Exploring Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis

Martin Pawelczyk[1] **Chirag Agarwal**[2,3] **Shalmali Joshi**[3]
Sohini Upadhyay[3] **Himabindu Lakkaraju**[3]
[1]University of Tübingen [2]Adobe Research [3]Harvard University

Abstract

As machine learning (ML) models become more widely deployed in high-stakes applications, counterfactual explanations have emerged as key tools for providing actionable model explanations in practice. Despite the growing popularity of counterfactual explanations, a deeper understanding of these explanations is still lacking. In this work, we systematically analyze counterfactual explanations through the lens of adversarial examples. We do so by formalizing the similarities between popular counterfactual explanation and adversarial example generation methods identifying conditions when they are equivalent. We then derive the upper bounds on the distances between the solutions output by counterfactual explanation and adversarial example generation methods, which we validate on several real world data sets. By establishing these theoretical and empirical similarities between counterfactual explanations and adversarial examples, our work raises fundamental questions about the design and development of existing counterfactual explanation algorithms.

1 INTRODUCTION

With the increasing use of Machine learning (ML) models in critical domains, such as health care and law enforcement, it becomes important to ensure that their decisions are robust and explainable. To this end, several approaches have been proposed in recent literature to explain the complex behavior of ML models (Simonyan et al., 2013; Ribeiro et al., 2016; Lundberg and Lee, 2017; Sundararajan et al., 2017). One such popular class of explanations designed to provide recourse to individuals adversely impacted by algorithmic decisions are *counterfactual explanations* (Wachter et al., 2017; Ustun et al., 2019; Barocas et al., 2020; Venkatasubramanian and Alfano, 2020). For example, in a credit scoring model where an individual loan application is denied, a counterfactual explanation can highlight the minimal set of changes the individual can make to obtain a positive outcome (Pawelczyk et al., 2020a; Karimi et al., 2020c). Algorithms designed to output counterfactual explanations often attempt to find a closest counterfactual for which the model prediction is positive (Wachter et al., 2017; Ustun et al., 2019; Pawelczyk et al., 2020a; Karimi et al., 2020c).

Adversarial examples, on the other hand, were proposed to highlight how vulnerabilities of ML models can be exploited by (malicious) adversaries (Szegedy et al., 2013; Ballet et al., 2019; Cartella et al., 2021). These adversarial examples are usually also obtained by finding minimal perturbations to a given data instance such that the model prediction changes (Goodfellow et al., 2014; Carlini and Wagner, 2017; Moosavi-Dezfooli et al., 2016).

Conceptually, adversarial examples and counterfactual explanations solve a similar optimization problem (Freiesleben, 2020; Wachter et al., 2017; Cartella et al., 2021). Techniques generating adversarial examples and counterfactual explanations use distance or norm constraints in the objective function to enforce the notion of minimal perturbations. While adversarial methods generate instances that are semantically indistinguishable from the original instance, counterfactual explanations or algorithmic recourse[1], encourage minimal changes to an input so that so that a

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

[1]Note that counterfactual explanations, contrastive explanations, and recourses are used interchangeably in prior literature. Counterfactual/contrastive explanations serve as a means to provide recourse to individuals with unfavorable algorithmic decisions. We use these terms interchangeably as introduced and defined by Wachter et al. (2017).

user can readily act upon these changes to obtain the desired outcome. In addition, some methods in both lines of work use manifold-based constraints to find natural adversarial examples (Zhao et al., 2018) or realistic counterfactual explanations by restricting them to lie on the data manifold (Joshi et al., 2019; Pawelczyk et al., 2020a,b).

While the rationale of producing a counterfactual close to the original instance is motivated by the desideratum that counterfactuals should be actionable and easily understandable, producing close instances on the other side of the decision boundary could just as easily indicate adversarial activity. This begs the question to what extent do counterfactual explanation algorithms return solutions that resemble adversarial examples. However, there has been little to no work on systematically analyzing the aforementioned connections between the literature on counterfactual explanations and adversarial examples.

Present Work. In this work, we approach the study of similarities between counterfactual explanations and adversarial examples from the perspective of counterfactual explanations for algorithmic recourse. Therefore, we consider consequential decision problems (e.g., loan applications) commonly employed in recourse literature and our choices of data modalities (i.e., tabular data) and algorithms are predominantly motivated by this literature. In particular, we make one of the first attempts at establishing theoretical and empirical connections between state-of-the-art counterfactual explanation and adversarial example generation methods.

More specifically, we analyze these similarities by bounding the distances between the solutions of salient counterfactual explanation and popular adversarial example methods for locally linear approximations. Our analysis demonstrates that several popular counterfactual explanation and adversarial example generation methods such as the ones proposed by Wachter et al. (2017) and Carlini and Wagner (2017); Moosavi-Dezfooli et al. (2016), are equivalent for certain hyperparameter choices. Moreover, we demonstrate that C-CHVAE and the natural adversarial attack (NAE) (Zhao et al., 2018) provide similar solutions for certain generative model choices.

Finally, we carry out extensive experimentation with multiple synthetic and real-world data sets from diverse domains such as financial lending and criminal justice to validate our theoretical findings. We further probe these methods empirically to validate the similarity between the counterfactuals and adversarial examples output by several state-of-the-art methods. Our results indicate that counterfactuals and adversarial examples output by manifold-based methods such as NAE and C-CHVAE are more similar compared to those generated by other techniques. By establishing these and other theoretical and empirical similarities, our work raises fundamental questions about the design and development of existing counterfactual explanation algorithms.

2 RELATED WORK

This work lies at the intersection of counterfactual explanations and adversarial examples in machine learning. Below we discuss related work for each of these topics and their connection.

Adversarial examples. Adversarial examples are obtained by making infinitesimal perturbations to input instances such that they force a ML model to generate adversary-selected outputs. Algorithms designed to successfully generate these examples are called Adversarial attacks (Szegedy et al., 2013; Goodfellow et al., 2014). Several attacks have been proposed in recent literature depending on the degree of knowledge/access of the model, training data, and optimization techniques. While gradient-based methods (Goodfellow et al., 2014; Kurakin et al., 2016; Moosavi-Dezfooli et al., 2016) find the minimum ℓ_p-norm perturbations to generate adversarial examples, generative methods (Zhao et al., 2018) constrain the search for adversarial examples to the training data-manifold. Finally, some methods (Cisse et al., 2017) generate adversarial examples for non-differentiable and non-decomposable measures in complex domains such as speech recognition and image segmentation. We refer to a well-established survey for a more comprehensive overview of adversarial examples (Akhtar and Mian, 2018).

Counterfactual explanations. Counterfactual explanation methods aim to provide explanations for a model prediction in the form of minimal changes to an input instance that changes the original prediction (Wachter et al., 2017; Ustun et al., 2019; Van Looveren and Klaise, 2019; Karimi et al., 2020b). These methods are categorized based on the access to the model (or gradients), sparsity of the generated explanation and whether the generated explanations are constrained to the manifold (Verma et al., 2020; Karimi et al., 2020b). To this end, Wachter et al. (2017) proposed a gradient-based method to obtain counterfactual explanations for models using a distance-based penalty and finding the nearest counterfactual explanation. Further, restrictions on attributes such as race, age, and gender are generally enforced to ensure that the output counterfactual explanations are realistic for users to act on them. In addition, manifold-based constraints are imposed in many

methods (Pawelczyk et al., 2020a; Joshi et al., 2019) so that the counterfactual explanations are faithful to the data distribution. Finally, causal approaches have recently been proposed to generate counterfactual explanations that adhere to causal constraints (Karimi et al., 2020a; Barocas et al., 2020; Karimi et al., 2021, 2020c).

Connections between adversarial examples and counterfactual explanations. Conceptual connections between adversarial examples and counterfactual explanations have been previously identified in the literature (Freiesleben, 2020; Browne and Swift, 2020). While Freiesleben (2020) highlight conceptual differences in aims, formulation and use-cases between both sub-fields suggesting that counterfactual explanations represent a broader class of examples of which adversarial examples represent a subclass, Browne and Swift (2020) focus on discussing the differences *w.r.t* semantics hidden layer representations of DNNs. Our goal, on the other hand, is to theoretically formalize and empirically analyze the (dis)similarity between these fields.

3 PRELIMINARIES

Notation. We denote a classifier $h : \mathcal{X} \to \mathcal{Y}$ mapping features $\mathbf{x} \in \mathcal{X}$ to labels $\mathcal{Y}$. Further, we define $h(\mathbf{x}) = g(f(\mathbf{x}))$, where $f : \mathcal{X} \to \mathbb{R}$ is a scoring function (*e.g.,* logits) and $g : \mathbb{R} \to \mathcal{Y}$ an activation function that maps output logit scores to discrete labels. Below we describe some representative methods used in this work to generate counterfactual explanations and adversarial examples.

3.1 Counterfactual explanation methods

Counterfactual explanations provide recourses by identifying which attributes to change for reversing a models' adverse outcome. Methods designed to output counterfactual explanations find a counterfactual $\mathbf{x}'$ that is "closest" to the original instance $\mathbf{x}$ and changes the models' prediction $h(\mathbf{x}')$ to the desired label. While several of these methods incorporate distance metrics (*e.g.,* ℓ_p-norm) or user preferences (Rawal and Lakkaraju, 2020) to find the desired counterfactuals, some efforts also impose causal (Karimi et al., 2020c) or data manifold constraints (Joshi et al., 2019; Pawelczyk et al., 2020a,b) to find realistic counterfactuals. We now describe counterfactual explanation methods from two broad categories: 1) Gradient- (Wachter et al., 2017) and 2) search-based (Pawelczyk et al., 2020a).

Score CounterFactual Explanations (SCFE). For a given classifier h and the corresponding scoring function f, and a distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, Wachter et al. (2017) formulate the problem of finding a counterfactual $\mathbf{x}'$ for $\mathbf{x}$ as:

$$\arg\min_{\mathbf{x}'} \ (f(\mathbf{x}') - s)^2 + \lambda\, d(\mathbf{x}, \mathbf{x}'), \tag{1}$$

where s is the target score for $\mathbf{x}$ and λ is set to iteratively increase until $f(\mathbf{x}')=s$. More specifically, to arrive at a counterfactual probability of 0.5, the target score for $g(\mathbf{x})$ for a sigmoid function is $s=0$, where the logit corresponds to a 0.5 probability for $y=1$.

C-CHVAE. Let I_γ and $\mathcal{G}_\theta$ denote the encoder and decoder of the VAE model used by C-CHVAE (Pawelczyk et al., 2020a) to generate realistic counterfactuals. Note that the counterfactuals for $\mathbf{x}$ are generated in the latent space of the encoder $\mathcal{Z}$, where $I_\gamma : \mathcal{X} \to \mathcal{Z}$. Let $\mathbf{z}$ and $\tilde{\mathbf{z}} = \mathbf{z}+\delta$ denote the latent representation and generated counterfactuals for the original instance $\mathbf{x}$. Intuitively, $\mathcal{G}_\theta$ is a generative model that projects the latent counterfactuals to the feature space and I_γ allows to search for counterfactuals in the data manifold. Thus, the objective function is defined as follows:

$$\begin{aligned} \delta^* &= \arg\min_{\delta \in \mathcal{Z}} \|\delta\| \\ &\text{s.t. } h(\mathcal{G}_\theta(I_\gamma(\mathbf{x}^f) + \delta), \mathbf{x}^p) \neq h(\mathbf{x}^f, \mathbf{x}^p), \end{aligned} \tag{2}$$

where $\mathbf{x}^p$ and $\mathbf{x}^f$ indicate the protected and non-protected features of $\mathbf{x}$ and Eqn. 2 finds the minimal perturbation δ by changing the non-protected features $\mathbf{x}^m$ constrained to the data-manifold.

3.2 Adversarial example generation methods

Similar to counterfactual explanation methods, most methods generating adversarial examples also solve a constrained optimization problem to find perturbations in the input manifold that cause models to misclassify. These methods are broadly categorized into *poisoning* (e.g., Shafahi et al. (2018)) and *exploratory* (e.g., Goodfellow et al. (2014)) methods. While *poisoning* methods attack the model during training and attempts to learn, influence, and corrupt the underlying training data or model, *Exploratory* methods do not tamper with the underlying model but instead generate specific examples that cause the model to produce the desired output. Like counterfactual explanation methods, evasion methods also use gradient-based optimization to generate adversarial examples. Below, we briefly outline three evasion techniques considered in this work.

C&W Attack. For a given input $\mathbf{x}$ and classifier $h(\cdot)$, Carlini and Wagner (2017) formulate the problem of finding an adversarial example $\mathbf{x}'=\mathbf{x}+\delta$ such that $h(\mathbf{x}') \neq h(\mathbf{x})$ as:

$$\arg\min_{\mathbf{x}'} c \cdot \ell(\mathbf{x}') + d(\mathbf{x}, \mathbf{x}') \ \text{ s.t. } \ \mathbf{x}' \in [0,1]^d \tag{3}$$

where c is a hyperparameter and $\ell(\cdot)$ is a loss function such that $h(\mathbf{x}')=y$ if and only if $\ell(\mathbf{x}') \leq 0$. The constraint $\mathbf{x}' \in [0,1]^d$ is applied so that the resulting $\mathbf{x}'$ is within a given data range.

DeepFool. For a given instance $\mathbf{x}$, DeepFool (Moosavi-Dezfooli et al., 2016) perturbs $\mathbf{x}$ by adding small perturbation δ_{DF} at each iteration of the algorithm. The perturbations from each iterations are finally aggregated to generate the final perturbation once the output label changes. The minimal perturbation to change the classification model's prediction is the solution to the following objective:

$$\begin{aligned} \delta^*_{\text{DF}}(\mathbf{x}) = \arg\min_{\delta} \; & ||\delta||_2 \\ \text{s.t. } & \text{sign}(f(\mathbf{x}+\delta)) \neq \text{sign}(f(\mathbf{x})), \end{aligned} \tag{4}$$

where $\mathbf{x}$ is the input sample. The closed-form step for each iteration is: $\delta^*_{\text{DF}} = -(f(\mathbf{x})/||\nabla f(\mathbf{x})||_2^2)\nabla f(\mathbf{x})$.

Natural Adversarial Example (NAE). Similar to C-CHVAE, Zhao et al. (2018) proposes NAE to search for adversarial examples using a generative model $\mathcal{G}_\theta$ where the similarity is measured in the latent space of $\mathcal{G}_\theta$. Thus, the objective is given by:

$$\mathbf{z}^* = \arg\min_{\tilde{\mathbf{z}} \in \mathcal{Z}} \|\tilde{\mathbf{z}} - I_\gamma(\mathbf{x})\| \text{ s.t. } h(\mathcal{G}_\theta(\tilde{\mathbf{z}})) \neq h(\mathbf{x}), \tag{5}$$

where $I_\gamma(\mathbf{x})$ corresponds to the latent representation of $\mathbf{x}$ and $\mathcal{G}_\theta(\tilde{\mathbf{z}})$ maps the latent sample to the feature space. NAE separately trains an inverter function from $\mathcal{G}_\theta$ by enforcing the latent representation to be normally distributed (*i.e.,* corresponding to the noise model of the generator) while minimizing the reconstruction error of the feature space.

4 THEORETICAL ANALYSIS

In this section, we provide theoretical connections between counterfactual explanation and adversarial example methods by leveraging similarities in the objective functions and optimization procedures. In particular, we compare: 1) SCFE and C&W (Sec. 4.1), 2) SCFE and DeepFool (Sec. 4.2), and 3) C-CHVAE and NAE (Sec. 4.3) due to their similarity in the objective functions. We do these comparisons either for a specific loss, solutions based on the classification model, or constraints imposed during optimization. We focus on locally linear model approximations as these are often studied as a first step (Hardt and Ma, 2017; Ustun et al., 2019; Rosenfeld et al., 2020; Garreau and Luxburg, 2020) towards understanding nonlinear model behaviour.

4.1 SCFE and C&W

Two popular gradient-based methods for generating adversarial and counterfactual samples are the C&W Attack and SCFE, respectively. Here, we first show the closed-form solutions for the minimum perturbation required by C&W (δ^*_{CW}) and SCFE (δ^*_{SCFE}) to generate adversarial examples and counterfactuals. We then leverage these solutions to derive an upper bound for the distance between the adversarial and counterfactual samples. Using the loss function $\ell^*(\cdot) = \max(0, \max_i(f(\mathbf{x})_i) - f(\mathbf{x})_y)$ recommended by Carlini and Wagner (2017), we derive an upper bound for the distance between the counterfactuals and adversarial examples generated using SCFE and C&W. For the upper bound, we first state a lemma that derives the closed-form solution for δ^*_{SCFE}.

Lemma 1. *(Optimal Counterfactual Perturbation) For a scoring function with weights* $\mathbf{w}$ *the SCFE method generates a counterfactual* $\mathbf{x}_{SCFE}$ *for an input* $\mathbf{x}$ *using the counterfactual perturbation* δ^*_{SCFE} *such that:*

$$\delta^*_{SCFE} = m \cdot (\mathbf{w}\mathbf{w}^\mathbf{T} + \lambda \mathbf{I})^{-1}\mathbf{w}, \tag{6}$$

where s *is the target score for* $\mathbf{x}$, $m = s - f(\mathbf{x})$ *is the target residual,* $f(x) = \mathbf{w}^\top \mathbf{x} + b$ *is a local linear score approximation, and* λ *is a given hyperparameter.*

Proof Sketch. We derive the closed-form solution for δ^*_{SCFE} by formulating the SCFE objective in its vector quadratic form. See Appendix B.1 for the complete proof. □

Using Lemma 1, we now formally state and derive the upper bound for the distance between the counterfactuals and adversarial examples.

Theorem 1. *(Difference between SCFE and C&W) Under the same conditions as stated in Lemma 1, the normed difference between the SCFE counterfactual* $\mathbf{x}_{SCFE}$ *and C&W adversarial example* $\mathbf{x}_{CW}$ *using the loss function* $\ell^*(\cdot)$ *is upper bounded by:*

$$\begin{aligned} &\|\mathbf{x}_{SCFE} - \mathbf{x}_{CW}\|_p \\ &\leq \left\| \frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \|\mathbf{w}\|_2^2}\right)(s - f(\mathbf{x})) - c\mathbf{I} \right\|_p ||\mathbf{w}||_p. \end{aligned} \tag{7}$$

Proof Sketch. We first derive the closed-form solution for the perturbation used by C&W. Intuitively, this solution is equivalent to shifting $\mathbf{x}$ in the direction of the models' decision boundary scaled by c. The upper bound follows by applying Lemma 1 and Cauchy-Schwartz inequality. Moreover, choosing the hyperparameter such that $\lambda \to 0$ and setting $c = m/\|\mathbf{w}\|_2^2$ yields equivalence, *i.e.,* $||\mathbf{x}_{\text{SCFE}} - \mathbf{x}_{\text{CW}}||_p \to 0$. See Appendix B.3 for the complete proof. □

We note that the upper bound is smaller when the original score $f(\mathbf{x})$ is close to the target score s, sug-

gesting that $\mathbf{x}_{\text{SCFE}}$ and $\mathbf{x}_{\text{CW}}$ are more similar when $\mathbf{x}$ is closer to the decision boundary.

4.2 SCFE and DeepFool

DeepFool is an adversarial attack that uses an iterative gradient-based optimization approach to generate adversarial examples. Despite the differences in the formulations of SCFE and DeepFool, our theoretical analysis reveals a striking similarity between the two methods. In particular, we provide an upper bound for the distance between the solutions output by counterfactuals and adversarial examples generated using SCFE and DeepFool, respectively.

Theorem 2. *(Difference between SCFE and DeepFool) Under the same conditions as stated in Lemma 1, the normed difference between the SCFE counterfactual $\mathbf{x}_{SCFE}$ and the DeepFool adversarial example $\mathbf{x}_{DF}$ is upper bounded by:*

$$\|\mathbf{x}_{SCFE} - \mathbf{x}_{DF}\|_p \leq \left\| \left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \|\mathbf{w}\|_2^2} \right) \frac{(s - f(\mathbf{x}))}{\lambda} + \mathbf{I} \frac{f(\mathbf{x})}{\|\mathbf{w}\|_2^2} \right\|_p \|\mathbf{w}\|_p. \tag{8}$$

Proof Sketch. We show the similarity between SCFE and DeepFool methods by comparing their closed-form solutions for the generated counterfactual and adversarial examples. Similar to Theorem 1, the results follow from Cauchy-Schwartz inequality, (see Appendix B.4 for the complete proof). Moreover, choosing the hyperparameter such that $\lambda \to 0$ and setting $s{:=}0$ yields equivalence, *i.e.,* $||\mathbf{x}_{\text{SCFE}} - \mathbf{x}_{\text{DF}}||_p \to 0$. □

The right term in the inequality (Eqn. 8) entails that the l_p-norm of the difference between the generated samples is bounded if: 1) the predicted score is closer to the target score of a given input, and 2) the gradients with respect to the logit scores of the underlying model are bounded.

4.3 Manifold-based methods

We formalize the connection between manifold-based methods by comparing NAE to C-CHVAE as both rely on generative models. While C-CHVAE uses variational autoencoders, NAE uses GANs, specifically Wasserstein GAN (Arjovsky et al., 2017), to generate adversarial example. To allow a fair comparison, we assume that both methods use the same generator $\mathcal{G}_\theta$ and inverter I_γ networks.

Proposition 1. *Let $p{=}\emptyset$ in C-CHVAE. Assuming that C-CHVAE and NAE use the same generator $\mathcal{G}_\theta$ and inverter functions $\mathcal{I}_\theta$. Then the proposed objectives of NAE and C-CHVAE are equivalent.*

Proof. Since $p{=}\emptyset$, equation 2 reduces to:

$$\delta^* {=} \arg\min_{\delta \in \mathcal{Z}} \|\delta\| \text{ s.t. } h(\mathcal{G}_\theta(I_\gamma(\mathbf{x}^f) + \delta)) \neq h(\mathbf{x}^f) \tag{9}$$

Also, $I_\gamma(\mathbf{x}){=}\mathbf{z}$. Replacing $\tilde{\mathbf{z}}{-}\mathbf{z}{=}\delta$ in eqn. 5, we get:

$$\delta^* = \arg\min_{\delta \in \mathcal{Z}} \|\delta\| \text{ s.t. } h(\mathcal{G}_\theta(I_\gamma(\mathbf{x}) + \delta)) \neq h(\mathbf{x}) \tag{10}$$

Since $\mathbf{x}^f{=}\mathbf{x}$, we get the equivalence. □

Both C-CHVAE and NAE use search methods to generate adversarial examples or counterfactuals using the above objective function. In particular, both NAE and C-CHVAE samples $\mathbf{z}$ using an ℓ_p-norm ball of radius range $(r_{\text{NAE}}, \Delta r_{\text{NAE}}]$ and r_{C}. $\tilde{\mathbf{z}}_{\text{NAE}}$ denotes the solution returned by Zhao et al. (2018) and $\tilde{\mathbf{z}}_{\text{C}}$ the solution returned by C-CHVAE. We denote r^*_{NAE} and r^*_{C} as the corresponding radius parameters from NAE and C-CHVAE, respectively, and restrict our analysis to the class of L-Lipschitz generative models:

Definition 1. *Bora et al. (2017): A generative model $\mathcal{G}_\theta(\cdot)$ is L-Lipschitz if $\forall\ \mathbf{z}_1,\ \mathbf{z}_2 \in \mathcal{Z}$, we have,*

$$\|\mathcal{G}_\theta(\mathbf{z}_1) - \mathcal{G}_\theta(\mathbf{z}_2)\|_p \leq L\|\mathbf{z}_1 - \mathbf{z}_2\|_p. \tag{11}$$

Note that commonly used DNN models comprise of linear, convolutional and activation layers, which satisfy Lipschitz continuity (Gouk et al., 2021).

Lemma 2. *(Difference between C-CHVAE and NAE) Let $\tilde{\mathbf{z}}_C$ and $\tilde{\mathbf{z}}_{NAE}$ be the output generated by C-CHVAE and NAE by sampling from ℓ_p-norm ball in the latent space using an L-Lipschitz generative model $\mathcal{G}_\theta(\cdot)$. Analogously, let $\mathbf{x}_{NAE}{=}\mathcal{G}_\theta(\tilde{\mathbf{z}}_{NAE})$ and $\mathbf{x}_C{=}\mathcal{G}_\theta(\tilde{\mathbf{z}}_C)$ generate perturbed samples by design of the two methods. Let r^*_{NAE} and r^*_C be the corresponding radii chosen by each algorithm such that they successfully return an adversarial example or counterfactual. Then, $\|\mathbf{x}_C - \mathbf{x}_{NAE}\|_p \leq L(r^*_C + r^*_{NAE})$.*

Proof Sketch. The proof follows from triangle inequality, L-Lipschitzness of the generative model, and the fact that the ℓ_p-norm of the method's outputs are known in the latent space. See Appendix B.5 for a detailed proof. □

Intuitively, the adversarial example and counterfactual explanation generated by the methods are bounded depending on the data manifold properties (captured by the Lipschitzness of the generative model) and the radius hyperparameters used by the search algorithms.

4.4 On the Underlying Assumptions

The analyzed counterfactual explanation objectives in our analysis slightly differ from their original implementations. Hence, the following remarks are in place. First, the original objective stated by Wachter et al. (2017) includes the median absolute deviation (MAD) for each input as a normalization term for the regularizer. However, the regularizer is independent of δ, and can therefore be incorporated into Lemma 1 and Theorem 1. Second, our analysis focuses on objective functions from SCFE and C-CHVAE explanation methods which consider generic distance functions. Hence, to facilitate a fair comparison across all counterfactual explanation and adversarial example algorithms, we use ℓ_2-norm in our analysis. Finally, immutability constraints can be readily incorporated. For instance, instead of taking the derivative of the SCFE objective function with respect to all available features, we take the derivative with respect to the mutable features only.

5 EXPERIMENTS

We now present the empirical analysis to demonstrate the similarities between counterfactual explanations and adversarial examples. More specifically, we verify the validity of our theoretical upper bounds using real-world datasets and determine the extent to which counterfactual explanations and adversarial examples similar to each other.

5.1 Experimental Setup

We first describe the synthetic and real-world datasets used to study the connections between counterfactual explanations and adversarial examples, and then we outline our experimental setup.

Synthetic Data. We generate 5000 samples from a mixture of Gaussians with pdfs $\mathcal{N}(\mu_1{=}[1.0, 1.0], \Sigma_1{=}\mathbf{I})$ and $\mathcal{N}(\mu_2{=}[-1.0, -1.0], \Sigma_2{=}\mathbf{I})$.

Real-world Data. We use three datasets in our experiments. 1) The *UCI Adult* dataset (Dua and Graff, 2017) consisting of 48842 individuals with demographic (*e.g.,* age, race, and gender), education (degree), employment (occupation, hours-per-week), personal (marital status, relationship), and financial (capital gain/loss) features. The task is to predict whether an individual's income exceeds \$50K per year or not. 2) The *COMPAS* dataset (Mattu et al., 2016) comprising of 10000 individuals representing defendants released on bail. The task is to predict whether to release a defendant on bail or not using features, such as criminal history, jail, prison time, and defendant's demographics. 3) The *German Credit* dataset from the UCI repository (Dua and Graff, 2017) consisting of demographic (age, gender), personal (marital status), and financial (income, credit duration) features from 1000 credit applications. The task is to predict whether an applicant qualifies for credit or not.

Methods. Following our analysis in Sec. 4, we compare the following pair of methods: i) SCFE (Wachter et al., 2017) vs. C&W (Carlini and Wagner, 2017), ii) SCFE vs. DeepFool (Moosavi-Dezfooli et al., 2016), and iii) C-CHVAE (Pawelczyk et al., 2020a) vs. NAE (Zhao et al., 2018).

Prediction Models. For the synthetic dataset, we train a logistic regression model (LR) to learn the mixture component (samples and corresponding decision boundary shown in Fig. 1), whereas for real-world datasets, we obtain adversarial examples and counterfactuals using LR and artificial neural network (ANN) models. See Appendix C for more details.

Implementation Details For all real-world data, adversarial examples and counterfactuals are generated so as to flip the target prediction label from unfavorable (y=0) to favorable (y=1). We use ℓ_2-norm as the distance function in all our experiments. We partition the dataset into train-test splits where the training set is used to train the predictor models. Adversarial examples and counterfactuals are generated for the trained models using samples in the test splits. For counterfactual explanation methods applied to generate recourse, all features are assumed actionable for fair comparison with adversarial example generation methods. See Appendix C for more implementation details.

5.2 Results

Validating our Theoretical Upper Bounds. We empirically validate the theoretical upper bounds obtained in Sec. 4. To this end, we first estimate the bounds for each instance in the test set according to Theorems 1 and 2, and compare them with the empirical estimates of the ℓ_2-norm differences (LHS of Theorems 1 and 2). We use the same procedure to validate the bounds from Lemma 3.

SCFE vs. C&W and DeepFool. In Fig. 2, we show the empirical evaluation of our theoretical bounds for all real-world datasets. For each dataset, we show four box-plots: empirical estimates (green) and theoretical upper bounds (blue) of the distance (ℓ_2-norm) between the resulting counterfactuals and adversarial examples for SCFE and C&W (labeled as SCFE vs. CW), and SCFE and DeepFool (labeled as SCFE vs. DF). Across all three datasets, we observe that no bounds were violated for both theorems. The gap between empirical and theoretical values is relatively small for German credit dataset as compared to COMPAS and Adult datasets. From Theorems 1 and 2, we see that the bound strongly depends on

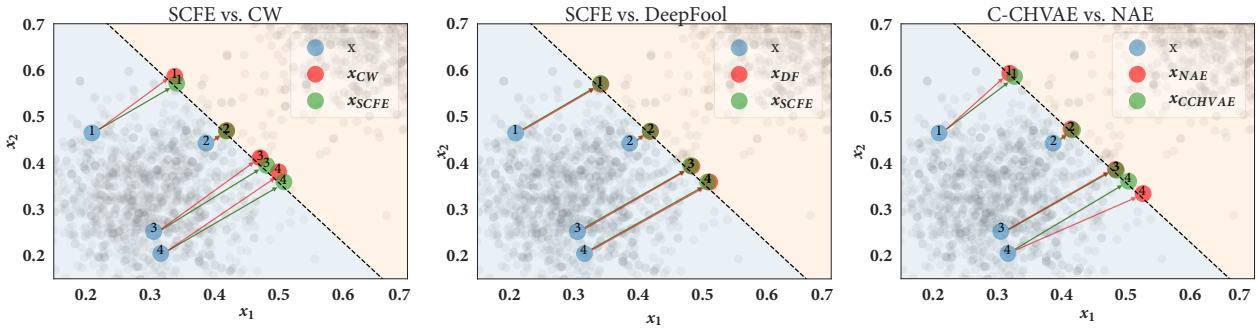
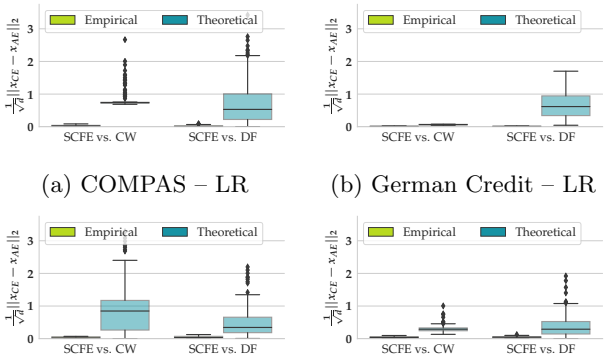


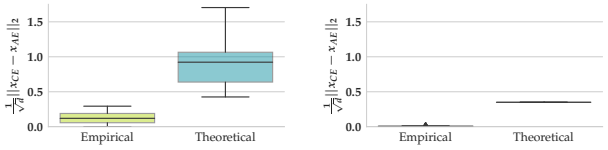
Figure 1: Similarity comparison of adversarial example and counterfactual explanation methods. Based on synthetic data, we generate adversarial examples (in red) and counterfactual explanations (in green) for some randomly chosen test set points (in blue) using methods described in Sec. 3. (**Left**) Both SCFE (in green) and C&W (in red) samples are close to each other, indicating strong similarity between these methods. (**Middle**) SCFE (in green) and DeepFool (in red) samples exactly coincide, indicating equivalence. (**Right**) C-CHVAE (in green) and NAE (in red) samples are closer if the blue factual points are closer to the boundary.



(a) COMPAS – LR (b) German Credit – LR

(c) COMPAS – ANN (d) German Credit – ANN

Figure 2: Verifying the theoretical bounds from Theorems 1 and 2. The green boxplots correspond to the empirical norm differences between SCFE (i.e., x_{CE}) and CW or DF (i.e., x_{AE}). The blue boxplots show the distribution of upper bounds, which we evaluated by plugging in the necessary quantities (hyperparameters, gradients, logit values) into equations 7 and 8. No bounds are violated. For ANNs, the upper bounds were computed using local linear model approximations.



(a) Adult – ANN (b) COMPAS – ANN

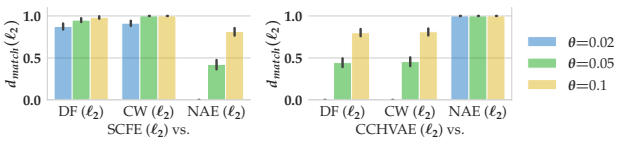
Figure 3: Verifying the theoretical bounds from Lemma 2. The green boxplots correspond to the empirical norm differences between CCHVAE (i.e., x_{CE}) and NAE (i.e., x_{AE}). The blue boxplots show the distribution of upper bounds, which we evaluated by plugging in the corresponding quantities (hyperparameters, Lipschitz constant) into the upper bound from Lemma 2. The Lipschitz constant is computed based on decoders and encoders using Lemma 4. No bounds are violated.

the norm of the logit score gradient $w{=}\nabla_x f(\mathbf{x})$, *e.g.,* for Adult dataset these norms are relatively higher leading to less tight bounds.

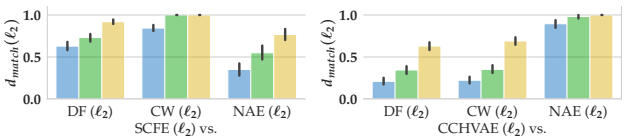
C-CHVAE vs.NAE. In Fig. 3, we validate the bounds obtained in Lemma 3 for all three datasets using an encoder-decoder framework. We observe that our upper bounds are tight, thus validating our theoretical analysis for comparing manifold-based counterfactual explanation (C-CHVAE) and adversarial example generation method (NAE).

Similarities between Counterfactuals and Adversarial examples. Here, we qualitatively and quantitatively show the similarities between counterfactuals and adversarial examples using several datasets.

Analysis with Synthetic Data. In Fig. 1, we show the similarity between counterfactual explanations and adversarial examples generated for a classifier trained on a two-dimensional mixture of Gaussian datasets. Across all cases, we observe that most output samples generated by counterfactual explanation and adversarial example methods overlap. In particular, for samples near the decision boundary, the solutions tend to be more similar. These results confirm our theoretical bounds, which depend on the difference between the logit sample prediction $f(\mathbf{x})$ and the target score s. If points are close to the decision boundary, $f(\mathbf{x})$ is closer to s, suggesting that the resulting counterfactual and adversarial example



(a) COMPAS



(b) Adult

Figure 4: Analyzing to what extent different counterfactual explanation methods and adversarial example generation methods are empirically equivalent for the logistic regression classifier. To do that, we compute d_{match} from Eqn. 12. Missing bars indicate that there was no match.

will be closer as implied by Theorems 1 and 2.

Analysis with Real Data. For real-world datasets, we define two additional metrics beyond those studied in our theoretical analysis to gain a more granular understanding about the similarities of counterfactuals and adversarial examples. First, we introduce d_{match} which quantifies the similarity between counterfactuals (i.e., $\mathbf{x}_{\text{CE}}$) and adversarial examples (i.e., $\mathbf{x}_{\text{AE}}$):

$$d_{\text{match}} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left[\frac{1}{\sqrt{d}}\|\mathbf{x}_{\text{CE}}^{(i)} - \mathbf{x}_{\text{AE}}^{(i)}\|_2 < \theta\right], \qquad (12)$$

where n is the total number of instances used in the analysis and $\theta \in \{0.02, 0.05, 0.1\}$ is a threshold determining when to consider counterfactual and adversarial examples as equivalent. d_{match} evaluates whether counterfactuals and adversarial examples are exactly the same with higher d_{match} implying higher similarity. Second, we complement d_{match} by Spearman rank ρ between δ_{CE} and δ_{AE}, which is a rank correlation coefficient measuring to what extent the perturbations' rankings agree, i.e., whether adversarial example generation methods and counterfactual explanation methods deem the same dimensions important in order to arrive at their final outputs. Here, $\rho(\delta_{\text{CE}}, \delta_{\text{AE}})=1$ implies that the rankings are same, 0 suggests that the rankings are independent, and -1 indicates reversely ordered rankings.

In Fig. 4, we compare a given counterfactual explanation method to salient adversarial example generation methods (DeepFool, C&W, and NAE) using d_{match}. We show the results for Adult and COMPAS datasets using LR models and relegate results for German Credit as well as neural network classifiers to Appendix D. Our results in Fig. 4 validate that the SCFE method is similar to DeepFool and C&W (higher d_{match} for lower θ). Across all datasets, this result aligns and validates with the similarity analysis in Sec. 4. Similarly, manifold-based methods demonstrate higher d_{match} compared to non-manifold methods (right panels in Fig. 4). Additionally, we show the results from the rank correlation analysis in Table 1 and observe that the maximum rank correlations (between 0.90 and 1.00) are obtained for methods that belong to the same categories, suggesting that the considered counterfactuals and adversarial examples are close to being equivalent.

6 CONCLUSION

In this work, we formally analyzed the connections between state-of-the-art adversarial example generation methods and counterfactual explanation methods. To this end, we first highlighted salient counterfactual explanation and adversarial example methods in literature, and leveraged similarities in their objective functions, optimization algorithms and constraints utilized in these methods to theoretically analyze conditions for equivalence and bound the distance between the solutions output by counterfactual explanation and adversarial example generation methods. For locally linear models, we bound the distance between the solutions obtained by C&W and SCFE using loss functions preferred in the respective works. We obtained similar bounds for the solutions of DeepFool and SCFE. We also demonstrated equivalence between the manifold-based methods of NAE and C-CHVAE and bounded the distance between their respective solutions. Finally, we empirically evaluated our theoretical findings on simulated and real-world data sets.

By establishing theoretically and empirically that several popular counterfactual explanation algorithms are generating extremely similar solutions as those of well known adversarial example algorithms, our work raises fundamental questions about the design and development of existing counterfactual explanation algorithms. *Do we really want counterfactual explanations to resemble adversarial examples, as our work suggests they do? How can a decision maker distinguish an adversarial attack from a counterfactual explanation? Does this imply that decision makers are tricking their own models by issuing counterfactual explanations? Can we do a better job of designing counterfactual explanations?* Moreover, by establishing connections between popular counterfactual explanation and adversarial example algorithms, our work opens up the possibility of using insights from adversarial robustness literature to improve the design and development of counterfactual explanation algorithms.

We hope our formal analysis helps carve a path for more robust approaches to counterfactual explana-

Table 1: Average Spearman rank correlation between counterfactual and adversarial perturbations. For every input $\mathbf{x}$, we compute the corresponding adversarial perturbation δ_{AE} and the counterfactual perturbation δ_{CE}. We then compute Spearman's $\rho(\delta_{\text{AE}}, \delta_{\text{CE}})$ and report their means (gradient-based: (g); manifold-based: (m)).

	COMPAS				Adult			
	LR		ANN		LR		ANN	
Model	SCFE (g)	CCHVAE (m)	SCFE (g)	CCHVAE (m)	SCFE (g)	CCHVAE (m)	SCFE (g)	CCHVAE (m)
CW (g)	0.88 ± 0.16	0.67 ± 0.30	0.93 ± 0.10	0.67 ± 0.22	$\mathbf{0.95 \pm 0.06}$	0.86 ± 0.10	0.92 ± 0.09	0.70 ± 0.16
DF (g)	$\mathbf{0.91 \pm 0.12}$	0.68 ± 0.31	$\mathbf{0.97 \pm 0.03}$	0.65 ± 0.22	0.92 ± 0.06	0.80 ± 0.13	$\mathbf{0.93 \pm 0.08}$	0.63 ± 0.20
NAE (m)	0.57 ± 0.35	$\mathbf{0.94 \pm 0.08}$	0.71 ± 0.19	$\mathbf{1.00 \pm 0.00}$	0.83 ± 0.12	$\mathbf{0.90 \pm 0.10}$	0.74 ± 0.13	$\mathbf{0.98 \pm 0.02}$

tions, a critical aspect for calibrating trust in ML. Improving our theoretical bounds using other strategies and deriving new theoretical bounds for other approaches is an interesting future direction.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful feedback. This work is supported in part by the NSF awards #IIS-2008461 and #IIS-2040989, and research awards from the Harvard Data Science Institute, Amazon, Bayer, and Google. HL would like to thank Mohan and Sujatha Lakkaraju, and Pracheer Gupta for all their inputs and support. The views expressed are those of the authors and do not reflect the official policy or position of the funding agencies.

References

Akhtar, N. and Mian, A. (2018). Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. arxiv 2017. *arXiv preprint arXiv:1701.07875*, 30.

Ballet, V., Renard, X., Aigrain, J., Laugel, T., Frossard, P., and Detyniecki, M. (2019). Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274*.

Barocas, S., Selbst, A. D., and Raghavan, M. (2020). The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 80–89.

Biggio, B., Nelson, B., and Laskov, P. (2012). Poisoning attacks against support vector machines. In *ICML*.

Bora, A., Jalal, A., Price, E., and Dimakis, A. G. (2017). Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR.

Browne, K. and Swift, B. (2020). Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks. *arXiv preprint arXiv:2012.10076*.

Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE.

Cartella, F., Anunciacao, O., Funabiki, Y., Yamaguchi, D., Akishita, T., and Elshocht, O. (2021). Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *arXiv preprint arXiv:2101.08030*.

Cisse, M., Adi, Y., Neverova, N., and Keshet, J. (2017). Houdini: Fooling deep structured prediction models. *arXiv preprint arXiv:1707.05373*.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Freiesleben, T. (2020). Counterfactual explanations & adversarial examples–common grounds, essential differences, and potential transfers. *arXiv preprint arXiv:2009.05487*.

Garreau, D. and Luxburg, U. (2020). Explaining the explainer: A first theoretical analysis of lime. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1287–1296. PMLR.

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. (2021). Regularisation of neural networks by enforcing lipschitz continuity. Springer.

Hardt, M. and Ma, T. (2017). Identity matters in deep learning. In *International Conference on Learning Representations (ICLR)*.

Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., and Ghosh, J. (2019). Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615*.

Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. (2020a). Model-agnostic counterfactual explanations for consequential decisions. In *International*

Conference on Artificial Intelligence and Statistics (AISTATS), pages 895–905. PMLR.

Karimi, A.-H., Barthe, G., Schölkopf, B., and Valera, I. (2020b). A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *arXiv preprint arXiv:2010.04050*.

Karimi, A.-H., Schölkopf, B., and Valera, I. (2021). Algorithmic recourse: from counterfactual explanations to interventions. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 353–362.

Karimi, A.-H., von Kügelgen, J., Schölkopf, B., and Valera, I. (2020c). Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. *arXiv preprint arXiv:2006.06831*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kurakin, A., Goodfellow, I., Bengio, S., et al. (2016). Adversarial examples in the physical world.

Lundberg, S. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

Mattu, S., Kirchner, L., and Angwin, J. (2016). How we analyzed the compas recidivism algorithm. *ProPublica*.

Moosavi-Dezfooli, S.-M., Fawzi, A., and Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582.

Pawelczyk, M., Bielawski, S., van den Heuvel, J., Richter, T., and Kasneci, G. (2021). Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms. *arXiv preprint arXiv:2108.00783*.

Pawelczyk, M., Broelemann, K., and Kasneci, G. (2020a). Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020*, pages 3126–3132.

Pawelczyk, M., Broelemann, K., and Kasneci, G. (2020b). On counterfactual explanations under predictive multiplicity. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, volume 124.

Rawal, K. and Lakkaraju, H. (2020). Interpretable and interactive summaries ofactionable recourses. *arXiv preprint arXiv:2009.07165*.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Rosenfeld, E., Winston, E., Ravikumar, P., and Kolter, Z. (2020). Certified robustness to label-flipping attacks via randomized smoothing. In *International Conference on Machine Learning*, pages 8230–8241. PMLR.

Shafahi, A., Huang, W. R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., and Goldstein, T. (2018). Poison frogs! targeted clean-label poisoning attacks on neural networks. In *NeurIPS*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Sundararajan, M., Taly, A., and Yan, Q. (2017). Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pages 3319–3328. PMLR.

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

Tavallali, P., Behzadan, V., Tavallali, P., and Singhal, M. (2021). Adversarial poisoning attacks and defense for general multi-class models based on synthetic reduced nearest neighbors. *arXiv*.

Ustun, B., Spangher, A., and Liu, Y. (2019). Actionable recourse in linear classification. *Proceedings of the Conference on Fairness, Accountability, and Transparency*.

Van Looveren, A. and Klaise, J. (2019). Interpretable counterfactual explanations guided by prototypes. *arXiv preprint arXiv:1907.02584*.

Venkatasubramanian, S. and Alfano, M. (2020). The philosophical basis of algorithmic recourse. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAT*)*, pages 284–293.

Verma, S., Dickerson, J., and Hines, K. (2020). Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*.

Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. In *Harvard Journal & Technology*.

Zhao, Z., Dua, D., and Singh, S. (2018). Generating natural adversarial examples. In *International Conference on Learning Representations (ICLR)*.

Supplementary Material: Exploring Counterfactual Explanations Through the Lens of Adversarial Examples: A Theoretical and Empirical Analysis

APPENDIX SUMMARY

Section A provides a categorization of counterfactual explanation and adversarial example methods. In Section B, we provide detailed proofs for Lemmas 1 and 3, and Theorems 1 and 2. In Section C, we provide implementation details for all models used in our experiments including (i) the supervised learning models, (ii) the counterfactual explanation and adversarial example methods, and the (iii) generative models required to run the manifold-based methods. Finally, in Section D, we present the remaining experiments we referred to in the main text.

A TAXONOMY OF COUNTERFACTUAL AND ADVERSARIAL EXAMPLE METHODS

In order to choose methods to compare across counterfactual explanation methods and adversarial example generation methods, we surveyed existing literature. We use a taxonomy to categorize each subset of methods based on various factors. The main characteristics we use are based on type of method, based on widely accepted terminology and specific implementation details. In particular, we distinguish between i) constraints imposed for generating adversarial examples or counterfactual explanations, ii) algorithms used for generating them. For the class of adversarial example generation methods, we further distinguish between *poisoning attacks* and *evasion attacks* and note that evasion attacks are most closely related to counterfactual explanation methods. The taxonomy for counterfactual explanation methods is provided in Table 2 and that for adversarial example generation methods is provided in Table 3.

The main algorithm types used for counterfactual explanation methods are search-based, gradient-based and one method that uses integer programming (Ustun et al., 2019). The main constraints considered are actionability i.e., only certain features are allowed to change, and counterfactual explanations are encouraged to be realistic using either causal and/or manifold constraints. Similarly, for adversarial example generation methods primarily, Greedy search-based and gradient-based methods are most common. Manifold constraints are also imposed in a few cases where the goal is to generate adversaries close to the data-distribution. Based on this taxonomy, we select the appropriate pairs of counterfactual explanation method and adversarial example generation method to compare to each other for theoretical analysis. This leads us to compare gradient-based methods SCFE and C&W attack, SCFE and DeepFool and finally, manifold-based methods C-CHVAE and NAE with their search-based algorithms.

Table 2: Taxonomy of counterfactual explanation methods

Algorithm	Constraints	Method
Search-based	Causal, Actionability	**MINT** (Karimi et al., 2020c)
	Manifold, Actionability	**C-CHVAE** (Pawelczyk et al., 2020a)
Gradient-based	Actionability	**CFE, SCFE** Wachter et al. (2017)
	Manifold, Actionability	**REVISE** (Joshi et al., 2019)
Integer-programming	Actionability/Linear black-box	**AR** (Ustun et al., 2019)

Table 3: Taxonomy of adversarial example generation methods

	Algorithm	Constraints	Method
Poisoning Attacks	Greedy Search Gradient-based	Manifold Data-domain	**Adv. Data Poisoning** (Tavallali et al., 2021) **SVM-attack** (Biggio et al., 2012) **One-Shot Kill** (Shafahi et al., 2018)
Evasion Attacks	Search-based Gradient-based	Manifold Data-domain	**NAE** (Zhao et al., 2018) **DeepFool** (Moosavi-Dezfooli et al., 2016) **C&W Attack** (Carlini and Wagner, 2017)

B PROOFS OF SECTION 4

B.1 Proof of Lemma 1

Lemma 1. *For a linear score function $f(x) = \mathbf{w}^\top \mathbf{x} + b$, the **SCFE** counterfactual for $\mathbf{x}$ on f is $\mathbf{x}' = \mathbf{x} + \delta^*$ where*

$$\delta^* = (\mathbf{w}\mathbf{w}^\mathbf{T} + \lambda \mathbf{I})^{-1}(s - \mathbf{w}^T\mathbf{x} - b)\mathbf{w}.$$

Proof. Reformulating Equation 1 using l_2-norm as the distance metric, we get:

$$\min_{\mathbf{x}'}(\mathbf{w}^T\mathbf{x}' + b - s)^2 + \lambda||\mathbf{x}' - \mathbf{x}||_2^2.$$

We can convert this minimization objective into finding the minimum perturbation δ by substituting $\mathbf{x}' = \mathbf{x} + \delta$, *i.e.,*

$$\min_{\delta}(\mathbf{w}^T\mathbf{x} + \mathbf{w}^T\delta + b - s)^2 + \lambda||\mathbf{x}' - \mathbf{x}||_2^2. \tag{13}$$

Using $s - \mathbf{w}^T\mathbf{x} - b = m$ as a dummy variable and $\mathbf{x}' - \mathbf{x} = \delta$, we get:

$$\begin{aligned}
&\min_{\delta}(\mathbf{w}^T\delta - m)^2 + \lambda||\delta||_2^2 \\
&\min_{\delta}(\mathbf{w}^T\delta - m)^T(\mathbf{w}^T\delta - m) + \lambda\delta^T\delta \\
&\min_{\delta}(\delta^T\mathbf{w} - m)(\mathbf{w}^T\delta - m) + \lambda\delta^T\delta && (m \text{ is a scalar, hence } m^T = m) \\
&\min_{\delta}\delta^T\mathbf{w}\mathbf{w}^T\delta - 2m\delta^T\mathbf{w} + m^2 + \lambda\delta^T\delta \\
&\min_{\delta}\delta^T(\mathbf{w}\mathbf{w}^T + \lambda\mathbf{I})\delta - 2m\delta^T\mathbf{w} + m^2 \\
&\min_{\delta}\delta^T\mathbf{M}\delta - 2m\mathbf{w}^T\delta + m^2 && (\text{where } \mathbf{M} = \mathbf{w}\mathbf{w}^T + \lambda\mathbf{I}) \\
&\min_{\delta}\delta^T\mathbf{M}\delta - 2\eta^T\delta + m^2 && (\text{where } m\mathbf{w} = \eta) \\
&\min_{\delta}\delta^T\mathbf{M}\delta - 2\eta^T\delta + \eta^T\mathbf{M}^{-1}\eta - \eta^T\mathbf{M}^{-1}\eta + m^2 \\
&\min_{\delta}(\delta - \mathbf{M}^{-1}\eta)^T\mathbf{M}(\delta - \mathbf{M}^{-1}\eta) - \eta^T\mathbf{M}^{-1}\eta + m^2
\end{aligned}$$

The closed form solution is given by,

$$\delta^* = \mathbf{M}^{-1}\eta, \tag{14}$$

where $M = \mathbf{w}\mathbf{w}^T + \lambda\mathbf{I}$.

The expression in equation 14 can further be simplified:

$$\begin{aligned}
\delta^* &= \frac{m}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \|\mathbf{w}\|_2^2}\right)\mathbf{w} && (\text{Sherman-Morrison Formula}) \\
&= \frac{m}{\lambda}\left(\mathbf{I}\mathbf{w} - \mathbf{w}\frac{\|\mathbf{w}\|_2^2}{\lambda + \|\mathbf{w}\|_2^2}\right) \\
&= \frac{m}{\lambda}\cdot\frac{\lambda}{\lambda + \|\mathbf{w}\|_2^2}\cdot\mathbf{w} = \frac{m}{\lambda + \|\mathbf{w}\|_2^2}\cdot\mathbf{w},
\end{aligned} \tag{15}$$

where $m := s - \mathbf{w}^T\mathbf{x} - b$. Finally, we note that as $\lambda \to 0$, we have:

$$\delta^{**} = \frac{m}{\|\mathbf{w}\|_2^2} \cdot \mathbf{w}. \tag{16}$$

□

B.2 Proof of Lemma 3

Lemma 3. *For a binary classifier $h(\mathbf{x}) = g(f(\mathbf{x}))$ such that $f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$, $g(\mathbf{x}) = \sigma(\mathbf{x})$, and $h(\mathbf{x})$ is the probability that $\mathbf{x}$ is in the class $y = 1$,*

$$\ell^*(\mathbf{x}) = \max(0, -2(\mathbf{w}^\top\mathbf{x} + b))$$

.

Proof. Given our formulation of $h(\mathbf{x})$, $f(\mathbf{x})$ is the score corresponding to class $y = 1$. By the definition of $\sigma(\mathbf{x})$,

$$f(\mathbf{x}) = \ln \frac{h(\mathbf{x})}{1 - h(\mathbf{x})} = \ln h(\mathbf{x}) - \ln(1 - h(\mathbf{x}))$$

Then the score corresponding to the class $y = 0$ is

$$\ln \frac{1 - h(\mathbf{x})}{1 - (1 - h(\mathbf{x}))} = \ln \frac{1 - h(\mathbf{x})}{h(\mathbf{x})} = \ln(1 - h(\mathbf{x})) - \ln h(\mathbf{x}) = -f(\mathbf{x})$$

Substituting back into definition of $\ell^*(\mathbf{x})$,

$$\begin{aligned}
\ell^*(\mathbf{x}) &= \max(0, \max_i(f(\mathbf{x})_i) - f(\mathbf{x})_y) \\
&= \max(0, (-f(\mathbf{x}) - f(\mathbf{x})) \\
&= \max(0, (-2f(\mathbf{x})) \\
&= \max(0, -2(\mathbf{w}^\top\mathbf{x} + b)).
\end{aligned}$$

□

B.3 Proof of Theorem 1

Theorem 1. *For a linear classifier $h(\mathbf{x}) = g(f(\mathbf{x}))$ such that $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, the difference between the **SCFE** counterfactual example $\mathbf{x}_{SCFE}$ and the **C&W** adversarial example $\mathbf{x}_{CW}$ using the recommended loss function $\ell^*(\cdot) = \max(0, \max_i(f(\mathbf{x})_i) - f(\mathbf{x})_y)$ is given by:*

$$\|\mathbf{x}_{SCFE} - \mathbf{x}_{CW}\|_p \leq \left\| \frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) - c\mathbf{I} \right\|_p \|\mathbf{w}\|_p$$

.

Proof. Consider a binary classifier $h(\mathbf{x}) = g(f(\mathbf{x}))$ such that $f(\mathbf{x}) = \mathbf{w}^\top\mathbf{x} + b$, $g(\mathbf{x}) = \sigma(\mathbf{x})$, and $h(\mathbf{x})$ is the probability that $\mathbf{x}$ is in the class $y = 1$. Then by Lemma 3 and using ℓ_2-nrom as the distance metric, we can write the **C&W Attack** objective as

$$\arg\min_{\mathbf{x}'} c \max(0, -2(\mathbf{w}^\top\mathbf{x}' + b)) + \|\mathbf{x} - \mathbf{x}'\|_2^2$$

We can convert this minimization objective into finding the minimum perturbation δ by substituting $\mathbf{x}' = \mathbf{x} + \delta$,

$$\arg\min_{\delta} c \max(0, -2(\mathbf{w}^\top\mathbf{x} + \mathbf{w}^\top\delta + b)) + \|\delta\|_2^2$$

The subgradients of this objective are

$$\begin{cases} 2\delta & \text{when } -2(\mathbf{w}^\top\mathbf{x} + \mathbf{w}^\top\delta + b) < 0 \\ -2c\mathbf{w} + 2\delta & \text{otherwise} \end{cases}$$

By Lemma 3, $-2(\mathbf{w}^\top \mathbf{x} + \mathbf{w}^\top \delta + b) = -f(\mathbf{x}) - f(\mathbf{x}) < 0$. This implies that $f(\mathbf{x}) > -f(\mathbf{x})$, i.e that the score for class $y = 1$ is greater than the score for $y = 0$. As this indicates an adversarial example has already been found, we focus on minimizing the other subgradient. Setting this subgradient equal to 0,

$$0 = -2c\mathbf{w} + 2\delta$$
$$\delta = c\mathbf{w}$$

Thus the minimum perturbation to generate and adversarial example using the **C&W Attack** is

$$\delta^*_{\text{CW}} = c\mathbf{w}$$

Now, taking the difference between the minimum perturbation to generate a **SCFE** counterfactual (Lemma 1) and DeepFool (equation 18), we get:

$$\begin{aligned}
\delta^*_{\text{SCFE}} - \delta^*_{\text{CW}} &= (\mathbf{w}\mathbf{w}^T + \lambda\mathbf{I})^{-1}(s - \mathbf{w}^T\mathbf{x} - b)\mathbf{w} - c\mathbf{w} \\
&= ((\mathbf{w}\mathbf{w}^T + \lambda\mathbf{I})^{-1}(s - f(\mathbf{x})) - c\mathbf{I})\mathbf{w} \\
&= \left(\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) - c\mathbf{I}\right)\mathbf{w} && \text{(Using Sherman–Morrison formula)}
\end{aligned}$$

Taking the l_p-norm on both sides, we get:

$$\begin{aligned}
\|\delta^*_{\text{SCFE}} - \delta^*_{\text{CW}}\|_p &= \left\|\left(\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) - c\mathbf{I}\right)\mathbf{w}\right\|_p \\
&\leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) - c\mathbf{I}\right\|_p \|\mathbf{w}\|_p && \text{(Using Cauchy-Schwartz)}
\end{aligned}$$

Adding and subtracting the input instance $\mathbf{x}$ in the left term, we get:

$$\begin{aligned}
\|\mathbf{x} + \delta^*_{\text{SCFE}} - (\mathbf{x} + \delta^*_{\text{CW}})\|_p &\leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) - c\mathbf{I}\right\|_p \|\mathbf{w}\|_p \\
\|\mathbf{x}_{\text{SCFE}} - \mathbf{x}_{\text{CW}}\|_p &\leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) - c\mathbf{I}\right\|_p \|\mathbf{w}\|_p,
\end{aligned}$$

where the final equation gives an upper bound on the difference between the **SCFE** counterfactual and the **C&W** adversarial example.

Furthermore, we ask under which conditions the normed difference becomes 0. We start with:

$$\delta^*_{\text{SCFE}} - \delta^*_{\text{CW}} = \frac{m}{\lambda + \|\mathbf{w}\|_2^2} \cdot \mathbf{w} - c \cdot \mathbf{w}$$

Taking the l_p-norm on both sides, we get:

$$\|\delta^*_{\text{SCFE}} - \delta^*_{\text{CW}}\|_p = \left|\frac{m}{\lambda + \|\mathbf{w}\|_2^2} - c\right| \cdot \|\mathbf{w}\|_p$$

If we were to choose $\lambda \to 0$ we would get:

$$\|\delta^{**}_{\text{SCFE}} - \delta^*_{\text{CW}}\|_p = \left|\frac{m - c \cdot \|\mathbf{w}\|_2^2}{\|\mathbf{w}\|_2^2}\right| \cdot \|\mathbf{w}\|_p,$$

where equality holds when the hyperparameter is chosen so that $c := \frac{m}{\|\mathbf{w}\|_2^2}$. □

B.4 Proof of Theorem 2

Theorem 2. *For a linear classifier $h(\mathbf{x}) = g(f(\mathbf{x}))$ such that $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, the difference between the counterfactual example $\mathbf{x}_{SCFE}$ generated by Wachter et al. (2017) and the adversarial example $\mathbf{x}_{DF}$ generated by Moosavi-Dezfooli et al. (2016) is given by:*

$$||\mathbf{x}_{SCFE} - \mathbf{x}_{DF}||_p \leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) + \left(\mathbf{I}\frac{f(\mathbf{x})}{\|\mathbf{w}\|_2^2}\right)\right\|_p \cdot ||\mathbf{w}||_p, \tag{17}$$

Proof. The minimal perturbation to change the classifier's decision for a binary model $f(\mathbf{x})$ is given by the closed-form formula (Moosavi-Dezfooli et al., 2016):

$$\delta^*_{\text{DF}} = -\frac{f(\mathbf{x})}{||\mathbf{w}||_2^2}\mathbf{w}. \tag{18}$$

Now, taking the difference between the minimum perturbation added to an input instance $\mathbf{x}$ by Wachter algorithm (Lemma 1) and DeepFool (equation 18), we get:

$$\begin{aligned}
\delta^*_{\text{SCFE}} - \delta^*_{\text{DF}} &= (\mathbf{w}\mathbf{w}^T + \lambda\mathbf{I})^{-1}(s - \mathbf{w}^T\mathbf{x} - b)\mathbf{w} - \left(-\frac{f(\mathbf{x})}{||\mathbf{w}||_2^2}\mathbf{w}\right) \\
\delta^*_{\text{SCFE}} - \delta^*_{\text{DF}} &= \left((\mathbf{w}\mathbf{w}^T + \lambda\mathbf{I})^{-1}(s - f(\mathbf{x})) + \frac{f(\mathbf{x})}{||\mathbf{w}||_2^2}\right)\mathbf{w} \\
\delta^*_{\text{SCFE}} - \delta^*_{\text{DF}} &= \left(\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) + \frac{f(\mathbf{x})}{||\mathbf{w}||_2^2}\right)\mathbf{w} \qquad \text{(Using Sherman–Morrison formula)}
\end{aligned}$$

Taking the l_p-norm on both sides, we get:

$$\begin{aligned}
\|\delta^*_{\text{SCFE}} - \delta^*_{\text{DF}}\|_p &= \left\|\left(\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) + \mathbf{I}\frac{f(\mathbf{x})}{\|\mathbf{w}\|_2^2}\right)\mathbf{w}\right\|_p \\
&\leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) + \mathbf{I}\frac{f(\mathbf{x})}{\|\mathbf{w}\|_2^2}\right\|_p \|\mathbf{w}\|_p \qquad \text{(Using Cauchy-Schwartz)}
\end{aligned}$$

Adding and subtracting the input instance $\mathbf{x}$ in the left term, we get:

$$\begin{aligned}
\|\mathbf{x} + \delta^*_{\text{SCFE}} - (\mathbf{x} + \delta^*_{\text{DF}})\|_p &\leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) + \mathbf{I}\frac{f(\mathbf{x})}{\|\mathbf{w}\|_2^2}\right\|_p \|\mathbf{w}\|_p \\
\|\mathbf{x}_{\text{SCFE}} - \mathbf{x}_{\text{DF}}\|_p &\leq \left\|\frac{1}{\lambda}\left(\mathbf{I} - \frac{\mathbf{w}\mathbf{w}^T}{\lambda + \mathbf{w}^T\mathbf{w}}\right)(s - f(\mathbf{x})) + \mathbf{I}\frac{f(\mathbf{x})}{\|\mathbf{w}\|_2^2}\right\|_p \|\mathbf{w}\|_p,
\end{aligned}$$

where the final equation gives an upper bound on the difference between the **SCFE** counterfactual and the adversarial example from DeepFool (Moosavi-Dezfooli et al., 2016).

Furthermore, we ask under which conditions the normed difference becomes 0. If we were to choose $\lambda \to 0$ we would get:

$$\begin{aligned}
\|\delta^{**}_{\text{SCFE}} - \delta^*_{\text{DF}}\|_p &= \left\|\frac{-f(\mathbf{x}) + s}{\|\mathbf{w}\|_2^2} \cdot \mathbf{w} + \frac{-f(\mathbf{x})}{\|\mathbf{w}\|_2^2} \cdot \mathbf{w}\right\|_p \\
&= \frac{|s|}{\|\mathbf{w}\|_2^2} \cdot \|\mathbf{w}\|_p,
\end{aligned}$$

where equality holds when the target score is chosen so that s=0, which corresponds to a probability of Y=1 of 0.5.

□

B.5 Proof of Lemma 2

Lemma 2. *Let $\tilde{\mathbf{z}}_{NAE}$ be the solution returned Zhao et al. (2018, Algorithm 1) and $\tilde{\mathbf{z}}_C$ the solution returned by the counterfactual search algorithm of Pawelczyk et al. (2020a) by sampling from ℓ_p-norm ball in the latent space using an L-Lipschitz generative model $\mathcal{G}_\theta(\cdot)$. Analogously, let $\mathbf{x}_{NAE} = \mathcal{G}_\theta(\tilde{\mathbf{z}}_{NAE})$ and $\mathbf{x}_C = \mathcal{G}_\theta(\tilde{\mathbf{z}}_C)$ by design of the two algorithms. Let r^*_{NAE} and r^*_C be the corresponding radius chosed by each algorithm respectively that successfully returns an adversarial example or counterfactual explanation. Then, $\|\mathbf{x}_{NAE} - \mathbf{x}_C\| \leq L(r^*_{NAE} + r^*_C)$.*

Proof. The proof straightforwardly follows from triangle inequality and L-Lipschitzness of the Generative model:

$$\begin{aligned}
\|\mathbf{x}_{\text{NAE}} - \mathbf{x}_C\| &= \|\mathcal{G}_\theta(\tilde{\mathbf{z}}_{\text{NAE}}) - \mathcal{G}_\theta(\tilde{\mathbf{z}}_C)\|_p && (19)\\
&\leq \|\mathcal{G}_\theta(\tilde{\mathbf{z}}_{\text{NAE}}) - \mathbf{x} + \mathbf{x} - \mathcal{G}_\theta(\tilde{\mathbf{z}}_C)\|_p && (20)\\
&\leq \|\mathcal{G}_\theta(\tilde{\mathbf{z}}_{\text{NAE}}) - \mathbf{x}\|_p + \|\mathbf{x} - \mathcal{G}_\theta(\tilde{\mathbf{z}}_C)\|_p && (21)\\
&= \|\mathcal{G}_\theta(\tilde{\mathbf{z}}_{\text{NAE}}) - \mathcal{G}_\theta(\mathbf{z})\|_p + \|\mathcal{G}_\theta(\mathbf{z}) - \mathcal{G}_\theta(\tilde{\mathbf{z}}_C)\|_p && (22)\\
&\leq L\|\tilde{\mathbf{z}}_{\text{NAE}} - \mathbf{z}\|_p + L\|\mathbf{z} - \tilde{\mathbf{z}}_C\|_p && (23)\\
&\leq L\{r^*_{\text{NAE}} + r^*_C\} && (24)
\end{aligned}$$

where equation 20 follows from triangle inequality in the ℓ_p-norm, equation 23 follows from the Lipschitzness assumption and equation 24 follows from properties of the counterfactual search algorithms. □

In the following we outline a lemma that allows us to estimate the Lipschitz constant of the generative model. This will be used for empirical validation of our theoretical claims.

Lemma 4 (Bora et al. (2017))**.** *If G is a d-layer neural network with at most c nodes per layer, all weights $\leq w_{\max}$ in absolute value, and M -Lipschitz non-linearity after each layer, then $G(\cdot)$ is L -Lipschitz with $L = (Mcw_{\max})^d$.*

C EXPERIMENTAL SETUP

C.1 Implementation Details for Counterfactual Explanation and Adversarial Example Methods

For all datasets, categorical features are one-hot encoded and data is scaled to lie between 0 and 1. We partition the dataset into train-test splits. The training set is used to train the classification models for which adversarial examples and counterfactual explanations are generated. adversarial examples and counterfactual explanations are generated for all samples in the test split for the fixed classification model. For counterfactual explanation methods applied to generate recourse examples, all features are assumed actionable for comparison with adversarial examples methods. Adversarial examples and counterfactuals are appropriately generated using the prescribed algorithm implementations in each respective method. Specifically,

i) **SCFE**: As suggested in Wachter et al. (2017), an Adam optimizer (Kingma and Ba, 2014) is used to obtain counterfactual explanations corresponding to the cost function of equation 14. We have based our implementation on the implementation provided by Pawelczyk et al. (2021).

ii) **C-CHVAE**: A (V)AE is additionally trained to model the data-manifold as prescribed in Pawelczyk et al. (2020a). As suggested in Pawelczyk et al. (2020a), a counterfactual search algorithm in the latent space of the (V)AEs. Particularly, a latent sample within an ℓ_p-norm ball with a fixed search radius is used until a counterfactual example is successfully obtained. The search radius of the norm ball is increased until a counterfactual explanation is found. The architecture of the generative model is provided in Appendix C.3. We have based our implementation on the implementation provided by Pawelczyk et al. (2021).

iv) **C&W Attack**: As prescribed in Carlini and Wagner (2017), we use gradient-based optimization to find Adversarial Examples using this attack.

v) **DeepFool**: We implement Moosavi-Dezfooli et al. (2016, Algorithm 1) to generate Adversarial Examples using DeepFool.

vi) **NAE**: This method trains a generative model and an inverter to generate Adversarial Examples. For consistency of comparison with **C-CHVAE**, we use the decoder of the same (V)AE as the generative model for this method. The inverter then corresponds to the encoder of the (V)AE. We use Zhao et al. (2018, Algorithm 1) which uses an iterative search method to find natural adversarial examples. The algorithm searches for adversarial examples in the latent space with radius between $(r, r + \Delta r]$. The search radius is iteratively increased until an Adversarial Example is successfully found.

We describe architecture and training details for real-world data sets in the following.

C.2 Supervised Classification Models

All models are implemented in PyTorch and use a $80-20$ train-test split for model training and evaluation. We evaluate model quality based on the model accuracy. All models are trained with the same architectures across the data sets:

	Neural Network	Logistic Regression
Units	[Input dim. , 18, 9, 3, 1]	[Input dim. , 1]
Type	Fully connected	Fully connected
Intermediate activations	ReLU	N/A
Last layer activations	Sigmoid	Sigmoid

Table 4: Classification model details

		Adult	COMPAS	German Credit
Batch-size	NN	512	32	64
	Logistic Regression	512	32	64
Epochs	NN	50	40	30
	Logistic Regression	50	40	30
Learning rate	NN	0.002	0.002	0.001
	Logistic Regression	0.002	0.002	0.001

Table 5: Training details

	Adult	COMPAS	German Credit
Logistic Regression	0.83	0.84	0.71
Neural Network	0.84	0.85	0.72

Table 6: Performance of models used for generating adversarial examples and counterfactual explanations

C.3 Generative model architectures used for C-CHVAE and NAE

For the results in Lemma 3, we used linear encoders and decoders. For the remaining experiments, we use the following architectures.

	Adult	COMPAS	German Credit
Encoder layers	[input dim, 16, 32, 10]	[input dim, 8, 10, 5]	[input dim, 16, 32, 10]
Decoder layers	[10, 16, 32, input dim]	[5, 10, 8, input dim]	[10, 16, 32, input dim]
Type	Fully connected	Fully connected	Fully connected
Intermediate activations	ReLU	ReLU	ReLU
Loss function	MSE	MSE	MSE

Table 7: Autoencoder details

D ADDITIONAL EMPIRICAL EVALUATION

D.1 Remaining Empirical Results from Section 5

In Table 8, we show the remaining results on the German Credit data pertaining to the Spearman rank correlation experiments, while Figure 5 depicts the remaining d_{match} results for the German Credit data set on the logistic regression classifier.

	German Credit			
	LR		ANN	
Model	SCFE	CCHVAE	SCFE	CCHVAE
CW	$\mathbf{0.92 \pm 0.04}$	0.52 ± 0.08	$\mathbf{0.98 \pm 0.02}$	0.72 ± 0.13
DF	$\mathbf{0.92 \pm 0.04}$	0.57 ± 0.08	0.97 ± 0.02	0.72 ± 0.13
NAE	0.44 ± 0.11	$\mathbf{0.99 \pm 0.01}$	0.71 ± 0.19	$\mathbf{0.99 \pm 0.01}$

Table 8: Average Spearman rank correlation between counterfactual perturbations and adversarial perturbations. For every input $\mathbf{x}$, we compute the corresponding adversarial perturbation δ_{AE} and the counterfactual perturbation δ_{CE}. We then compute the rank correlation of δ_{AE} and δ_{CE} and report their means. The maximum rank correlation is obtained for methods that belong to the same categories (gradient based vs. manifold-based).

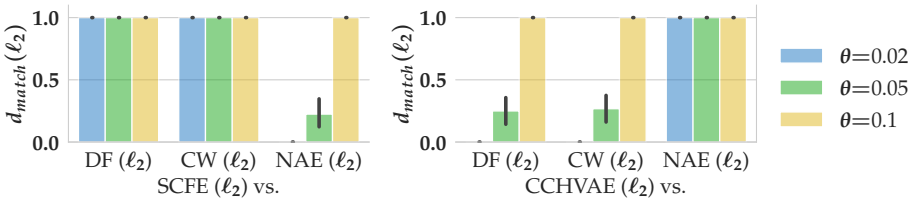
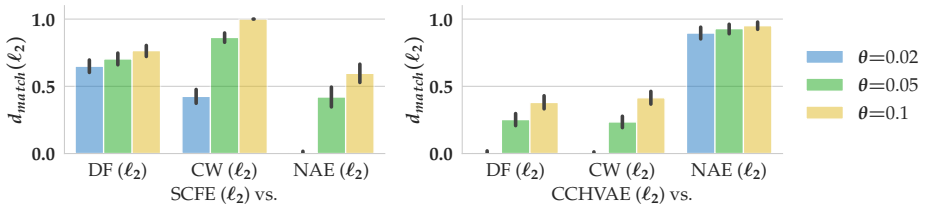


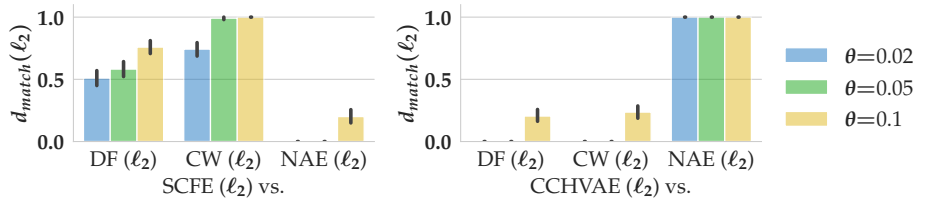
Figure 5: Analyzing to what extent different counterfactual explanation methods and adversarial example generation methods are empirically equivalent for the logistic regression classifier with German Credit data. We compute d_{match} from equation 12 with varying thresholds $\theta = \{0.02, 0.05, 0.1\}$. Missing bars indicate that there was no match.

We also include results for Neural Networks in Appendix D.2.

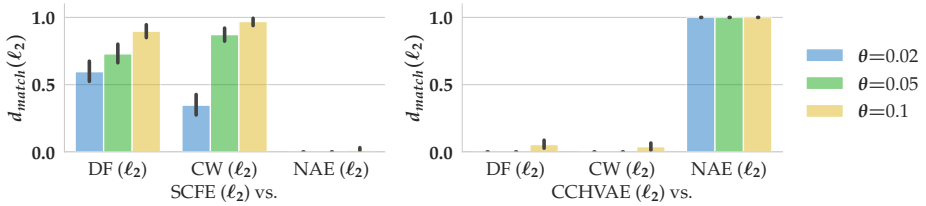
D.2 Empirical Evaluation with ANN



(a) Adult

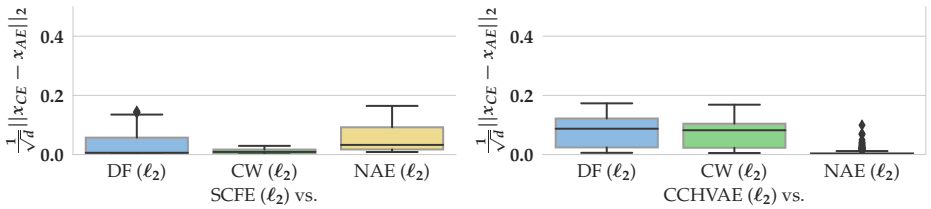


(b) COMPAS

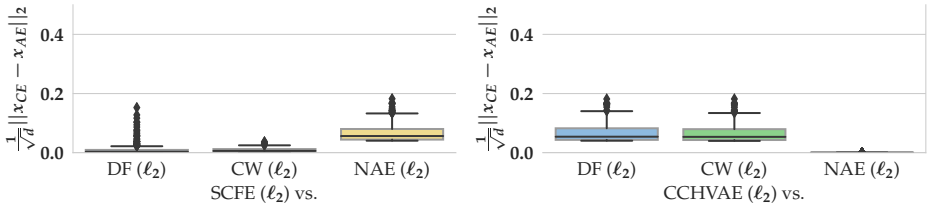


(c) German Credit

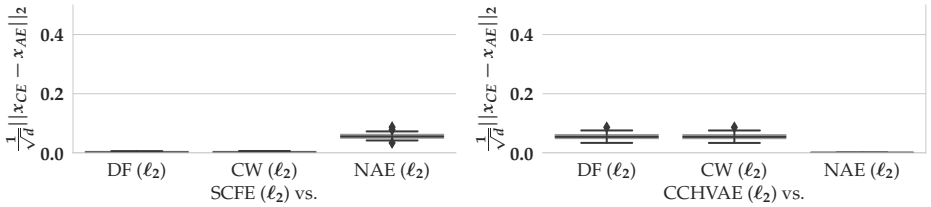
Figure 6: Analyzing to what extent different counterfactual explanations and adversarial examples are empirically equivalent for the 2-layer ANN classifier. To do that, we compute d_{match} from equation 12 with varying thresholds $\theta = \{0.02, 0.05, 0.1\}$. Missing bars indicate that there was no match.



(a) Adult

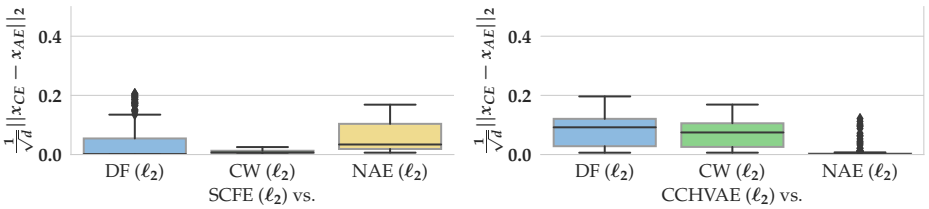


(b) COMPAS

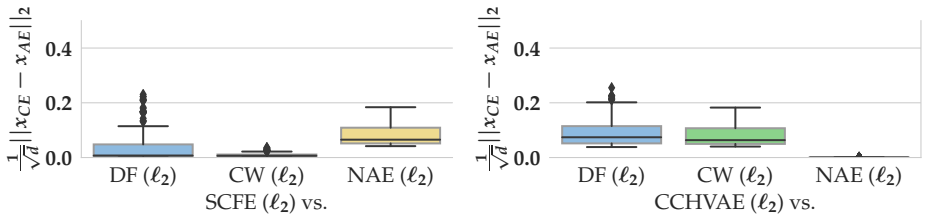


(c) German Credit

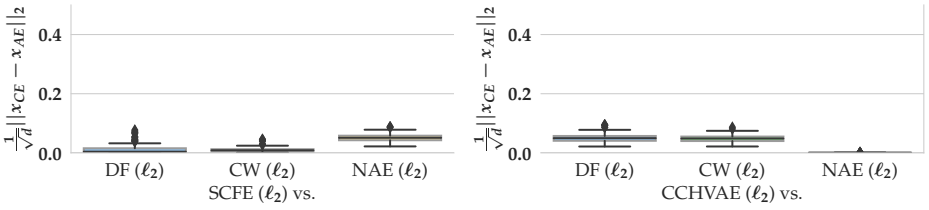
Figure 7: Distribution of instance wise norm comparisons for the logistic regression model. We show the distribution of cost comparisons across negatively predicted instances ($\hat{y} = 0$) for which we computed adversarial examples and counterfactual explanations.



(a) Adult



(b) COMPAS



(c) German Credit

Figure 8: Distribution of instance wise norm comparisons for the 2-layer ANN. We show the distribution of cost comparisons across negatively predicted instances ($\hat{y} = 0$) for which we computed adversarial examples and counterfactual explanations.