*Proof.* Note that

$$\mathbb{P}[\text{Reconstruction}] = \mathbb{P}[\text{There is a counterfactual in every decision boundary cell}] \quad (20)$$

$$= 1 - \mathbb{P}[\text{At least one decision boundary cell does not have a counterfactual}] \quad (21)$$

$$= 1 - \sum_{i=1}^{k(\epsilon)} \mathbb{P}[i^{\text{th}} \text{ decision boundary cell does not have a counterfactual}] \quad (22)$$

Let  $\mathcal{M}_i$  denote the event “ $i^{\text{th}}$  decision boundary cell does not have a counterfactual”. At the end of  $n$  queries,

$$\mathbb{P}[\mathcal{M}_i] = \prod_{j=1}^n \underbrace{\mathbb{P}[j^{\text{th}} \text{ query falling outside of } \mathbb{G}_{m,g_m}(\mathbb{H}_i)]}_{=1-v_i(\epsilon) \text{ for uniform queries}} \quad (23)$$

$$= (1 - v_i(\epsilon))^n. \quad (24)$$

Therefore,

$$\mathbb{P}[\text{Reconstruction}] = 1 - \sum_{i=1}^{k(\epsilon)} (1 - v_i(\epsilon))^n \quad (25)$$

$$\geq 1 - k(\epsilon)(1 - v^*(\epsilon))^n \quad \left( \because v_i(\epsilon) \geq v^*(\epsilon) = \min_j v_j(\epsilon) \right). \quad (26)$$

□

#### A.5 Proof of Theorem 3.10 and Corollary 3.11

**Theorem 3.10.** Let the target  $m$  and surrogate  $\tilde{m}$  be ReLU classifiers such that  $m(\mathbf{w}) = \tilde{m}(\mathbf{w})$  for every counterfactual  $\mathbf{w}$ . For any point  $\mathbf{x}$  that lies in a decision boundary cell,  $|\tilde{m}(\mathbf{x}) - m(\mathbf{x})| \leq \sqrt{d}(\gamma_m + \gamma_{\tilde{m}})\epsilon$  holds with probability  $p \geq 1 - k(\epsilon)(1 - v^*(\epsilon))^n$ .

**Corollary 3.11.** Suppose the target  $m$  and surrogate  $\tilde{m}$  are locally Lipschitz (not necessarily ReLU) such that  $m(\mathbf{w}) = \tilde{m}(\mathbf{w})$  for every counterfactual  $\mathbf{w}$ . Assume the counterfactuals are well-spaced out and forms a  $\delta$ -cover over the decision boundary. Then  $|\tilde{m}(\mathbf{x}) - m(\mathbf{x})| \leq (\gamma_m + \gamma_{\tilde{m}})\delta$ , over the target decision boundary.

*Proof.* Note that from Theorem 3.6, with probability  $p \geq 1 - k(\epsilon)(1 - v^*(\epsilon))^n$  at least one counterfactual exists within each decision boundary cell. When this is the case, we have

$$|\tilde{m}(\mathbf{x}) - m(\mathbf{x})| = |\tilde{m}(\mathbf{x}) - \tilde{m}(\mathbf{w}) - (m(\mathbf{x}) - \tilde{m}(\mathbf{w}))| \quad (27)$$

$$= |\tilde{m}(\mathbf{x}) - \tilde{m}(\mathbf{w}) - (m(\mathbf{x}) - m(\mathbf{w}))| \quad (28)$$

$$\leq \underbrace{|\tilde{m}(\mathbf{x}) - \tilde{m}(\mathbf{w})|}_{\leq \gamma_{\tilde{m}} \|\mathbf{x} - \mathbf{w}\|_2} + \underbrace{|m(\mathbf{x}) - m(\mathbf{w})|}_{\leq \gamma_m \|\mathbf{x} - \mathbf{w}\|_2} \quad (29)$$

$$\leq (\gamma_m + \gamma_{\tilde{m}}) \|\mathbf{x} - \mathbf{w}\|_2 \quad (30)$$

$$\leq \sqrt{d}(\gamma_m + \gamma_{\tilde{m}})\epsilon \quad (31)$$

where the first inequality is a result of applying the triangle inequality and the second follows from the definition of local Lipschitz continuity (Definition 3.9). The final inequality is due to the availability of a counterfactual within each decision boundary cell, which ensures  $\|\mathbf{x} - \mathbf{w}\|_2 \leq \sqrt{d}\epsilon$ . Corollary 3.11 follows directly from the second inequality, since the  $\delta$ -cover of  $\mathbf{w}$ 's ensure  $\|\mathbf{x} - \mathbf{w}\|_2 \leq \delta$   $\square$

## B Experimental Details and Additional Results

All the experiments were carried-out on two computers, one with a NVIDIA RTX A4500 GPU and the other with a NVIDIA RTX 3050 Mobile.

### B.1 Details of Real-World Datasets

We use four publicly available real-world tabular datasets (namely, Adult Income, COMPAS, DCCC, and HELOC) to evaluate the performance of the attack proposed in Section 3.3. The details of these datasets are as follows:

- **Adult Income:** The dataset is a 1994 census database with information such as educational level, marital status, age and annual income of individuals [Becker and Kohavi, 1996]. The target is to predict “income”, which indicates whether the annual income of a given person exceeds \$50000 or not (i.e.,  $y = \mathbb{1}[\text{income} \geq 0.5]$ ). It contains 32561 instances in total (the training set), comprising of 24720 from  $y = 0$  and 7841 from  $y = 1$ . To make the dataset class-wise balanced we randomly sample 7841 instances from class  $y = 0$ , giving a total effective size of 15682 instances. Each instance has 6 numerical features and 8 categorical features. During pre-processing, categorical features are encoded as integers. All the features are then normalized to the range  $[0, 1]$ .
- **Home Equity Line of Credit (HELOC):** This dataset contains information about customers who have requested a credit line as a percentage of home equity FICO [2018]. It contains 10459 instances with 23 numerical features each. Prediction target is “is\_at\_risk” which indicates whether a given customer would pay the loan in the future. Dataset is slightly unbalanced with class sizes of 5000 and 5459 for  $y = 0$  and  $y = 1$ , respectively. Instead of using all 23 features, we use the following subset of 10 for our experiments; “estimate\_of\_risk”, “net\_fraction\_of\_revolving\_burden”, “percentage\_of\_legal\_trades”, “months\_since\_last\_inquiry\_not\_recent”, “months\_since\_last\_trade”, “percentage\_trades\_with\_balance”, “number\_of\_satisfactory\_trades”, “average\_duration\_of\_resolution”, “nr\_total\_trades”, “nr\_banks\_with\_high\_ratio”. All the features are normalized to lie in the range  $[0, 1]$ .
- **Correctional Offender Management Profiling for Alternative Sanctions (COMPAS):** This dataset has been used for investigating racial biases in a commercial algorithm used for evaluating reoffending risks of criminal defendants [Angwin et al., 2016]. It includes 6172 instances and 20 numerical features. The target variable is “is\_recid”. Class-wise counts are 3182 and 2990 for  $y = 0$  and  $y = 1$ , respectively. All the features are normalized to the interval  $[0, 1]$  during pre-processing.
- **Default of Credit Card Clients (DCCC):** The dataset includes information about credit card clients in Taiwan Yeh [2016]. The target is to predict whether a client will default on the credit or not, indicated by “default.payment.next.month”. The dataset contains 30000 instances with 24 attributes each. Class-wise counts are 23364 from  $y = 0$  and 6636 from  $y = 1$ . To alleviate the imbalance, we randomly select 6636 instances from  $y = 0$  class, instead of using all the instances. Dataset has 3 categorical attributes, which we encode into integer values. All the attributes are normalized to  $[0, 1]$  during pre-processing.

### B.2 Experiments on the attack proposed in Section 3.3

In this section, we provide details about our experimental setup with additional results. For convenience, we present the neural network model architectures by specifying the number of neurons in each hidden layer as a tuple, where the leftmost element corresponds to the layer next to the input; e.g.: a model specified as (20,30,10) has the following architecture:

Input  $\rightarrow$  Dense(20, ReLU)  $\rightarrow$  Dense(30, ReLU)  $\rightarrow$  Dense(10, ReLU)  $\rightarrow$  Output(Sigmoid)

Other specifications of the models, as detailed below, are similar across most of the experiments. Changes are specified specifically. The hidden layer activations are ReLU and the layer weights are  $L_2$ -regularized. The regularization coefficient is 0.001. Each model is trained for 200 epochs, with a batch size of 32.

Fidelity is evaluated over a uniformly sampled set of input instances (uniform data) as well as a held-out portion of the original data (test data). The experiments were carried out as follows:

1. Initialize the target model. Train using  $\mathbb{D}_{\text{train}}$ .
2. For  $t = 1, 2, \dots, T$  :
  - (a) Sample  $N \times t$  data points from the dataset to create  $\mathbb{D}_{\text{attack}}$ .
  - (b) Carry-out the attack given in Algorithm 1 with  $\mathbb{D}_{\text{attack}}$ . Use  $k = 1$  for “Baseline” models and  $k = 0.5$  for “Proposed” models.
  - (c) Record the fidelity over  $\mathbb{D}_{\text{ref}}$  along with  $t$ .
3. Repeat steps 1 and 2 for  $S$  number of times and calculate average fidelities for each  $t$ , across repetitions.

Based on the experiments of Aïvodji et al. [2020] and Wang et al. [2022], we select  $T = 20, 50, 100$ ;  $N = 20, 8, 4$  and  $S = 100, 50$ , in different experiments. We note that the exact numerical results are often variable due to the multiple random factors affecting the outcome such as the test-train-attack split, target and surrogate model initialization, and the randomness incorporated in the counterfactual generating methods. Nevertheless, the advantage of CCA over the baseline attack is observed across different realizations.

### B.2.1 Visualizing the attack using synthetic data

This experiment is conducted on a synthetic dataset which consists of 1000 samples generated using the `make_moons` function from the `sklearn` package. Features are normalized to the range  $[0, 1]$  before feeding to the classifier. The target model has 4 hidden layers with the architecture (10, 20, 20, 10). The surrogate model is 3-layered with the architecture (10, 20, 20). Each model is trained for 100 epochs. Since the intention of this experiment is to demonstrate the functionality of the modified loss function given in (3), a large query of size 200 is used, instead of performing multiple small queries. Fig. 7 shows how the original model reconstruction proposed by Aïvodji et al. [2020] suffers from the boundary shift issue, while the model with the proposed loss function overcomes this problem. Fig. 10 illustrates the instances misclassified by the two surrogate models.

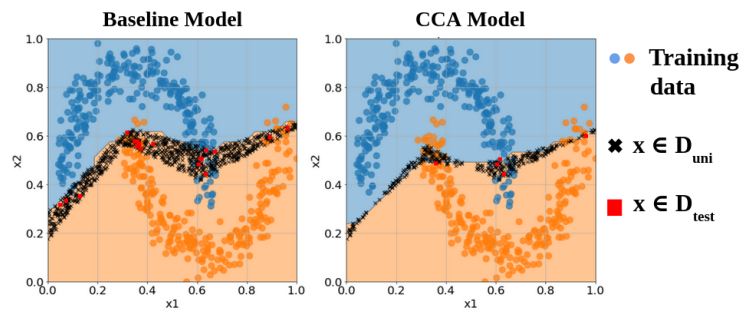


Figure 10: Misclassifications w.r.t. to the target model, over  $\mathbb{D}_{\text{uni}}$  and  $\mathbb{D}_{\text{test}}$  as the reference datasets for the 2-dimensional demonstration in Fig. 7. “Baseline” model causes a large number of misclassifications w.r.t. the “CCA” model.

### B.2.2 Comparing performance over four real-world dataset

We use a target model having 2 hidden layers with the architecture (20,10). Two surrogate model architectures, one exactly similar to the target architecture (model 0 - known architecture) and the other slightly different (model 1 - unknown architecture), are tested. Model 1 has 3 hidden layers with the architecture (20,10,5).

Fig. 11 illustrates the fidelities achieved by the two model architectures described above. Fig. 12 shows the corresponding variances of the fidelity values over 100 realizations. It can be observed that

the variances diminish as the query size grows, indicating more stable model reconstructions. Fig. 13 demonstrates the effect of the proposed loss function in mitigating the decision boundary shift issue.

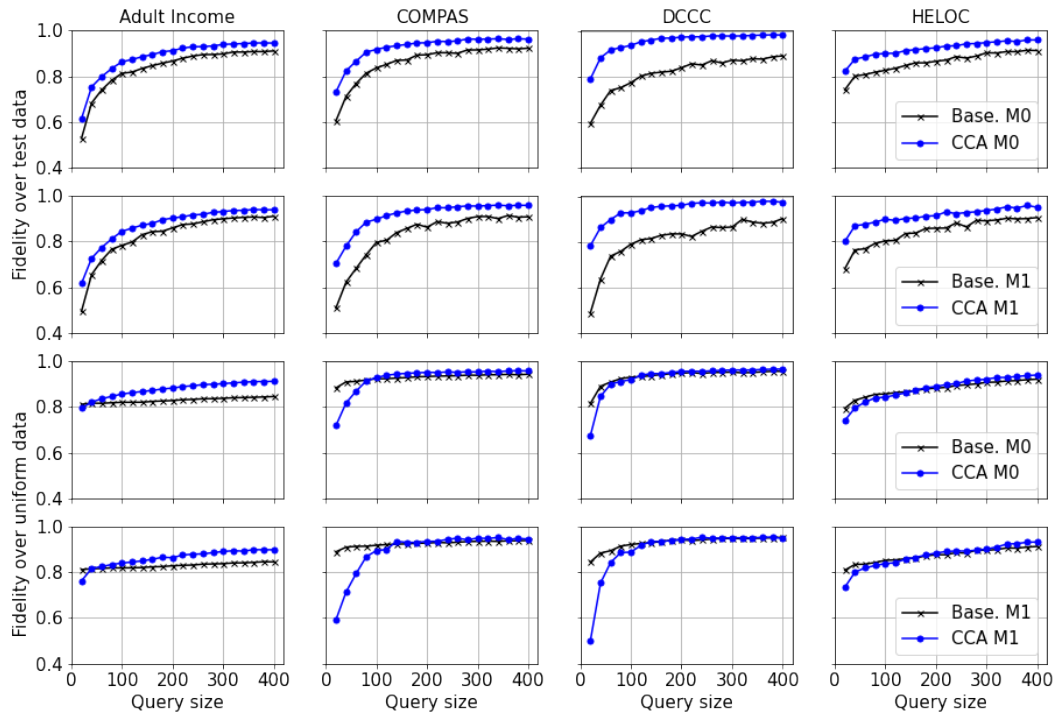


Figure 11: Fidelity for real-world datasets. Blue lines indicate “CCA” models. Black lines indicate “Baseline” models.

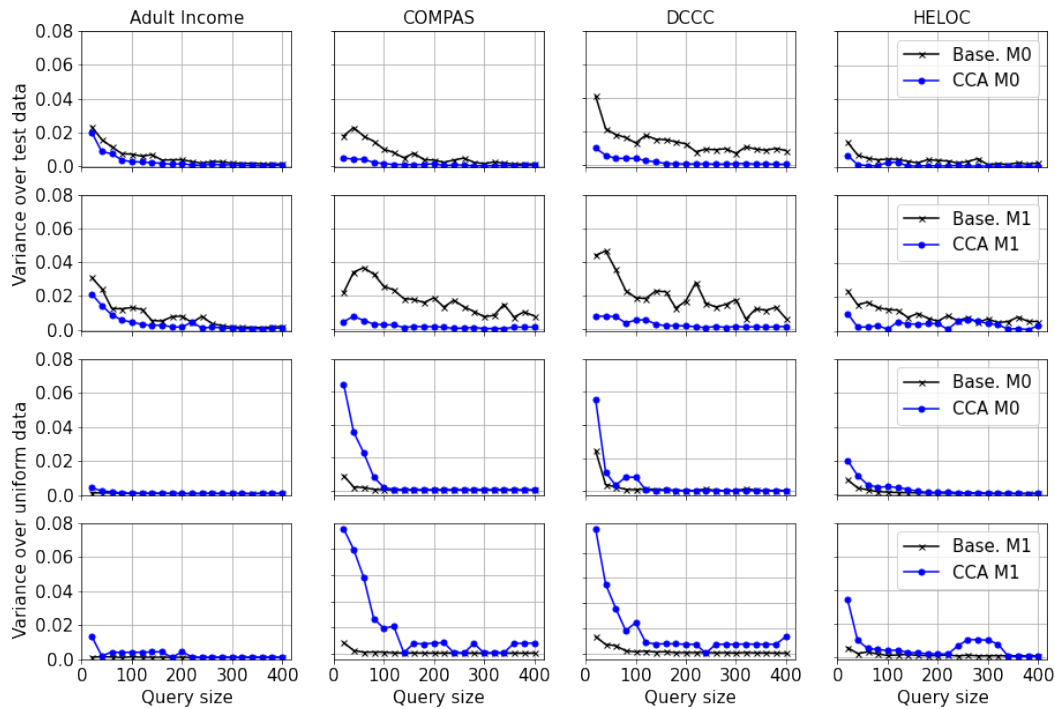


Figure 12: Variance of fidelity for real-world datasets. Blue lines indicate “CCA” models. Black lines indicate “Baseline” models.

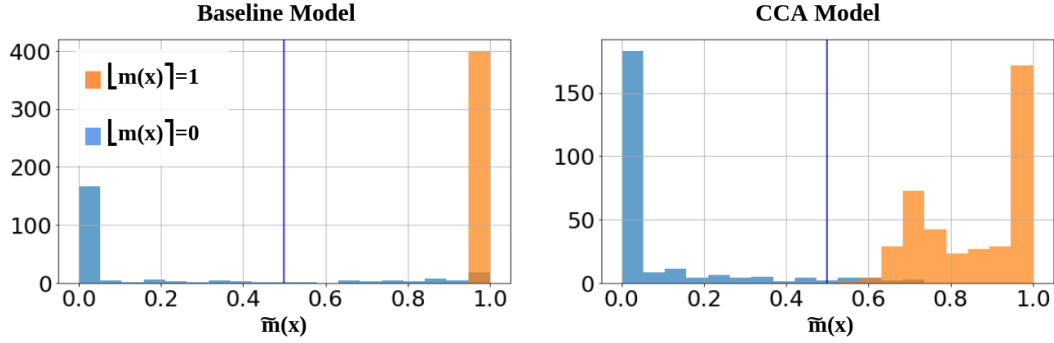


Figure 13: Histograms of probabilities predicted by “Baseline” and “CCA” models under the “Unknown Architecture” scenario (model 1) for the HELOC dataset. Note how the “Baseline” model provides predictions higher than 0.5 for a comparatively larger number of instances with  $\lfloor m(x) \rfloor = 0$  due to the boundary shift issue. The clamping effect of the novel loss function is evident in the “CCA” model’s histogram, where the decision boundary being held closer to the counterfactuals is causing the two prominent modes in the favorable region. The mode closer to 0.5 is due to counterfactuals and the mode closer to 1.0 is due to instances with  $\lfloor m(x) \rfloor = 1$ .

### B.2.3 Empirical and theoretical rates of convergence

Fig. 14 compares the rate of convergence of the empirical approximation error i.e.,  $1 - \mathbb{E} [\text{Fid}_{m, \mathbb{D}_{\text{ref}}}(\tilde{M}_n)]$  for two of the above experiments with the rate predicted by Theorem 3.2. Notice how the empirical error decays faster than  $n^{-2/(d-1)}$ .

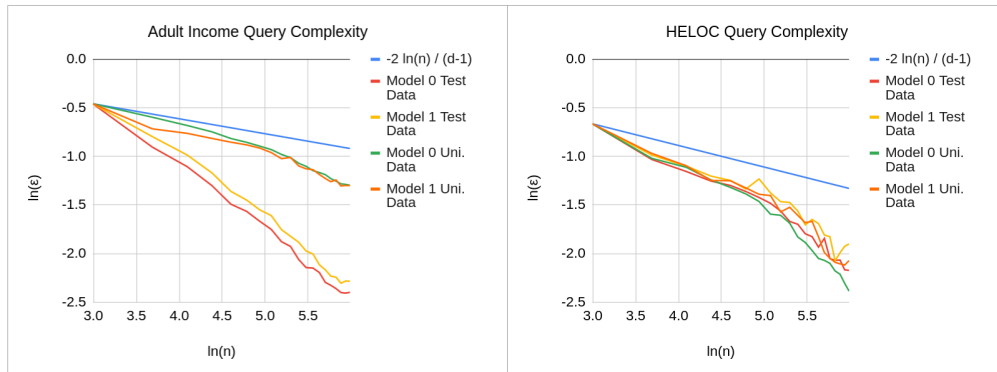


Figure 14: A comparison of the query complexity derived in Theorem 3.2 with the empirical query complexities obtained on the Adult Income and HELOC datasets. The graphs are on a log-log scale. We observe that the analytical query complexity is an upper bound for the empirical query complexities. All the graphs are recentered with an additive constant for presentational convenience. However, this does not affect the slope of the graph, which corresponds to the complexity.

### B.2.4 Studying effects of Lipschitz constants

For this experiment, we use a target model having 3 hidden layers with the architecture (20, 10, 5) and a surrogate model having 2 hidden layers with the architecture (20, 10). The surrogate model layers are  $L_2$ -regularized with a fixed regularization coefficient of 0.001. We achieve different Lipschitz constants for the target models by controlling their  $L_2$ -regularization coefficients during the target model training step. Following Gouk et al. [2021], we approximate the Lipschitz constant of target models by the product of the spectral norms of the weight matrices.

Fig. 15 illustrates the dependence of the attack performance on the Lipschitz constant of the target model. The results lead to the conclusion that target models with larger Lipschitz constants are more difficult to extract. This follows the insight provided by Theorem 3.10.



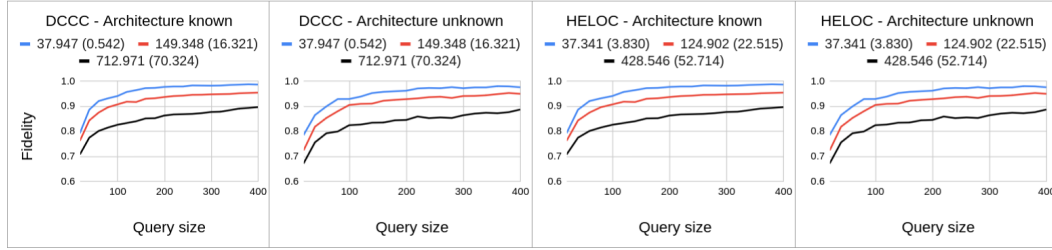


Figure 15: Dependence of fidelity on the target model’s Lipschitz constant. The approximations of the Lipschitz constants are shown in the legend with standard deviations within brackets. Lipschitz constants are approximated as the product of the spectral norm of weight matrices in each model. With a higher Lipschitz constant, the fidelity achieved by a given number of queries tend to degrade.

### B.2.5 Studying different model architectures

We observe the effect of the model architectures on the attack performance over Adult Income, COMPAS and HELOC datasets. Tables 3, 4, and 5, respectively, present the results.

Table 3: Fidelity over  $\mathbb{D}_{\text{test}}$  and  $\mathbb{D}_{\text{uni}}$  for Adult Income dataset

| Target $\rightarrow$<br>$\mathbb{D}_{\text{test}}$ | (20,10) |      |       |      | (20,10,5) |      |       |      | (20,20,10,5) |      |       |      |
|----------------------------------------------------|---------|------|-------|------|-----------|------|-------|------|--------------|------|-------|------|
|                                                    | n=100   |      | n=200 |      | n=100     |      | n=200 |      | n=100        |      | n=200 |      |
| Surrogate $\downarrow$                             | Base.   | CCA  | Base. | CCA  | Base.     | CCA  | Base. | CCA  | Base.        | CCA  | Base. | CCA  |
| (20,10)                                            | 0.88    | 0.89 | 0.92  | 0.93 | 0.82      | 0.84 | 0.91  | 0.93 | 0.94         | 0.95 | 0.95  | 0.96 |
| (20,10,5)                                          | 0.87    | 0.88 | 0.91  | 0.93 | 0.79      | 0.82 | 0.90  | 0.92 | 0.93         | 0.94 | 0.95  | 0.96 |
| (20,20,10,5)                                       | 0.85    | 0.86 | 0.91  | 0.91 | 0.79      | 0.81 | 0.89  | 0.92 | 0.93         | 0.92 | 0.95  | 0.95 |

| Target $\rightarrow$<br>$\mathbb{D}_{\text{uni}}$ | (20,10) |      |       |      | (20,10,5) |      |       |      | (20,20,10,5) |      |       |      |
|---------------------------------------------------|---------|------|-------|------|-----------|------|-------|------|--------------|------|-------|------|
|                                                   | n=100   |      | n=200 |      | n=100     |      | n=200 |      | n=100        |      | n=200 |      |
| Surrogate $\downarrow$                            | Base.   | CCA  | Base. | CCA  | Base.     | CCA  | Base. | CCA  | Base.        | CCA  | Base. | CCA  |
| (20,10)                                           | 0.71    | 0.81 | 0.75  | 0.87 | 0.78      | 0.84 | 0.79  | 0.87 | 0.84         | 0.88 | 0.85  | 0.91 |
| (20,10,5)                                         | 0.71    | 0.78 | 0.74  | 0.83 | 0.77      | 0.82 | 0.78  | 0.85 | 0.82         | 0.88 | 0.84  | 0.90 |
| (20,20,10,5)                                      | 0.71    | 0.75 | 0.74  | 0.81 | 0.77      | 0.81 | 0.78  | 0.84 | 0.82         | 0.86 | 0.84  | 0.90 |

Table 4: Fidelity over  $\mathbb{D}_{\text{test}}$  and  $\mathbb{D}_{\text{uni}}$  for COMPAS dataset

| Target $\rightarrow$<br>$\mathbb{D}_{\text{test}}$ | (20,10) |      |       |      | (20,10,5) |      |       |      | (20,20,10,5) |      |       |      |
|----------------------------------------------------|---------|------|-------|------|-----------|------|-------|------|--------------|------|-------|------|
|                                                    | n=100   |      | n=200 |      | n=100     |      | n=200 |      | n=100        |      | n=200 |      |
| Surrogate $\downarrow$                             | Base.   | CCA  | Base. | CCA  | Base.     | CCA  | Base. | CCA  | Base.        | CCA  | Base. | CCA  |
| (20,10)                                            | 0.93    | 0.96 | 0.94  | 0.97 | 0.92      | 0.94 | 0.94  | 0.96 | 0.94         | 0.96 | 0.95  | 0.97 |
| (20,10,5)                                          | 0.92    | 0.95 | 0.94  | 0.97 | 0.92      | 0.93 | 0.95  | 0.95 | 0.94         | 0.96 | 0.95  | 0.97 |
| (20,20,10,5)                                       | 0.92    | 0.95 | 0.92  | 0.97 | 0.84      | 0.91 | 0.89  | 0.94 | 0.92         | 0.94 | 0.94  | 0.96 |

| Target $\rightarrow$<br>$\mathbb{D}_{\text{uni}}$ | (20,10) |      |       |      | (20,10,5) |      |       |      | (20,20,10,5) |      |       |      |
|---------------------------------------------------|---------|------|-------|------|-----------|------|-------|------|--------------|------|-------|------|
|                                                   | n=100   |      | n=200 |      | n=100     |      | n=200 |      | n=100        |      | n=200 |      |
| Surrogate $\downarrow$                            | Base.   | CCA  | Base. | CCA  | Base.     | CCA  | Base. | CCA  | Base.        | CCA  | Base. | CCA  |
| (20,10)                                           | 0.94    | 0.95 | 0.94  | 0.96 | 0.95      | 0.95 | 0.95  | 0.96 | 0.96         | 0.95 | 0.96  | 0.96 |
| (20,10,5)                                         | 0.93    | 0.95 | 0.94  | 0.95 | 0.94      | 0.92 | 0.95  | 0.92 | 0.95         | 0.96 | 0.96  | 0.96 |
| (20,20,10,5)                                      | 0.93    | 0.94 | 0.94  | 0.95 | 0.94      | 0.85 | 0.94  | 0.90 | 0.95         | 0.92 | 0.95  | 0.94 |

Table 5: Fidelity over  $\mathbb{D}_{\text{test}}$  and  $\mathbb{D}_{\text{uni}}$  for HELOC dataset

| Target $\rightarrow$<br>$\mathbb{D}_{\text{test}}$ | (20,10) |      |       |      | (20,10,5) |      |       |      | (20,20,10,5) |      |       |      |
|----------------------------------------------------|---------|------|-------|------|-----------|------|-------|------|--------------|------|-------|------|
|                                                    | n=100   |      | n=200 |      | n=100     |      | n=200 |      | n=100        |      | n=200 |      |
| Surrogate $\downarrow$                             | Base.   | CCA  | Base. | CCA  | Base.     | CCA  | Base. | CCA  | Base.        | CCA  | Base. | CCA  |
| (20,10)                                            | 0.90    | 0.94 | 0.91  | 0.95 | 0.90      | 0.94 | 0.92  | 0.95 | 0.98         | 0.99 | 0.98  | 0.99 |
| (20,10,5)                                          | 0.88    | 0.92 | 0.92  | 0.95 | 0.89      | 0.92 | 0.92  | 0.95 | 0.98         | 0.98 | 0.98  | 0.99 |
| (20,20,10,5)                                       | 0.87    | 0.93 | 0.91  | 0.93 | 0.87      | 0.89 | 0.91  | 0.94 | 0.98         | 0.98 | 0.98  | 0.98 |

| Target $\rightarrow$<br>$\mathbb{D}_{\text{uni}}$ | (20,10) |      |       |      | (20,10,5) |      |       |      | (20,20,10,5) |      |       |      |
|---------------------------------------------------|---------|------|-------|------|-----------|------|-------|------|--------------|------|-------|------|
|                                                   | n=100   |      | n=200 |      | n=100     |      | n=200 |      | n=100        |      | n=200 |      |
| Surrogate $\downarrow$                            | Base.   | CCA  | Base. | CCA  | Base.     | CCA  | Base. | CCA  | Base.        | CCA  | Base. | CCA  |
| (20,10)                                           | 0.92    | 0.92 | 0.94  | 0.95 | 0.91      | 0.91 | 0.94  | 0.95 | 0.98         | 0.98 | 0.99  | 0.99 |
| (20,10,5)                                         | 0.91    | 0.90 | 0.94  | 0.93 | 0.91      | 0.89 | 0.93  | 0.94 | 0.97         | 0.97 | 0.98  | 0.99 |
| (20,20,10,5)                                      | 0.91    | 0.91 | 0.93  | 0.94 | 0.91      | 0.87 | 0.93  | 0.92 | 0.97         | 0.97 | 0.98  | 0.98 |

### B.2.6 Studying alternate counterfactual generating method

Counterfactuals can be generated such that they satisfy additional desirable properties such as actionability, sparsity and closeness to the data manifold, other than the proximity to the original instance. In this experiment, we observe how counterfactuals with above properties affect the attack performance. HELOC is used as the dataset. Target model has the architecture (20, 30, 10) and the architecture of the surrogate model is (10, 20).

To generate actionable counterfactuals, we use Diverse Counterfactual Explanations (DiCE) by Mothilal et al. [2020] with the first four features, i.e., “estimate\_of\_risk”, “months\_since\_last\_trade”, “average\_duration\_of\_resolution”, and “number\_of\_satisfactory\_trades” kept unchanged. The diversity factor of DiCE generator is set to 1 in order to obtain only a single counterfactual for each query. Sparse counterfactuals are obtained by the same MCCF generator used in other experiments, but now with  $L_1$  norm as the cost function  $c(\mathbf{x}, \mathbf{w})$ . Counterfactuals from the data manifold (i.e., realistic counterfactuals, denoted by 1-NN) are generated using a 1-Nearest-Neighbor algorithm. We use ROAR [Upadhyay et al., 2021] and C-CHVAE [Pawelczyk et al., 2020] to generate robust counterfactuals. Table 2 summarizes the performance of the attack. Fig. 8 shows the distribution of the counterfactuals generated using each method w.r.t. the decision boundary of the target model. We observe that the sparse, realistic, and robust counterfactuals have a tendency to lie farther away from the decision boundary, within the favorable region, when compared to the closest counterfactuals under  $L_2$  norm.

### B.2.7 Comparison with DualCFX Wang et al. [2022]

Wang et al. [2022] is one of the few pioneering works studying the effects of counterfactuals on model extraction, which proposes the interesting idea of using counterfactuals of counterfactuals to mitigate the decision boundary shift. This requires the API to provide counterfactuals for queries originating from both sides of the decision boundary. However, the primary focus of our work is on the one-sided scenario where an institution might be giving counterfactuals only to the rejected applicants to help them get accepted, but not to the accepted ones. Hence, a fair comparison cannot be achieved between CCA and the strategy proposed in Wang et al. [2022] in the scenario where only one-sided counterfactuals are available.

Therefore, in the two-sided scenario, we compare the performance of CCA with the DualCFX strategy proposed in Wang et al. [2022] under two settings:

1. only one sided counterfactuals are available for CCA (named CCA1)
2. CCA has all the data that DualCFX has (named CCA2)

We also include another baseline (following Aïvodji et al. [2020]) for the two-sided scenario where the models are trained only on query instances and counterfactuals, but not the counterfactuals of the counterfactuals. Results are presented in Table 6. Note that even for the same number of initial

query instances, the total number of actual training instances change with the strategy being used (CCA1 < Baseline < DualCFX = CCA2 – e.g.: queries+CFs for the baseline but queries+CFs+CCFs for DualCFX).

Table 6: Comparison with DualCFX. Legend: Base.=Baseline model based on [Aïvodji et al., 2020], Dual=DualCFX, CCA1=CCA with one-sided counterfactuals, CCA2=CCA with counterfactuals from both sides.

| Dataset | Query size | Architecture known (model 0) |       |      |      |                           |       |      |      |
|---------|------------|------------------------------|-------|------|------|---------------------------|-------|------|------|
|         |            | $\mathbb{D}_{\text{test}}$   |       |      |      | $\mathbb{D}_{\text{uni}}$ |       |      |      |
|         |            | Base.                        | Dual. | CCA1 | CCA2 | Base.                     | Dual. | CCA1 | CCA2 |
| DCCC    | n=100      | 0.95                         | 0.99  | 0.94 | 0.99 | 0.90                      | 0.95  | 0.92 | 0.97 |
|         | n=200      | 0.96                         | 0.99  | 0.98 | 0.99 | 0.90                      | 0.96  | 0.95 | 0.98 |
| HELOC   | n=100      | 0.94                         | 0.97  | 0.90 | 0.98 | 0.91                      | 0.98  | 0.84 | 0.98 |
|         | n=200      | 0.96                         | 0.98  | 0.92 | 0.98 | 0.93                      | 0.98  | 0.89 | 0.99 |

| Dataset | Query size | Architecture unknown (model 1) |       |      |      |                           |       |      |      |
|---------|------------|--------------------------------|-------|------|------|---------------------------|-------|------|------|
|         |            | $\mathbb{D}_{\text{test}}$     |       |      |      | $\mathbb{D}_{\text{uni}}$ |       |      |      |
|         |            | Base.                          | Dual. | CCA1 | CCA2 | Base.                     | Dual. | CCA1 | CCA2 |
| DCCC    | n=100      | 0.92                           | 0.98  | 0.93 | 0.98 | 0.88                      | 0.92  | 0.89 | 0.93 |
|         | n=200      | 0.96                           | 0.99  | 0.96 | 0.99 | 0.89                      | 0.94  | 0.94 | 0.96 |
| HELOC   | n=100      | 0.92                           | 0.91  | 0.90 | 0.96 | 0.88                      | 0.92  | 0.84 | 0.96 |
|         | n=200      | 0.95                           | 0.92  | 0.91 | 0.97 | 0.93                      | 0.94  | 0.88 | 0.97 |

## B.2.8 Studying other machine learning models

We explore the effectiveness of the proposed attack when the target model is no longer a neural network classifier. The surrogate models are still neural networks with the architectures (20, 10) for model 0 and (20, 10, 5) for model 1. A random forest classifier with 100 estimators and a linear regression classifier, trained on Adult Income dataset are used as the targets. Ensemble size  $S$  used is 20. Results are shown in Fig. 16, where the proposed attack performs better or similar to the baseline attack.

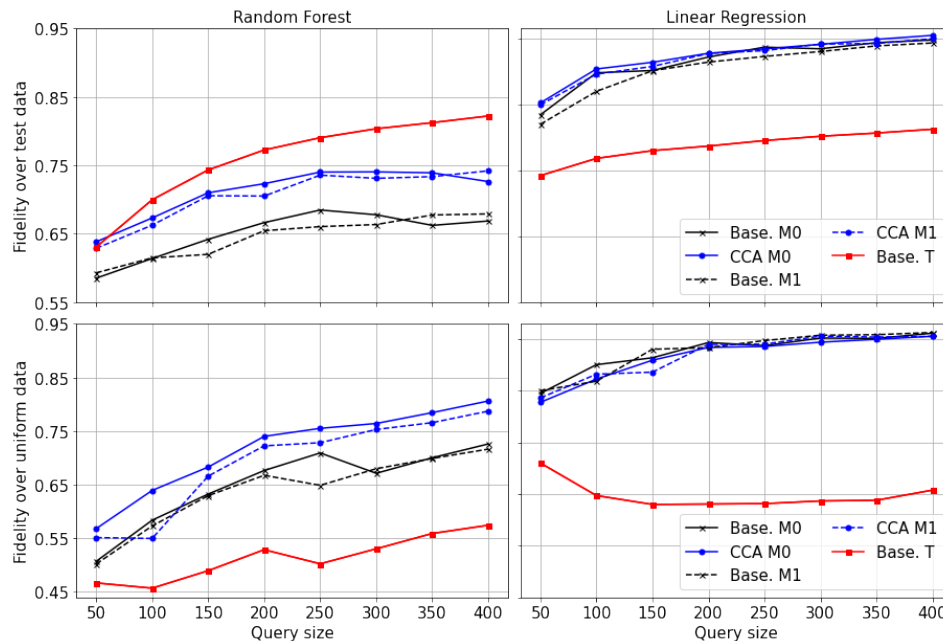


Figure 16: Performance of the attack when the target model is not a neural network. Surrogates M0 and M1 are neural networks with the architectures (20,10) and (20,10,5) respectively. Baseline T is a surrogate model from the same class as the target model.

### B.2.9 Studying effect of unbalanced $\mathbb{D}_{\text{attack}}$

In all the other experiments, the attack dataset  $\mathbb{D}_{\text{attack}}$  used by the adversary is sampled from a class-wise balanced dataset. In this experiment we explore the effect of querying using an unbalanced  $\mathbb{D}_{\text{attack}}$ . Model architectures used are (20, 10) for the target model and surrogate model 0, and (20, 10, 5) for surrogate model 1. While the training set of the teacher and the test set of both the teacher and the surrogates were kept constant, the proportion of the samples in the attack set  $\mathbb{D}_{\text{attack}}$  was changed. In the first case, examples from class  $y = 1$  were dominant (80%) and in the second case, the majority of the examples were from class  $y = 0$  (80%). The results are shown in Fig. 17.

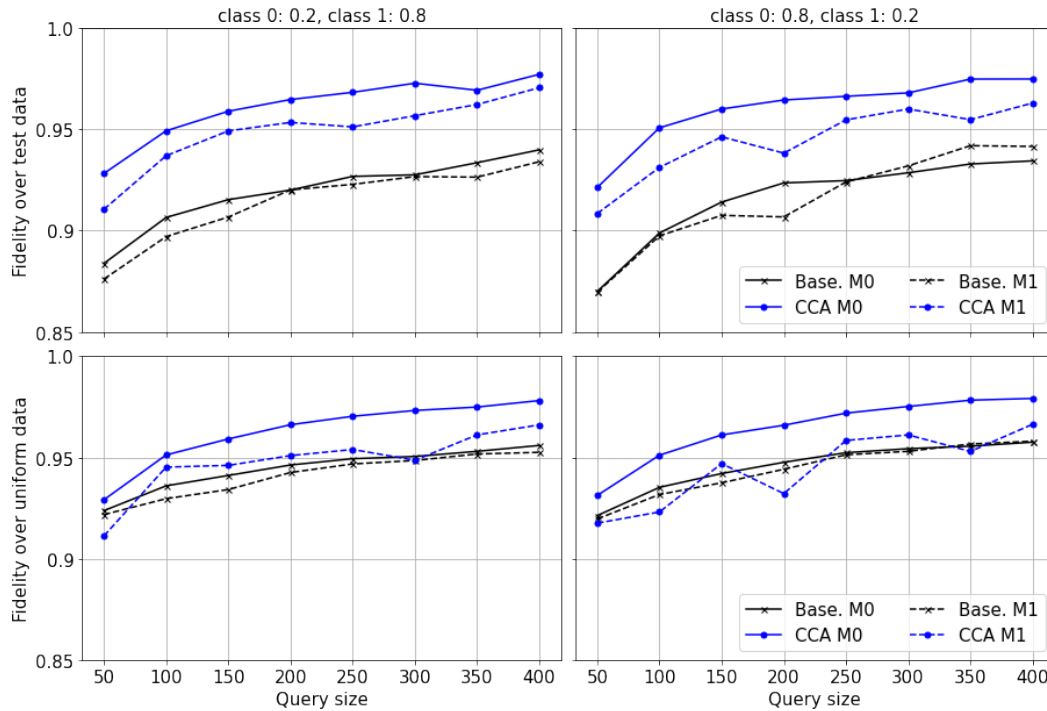


Figure 17: Results corresponding to the HELOC dataset with queries sampled from biased versions of the dataset (i.e., a biased  $\mathbb{D}_{\text{attack}}$ ). The version on the left uses a  $\mathbb{D}_{\text{attack}}$  with 20% and 80% examples from classes  $y = 0$  and  $y = 1$ , respectively. The version on the right was obtained with a  $\mathbb{D}_{\text{attack}}$  comprising of 80% and 20% examples from classes  $y = 0$  and  $y = 1$ , respectively.

### B.2.10 Studying alternate loss functions

We explore using binary cross-entropy loss function directly with labels 0, 1 and 0.5 in place of the proposed loss. Precisely, the surrogate loss is now defined as

$$L(\tilde{n}, y) = -y(\mathbf{x}) \log(\tilde{n}(\mathbf{x})) - (1 - y(\mathbf{x})) \log(1 - \tilde{n}(\mathbf{x})) \quad (32)$$

which is symmetric around 0.5 for  $y(\mathbf{x}) = 0.5$ . Two surrogate models are observed, with architectures (20, 10) for model 0 and (20, 10, 5) for model 1. The target model's architecture is similar to that of model 0. The ensemble size is  $S = 20$ .

The results (in Fig. 18) indicate that the binary cross-entropy loss performs worse than the proposed loss. The reason might be the following: As the binary cross-entropy loss is symmetric around 0.5 for counterfactuals, it penalizes the counterfactuals that are farther inside the favorable region. This in turn pulls the surrogate decision boundary towards the favorable region more than necessary, causing a decision boundary shift.

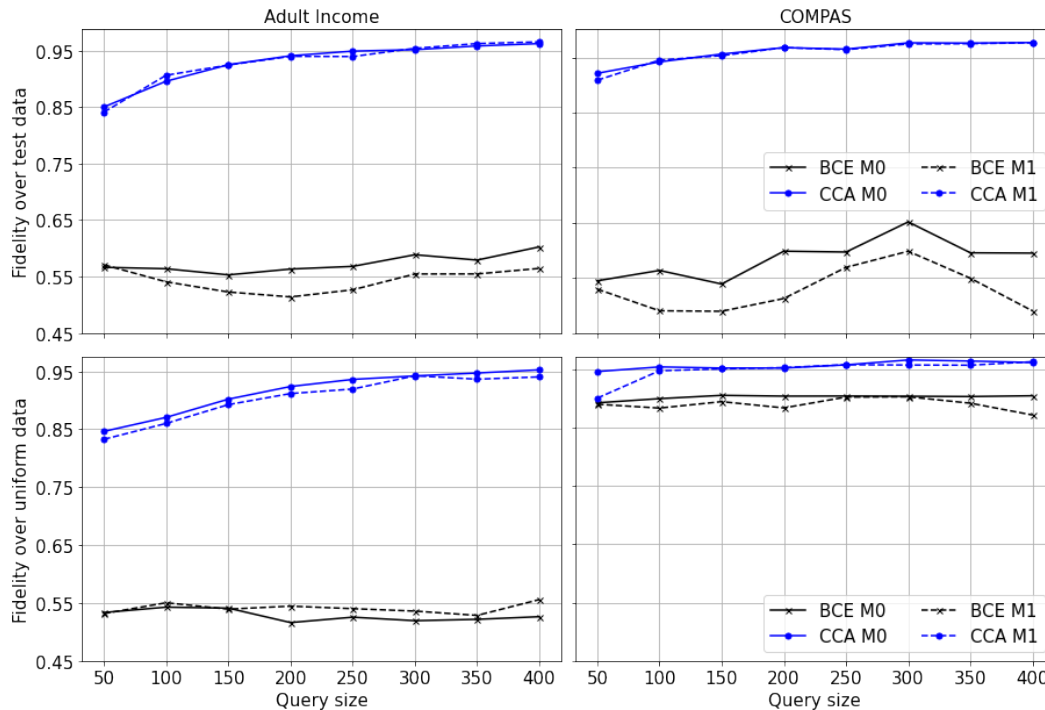


Figure 18: Performance of binary cross-entropy loss with labels 0, 0.5 and 1. Black lines corresponding to binary cross entropy (BCE) loss and blue lines depict the performance of the CCA loss.

### B.3 Experiments for verifying Theorem 3.2

This experiment includes approximating a spherical decision boundary in the first quadrant of a  $d$ -dimensional space. The decision boundary is a portion of a sphere with radius 1 and the origin at  $(1, 1, \dots, 1)$ . The input space is assumed to be normalized, and hence, restricted to the unit hypercube. See Section 3.1 for a description of the attack strategy. Fig. 19 presents a visualization of the experiment in the case where the dimensionality  $d = 2$ . Fig. 20 presents a comparison of theoretical and empirical query complexities for higher dimensions. Experiments agree with the theoretical upper-bound.

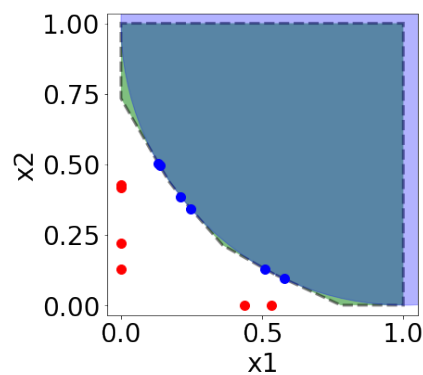


Figure 19: Synthetic attack for verifying Theorem 3.2 in the 2-dimensional case. Red dots represent queries and blue dots are the corresponding closest counterfactuals. Dashed lines indicate the boundary of the polytope approximation.

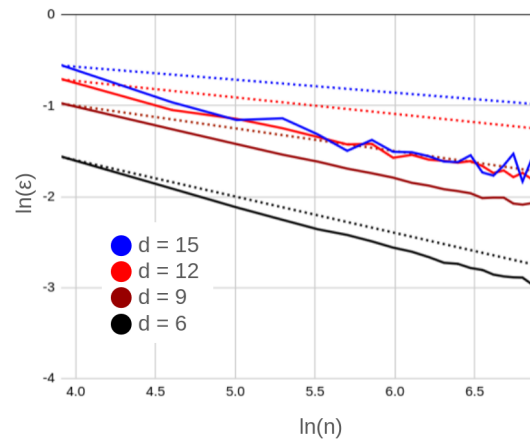


Figure 20: Verifying Theorem 3.2: Dotted and solid lines indicate the theoretical and empirical rates of convergence.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: Abstract and introduction correctly summarize and highlight the main claims made.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Assumptions of each theorem are specified on the spot. Paper also includes a discussion on limitations and future work in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.

- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[Yes\]](#)

Justification: Assumptions corresponding to each theoretical result are stated immediately before the theorems. Proofs for all the results are presented in the appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: Details pertaining to all the experiments, including model architectures, datasets, querying strategy, no. of training epochs are presented in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case



of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: While all the datasets used in the experiments are publicly available, we have made the code public as a Github repository, of which the link is given in the Introduction.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Details pertaining to each experiment are given in the corresponding sections of the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In the main experiments supporting the claim, we report standard deviations or variance along with the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: While the GPU models on which the experiments were done are listed in the appendix, compute times have not been reported.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Implications of the theoretical results on privacy of the models are discussed next to the theorems. Appendix contains a separate section on the broader impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This research does not release any data or models.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [\[Yes\]](#)

Justification: All publicly available datasets that were used in the experiments have been cited.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[NA\]](#)

Justification: Paper does not release any assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[NA\]](#)

Justification: Paper does not include experiments involving crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Paper does not include experiments involving crowdsourcing or human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.