

Enhancing the Robustness of Counterfactual Explanations via Adversarial Training

Rithik Appachi Senthilkumar¹, Francesco Leofante², and Vijay Ganesh¹

¹ Georgia Institute of Technology, Atlanta, USA
{rsenthilkumar8,vganesh45}@gatech.edu

² Imperial College London, London, UK
f.leofante@imperial.ac.uk

Abstract. Counterfactual explanations (CEs) provide a simple, yet powerful interpretive approach to understanding neural network behavior, envisioning hypothetical scenarios by systematically altering input features and analyzing the resulting changes in model predictions. However, CEs are useful only if they are robust, i.e., they remain *consistent and meaningful* even when adversarially perturbed. While prior work has explored the development of generators for robust CEs, checking the robustness of CEs produced for DNNs has not been explored within the formal verification context, nor has it been studied under adversarial training. We present a systematic study utilizing adversarial training to fortify the underlying neural network, observing its effect on the formal robustness of DNNs with respect to CEs using α, β -CROWN, a state-of-the-art NNV. Our experiments across multiple datasets, network architectures, and CE generators indicate that adversarial training has a positive impact on the number of formally verified robust CEs. We also measure the impact of adversarial training on other desirable properties of CEs, such as plausibility and proximity. While plausibility does not change, there is a trade-off with proximity when using a gradient-based CE generator.

Keywords: Verification of Neural Networks · Counterfactual Explanations · Adversarial Training.

1 Introduction

Given the prevalence of neural networks in high-stakes domains such as healthcare [10] and criminal justice [33], their interpretability has emerged as a matter of concern. Interpretability allows users to understand and trust the decisions made by machine learning systems. Moreover, it aids in debugging models [44], ensuring regulatory compliance, and providing insights that can lead to new discoveries or improved model designs.

Counterfactual explanations (CEs) have emerged as a promising approach to making machine learning models more interpretable. They illustrate how specific changes to input features can alter the model’s prediction, providing users with

actionable insights into the model’s decision-making process and the factors affecting model outcomes [35,25]. Assuming a machine learning model M predicts class c for an input x , a CE can be described as a modified input x' that yields a prediction $c' \neq c$ by M , produced by changing some features of x . By explaining model predictions, this approach offers avenues into algorithmic recourse [17] for people who deal with real-life outcomes of machine learning model predictions.

1.1 Problem Statement

We seek to explore the generation of CEs robust to noisy execution [14] i.e. tiny perturbations applied on the CE instance. In real-life algorithmic recourse scenarios, non-robust CEs could mislead users, eroding user trust in the underlying predictive models. Prior work in this space has emphasized the design of generators for robust CEs [21,28] for neural networks, but there has been little work on how one can train the underlying neural network to induce the production of robust CE, invariant to the generator algorithm used to produce said CEs.

1.2 Our Approach

We leverage adversarial training [22] to train identically initialized neural networks, each with a different level of adversarial training by varying the strength of the Projected Gradient Descent (PGD) [22] adversary. For each trained network, we generate multiple CEs using a variety of generator algorithms. Finally, using the α, β -CROWN [43,38,39,36,42,31,41,45,20] neural network verifier, we assess the formal robustness of these CEs by measuring the proportion that can be adversarially perturbed within a predefined perturbation bound. This analysis provides a systematic understanding of how adversarial training affects the vulnerability of CEs. This approach differs from previous work in this space, as we investigate whether enforcing robustness on the networks also affects the quality of explanations, both in terms of their robustness and other desiderata (cost and plausibility).

1.3 Contributions.

We make the following contributions through this paper:

1. We introduce a model-centric framing towards generating robust CEs that aims to explore how we can train the underlying network differently, as opposed to building a dedicated generator for robust CEs.
2. To this end, we present an extensive experimental evaluation³ of the effect of adversarially training neural networks on the formal robustness of CEs (measured by α, β -CROWN) produced, spanning multiple datasets, model architectures and geometrically diverse CE generators. We observe that increased adversarial training generally leads to more robust CEs.

³ Our experiments can be found on github.com/rithik83/robust-ce

3. We investigate potential trade-offs between the robustness of CEs and their plausibility and proximity that arise due to adversarially training networks. We observe a slight trade-off between proximity and robustness for CEs produced with a gradient-based generator, but no general effect otherwise.

2 Background

2.1 Counterfactual Explanations in Machine Learning

Formally, existing literature characterises CEs in terms of the solutions to the following optimisation problem:

Definition 1. Consider an input $x \in \mathbb{R}^N$ and a binary classifier M s.t. $M(x) = c$. Given a distance metric $d : \mathbb{R}^N \times \mathbb{R}^N \rightarrow \mathbb{R}$, a CE is any x' such that:

$$\arg \min_{x'} d(x, x') \quad (1a)$$

$$\text{subject to } M(x') = 1 - c, x' \in \mathbb{R}^N \quad (1b)$$

Producing CEs for machine learning (ML) models therefore entails changing the feature values we input to the model, and analyzing their effect on predictions, with changes in prediction being of particular interest [25]. Alternate notions of counterfactuals exist in the literature such as that proposed by Izza et al. [12] who define counterfactual explanations (CXps) as a set of features such that if only those features are allowed to change, there exists some point x' within a distance bound which causes the network to predict a different class, with distance-restricted CXps generalizing standard CXps by adding an l_p constraint. Our notion of CEs is, by contrast, concerned with finding a point that changes network predictions, rather than a minimal set of features. Moreover, while our definition and experiments assume a binary classification, the concept can be extended to a multi-class setting, where a CE is simply a point that causes the underlying network to predict a target class different from the original (factual) point's class label.

CEs in ML offer actionable insights for users to improve their results when faced with automated decision-making systems, a process known as *algorithmic recourse* [17]. Consider an ML model that decides loan provision. The model may reject an individual's application for a loan given their attributes such as income, age and assets. A CE generated for this prediction can suggest an increase in income by 5000\$ causing the underlying model to predict a loan acceptance.

Given the value of CEs in algorithmic recourse, research has identified desiderata of CEs such as proximity [35], sparsity [29], faithfulness [1], plausibility [16] and robustness [14]. CE search is often guided by constraints that seek to promote these desiderata while ensuring validity of the CE. Depending on the approach, CE generation may rely on, e.g. gradient-based optimization [35,1], nearest-neighbor search [4], mixed-integer linear programming (MILP) [24].

2.2 Adversarial Machine Learning

Adversarial ML deals with the vulnerability of machine learning models to tiny and imperceptible perturbations (called *adversarial attacks*) to input data that cause a misclassification by the model [9]. Mechanisms have been developed to defend against adversarial attacks [9,22,30], and to evaluate the adversarial robustness of neural networks [5,6,19].

Adversarial training [22] has emerged as the most reliable, scalable, and empirically validated defense for neural networks against adversarial attacks. It involves exposing the model to adversarially perturbed inputs during training, enabling it to learn representations that are robust to input perturbations.

In practice, adversarial training is most commonly performed using the multi-step *Projected Gradient Descent* (PGD) attack [22], which generates adversarial examples by iteratively perturbing an input in the direction of the gradient of the loss with respect to the input:

$$x_{t+1} = \Pi_{x+S}(x_t + \alpha \text{sgn}(\nabla_x \mathcal{L}(\theta, x_t, y))),$$

where x_t is the perturbed input at step t , α is the step size, and Π_{x+S} denotes projection onto the ℓ_∞ -ball of radius ϵ . The single-step *Fast Gradient Sign Method* (FGSM) [9] can be viewed as a special case of PGD with a single iteration. Madry et al. [22] showed that PGD-based adversarial training leads to smoother loss gradients and a more stable loss landscape, improving robustness to adversarial perturbations.

Formal Verification of Neural Network Robustness: Verification of Neural Networks (VNN) helps provide formal guarantees about neural network robustness to perturbations, either guaranteeing that the input point cannot be successfully perturbed within a certain bound to produce a different classification, or producing a counterexample otherwise. These methods can be broadly categorized into constraint-based and abstraction-based approaches. Constraint-based verification, exemplified by Reluplex [18] and Marabou [37], uses Satisfiability Modulo Theory (SMT) or Mixed Integer Linear Programming (MILP) solvers to provide exact results but can be computationally expensive for large networks. Abstraction-based methods, such as CROWN [43] and DeepPoly [32], improve scalability by using abstract domains to evaluate networks on input sets, sacrificing completeness for efficiency. Recent advancements have led to hybrid approaches like α, β -CROWN [43,38,39,36,42,31,41,45,20], combining bound propagation with branch-and-bound algorithms for precision and scalability.

3 Robustness of Counterfactual Explanations

Jiang et al. [14] present a systematic survey laying out various notions of robustness of CEs: robustness against model changes (the CE remains valid despite changes to the underlying model’s parameters), model multiplicity (CE for an

aggregate prediction of a set of models is a valid CE for some subset of models), noisy execution (the CE for the underlying model remains valid despite tiny perturbations applied to it) and input changes (CEs produced for two similar factual input points, should be similar themselves).

Our paper subscribes to the notion of robustness of CEs against noisy execution (NE). According to Jiang et al. [14], "Assume an input x and a model M . Let x' be a CE for x . Robustness against NE requires that whenever a **small perturbation** σ is applied to x' , validity is not affected, i.e. $M(x') = M(x' + \sigma)$." Consider our loan approval ML model example in Section 2.1. While the CE here suggested an increase in income by exactly 5000\$, the loan applicant may not be able to control their circumstances to achieve this exact change. Had this CE not been robust to noisy execution, a slightly different income increase, say 4991\$ may yield a rejection, rendering the CE unreliable and potentially frustrating the applicant. A robust counterfactual, on the other hand, would ensure that small deviations from the suggested change still result in the desired outcome.

Our paper hones in on CE robustness against noisy execution as our objective is to explore how fortifying the underlying neural network via adversarial training can result in the production of CEs that end-users of ML-based decision-making systems can rely on, as a means of algorithmic recourse. A CE not robust against noisy execution is not a reliable CE for algorithmic recourse, as our real-life example highlights.

3.1 Pathways to Robust CEs: The CE Generator vs. Fortifying the Underlying Network

Existing approaches to improve the robustness of CEs to NE have emphasized the design of CE generation algorithms that produce robust CEs, without any bearing on the design or training process for the underlying network [21,28]. Our work poses a question unexplored in past research, **of whether there exist processes that can be applied on the underlying network to induce the generation of CEs robust to NE, instead of impacting the CE generator**. This model-centric approach can help us test whether CE robustness may be an intrinsic property of the underlying model, due to which improving CE robustness may not necessitate developing complex generators. Moreover, this can introduce possibilities of hybrid approaches that rely on both model training regime and CE generator design to further strengthen CE robustness.

To this end, we empirically study the impact of adversarial training [22] the underlying network. Both adversarial perturbations and CEs in machine learning share similar technical foundations – they involve solving a similar optimization problem of finding a perturbation to the original input datapoint to change the model’s predictions [8]. They often employ similar techniques such as gradient-based optimization to generate examples that cross the classifier’s decision boundary, with several popular CE generators producing very similar examples as adversarial attack algorithms [27].

However, there are important differences between these two concepts [8]. Adversarial perturbations are crafted with the sole aim of fooling the underlying

model to produce a misclassification despite the perturbation being imperceptible. Counterfactual explanations, on the other hand, seek to provide actionable explanations for model decisions by producing realistic changes meaningful to humans. Nonetheless, due to the similarity in their optimization objectives, techniques developed to improve the robustness of neural networks to adversarial examples [22] can potentially be helpful towards enhancing the robustness of counterfactual explanations themselves, which we explore in this paper.

4 Experimental Evaluation and Results

This section of the paper comprehensively describes our experimental process, elaborating on the datasets and models used, configurations of adversaries for training, CE generation and formal robustness evaluation of said CEs using α, β -CROWN. We describe the software used for the experiments, and finally discuss our experimental results and key insights.

4.1 Experimental Setup

Datasets We experiment with three tabular datasets for binary classification:

- *California Housing* [26], a dataset of 20640 instances and 8 features describing housing characteristics of California districts. We split the numerical target (median house value for each district) along its median to two classes.
- *Spambase* [11], a dataset of 4601 instances and 57 features describing e-mails and word frequencies. Target classes are 1 (spam) and 0 (non-spam).
- *Single Elder Home Monitoring: Gas and Position* [23], a dataset of 444631 instances and 10 features corresponding to gas and temperature sensor or movement sensor readings. We consider the first 20000 instances for our experiments and dropped the timestamp feature. Moreover, we split the numerical target (COValue) along its median to two classes.

Their relatively low-dimensional input space makes them a computationally efficient choice for training networks and generating CEs. Besides, the datasets consist purely of continuous and not categorical features, which was necessary for adversarial training. Moreover, the chosen datasets strike a deliberate balance between simplicity and representativeness: they are sufficiently expressive to capture non-trivial decision boundaries, yet structured in a way that enables reliable adversarial training and CE generation, making them particularly well-suited for isolating and studying the relationship between model robustness and the robustness of counterfactual explanations. 80% of the dataset was used as the train-set, and standard scaling was applied on them prior to use.

Neural Network Initialization and Training: We train multi-layer perceptrons (MLPs) with four hidden layer configurations: 1 32-neuron layer, 1

64-neuron layer, 2 32-neuron layers and 2 64-neuron layers, aiming to demonstrate that the relationship between adversarial training and CE robustness is a general phenomenon rather than an artifact of a specific network architecture. We use Rectified Linear Unit (ReLU) as the intermediate activation, sigmoid as the final activation and binary cross-entropy as the loss function.

For each dataset and network architecture combination, we initialized four MLPs with identical initial weights. One network was trained using standard training, while the remaining 3 were trained with varying degrees of PGD adversaries with ℓ_∞ perturbation bounds $\epsilon \in \{0.01, 0.02, 0.03\}$, and step sizes 0.001, 0.002 and 0.003 respectively, with 10 PGD steps for all.

Generating Counterfactual Explanations: To evaluate our approach, we select a diverse suite of generators with distinct geometric properties:

- A gradient-based generator proposed by Wachter et al. [35], which iteratively moves in the direction minimizing the loss of the input w.r.t the CE target class, with a distance-based regularization term to induce proximity. This technique produces CEs often indistinguishable from adversarial attacks. CE search stops either after 10000000 iterations, or once the CE crosses the decision-boundary with a prediction ≥ 0.6 for class 1 or ≤ 0.4 for class 0, to ensure the search does not stop exactly on the boundary line.
- A KD-Tree based nearest neighbor algorithm [4] that finds the closest instance to the original point that corresponds to the target class, from the training data. This tends to produce more robust CEs by situating them deeper within the data manifold.
- A Mixed-Integer Linear Programming (MILP) based CE generator [24]. A limit of 10 seconds was provided for each instance.
- A generator that performs binary search between the original point and a random target class instance, finding a point that just crosses the boundary.

The CE generators remain fixed across all training schemes (standard and adversarial training), keeping findings unbiased.

An experimental run consists of randomly selecting 200 points from an unseen test set and generating one CE for each factual point with the target class opposite to the factual class. Note that a CE only contributes to the evaluation metrics if the model (1) correctly predicts the original datapoint, and (2) correctly predicts the CE as belonging to the target class (i.e. the CE is valid).

Evaluation Metrics: We formally verify the robustness of generated CEs using the state-of-the-art VNN solver α, β -CROWN [43,38,39,36,42,31,41,45,20]. We assess whether each CE is robust to perturbations within an ℓ_∞ norm of 0.05 i.e. the model’s prediction does not change from the target back to original class. Instances for which the solver returns an error are omitted in our analysis. The solver returns one of three possible results for each CE:

Table 1. Results of adversarial robustness evaluation for CEs produced by Wachter (gradient-based) [35] and nearest neighbor [4] generators. CE robustness tested within an ℓ_∞ perturbation bound of 0.05. Accuracy over all 200 samples, number of robust CEs, mean implausibility and mean euclidean distance between factual and CE over all valid CEs presented. AT-1 ($\epsilon = 0.01$), AT-2 ($\epsilon = 0.02$) and AT-3 ($\epsilon = 0.03$) are the adv. trained networks. Clean is the standard (non-adversarial) trained model.

Dataset	Hidden Neurons	Train. Acc.	Wachter CE Gen.			KD-Tree NN CE Gen.		
			Robust	Impl. Dist.		Robust	Impl. Dist.	
Cali. Housing	32	Clean	84%	34/167	1.305 0.475	15/168	1.172	0.786
		AT-1	85%	62/170	1.324 0.539	14/170	1.089	0.807
		AT-2	86%	127/172	1.351 0.690	20/172	1.099	0.833
		AT-3	85%	170/170	1.412 0.908	28/170	1.090	0.872
	64	Clean	85%	21/169	1.296 0.464	19/170	1.161	0.775
		AT-1	86%	65/171	1.304 0.546	18/172	1.103	0.798
		AT-2	85.5%	135/171	1.369 0.747	14/171	1.103	0.822
		AT-3	85.5%	171/171	1.396 0.886	27/171	1.100	0.852
	32-32	Clean	87.5%	31/175	1.360 0.547	14/175	1.115	0.726
		AT-1	87%	74/174	1.341 0.605	28/174	1.150	0.758
		AT-2	86.5%	109/173	1.349 0.700	27/173	1.108	0.789
		AT-3	88%	151/176	1.369 0.765	31/176	1.099	0.797
	64-64	Clean	89%	23/178	1.314 0.470	20/178	1.138	0.731
		AT-1	89.5%	59/178	1.312 0.602	27/179	1.140	0.754
		AT-2	89%	85/177	1.317 0.685	40/178	1.162	0.770
		AT-3	88%	133/176	1.368 0.806	32/176	1.103	0.798
Spambase	32	Clean	92%	15/175	2.076 1.687	28/184	1.647	4.865
		AT-1	91.5%	27/174	1.987 1.901	34/182	1.439	4.881
		AT-2	92%	38/173	2.001 2.048	43/184	1.441	4.892
		AT-3	92%	51/168	2.012 2.237	52/184	1.497	4.899
	64	Clean	93%	4/184	1.990 1.620	28/184	1.558	4.798
		AT-1	93.5%	14/178	1.970 1.808	41/186	1.484	4.790
		AT-2	93%	20/169	1.951 1.983	52/185	1.389	4.844
		AT-3	93%	30/166	1.992 2.249	57/179	1.342	4.899
	32-32	Clean	93.5%	5/187	2.067 1.559	23/186	1.872	4.741
		AT-1	93%	5/183	2.083 1.774	39/185	2.008	4.714
		AT-2	93%	13/177	1.975 1.959	49/181	1.543	4.762
		AT-3	91.5%	17/176	1.972 2.094	64/178	1.374	4.884
	64-64	Clean	94%	0/181	2.009 1.343	27/187	1.628	4.600
		AT-1	94%	0/184	2.034 1.610	34/184	1.631	4.563
		AT-2	94%	0/185	1.985 1.856	52/172	1.529	4.716
		AT-3	92.5%	5/179	1.997 2.056	58/171	1.426	4.762
Elder Home	32	Clean	96%	13/180	4.030 0.487	17/192	1.511	0.952
		AT-1	96%	40/181	4.153 0.572	26/192	1.590	0.971
		AT-2	95%	60/179	4.212 0.685	37/190	1.361	1.048
		AT-3	95.5%	71/179	4.230 0.741	41/191	1.604	0.989
	64	Clean	95.5%	4/175	3.900 0.474	28/191	1.372	0.970
		AT-1	95.5%	22/175	3.956 0.557	36/191	1.299	1.015
		AT-2	97%	36/180	4.027 0.608	37/194	1.346	0.995
		AT-3	96.5%	53/179	4.105 0.694	42/193	1.296	1.018
	32-32	Clean	98%	0/155	3.399 0.351	31/196	1.327	0.911
		AT-1	98.5%	5/163	3.490 0.450	38/197	1.301	0.952
		AT-2	97.5%	6/169	3.680 0.534	60/195	1.380	0.926
		AT-3	97.5%	17/172	3.843 0.581	51/195	1.268	0.938
	64-64	Clean	98%	0/163	3.654 0.401	45/196	1.379	0.902
		AT-1	98.5%	1/165	3.667 0.490	54/197	1.222	0.915
		AT-2	98.5%	5/171	3.695 0.549	61/197	1.230	0.933
		AT-3	98.5%	9/168	3.536 0.584	70/197	1.259	0.921

- *unsafe*: If the solver identifies a perturbation to the CE (a counterexample) causing the network to flip its prediction, i.e. the CE was not robust. Can either be *unsafe-pgd* (PGD attack) or *unsafe-bab* (branching-and-bound) depending on the strategy used by the solver to arrive at that conclusion.
- *safe/verified*: If the solver successfully proved that no such perturbation exists, indicating that the counterfactual explanation is robust (for our specified perturbation bound), i.e. the CE is verified robust. When incomplete verification suffices to prove this property, *safe-incomplete* is returned.
- *unknown*: If the search stops early due to timeouts or resource limits.

The proportion of robust CEs generated gives us an estimate of the robustness of CEs against noisy execution. Our work, however, also seeks to explore the impact of adversarial training on other CE desiderata, namely plausibility [16], i.e. the extent to which the proposed counterfactual corresponds to a realistic, feasible, and domain-consistent data point, and proximity [35], the distance between the original point and its CE. Prior research [34,13,7] has identified inherent correlations and potential tensions between these metrics; for instance, increasing robustness often necessitates a trade-off with proximity [28]. Consequently, we aim to ensure that our model-centric approach to robustness does not disproportionately degrade these other essential properties, maintaining the practical utility of the generated recourse. For plausibility, we used Local Outlier Factor [3], measured as *implausibility* (higher this value, lower the plausibility). For proximity, we used the Euclidean distance between the original point and its corresponding CE (higher the distance, lower the proximity). We collect the total number of robust CEs, mean implausibility and mean distance as our metrics.

Our evaluation prioritized the factors most central to our framework’s performance: adversarial training intensity, network architecture, and generator diversity across three distinct datasets. We focused our analysis on these high-level hyperparameters to establish the fundamental relationship between adversarial training and CE robustness. Fixing perturbation radius and seeds across our comparative benchmarks ensured a controlled environment to isolate the effects of adversarial training on CE robustness/plausibility.

The experiments were carried out on the NVIDIA Quatro RTX 6000 GPU made available through the Phoenix cluster provided by Georgia Institute of Technology’s Partnership for an Advanced Computing Environment (PACE)⁴. All experiments were conducted in Python, with networks trained using PyTorch [2], CEs generated using the implementations provided by the RobustX library [15], and the robustness verified using the Python API of α, β -CROWN.

4.2 Results and Discussion

Table 1 demonstrates the results of our experiments using the gradient-based [35] and nearest neighbor-based [4] CE generators. Due to space constraints and the uniform lack of robust results for experiments involving the MILP [24] and

⁴ pace.gatech.edu/phoenix-cluster/

binary search based CE generators, we report the experimental data for these two explainers in Appendix A, focusing our primary discussion on the generators that are positively affected by adversarial training.

Across nearly all configurations of dataset, network architecture and CE generator, our results clearly indicate a positive trend between the extent of adversarial training (measured by the strength of PGD adversary used during training) of the underlying neural network, and the proportion of CEs generated that are robust to noisy execution. This is perhaps the most encouraging result our work presents: keeping the CE generation process constant, it is possible to train the underlying neural network in a manner that induces the production of robust CEs, without affecting the overall accuracy of the model. This trend is most starkly observable in the case of the California Housing dataset with the gradient-based generator, where particularly in the smaller networks, all generated CEs when using the highest extent of adversarial training turn out to be robust. While the proportion of robust CEs is comparatively lower in other datasets and the nearest neighbor generator, the trend of stronger adversarial training improving said proportion still holds true in almost all cases, which was the motive of our exploration of the adversarial training strategy.

Intuitively, our results can be understood as an effect of adversarial training smoothening decision boundaries. A model obtained with vanilla training can develop highly irregular boundaries where small perturbations easily push points back across the boundary. Adversarial training penalizes this irregularity by forcing the model to maintain consistent predictions in local neighborhoods, effectively flattening and regularizing the boundary. On a smoother boundary, the nearest counterfactual point is less likely to sit on a "spike" in the decision region, so the explanation naturally tolerates more perturbation before becoming invalid. This is why robustness improves broadly across the domain rather than just at specific points.

While adversarial training significantly enhances the certified robustness of CEs generated via gradient-based and heuristic methods, we observe a divergence when employing exact or complete solvers like MILP and Binary Search. As detailed in Appendix A, these generators consistently identify CEs that remain non-robust even under the fortified model. This phenomenon underscores a fundamental limitation: the 'robustness gap' persists when the generator is capable of exhaustive search within the input space. We argue that the efficacy of adversarial training as a defense is intrinsically linked to the 'incompleteness' of the generator; essentially, a heuristic defense is most effective against heuristic-based explorations.

Moreover, adversarial training provides robustness on average rather than worst-case guarantees, which is why complete CE solvers can still find non-robust CEs. However, completeness is not always needed in practice. Users care that counterfactual explanations hold under realistic shifts. Adversarial training targets exactly this. Complete verification only becomes essential in safety-critical or strict regulatory settings where even a single failure matters, and in those

cases it often comes at prohibitive computational cost. So the gap with complete solvers is better understood as a pragmatic trade-off.

A slight trade-off between robustness and proximity of CEs is observed when applying the gradient-based CE generator. As the extent of adversarial training increases, so does CE robustness, however the resultant CEs are on average further away from their corresponding original points. This trade-off has been noted in prior research [28]. Since both adversarial training and CE generation in this case rely on gradients, their interaction is geometrically meaningful: adversarial training enlarges the region of label invariance around data points and smooths the local loss landscape, effectively increasing the margin to the decision boundary. Consequently, the gradient-based CE search must traverse a larger region of stable predictions before reaching a class-changing boundary, resulting in more distant but more robust CE.

There is a natural tension between proximity and robustness: making a counterfactual explanation more robust typically pushes it further from the original input. At first glance this seems undesirable, as users generally prefer minimal changes. But a slightly less proximal explanation that remains valid under realistic perturbations is arguably more actionable than a closer one that breaks under small shifts. If a user is told "increase your income by \$200" but that explanation is fragile, they might act on it and still be denied. A more robust alternative, say "increase by \$500", gives them a reliable target they can actually trust. In practice, a small sacrifice in proximity buys meaningful gains in reliability, which is what matters when explanations are meant to guide real decisions.

We do not observe a meaningful effect on plausibility in either direction with adversarial training, demonstrating that we can improve robustness of CEs without impacting their plausibility. Moreover, adversarial training did not negatively impact the accuracy of the underlying model, indicating that we can achieve gains on the robustness of CEs generated without affecting the reliability and performance of the model.

5 Conclusions and Future Work

In this paper, we set out to explore the concept of formal robustness of CEs for neural networks to noisy execution, particularly how adversarially training neural networks can impact this type of robustness. By analyzing neural networks trained with varying levels of adversarial robustness using formal verification techniques, we found that adversarially trained neural networks (that are robust against adversarial attacks) also produce robust CEs as compared to those that are trained classically, and that this effect comes at a cost of slightly lower proximity between the CE and its corresponding factual instance (i.e. an increased cost in producing the CE) when using a gradient-based CE generation approach.

As opposed to prior research in this space which tackled the problem of generating CEs robust to noisy execution by developing specialized CE generators, our work approached this problem from the point of view of the underlying neu-

ral network, investigating whether the network could be trained differently so as to induce the production of robust CEs keeping the CE generator constant.

To carry this work forward, a next step we have identified is the application of different robust training methodologies such as CROWN-IBP [40] that leverages interval bound propagation to train verifiably robust neural networks. We also plan to explore how certified training affects the robustness of CEs generated via complete solvers, for which adversarial training was unable to induce the production of robust CEs.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A Experiments with MILP and Binary Linear Search CE Generators

Table 2. Results of adversarial robustness evaluation for MILP based [24] and Binary Linear Search (BLS) CE generator. MILP CE generator for *Spambase* with 64-64 network did not return CE instances within the prescribed time limit of 10s per CE, and are hence excluded from the table.

Dataset	Hidden Neurons	Train. Acc.	MILP CE Gen.			BLS CE Gen.		
			Robust	Impl.	Dist.	Robust	Impl.	Dist.
Cali. Housing	32	Clean	84%	0/168	1.379	0.363	0/168	1.175 1.727
		AT-1	85%	0/170	1.373	0.435	0/170	1.171 1.639
		AT-2	86%	0/172	1.385	0.610	0/172	1.176 1.451
		AT-3	85%	0/170	1.399	0.794	0/170	1.185 1.351
	64	Clean	85%	0/170	1.369	0.348	0/170	1.174 1.697
		AT-1	86%	0/172	1.370	0.464	0/172	1.181 1.516
		AT-2	85.5%	0/171	1.385	0.649	0/171	1.179 1.451
		AT-3	85.5%	0/171	1.375	0.736	0/171	1.181 1.365
	32-32	Clean	87.5%	0/175	1.351	0.362	0/175	1.166 1.743
		AT-1	87%	0/174	1.346	0.444	0/174	1.178 1.587
		AT-2	86.5%	0/173	1.318	0.518	0/173	1.179 1.512
		AT-3	88%	0/176	1.315	0.555	0/176	1.185 1.493
	64-64	Clean	89%	0/83	1.186	0.120	0/178	1.183 1.616
		AT-1	89.5%	0/71	1.180	0.154	0/179	1.183 1.436
		AT-2	89%	0/98	1.157	0.228	0/178	1.189 1.356
		AT-3	88%	0/92	1.175	0.256	0/176	1.180 1.395
Spambase	32	Clean	92%	0/184	14.100	2.982	0/184	1.562 3.882
		AT-1	91.5%	0/176	2.080	3.047	0/183	1.481 3.822
		AT-2	92%	0/183	1.964	2.879	0/184	1.471 3.759
		AT-3	92%	0/182	1.965	2.851	0/184	1.488 3.730
	64	Clean	93%	0/183	5.513	2.615	0/186	1.424 4.013
		AT-1	93.5%	0/182	1.931	2.634	0/187	1.501 3.920
		AT-2	93%	0/184	1.958	2.676	0/186	1.496 3.821
		AT-3	93%	0/184	1.941	2.750	0/186	1.497 3.748
	32-32	Clean	93.5%	0/187	55759.432	2.206	0/187	1.523 3.885
		AT-1	93%	0/184	2.001	2.239	0/186	1.556 3.888
		AT-2	93%	0/186	2.188	2.315	0/186	1.469 3.867
		AT-3	91.5%	0/181	118655.701	2.390	0/183	1.527 3.754
	64-64	Clean	94%	-	-	-	0/188	1.512 3.915
		AT-1	94%	-	-	-	0/188	1.529 3.780
		AT-2	94%	-	-	-	0/188	1.459 3.645
		AT-3	92.5%	-	-	-	0/185	1.426 3.745
Elder Home	32	Clean	96%	0/192	4.462	0.498	0/192	2.724 1.625
		AT-1	96%	0/192	4.474	0.577	0/192	2.684 1.722
		AT-2	95%	0/190	4.584	0.674	0/190	3.060 1.824
		AT-3	95.5%	0/191	4.532	0.722	0/191	3.078 1.754
	64	Clean	95.5%	0/191	4.532	0.502	0/191	2.697 1.670
		AT-1	95.5%	0/191	4.594	0.591	0/191	2.724 1.655
		AT-2	97%	0/194	4.514	0.628	0/194	2.908 1.668
		AT-3	96.5%	0/193	4.591	0.722	0/193	3.062 1.756
	32-32	Clean	98%	0/196	4.414	0.443	0/196	2.767 1.644
		AT-1	98.5%	0/197	4.411	0.524	0/197	2.813 1.653
		AT-2	97.5%	0/195	4.374	0.656	0/195	2.868 1.813
		AT-3	97.5%	0/195	4.368	0.677	0/195	3.062 1.766
	64-64	Clean	98%	0/132	3.380	0.384	0/196	2.829 1.534
		AT-1	98.5%	0/119	3.121	0.424	0/197	2.830 1.545
		AT-2	98.5%	0/94	2.858	0.423	0/197	3.115 1.661
		AT-3	98.5%	0/79	2.798	0.402	0/197	2.783 1.743

References

1. Altmeyer, P., Farmanbar, M., van Deursen, A., Liem, C.C.: Faithful model explanations through energy-constrained conformal counterfactuals. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 10829–10837 (2024)
2. Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C., Maher, B., Pan, Y., Puhersch, C., Reso, M., Saroufim, M., Siraichi, M.Y., Suk, H., Suo, M., Tillet, P., Wang, E., Wang, X., Wen, W., Zhang, S., Zhao, X., Zhou, K., Zou, R., Mathews, A., Chanan, G., Wu, P., Chintala, S.: PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24). ACM (Apr 2024). <https://doi.org/10.1145/3620665.3640366>, <https://docs.pytorch.org/assets/pytorch2-2.pdf>
3. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
4. Brughmans, D., Leyman, P., Martens, D.: Nice: an algorithm for nearest instance counterfactual explanations. *Data mining and knowledge discovery* **38**(5), 2665–2703 (2024)
5. Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., Madry, A., Kurakin, A.: On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705* (2019)
6. Croce, F., Andriushchenko, M., Sehwal, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., Hein, M.: Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670* (2020)
7. Dutta, S., Long, J., Mishra, S., Tilli, C., Magazzeni, D.: Robust counterfactual explanations for tree-based ensembles. In: International Conference on Machine Learning, ICML 2022, 17–23 July 2022, Baltimore, Maryland, USA. Proceedings of Machine Learning Research, vol. 162, pp. 5742–5756. PMLR (2022)
8. Freiesleben, T.: The intriguing relation between counterfactual explanations and adversarial examples. *Minds and Machines* **32**(1), 77–109 (2022)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014)
10. Habehh, H., Gohel, S.: Machine learning in healthcare. *Current genomics* **22**(4), 291 (2021)
11. Hopkins, M., Reeber, E., Forman, G., Suermondt, J.: Spambase. UCI Machine Learning Repository (1999), DOI: <https://doi.org/10.24432/C53G6X>
12. Izza, Y., Huang, X., Morgado, A., Planes, J., Ignatiev, A., Marques-Silva, J.: Distance-restricted explanations: theoretical underpinnings & efficient implementation. *arXiv preprint arXiv:2405.08297* (2024)
13. Jiang, J., Leofante, F., Rago, A., Toni, F.: Formalising the robustness of counterfactual explanations for neural networks. In: Thirty-Seventh AAAI Conference on Artificial Intelligence. pp. 14901–14909. AAAI Press (2023)
14. Jiang, J., Leofante, F., Rago, A., Toni, F.: Robust counterfactual explanations in machine learning: A survey. In: Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3–9, 2024. pp. 8086–8094. ijcai.org (2024)

15. Jiang, J., Marzari, L., Purohit, A., Leofante, F.: Robustx: Robust counterfactual explanations made easy. In: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025. pp. 11067–11071. ijcai.org (2025)
16. Joshi, S., Koyejo, O., Vijitbenjaronk, W., Kim, B., Ghosh, J.: Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv preprint arXiv:1907.09615* (2019)
17. Karimi, A.H., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *arXiv preprint arXiv:2010.04050* (2020)
18. Katz, G., Barrett, C., Dill, D.L., Julian, K., Kochenderfer, M.J.: Reluplex: An efficient smt solver for verifying deep neural networks. In: Computer Aided Verification: 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part I 30. pp. 97–117. Springer (2017)
19. König, M., Bosman, A.W., Hoos, H.H., van Rijn, J.N.: Critically assessing the state of the art in neural network verification. *Journal of Machine Learning Research* **25**(12), 1–53 (2024)
20. Kotha, S., Brix, C., Kolter, J.Z., Dvijotham, K., Zhang, H.: Provably bounding neural network preimages. In: Advances in Neural Information Processing Systems. vol. 36, pp. 80270–80290. Curran Associates, Inc. (2023)
21. Leofante, F., Lomuscio, A.: Towards robust contrastive explanations for human-neural multi-agent systems. In: Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023. pp. 2343–2345. ACM (2023)
22. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017)
23. Marín López, D., Marín, D., Fonollosa, J., Llano, J., Perera, A., Haddi, Z.: Single Elder Home Monitoring: Gas and Position. *UCI Machine Learning Repository* (2023), DOI: <https://doi.org/10.24432/C5762W>
24. Mohammadi, K., Karimi, A.H., Barthe, G., Valera, I.: Scaling guarantees for nearest counterfactual explanations. In: Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society. pp. 177–187 (2021)
25. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
26. Pace, R.K., Barry, R.: Sparse spatial autoregressions. *Statistics & Probability Letters* **33**(3), 291–297 (1997)
27. Pawelczyk, M., Agarwal, C., Joshi, S., Upadhyay, S., Lakkaraju, H.: Exploring counterfactual explanations through the lens of adversarial examples: A theoretical and empirical analysis. In: International Conference on Artificial Intelligence and Statistics. pp. 4574–4594. PMLR (2022)
28. Pawelczyk, M., Datta, T., van den Heuvel, J., Kasneci, G., Lakkaraju, H.: Probabilistically robust recourse: Navigating the trade-offs between costs and robustness in algorithmic recourse. In: The Eleventh International Conference on Learning Representations, ICLR. OpenReview.net (2023)
29. Schut, L., Key, O., Mc Grath, R., Costabello, L., Sacaleanu, B., Gal, Y., et al.: Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In: International Conference on Artificial Intelligence and Statistics. pp. 1756–1764. PMLR (2021)
30. Shafahi, A., Najibi, M., Ghiasi, M.A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! *Advances in neural information processing systems* **32** (2019)

31. Shi, Z., Jin, Q., Kolter, Z., Jana, S., Hsieh, C.J., Zhang, H.: Neural network verification with branch-and-bound for general nonlinearities. In: International Conference on Tools and Algorithms for the Construction and Analysis of Systems (2025)
32. Singh, G., Gehr, T., Püschel, M., Vechev, M.: An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages* **3**(POPL), 1–30 (2019)
33. Travaini, G.V., Pacchioni, F., Bellumore, S., Bosia, M., De Micco, F.: Machine learning and criminal justice: A systematic review of advanced methodology for recidivism risk prediction. *International journal of environmental research and public health* **19**(17), 10594 (2022)
34. Upadhyay, S., Joshi, S., Lakkaraju, H.: Towards robust and reliable algorithmic recourse. In: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (eds.) *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6–14, 2021, virtual*. pp. 16926–16937 (2021)
35. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.* **31**, 841 (2017)
36. Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.J., Kolter, J.Z.: Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems* **34**, 29909–29921 (2021)
37. Wu, H., Isac, O., Zeljić, A., Tagomori, T., Daggett, M., Kokke, W., Refaeli, I., Amir, G., Julian, K., Bassan, S., Huang, P., Lahav, O., Wu, M., Zhang, M., Komentanskaya, E., Katz, G., Barrett, C.: Marabou 2.0: A versatile formal analyzer of neural networks (2024), <https://arxiv.org/abs/2401.14461>
38. Xu, K., Shi, Z., Zhang, H., Wang, Y., Chang, K.W., Huang, M., Kailkhura, B., Lin, X., Hsieh, C.J.: Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems* **33** (2020)
39. Xu, K., Zhang, H., Wang, S., Wang, Y., Jana, S., Lin, X., Hsieh, C.J.: Fast and Complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. In: *International Conference on Learning Representations* (2021), <https://openreview.net/forum?id=nVZtXBI6LNn>
40. Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., Hsieh, C.J.: Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316* (2019)
41. Zhang, H., Wang, S., Xu, K., Li, L., Li, B., Jana, S., Hsieh, C.J., Kolter, J.Z.: General cutting planes for bound-propagation-based neural network verification. *Advances in Neural Information Processing Systems* (2022)
42. Zhang, H., Wang, S., Xu, K., Wang, Y., Jana, S., Hsieh, C.J., Kolter, Z.: A branch and bound framework for stronger adversarial attacks of ReLU networks. In: *Proceedings of the 39th International Conference on Machine Learning*. vol. 162, pp. 26591–26604 (2022)
43. Zhang, H., Weng, T.W., Chen, P.Y., Hsieh, C.J., Daniel, L.: Efficient neural network robustness certification with general activation functions. In: *Advances in Neural Information Processing Systems (NuerIPS)* (dec 2018)
44. Zhang, Q.s., Zhu, S.C.: Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering* **19**(1), 27–39 (2018)
45. Zhou, D., Brix, C., Hanasusanto, G.A., Zhang, H.: Scalable neural network verification with branch-and-bound inferred cutting planes. In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems* (2024)