
ElliCE: Efficient and Provably Robust Algorithmic Recourse via the Rashomon Sets

Bohdan Turbal¹ Iryna Voitsitska² Lesia Semenova^{3*}

¹ Princeton University ² Ukrainian Catholic University ³ Rutgers University
bt4811@princeton.edu, voitsitska.pn@ucu.edu.ua, lesia.semenova@rutgers.edu

Abstract

Machine learning models now influence decisions that directly affect people’s lives, making it important to understand not only their predictions, but also how individuals could act to obtain better results. Algorithmic recourse provides actionable input modifications to achieve more favorable outcomes, typically relying on counterfactual explanations to suggest such changes. However, when the Rashomon set – the set of near-optimal models – is large, standard counterfactual explanations can become unreliable, as a recourse action valid for one model may fail under another. We introduce ElliCE, a novel framework for robust algorithmic recourse that optimizes counterfactuals over an ellipsoidal approximation of the Rashomon set. The resulting explanations are provably valid over this ellipsoid, with theoretical guarantees on uniqueness, stability, and alignment with key feature directions. Empirically, ElliCE generates counterfactuals that are not only more robust but also more flexible, adapting to user-specified features constraints while being substantially faster than existing baselines. This provides a principled and practical solution for reliable recourse under model uncertainty, ensuring stable recommendations for users even as models evolve.

1 Introduction

When an algorithmic decision denies someone a loan, a job, or insurance coverage, a natural question follows: *What could I change to obtain a better outcome next time?* Algorithmic recourse answers this question by providing concrete, actionable changes that could lead to a more favorable decision. A common way to generate such recommendations is through counterfactual explanations – small modifications to an individual’s features that flip the model’s prediction. Yet, even when the recommendation looks specific (e.g. “increase your income by \$5000”), one must ask: *Would that same change still work tomorrow if the institution retrain or replaces its model?* or *How stable are these suggestions across equally good models that explain the data in different ways?*

Most existing counterfactual generation methods [25, 46, 50, 52, 54, 57, 61, 66] implicitly assume that the underlying model is fixed and perfectly known. In practice, models evolve: banks regularly retrain risk predictors, healthcare institutions update diagnostic classifiers, and regulators may require model re-validation under new privacy or transparency constraints. Small shifts in data or regularization can result in very different-yet-equally-accurate models. This phenomenon, known as the Rashomon effect [8, 18, 23, 56, 58], implies that many distinct predictors achieve nearly optimal performance. In such settings, a counterfactual valid for one model can fail under another, undermining the reliability and consistency of algorithmic recourse.

*The majority of this work was conducted while Bohdan was at Taras Shevchenko National University of Kyiv and Lesia at Microsoft Research NYC.

Recent approaches have attempted to produce robust counterfactuals, meaning counterfactuals that are valid under small parameter perturbations or across predefined ensembles [17, 22, 27, 34, 36, 37, 41, 63]. However, these methods either rely on heavy-weight mixed-integer solvers, restrict robustness to local neighborhoods around a single model, or lack formal guarantees of validity across the full space of near-optimal solutions known as the Rashomon set. None of them directly leverages the geometry of this Rashomon set itself.

We introduce ElliCE, an efficient and provably robust framework for algorithmic recourse that works over an ellipsoidal approximation of the Rashomon set. By modeling the space of near-optimal models as an ellipsoid derived from the curvature (Hessian) of the loss landscape, ElliCE reformulates robust counterfactual generation as a tractable convex optimization problem. The resulting counterfactuals are valid for every model inside the ellipsoid, ensuring that a user’s recommended action remains meaningful even if the deployed model is replaced by another equally accurate one from the approximated Rashomon set.

Our contributions are fourfold: (1) *Theoretical foundation*. We derive a closed-form expression for the worst-case prediction, which allows us to formulate the robust recourse problem as a convex optimization and establish formal guarantees of validity, uniqueness, and stability for ElliCE’s counterfactuals. (2) *Geometric intuition*. We show that ElliCE’s robustness term connects the counterfactual’s stability with the importance of the features it modifies as the optimization naturally aligns recourse directions with the principal curvature axes of the loss landscape. (3) *Actionability*. ElliCE supports feature-level constraints, such as sparsity constraints, immutable or range-restricted attributes, allowing users to generate realistic, actionable recourse tailored to specific application or user settings. (4) *Empirical validation*. Across multiple high-stakes tabular datasets, ElliCE achieves higher robustness and remains one to three orders of magnitude faster than competing baselines, while maintaining proximity and plausibility.

Ultimately, ElliCE looks at algorithmic recourse through the lens of model multiplicity. Instead of relying on a single model’s decision boundary, it offers explanations that stay consistent across many models that fit the data almost equally well. This perspective treats the Rashomon Effect not as a flaw to eliminate, but as an inherent source of uncertainty to account for, leading to stable recourse in the presence of model diversity.

2 Related works

Rashomon Effect. The Rashomon Effect, a term popularized by Breiman [8] in the context of machine learning, describes the phenomenon where multiple distinct models can achieve near-optimal empirical risk (these models form a Rashomon set). This effect is also referred to as model multiplicity [5, 48]. The existence of the Rashomon set has implications for the trustworthiness and reliability of machine learning systems, influencing feature importance [14, 15, 20, 51], fairness [13, 45, 49], the existence of simple yet accurate models [6, 58, 59] to name a few. Significant research has focused on measuring and characterizing the Rashomon set for different model classes [29, 31, 32, 68, 70]. Our work leverages insights into the geometry of the Rashomon set, explored by works like Donnelly et al. [16], Zhong et al. [70], but applies them to the distinct challenge of generating robust algorithmic recourse across this set.

Counterfactual Explanations. Counterfactual Explanations (CEs) have emerged as a prominent tool for providing algorithmic recourse. Numerous approaches exist for generating CEs. Proximity-based methods aim for counterfactuals requiring minimal feature space perturbations [9, 50, 65, 66]. Sparsity techniques prioritize modifying the fewest features possible to enhance actionability [25, 61], while some methods attempt to balance both objectives [46]. Another research direction emphasizes plausibility, ensuring generated CEs represent realistic instances by constraining them to the data manifold, for example, using guidance from generative models [39, 52, 53], encoding feasibility rules [40], or tracing density-aware paths [54]. Recent extensions also incorporate temporal reasoning [11] and fairness objectives [4, 43, 69]. A key limitation across these approaches (which ElliCE directly addresses) is the assumption of a fixed, perfectly known predictive model, as counterfactuals constructed near a specific decision boundary can become unstable under model updates or perturbations.

Robustness to Local Model Perturbations. Building upon the limitation of fixed models, one line of work has focused specifically on achieving robustness against small, local changes or per-

turbations in the model’s parameters. For instance, ROAR [63] optimizes CEs considering local Δ -perturbations of the model. Jiang et al. [33] introduced Δ -robustness, a formal metric to assess CE validity under bounded parameter perturbations in neural networks, with subsequent works developing provably robust MILP-based methods [34, 36]. While these methods offer formal guarantees for Δ -robustness, MILP-based approaches can face scalability challenges, and the focus is generally on local parameter stability rather than the broader implications of the Rashomon effect.

Robustness under the Rashomon Effect. A growing body of work addresses counterfactual robustness under model multiplicity, aligning closely with the Rashomon Effect. Several approaches evaluate stability across predefined sets or ensembles of models, introducing heuristic stability measures (e.g., TRex [27] and RobX [17]), probabilistic frameworks [22, 41], or guarantees under specific norms and conditions like distribution shift [24, 42, 44]. Foundational work by Pawelczyk et al. [53] conceptually linked the Rashomon Effect to counterfactuals, though primarily enhancing input perturbation robustness. More recent methods use argumentative ensembling [37] or aggregate explanations across AutoML-generated sets [10] to handle model multiplicity.

Our work takes a distinct approach. Rather than relying on ensemble agreement, heuristic stability metrics, local perturbations, or argumentative aggregation, ElliCE leverages the local geometry of the Rashomon set, approximated by an ellipsoid, to derive theoretically grounded, robust recourse valid across all models within the approximation.

3 Background and Notation

Dataset and hypothesis space. Consider n i.i.d. samples $\mathcal{S}_n = \{\mathbf{z}_i = (\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y_i \in \mathcal{Y} = \{0, 1\}$ are generated from an unknown distribution \mathcal{D} on $\mathcal{X} \times \mathcal{Y}$. Let \mathcal{Y}_{pred} be an output space, where $\mathcal{Y}_{pred} \subseteq \mathbb{R}$ for scores (logits) or $\mathcal{Y}_{pred} \subseteq [0, 1]$ for probabilities. Then $\mathcal{F} = \{f_\theta : \theta \in \Theta\}$ is a hypothesis space of functions $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}_{pred}$, parameterized by a vector $\theta \in \Theta \subseteq \mathbb{R}^p$. For example, \mathcal{F} can represent linear models or multilayer perceptrons. We denote a specific function by f_θ . As our analysis focuses on the parameter space Θ , we will often refer to the model directly by its parameter vector θ .

Loss and objective function. Let $\phi : \mathcal{Y}_{pred} \times \mathcal{Y} \rightarrow \mathbb{R}_+$ be a loss function. In this work, we consider logistic loss $\phi(f_\theta(\mathbf{x}), y) = -[y \log(\sigma(f_\theta(\mathbf{x}))) + (1 - y) \log(1 - \sigma(f_\theta(\mathbf{x})))]$, which is applied to the model’s raw score (logit), $s = f_\theta(\mathbf{x})$, where $\sigma(s) = \frac{1}{1 + \exp(-s)}$ is the sigmoid function. However, our results generalize to other convex losses. The true risk is the expected loss $J(\theta) = \mathbb{E}_{\mathbf{z}}[\phi(f_\theta(\mathbf{x}), y)]$ that we approximate with the empirical risk, which is the average loss, $\hat{J}(\theta) = \frac{1}{n} \sum_{i=1}^n \phi(f_\theta(\mathbf{x}_i), y_i)$. We also define an ℓ_2 -regularized objective function: $\hat{L}(\theta) = \hat{J}(\theta) + \frac{\lambda}{2} \|\theta\|_2^2$, where $\lambda \geq 0$ is the regularization strength. The empirical risk minimizer (ERM) is $\hat{\theta} \in \arg \min_{\theta \in \Theta} \hat{L}(\theta)$. When $\lambda = 0$, the ERM is $\hat{\theta} \in \arg \min_{\theta \in \Theta} \hat{J}(\theta)$.

Rashomon set. Following [21, 58, 67], we define the ϵ -Rashomon set within the parameter space Θ as the set of parameter vectors whose corresponding models f_θ have objective value close to the minimum:

$$\mathcal{R}(\epsilon) := \{\theta \in \Theta : \hat{L}(\theta) \leq \hat{L}(\hat{\theta}) + \epsilon\},$$

where $\epsilon \geq 0$ is the Rashomon parameter defining the allowable tolerance in objective compared to the ERM, $\hat{L}(\hat{\theta})$. It is typically a small value. The existence of the Rashomon set with multiple, distinct parameter vectors θ (corresponding to potentially distinct functions f_θ) achieving similar performance implies that different underlying logic (how features contribute to predictions) can explain the data equally well. It is important to be aware of this variability among near-optimal models when generating explanations for individual predictions, as different models in $\mathcal{R}(\epsilon)$ might suggest different ways an outcome could be changed.

Counterfactual explanations. Let $g : \mathcal{Y}_{pred} \rightarrow \{0, 1\}$ be the decision function that converts a model’s score output $s = f_\theta(\mathbf{x})$ to a final binary class label by applying a threshold t , such that $g(s) = 1[s \geq t]$. For an ERM $\hat{\theta}$ and for an input vector \mathbf{x}_0 with prediction $g(f_{\hat{\theta}}(\mathbf{x}_0)) = \hat{y}_0$, a counterfactual explanation \mathbf{x}_c is a data point such that its predicted class is the opposite, i.e., $g(f_{\hat{\theta}}(\mathbf{x}_c)) = 1 - \hat{y}_0$. The set of all counterfactual explanations for \mathbf{x}_0 under the model $\hat{\theta}$ and decision function g is defined as:

$$\mathcal{C}(\mathbf{x}_0, \hat{\theta}) = \{\mathbf{x}_c \in \mathcal{X} : g(f_{\hat{\theta}}(\mathbf{x}_c)) = 1 - g(f_{\hat{\theta}}(\mathbf{x}_0))\}.$$

For instance, in a credit loan application scenario, if an applicant \mathbf{x}_0 is denied a loan (e.g., $g(f_{\hat{\theta}}(\mathbf{x}_0)) = 0$), a counterfactual explanation \mathbf{x}_c would be a modified version of their application details (e.g., increased income, reduced debt) such that the model predicts approval, $g(f_{\hat{\theta}}(\mathbf{x}_c)) = 1$. While many such \mathbf{x}_c might exist, practical algorithmic recourse aims to find explanations that require minimal change for the user. This means finding the “closest” counterfactual: $\mathbf{x}_c^* = \arg \min_{\mathbf{x}_c \in \mathcal{C}(\mathbf{x}_0, \hat{\theta})} \nu(\mathbf{x}_c, \mathbf{x}_0)$, where $\nu(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is a defined distance function or cost metric.

Distance Metrics. In our framework, we primarily focus on the two distance metrics for generating actionable and interpretable counterfactuals: ℓ_2 or Euclidean distance and mixed distance ℓ_{mix} . Note that ℓ_2 is a natural geometric measure of proximity, that penalizes large differences in any feature, $\ell_2(\mathbf{x}_c, \mathbf{x}_0) = \|\mathbf{x}_c - \mathbf{x}_0\|_2^2 = \sum_{j=1}^d (x_{cj} - x_{0j})^2$. For practical applications where features have different natures (continuous and categorical), one can also consider the mixed distance ℓ_{mix} , inspired by Gower’s distance. Assuming that the data are standardized, it is defined as: $\ell_{mix}(\mathbf{x}_c, \mathbf{x}_0) = \sqrt{\sum_{j \in \mathcal{I}_{cont}} (x_{cj} - x_{0j})^2 + \sum_{j \in \mathcal{I}_{cat}} w_j \mathbf{1}[x_{cj} \neq x_{0j}]}$, where \mathcal{I}_{cont} and \mathcal{I}_{cat} denote the sets of continuous and categorical feature indices respectively, $\mathbf{1}[\cdot]$ is the indicator function, and w_j are optional weights reflecting the cost of changing feature j . We use ℓ_2 distance for our theoretical analysis in the subsequent sections.

Next, we describe our approximating framework and outline the optimization process.

4 A Framework for Robust Recourse over the Rashomon Set

We focus our theoretical analysis on linear predictors of the form $f_{\theta}(\mathbf{x}) = \theta^\top \mathbf{x}$. However, the same methodology applies in the final embedding space of multilayer perceptrons (MLPs) by writing the model as $f_{\theta}(\mathbf{x}) = \theta^\top h(\mathbf{x})$, where $h(\mathbf{x})$ is the penultimate-layer embedding and θ are the last-layer parameters. We freeze $h(\cdot)$ and apply the same ellipsoidal procedure to θ as in the linear case (equivalently, replace \mathbf{x} by $h(\mathbf{x})$ in the formulas below).

Approximated Rashomon set. For certain objectives, such as ℓ_2 -regularized mean-squared error on linear models, the Rashomon set is exactly an ellipsoid in the parameter space [58]: $\mathcal{R}(\epsilon) = \{\theta : (\theta - \hat{\theta})^\top (X^\top X + \lambda I_p)(\theta - \hat{\theta}) \leq \epsilon\}$, where $X \in \mathbb{R}^{n \times d}$ is the data matrix, whose i -th row is the feature vector \mathbf{x}_i^\top , I_p is an identity matrix of size $p \times p$, and $\lambda \in \mathbb{R}_+$ is the regularization parameter. Because mean-squared error provides a local quadratic approximation to other convex losses, the exact ellipsoidal form of its Rashomon set serves as strong motivation for the Rashomon set approximation. Building on this and on similar geometric intuition [70], we approximate the ϵ -Rashomon set with an ellipsoid defined by the local geometry of the loss landscape:

$$\hat{\mathcal{R}}(\epsilon) = \{\theta : \frac{1}{2}(\theta - \hat{\theta})^\top H(\theta - \hat{\theta}) \leq \epsilon\},$$

where $H = X^\top W X + \lambda I_p$ is the Hessian of the ℓ_2 -regularized loss function, evaluated at $\hat{\theta}$. For logistic loss, W is an $n \times n$ diagonal matrix of weights where $w_{ii} = \sigma(f_{\hat{\theta}}(\mathbf{x}_i))(1 - \sigma(f_{\hat{\theta}}(\mathbf{x}_i)))$. Recall from Section 3 that $\sigma(\cdot)$ is the sigmoid function.

The Hessian matrix H of the regularized objective $\hat{L}(\theta)$ is strictly positive definite. This is because it is the sum of the positive semidefinite (PSD) Hessian from the convex logistic loss and the positive definite (PD) Hessian from the ℓ_2 regularization term (λI_p), assuming $\lambda > 0$. A positive definite Hessian is important for our framework, as it guarantees the approximating ellipsoid is well-defined and bounded, and ensures that H is invertible for our closed-form solution.

In cases where the unregularized risk $\hat{J}(\theta)$ is minimized (e.g., for neural networks), the resulting Hessian is only guaranteed to be PSD and may be singular. For these models, we ensure positive definiteness in practice by adding a small stabilization term, αI_p , $\alpha > 0$, to the computed Hessian, which is a standard technique to guarantee invertibility.

Optimization. To find a robust counterfactual explanation, we want to compute an explanation \mathbf{x}_c that is closest to the original point \mathbf{x}_0 while ensuring that its predicted outcome is above a target threshold t for all models within the approximated Rashomon set. In other words, for a given \mathbf{x}_0 , we look for a minimally modified (measured in some distance; we will use ℓ_2 here) \mathbf{x}_c , such that its predicted outcome achieves t even when evaluated by the least favorable model θ within the

approximated Rashomon set $\hat{\mathcal{R}}(\epsilon)$. Formally, this requirement leads to the following optimization problem:

$$\min_{\mathbf{x}_c} \|\mathbf{x}_c - \mathbf{x}_0\|_2^2 \quad \text{s.t.} \quad \min_{\boldsymbol{\theta} \in \hat{\mathcal{R}}(\epsilon)} \boldsymbol{\theta}^\top \mathbf{x}_c \geq t. \quad (1)$$

The inner minimization problem admits a closed-form solution, as we show next in Theorem 1. By reformulating the problem in this way, we get a tractable optimization framework that supports more efficient computation and analytical analysis of solution properties.

Theorem 1 (Closed-form solution). *For positive-definite Hessian H , the inner minimization problem over the ellipsoid-approximated Rashomon set $\hat{\mathcal{R}}(\epsilon)$ has the closed-form solution $\min_{\boldsymbol{\theta} \in \hat{\mathcal{R}}(\epsilon)} \boldsymbol{\theta}^\top \mathbf{x}_c = \hat{\boldsymbol{\theta}}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$. Moreover, for a given \mathbf{x}_c , the worst-case model $\boldsymbol{\theta}_{worst}(\mathbf{x}_c)$ that achieves this minimum is: $\boldsymbol{\theta}_{worst}(\mathbf{x}_c) = \hat{\boldsymbol{\theta}} - \sqrt{2\epsilon} \frac{H^{-1} \mathbf{x}_c}{\sqrt{\mathbf{x}_c^\top H^{-1} \mathbf{x}_c}}$.*

We prove Theorem 1 in Appendix A.1. As a direct consequence of Theorem 1, we obtain a practical criterion for verifying the robustness of a potential counterfactual. Specifically, since the theorem provides an explicit characterization of the output generated by the least favorable model $\boldsymbol{\theta} \in \hat{\mathcal{R}}(\epsilon)$ for a given \mathbf{x}_c , we can immediately determine if this \mathbf{x}_c achieves the target t across the entire set as we show in the following corollary.

Corollary 1. *A given counterfactual explanation \mathbf{x}_c is robust with respect to all models in the ellipsoid-approximated Rashomon set $\hat{\mathcal{R}}(\epsilon)$ against a target score t if and only if: $\hat{\boldsymbol{\theta}}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c} \geq t$.*

By substituting the closed-form solution from Theorem 1 into the original optimization problem (1), the robust counterfactual optimization problem becomes:

$$\min_{\mathbf{x}_c} \|\mathbf{x}_c - \mathbf{x}_0\|_2^2 \quad \text{s.t.} \quad \hat{\boldsymbol{\theta}}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c} \geq t. \quad (2)$$

The resulting problem is a quadratically constrained quadratic program (QCQP), which is a class of tractable convex optimization problems. We solve it efficiently using a gradient-based method. Leveraging the formulation (2), we implement two approaches for generating counterfactuals: a search-based method for generating data-supported counterfactuals lying on the data manifold, and a continuous optimization method for exploring potentially novel non-data supported solutions.

Continuous CE generation. For non-data supported counterfactuals, we solve the convex optimization problem in Equation (2) using a gradient-based approach for both linear models and multilayer perceptrons. This method directly optimizes for a counterfactual \mathbf{x}_c in the input space. For neural networks, the process is guided by the worst-case model $\boldsymbol{\theta}_{worst}(\mathbf{x}_c)$ identified in the final layer’s embedding space using Theorem 1, with the resulting gradients mapped back to the input features. The full details of this procedure are available in Appendix B.4.

Data-supported CE generation. For practical applications where counterfactuals should remain on the data manifold, we generate data-supported explanations based on the training set. Specifically, we evaluate the robust logit $\hat{\boldsymbol{\theta}}^\top \mathbf{x}_i - \sqrt{2\epsilon \mathbf{x}_i^\top H^{-1} \mathbf{x}_i}$ for each training data point \mathbf{x}_i using Theorem 1. Then, we compute the subset S_{stable} by filtering out points where this robust prediction exceeds the target threshold t . Finally, we use k-d tree nearest neighbor search within S_{stable} to identify the points closest to the input point \mathbf{x}_0 in terms of defined distance (for example, ℓ_2), which gives us a counterfactual that is both robust and lies on the data manifold.

The continuous approach offers flexibility by exploring the entire feature space for new solutions, while the data-supported approach guarantees plausibility by restricting solutions to observed examples. We evaluate the performance of both approaches in Section 6 and focus on theoretical guarantees of our framework next.

5 Theoretical Guarantees of Ellice Counterfactuals

In this section, we explore key theoretical properties of the counterfactual explanations generated under our framework. Note that we use ℓ_2 distance as target distance between \mathbf{x}_0 and \mathbf{x}_c . We show that the counterfactual explanations generated by our method are valid, unique, stable, and align

with important directions in the feature space. We focus on each of these properties separately and proofs of theorems provided in this section are in Appendix A.2.

Validity. By explicitly optimizing for the worst-case model θ_{worst} within the defined ellipsoid, any counterfactual \mathbf{x}_c generated by ElliCE is, by construction, valid for all models in the approximated Rashomon set. This inherent validity ensures that the provided recourse is faithful, regardless of which model from the approximated Rashomon set was selected.

Uniqueness. By Theorem 2, that we state next, any solution \mathbf{x}_c to the modified optimization problem (2) is unique. Because our objective is strictly convex and the approximated Rashomon set is characterized as an ellipsoid, for a given \mathbf{x}_0 , there can never be two distinct counterfactuals at the same ℓ_2 distance from the original \mathbf{x}_0 . In practical terms, this uniqueness guarantees that ElliCE provides a single solution for a given input and desired robustness level. This directly addresses and resolves “explanation multiplicity” [26], where multiple, distinct explanation paths might exist for a single input (at least for ℓ_2 distance).

Theorem 2 (Uniqueness). *If a solution \mathbf{x}_c to the optimization problem (2) exists, then \mathbf{x}_c is unique.*

Stability. The input data \mathbf{x}_0 is often subject to noise or minor variations. A desirable property is that such small changes in the input do not lead to drastically different counterfactuals. Our framework ensures this stability. Theorem 3 formally states that the process of generating robust counterfactuals is Lipschitz continuous with a constant of 1. This means that if the original input \mathbf{x}_0 is perturbed by a small amount δ to become \mathbf{x}'_0 , the resulting robust counterfactual \mathbf{x}'_c will not deviate from the original counterfactual \mathbf{x}_c by more than the magnitude of the initial perturbation $\|\delta\|_2$. This property guarantees the reliability of the explanations.

Theorem 3 (Stability). *Given an input \mathbf{x}_0 , let \mathbf{x}_c be the robust counterfactual solution for \mathbf{x}_0 . If the input is perturbed to $\mathbf{x}'_0 = \mathbf{x}_0 + \delta$, where $\delta \in \mathbb{R}^d$, and \mathbf{x}'_c is the robust counterfactual solution for \mathbf{x}'_0 , then $\|\mathbf{x}_c - \mathbf{x}'_c\|_2 \leq \|\delta\|_2$.*

Alignment with Important Feature Directions. An insightful explanation should not only provide a path to a different outcome but also highlight which features are most critical in achieving that change, particularly under model uncertainty. The robustness penalty term, $C_{rob}(\epsilon, \mathbf{x}_c) = \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$, plays a key role in this alignment. Theorem 4 formalizes the intuition that to minimize this penalty (and thus find an efficient robust counterfactual), the recourse direction \mathbf{x}_c should align with directions in the feature space that are most sensitive or influential, as captured by the eigenvectors of the Hessian matrix H . Specifically, under certain conditions, the penalty is minimized when the counterfactual aligns with the leading eigenvector of H , which often corresponds to the direction of greatest sensitivity. This encourages the counterfactual to suggest changes along features that have a significant impact, making the explanation more informative.

Theorem 4 (Alignment with Important Feature Directions). *Define the robustness penalty as $C_{rob}(\epsilon, \mathbf{x}_c) = \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$ for a symmetric positive definite Hessian H . Let λ_1 be the largest eigenvalue of H with corresponding eigenvector \mathbf{q}_1 , and assume that λ_1 is unique. Then the robustness penalty term $C_{rob}(\epsilon, \mathbf{x}_c)$ is minimized when the counterfactual vector \mathbf{x}_c is aligned with the eigenvector \mathbf{q}_1 .*

Price of robustness. Previous literature has observed the trade-off between robustness and proximity [22]. Indeed, intuitively, increasing robustness and ensuring validity across a larger set of potential models may require more changes to the input features, effectively increasing the proximity. This implies a “cost” for greater robustness that Theorem 5 formalizes.

Theorem 5 (Robustness-Proximity Trade-off). *For an input \mathbf{x}_0 such that $\hat{\theta}^\top \mathbf{x}_0 \leq t$, where $\hat{\theta}$ is ERM, let $\mathbf{x}_c^*(\epsilon)$ be the optimal robust counterfactual for a given robustness level $\epsilon > 0$, and let $\nu(\epsilon) = \|\mathbf{x}_c^*(\epsilon) - \mathbf{x}_0\|_2^2$ be its ℓ_2 distance from \mathbf{x}_0 . If $\nu(\epsilon_1) > 0$ and $\mathbf{x}_c^*(\epsilon_1) \neq \mathbf{0}$, then for any two robustness levels $0 < \epsilon_1 < \epsilon_2$, $\nu(\epsilon_1) < \nu(\epsilon_2)$.*

The practical impact of this trade-off is significant. Overly robust counterfactuals may become distant and unactionable, while insufficient robustness compromises recourse reliability under model shifts. This underscores the need for methods that efficiently explore this trade-off by achieving substantial robustness with reasonable proximity – a goal that ElliCE effectively meets.

When applying our theoretical results to MLPs, the validity guarantee is fully preserved in the input space, which is a key result. The formal guarantees for uniqueness (Theorem 2), stability

(Theorem 3), and the robustness-proximity trade-off (Theorem 5), however, depend on the convexity of the feasible set (see proof of Theorem 2). While this convexity is guaranteed in the embedding space $h(\mathbf{x})$, the nonlinear mapping from the input space ($\mathbf{x} \mapsto h(\mathbf{x})$) means it is not guaranteed to hold there. This distinction highlights a fundamental challenge for robust recourse in deep models and underscores that extending these formal guarantees to the input space is a promising direction for future work. Nonetheless, these theorems provide a principled geometric foundation for our approach and hold for linear models and embedding spaces. Next, we present empirical results showing that ElliCE’s performance is consistent with its theoretical guarantees.

6 Evaluation Pipeline and Experimental Results

In our evaluation pipeline, we work with the hypothesis space of linear models and multi-layer perceptrons. However, our results can be extended to other hypothesis spaces that can be optimized with gradient descent, such as neural additive models [1]. In this section, we empirically show that ElliCE is faster and more robust as compared to other methods that produce robust counterfactuals. Please see Appendix B for additional details and results.

Datasets. We consider nine datasets from high-stakes decision domains such as lending (Australian Credit [55], FICO [19], German Credit [28], Banknote [47]), healthcare (Parkinson’s [62], Diabetes [60]), and recidivism (COMPAS [2]), as well as benchmark datasets (Wine Quality [12], Extended Iris [3]). Please see Table 3 for detailed dataset descriptions and preprocessing notes. We used datasets with predominantly categorical features (FICO, Australian Credit, COMPAS, German Credit, Diabetes) for data-supported CE generation, and datasets with continuous features (Diabetes, Parkinson’s, Banknote, Iris, and Wine Quality) for continuous methods. We balanced the datasets, standardized continuous features, and, for some datasets, dropped rows with missing values.

Baselines. We compare ElliCE to other methods that are designed to generate robust counterfactual explanations, such as T:Rex, Interval Abstractions (we refer to it as Delta-robustness [33]), PROPLACE, and ROAR. *T:Rex* [27] generates robust counterfactuals for neural networks using a Stability measure that depends on variance. It quantifies robustness to naturally occurring model changes, providing probabilistic validity guarantees. It is a successor of RobX [17], which targets tree-based ensembles. *Interval Abstractions* [36] ensures that counterfactuals are robust to bounded changes in model parameters (weights and biases). It uses interval neural networks and mixed-integer linear programming. *PROPLACE* [35] formulates counterfactual generation as a bi-level robust optimization problem: it enforces plausibility by restricting solutions to the convex hull of realistic samples and uses interval bounds on neural networks to ensure robustness. *ROAR* [64] optimizes counterfactual validity under bounded model parameter perturbations using a robustness-constrained loss formulation. Most implementations of our baselines follow Jiang et al. [38].

Evaluators. Precisely computing the entire Rashomon set for the hypothesis spaces that we consider is intractable. Therefore, to evaluate the robustness and validity of counterfactual explanations generated by ElliCE and the baselines, we rely on established techniques that approximate or characterize this set. These approaches generate diverse collections of near-optimal models, each serving as a proxy for the true Rashomon set. Our evaluators include: *Random Retrain*, which retrains models multiple times with different random seeds to capture procedural variability. *Rashomon Dropout* [32], which applies binary dropout masks to a single trained neural network’s weights during inference, creating an ensemble of thinned sub-models. *Adversarial Weight Perturbation (AWP)* [30], which generates diverse models from an initially trained model by applying small perturbations to its weights. We define the objective tolerance (Rashomon parameter) for the evaluators as $\varepsilon_{\text{target}}$, which is distinct from ϵ . This separation ensures that the Rashomon set used for evaluation is controlled independently from the robustness tolerance ϵ used by ElliCE.

Metrics. We evaluate the generated counterfactual explanations based on four metrics: validity, proximity, robustness, and plausibility. *Validity* measures whether a generated counterfactual \mathbf{x}_c for a given input \mathbf{x}_0 successfully achieves the desired outcome c when evaluated on the original model f_{baseline} for which it was generated, $\text{Validity} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[f_{\text{baseline}}(\mathbf{x}_{ci}) = c]$. *Proximity* measures the closeness of a counterfactual \mathbf{x}_c to the original instance \mathbf{x}_0 . We primarily report the ℓ_2 distance: $\|\mathbf{x}_c - \mathbf{x}_0\|_2$. Lower values indicate less change required and are thus better. *Plausibility* checks whether the generated counterfactuals lie in realistic regions of the feature space. Our data-supported counterfactuals are inherently plausible, as they lie on the data manifold. For con-

tinuous approach, because ElliCE enforces robustness by pushing counterfactuals away from the decision boundary, the resulting counterfactuals tend to shift toward higher-density regions of the target class. Nevertheless, we evaluate plausibility using the Local Outlier Factor (LOF) [35], a standard outlier-detection metric. LOF values close to 1 indicate high plausibility, whereas larger values suggest the counterfactual is in a low-density region. *Robustness* computes whether the generated counterfactual \mathbf{x}_c remains valid (i.e., still achieves the desired outcome c) for all models within an evaluator ensemble $\tilde{\mathcal{R}}(\varepsilon_{\text{target}})$. Total is calculated as the average across all n counterfactual points: $\text{Robustness} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}[\forall f_{\theta} \in \tilde{\mathcal{R}}(\varepsilon_{\text{target}}), f_{\theta}(\mathbf{x}_{c_i}) = c]$. A higher robustness score (closer to 1) is better, indicating that more counterfactual explanations are robust to model changes.

Experimental Setup. For evaluators, we define a target multiplicity tolerance globally in range $\varepsilon_{\text{target}} \in [0, 0.1]$. We provide discussion on how to choose ElliCE’s ϵ in Appendix B. For every dataset, we performed 4-fold stratified cross-validation. Within each fold, the training data are further split into 80% for training and 20% for validation. The procedure within each inner fold is as follows: (1) We train a base model f_{baseline} , which serves as a reference model for all counterfactual generation methods. (2) Using f_{baseline} as a reference (if required by the evaluation method), we generate $\varepsilon_{\text{target}}$ -Rashomon set. (3) Multiplicity parameters (ϵ for ElliCE, δ for Delta Robustness, ROAR and PROPLACE, or τ for T:Rex) for each baseline are tuned via grid search on the validation set with a goal of maximizing validity. We allocate approximately the same amount of time for each method to tune its parameters with a hard maximum of 8 hours per method per data fold (as a result, we could not run PROPLACE for Parkinsons dataset). (4) Final performance metrics are reported on the held-out split of the outer fold. Note that due to our tuning procedure, we expect high validity metric for ElliCE and baselines. Indeed, for data-supported methods validity is consistently 100% across datasets, so we do not report it.

We conducted experiments on logistic regression and multilayer perceptrons. Consistent with prior work [34, 63], we focus on generating counterfactuals that change predicted labels from 0 to 1. Linear models are trained using Scikit-learn’s LBFGS solver with an ℓ_2 penalty (regularization parameter 0.001). MLPs are trained with the Adam optimizer (learning rate 0.001), early stopping, and ℓ_2 regularization parameter 0.001. For evaluation, we generate one counterfactual per method for each data point in the held-out set. Each counterfactual is then evaluated against the three evaluators (Random Retrain, Rashomon Dropout, AWP). The exact construction algorithms for these evaluators are described in Appendix B.2. Reported metrics are averaged across data points and folds, with plots displaying the mean and standard error.

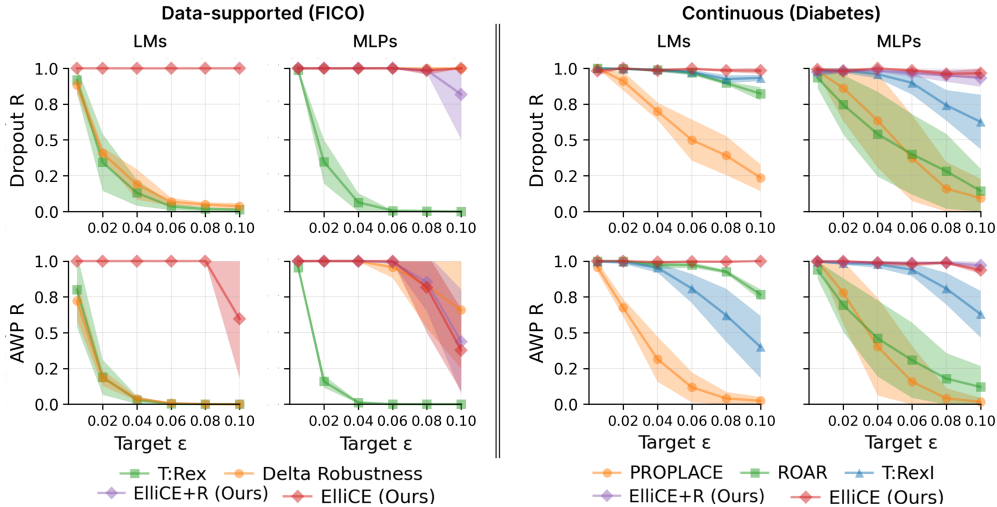


Figure 1: Robustness evaluation of ElliCE against baselines. The plot displays the robustness metric (y-axis) as a function of the target robustness level $\varepsilon_{\text{target}}$ for the evaluators (x-axis). ElliCE consistently outperforms all baselines across all robustness levels. See Appendix B for more figures. With ElliCE+R for MLPs we apply additional regularization to the Hessian, $\lambda = 0.1$, instead of 0.001.

Table 1: Performance of counterfactual methods on MLPs. For evaluators, we set $\varepsilon_{\text{target}}$ to 10% of the training objective ($\varepsilon_{\text{target}} = 0.1 \times \hat{L}(f_{\text{baseline}})$). **R** here stands for Robustness, **L2** for proximity, and PROP stands for PROPLACE. See Appendix B for results on other datasets.

Data	Method	Evaluation Metric					
		Retrain		Dropout Rashomon		AWP	
		R ↑	L2 ↓	R ↑	L2 ↓	R ↑	L2 ↓
Data-supported (DS)							
FICO	Ellice	1.00 ± 0.00	3.53 ± 0.17	1.00 ± 0.00	4.91 ± 0.22	1.00 ± 0.00	5.06 ± 0.29
	DeltaRob	1.00 ± 0.00	4.00 ± 0.10	1.00 ± 0.00	5.67 ± 0.58	0.96 ± 0.07	5.70 ± 0.72
	T:Rex	0.83 ± 0.08	3.12 ± 0.07	0.01 ± 0.00	3.07 ± 0.11	0.00 ± 0.00	2.77 ± 0.19
German	Ellice	1.00 ± 0.00	3.48 ± 0.10	1.00 ± 0.00	4.32 ± 0.31	1.00 ± 0.00	4.00 ± 0.24
	DeltaRob	0.98 ± 0.01	3.45 ± 0.06	0.99 ± 0.02	4.00 ± 0.15	1.00 ± 0.00	3.99 ± 0.22
	T:Rex	0.99 ± 0.01	3.47 ± 0.04	0.97 ± 0.02	4.03 ± 0.20	0.99 ± 0.01	4.23 ± 0.24
Continuous (CNT)							
Diabetes	Ellice	0.98 ± 0.01	2.15 ± 0.39	0.99 ± 0.02	3.05 ± 0.34	0.98 ± 0.02	3.22 ± 0.40
	PROP	0.48 ± 0.48	2.01 ± 0.05	0.19 ± 0.28	2.01 ± 0.05	0.08 ± 0.19	2.01 ± 0.05
	ROAR	0.86 ± 0.11	1.86 ± 0.24	0.40 ± 0.28	1.86 ± 0.24	0.31 ± 0.26	1.86 ± 0.24
	T:Rex	0.94 ± 0.03	2.47 ± 0.86	0.90 ± 0.08	4.18 ± 0.36	0.94 ± 0.04	4.18 ± 0.36

Table 2: Runtime performance and speedups for data-supported CE for MLP.

Dataset	Absolute (seconds)			Relative (speedup)	
	Ellice	T:Rex	Delta Rob	Over T:Rex	Over Delta Rob
FICO	1.792 ± 0.123	7.006 ± 0.058	242.035 ± 1.161	3.91×	135.04×
COMPAS	0.526 ± 0.011	3.534 ± 0.128	360.480 ± 6.701	6.72×	685.34×
Australian	0.057 ± 0.011	0.281 ± 0.006	2.783 ± 0.032	4.92×	48.64×
Diabetes	0.053 ± 0.001	0.296 ± 0.006	1.922 ± 0.032	5.60×	36.33×
German	0.101 ± 0.001	0.432 ± 0.013	9.905 ± 0.068	4.27×	97.88×

6.1 Ellice Generates Robust Counterfactuals

Figure 1 illustrates the relationship between the evaluators’ multiplicity level $\varepsilon_{\text{target}}$ and the achieved robustness for the baselines. We report results for both linear models and MLPs for data-supported and continuous methods. Across different settings, we observe that Ellice consistently produces more robust counterfactuals than baselines. Notably, Ellice’s counterfactuals generally do not exhibit a decrease in robustness as $\varepsilon_{\text{target}}$ increases, demonstrating stability under different levels of target multiplicity. This robustness, however, can sometimes come with a greater distance from the original instance (i.e., longer CEs), a trade-off that we saw in Section 5 and report in Table 1. For the MLP setting, our empirical results in Figure 1 and Table 1 suggest that Ellice’s ellipsoidal approximation offers good flexibility, allowing it to adapt to the underlying loss function’s shape.

6.2 Ellice is Efficient

Tables 2, 5 and 6 clearly demonstrates Ellice’s advantage in computational efficiency. Our method is consistently one to three orders of magnitude faster than baselines. Moreover, the runtimes of both T:Rex and Delta Robustness tend to grow substantially with the dataset size. In contrast, Ellice remains lightweight and exhibits better scalability. Across all datasets tested, Ellice’s absolute runtimes for generating a counterfactual remain under two seconds. This efficiency comes from a closed-form solution for the inner optimization problem (Theorem 1). The primary preprocessing cost involves computing and inverting the Hessian matrix H , requiring $O(np^2)$ for computation and $O(p^3)$ for inversion, performed once per model (where n is the training set size and p is the parameter dimension). Per-instance counterfactual generation then requires only $O(p^2)$ operations.

6.3 Sensitivity Analysis

Figure 2 (a,b) shows an empirical sensitivity analysis of Ellice’s robustness with respect to its internal Rashomon parameter ϵ . The plots show how the achieved robustness (evaluated against the Random Retrain and Ellipsoidal Rashomon set evaluators, respectively) varies as Ellice’s internal ϵ changes. These results illustrate that Ellice can achieve high levels of robustness even for rel-

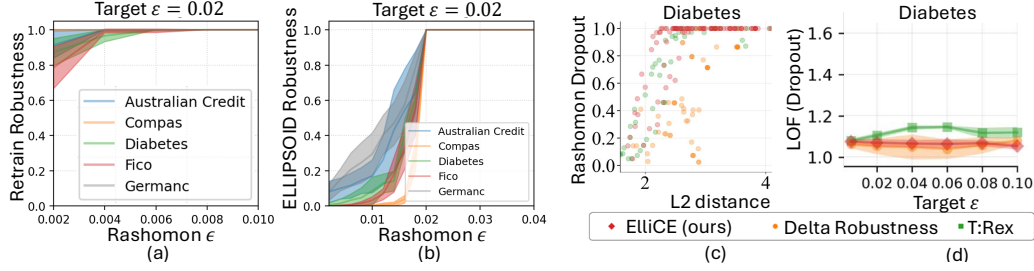


Figure 2: (a,b) Sensitivity of ElliCE’s robustness (y-axis) to its internal ϵ hyperparameter (x-axis). Robustness is evaluated against Random Retrain (a) and an Ellipsoidal Rashomon set approximation defined with a fixed ϵ_{target} (b). (c, d) Robustness vs. ℓ_2 proximity trade-off (c) and plausibility (d) of counterfactuals generated by ElliCE and baselines on Diabetes dataset.

atively small values of its internal ϵ when evaluated against the Retrain ensemble. For the middle plot (Ellipsoidal evaluator), while initial robustness may be lower for smaller internal ϵ values, the performance increases sharply, as ϵ approaches the targeted robustness level.

6.4 Robustness-Proximity Trade-off and Plausibility

Figure 2(c) illustrates the inherent trade-off between robustness and proximity for CEs generated by ElliCE, supporting our discussion in Section 5. While the trade-off occurs for all baselines, ElliCE achieves the highest robustness at a given length level. Understanding this trade-off is key to selecting counterfactuals that balance reliability under model shifts with practical user actionability. ElliCE provides a mechanism to navigate this by allowing control over its Rashomon parameter. We also observed good plausibility across all baselines and datasets, as supported by Figure 2(d) and 8. All LOF values tend to be close to 1, thus the generated counterfactuals lie on the data manifold.

6.5 Actionability

To ensure that generated recourse remains realistic and feasible, we incorporate actionability constraints that specify which features can change and within what ranges. ElliCE supports restrictions on features, including immutable features (e.g., age, citizenship) as well as range and direction constraints such as income or loan duration. It also allows for sparse counterfactuals by adding an optional penalty on the number of modified features. For instance, before applying actionability, one robust counterfactual on the German Credit dataset suggested changing the applicant’s age, an immutable feature. After enforcing immutability and sparsity constraints, ElliCE instead adjusted the credit amount and credit length, reducing both and thus lowering the predicted credit risk, which is reasonable in the lending context. Further details are provided in Appendix D.

7 Conclusions, Implications and Limitations

Standard algorithmic recourse is fragile. A recommendation given to a user today may become invalid tomorrow if the underlying model is retrained or replaced – a common scenario under the Rashomon effect. This paper addressed this reliability gap by introducing ElliCE, a framework that provides recourse with provable robustness guarantees. ElliCE approximates the set of near-optimal models with an ellipsoid and computes counterfactuals that remain valid across this approximated Rashomon set. A strength of ElliCE is its support for actionability. Users can specify immutable features, range or direction constraints, and optional sparsity penalties, ensuring that the resulting recourse is both robust and realistic. This flexibility might help prevent impractical or unethical recommendations and gives users greater control over actions. While robustness alone does not ensure fairness, user-specified actionability constraints can help to ensure that counterfactuals remain feasible and ethically sound. A comprehensive fairness analysis remains an important direction for future work. The ellipsoidal approximation, while efficient, is a simplification of the true Rashomon set, and for neural networks our analysis currently captures local rather than global model multiplicity. Despite these limitations, ElliCE provides a practical and theoretically grounded tool for robust and actionable recourse, providing stable and trustworthy advice.

Acknowledgments

We thank the RAI for Ukraine program, led by the Center for Responsible AI at New York University in collaboration with Ukrainian Catholic University in Lviv, for supporting Bohdan’s and Iryna’s participation in this research.

Code Availability

Implementations of ElliCE are available at https://github.com/BogdanTurbal/ElliCE_EXPERIMENTS.

References

- [1] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:4699–4711, 2021.
- [2] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There’s software used across the country to predict future criminals. and it’s biased against blacks. *ProPublica*, May 2016. [Newspaper article].
- [3] Samy Baladram. Iris dataset extended, 2023.
- [4] Ainhize Barrainkua, Giovanni De Toni, Jose Antonio Lozano, and Novi Quadrianto. Who pays for fairness? rethinking recourse under social burden. *arXiv preprint arXiv:2509.04128*, 2025.
- [5] Emily Black, Manish Raghavan, and Solon Barocas. Model multiplicity: Opportunities, concerns, and solutions. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 850–863, 2022.
- [6] Zachery Boner, Harry Chen, Lesia Semenova, Ronald Parr, and Cynthia Rudin. Using noise to infer aspects of simplicity without learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- [7] Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [8] Leo Breiman. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199–231, 2001.
- [9] Dieter Brughmans, Pieter Leyman, and David Martens. NICE: An algorithm for nearest instance counterfactual explanations. *CoRR*, abs/2104.07411, 2021. doi: 10.48550/arXiv.2104.07411.
- [10] Mustafa Cavus, Jan N van Rijn, and Przemysław Biecek. Beyond the single-best model: Rashomon partial dependence profile for trustworthy explanations in automl. In *International Conference on Discovery Science*, pages 445–459. Springer, 2025.
- [11] Marina Ceccon, Alessandro Fabris, Goran Radanović, Asia J Biega, and Gian Antonio Susto. Reinforcement learning for durable algorithmic recourse. *arXiv preprint arXiv:2509.22102*, 2025.
- [12] Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: <https://doi.org/10.24432/C56S3T>.
- [13] Gordon Dai, Pavan Ravishankar, Rachel Yuan, Daniel B. Neill, and Emily Black. Be intentional about fairness!: Fairness, size, and multiplicity in the rashomon set. *CoRR*, abs/2501.15634, 2025. doi: 10.48550/arXiv.2501.15634.
- [14] Jiayun Dong and Cynthia Rudin. Exploring the cloud of variable importance for the set of all good models. *Nature Machine Intelligence*, 2(12):810–824, 2020.

- [15] Jon Donnelly, Srikar Katta, Cynthia Rudin, and Edward P Browne. The rashomon importance distribution: Getting RID of unstable, single model-based variable importance. In *Neural Information Processing Systems (NeurIPS)*, 2023.
- [16] Jon Donnelly, Zhicheng Guo, Alina Jade Barnett, Hayden McTavish, Chaofan Chen, and Cynthia Rudin. Rashomon sets for prototypical-part networks: Editing interpretable models in real-time. *arXiv preprint arXiv:2503.01087*, 2025.
- [17] Sanghamitra Dutta, Jason Long, Saumitra Mishra, Cecilia Tilli, and Daniele Magazzeni. Robust counterfactual explanations for tree-based ensembles. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5742–5756. PMLR, Jul 2022.
- [18] Alexander D’Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 2020.
- [19] Fair Isaac Corporation (FICO). FICO explainable machine learning challenge: Home equity line of credit (heloc) dataset, 2018.
- [20] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.
- [21] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.
- [22] Alexandre Forel, Axel Parmentier, and Thibaut Vidal. Don’t explain noise: Robust counterfactuals for randomized ensembles. *CoRR*, abs/2205.14116, 2022. doi: 10.48550/arXiv.2205.14116.
- [23] Prakhar Ganesh, Afaf Taik, and Golnoosh Farnadi. Systemizing multiplicity: The curious case of arbitrariness in machine learning. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 8, pages 1032–1048, 2025.
- [24] Prateek Garg, Lokesh Nagalapatti, and Sunita Sarawagi. From search to sampling: Generative models for robust algorithmic recourse. *arXiv preprint arXiv:2505.07351*, 2025.
- [25] Rory Mc Grath, Luca Costabello, Chan Le Van, Paul Sweeney, Farbod Kamiab, Zhao Shen, and Freddy Lecue. Interpretable credit application predictions with counterfactual explanations. *CoRR*, abs/1811.05245, 2018. doi: 10.48550/arXiv.1811.05245.
- [26] Abirami Gunasekaran, Pritesh Mistry, and Minsi Chen. Which explanation should be selected: A method agnostic model class reliance explanation for model and explanation multiplicity. *SN Computer Science*, 5:503, 2024. doi: 10.1007/s42979-024-02810-8.
- [27] Faisal Hamman, Erfaun Noorani, Saumitra Mishra, Daniele Magazzeni, and Sanghamitra Dutta. Robust counterfactual explanations for neural networks with probabilistic guarantees. *CoRR*, abs/2305.11997, 2023. doi: 10.48550/arXiv.2305.11997. International Conference on Machine Learning (ICML), 2023.
- [28] Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994. DOI: <https://doi.org/10.24432/C5NC77>.
- [29] Hsiang Hsu and Flavio Calmon. Rashomon capacity: A metric for predictive multiplicity in classification. In *Neural Information Processing Systems (NeurIPS)*, volume 35, pages 28988–29000, 2022.
- [30] Hsiang Hsu and Flavio du Pin Calmon. Rashomon capacity: A metric for predictive multiplicity in classification. *CoRR*, abs/2206.01295, 2022. doi: 10.48550/arXiv.2206.01295. NeurIPS 2022 camera-ready version.

- [31] Hsiang Hsu, Ivan Brugere, Shubham Sharma, Freddy Lecue, and Richard Chen. Rashomongb: Analyzing the rashomon effect and mitigating predictive multiplicity in gradient boosting. *Advances in Neural Information Processing Systems*, 37:121265–121303, 2024.
- [32] Hsiang Hsu, Guihong Li, Shaohan Hu, and Chun-Fu Chen. Dropout-based rashomon set exploration for efficient predictive multiplicity estimation. In *The Twelfth International Conference on Learning Representations*, 2024.
- [33] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Formalising the robustness of counterfactual explanations for neural networks. *CoRR*, abs/2208.14878, 2022. doi: 10.48550/arXiv.2208.14878.
- [34] Junqi Jiang, Jianglin Lan, Francesco Leofante, Antonio Rago, and Francesca Toni. Provably robust and plausible counterfactual explanations for neural networks via robust optimisation. *CoRR*, abs/2309.12545, 2023. doi: 10.48550/arXiv.2309.12545. Accepted at ACML 2023, camera-ready version.
- [35] Junqi Jiang, Jianglin Lan, Francesco Leofante, Antonio Rago, and Francesca Toni. Provably robust and plausible counterfactual explanations for neural networks via robust optimisation. In Berrin Yanikoğlu and Wray Buntine, editors, *Proceedings of the 15th Asian Conference on Machine Learning*, volume 222 of *Proceedings of Machine Learning Research*, pages 582–597. PMLR, 11–14 Nov 2024.
- [36] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Interval abstractions for robust counterfactual explanations. *CoRR*, abs/2404.13736, 2024. doi: 10.48550/arXiv.2404.13736. Published in Artificial Intelligence Journal.
- [37] Junqi Jiang, Francesco Leofante, Antonio Rago, and Francesca Toni. Recourse under model multiplicity via argumentative ensembling. In Natasha Alechina, Virginia Dignum, Mehdi Dastani, and Juan S. Sichman, editors, *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024)*. International Foundation for Autonomous Agents and Multiagent Systems, 2024.
- [38] Junqi Jiang, Luca Marzari, Aaryan Purohit, and Francesco Leofante. Robustx: Robust counterfactual explanations made easy. *arXiv preprint arXiv:2502.13751*, 2025.
- [39] Shalmali Joshi, Oluwasanmi Koyejo, Warut Vijitbenjaronk, Been Kim, and Joydeep Ghosh. Towards realistic individual recourse and actionable explanations in black-box decision making systems. *CoRR*, abs/1907.09615, 2019. doi: 10.48550/arXiv.1907.09615.
- [40] Amir-Hossein Karimi, Gilles Barthe, Borja Balle, and Isabel Valera. Model-agnostic counterfactual explanations for consequential decisions. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pages 895–905. PMLR, Aug 2020.
- [41] Keita Kinjo. Robust counterfactual explanations under model multiplicity using multi-objective optimization. *CoRR*, abs/2501.05795, 2025. doi: 10.48550/arXiv.2501.05795. Submitted to ICLR 2025.
- [42] Gunnar König, Hidde Fokkema, Timo Freiesleben, Celestine Mandler-Dünner, and Ulrike von Luxburg. Performative validity of recourse explanations. *arXiv preprint arXiv:2506.15366*, 2025.
- [43] Alejandro Kuratomi, Zed Lee, Panayiotis Tsaparas, Evaggelia Pitoura, Tony Lindgren, Guilherme Dinis Junior, and Panagiotis Papapetrou. Subgroup fairness based on shared counterfactuals. *Knowledge and Information Systems*, pages 1–39, 2025.
- [44] Phone Kyaw, Kshitij Kayastha, and Shahin Jabbari. Optimal robust recourse with l^p - bounded model change. *arXiv preprint arXiv:2509.21293*, 2025.
- [45] Lucas Langlade, Julien Ferry, Gabriel Laberge, and Thibaut Vidal. Fairness and sparsity within rashomon sets: Enumeration-free exploration and characterization. *arXiv preprint arXiv:2502.05286*, 2025.

- [46] Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detryniecki. Inverse classification for comparison-based interpretability in machine learning. *CoRR*, abs/1712.08443, 2017. doi: 10.48550/arXiv.1712.08443.
- [47] Volker Lohweg. Banknote Authentication. UCI Machine Learning Repository, 2012. DOI: <https://doi.org/10.24432/C55P57>.
- [48] Charles Marx, Flavio Calmon, and Berk Ustun. Predictive multiplicity in classification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 6765–6774, 2020.
- [49] Anna P Meyer, Yea-Seul Kim, Aws Albarghouthi, and Loris D’Antoni. Perceptions of the fairness impacts of multiplicity in machine learning. *arXiv preprint arXiv:2409.12332*, 2024.
- [50] Kiarash Mohammadi, Amir-Hossein Karimi, Gilles Barthe, and Isabel Valera. Scaling guarantees for nearest counterfactual explanations. *CoRR*, abs/2010.04965, 2020. doi: 10.48550/arXiv.2010.04965.
- [51] Sebastian Müller, Vanessa Toborek, Katharina Beckh, Matthias Jakobs, Christian Bauckhage, and Pascal Welke. An empirical evaluation of the rashomon effect in explainable machine learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 462–478. Springer, 2023.
- [52] Daniel Nemirowsky, Nicolas Thiebaud, Ye Xu, and Abhishek Gupta. CounterGAN: Generating counterfactuals for real-time recourse and interpretability using residual GANs. In James Cussens and Kun Zhang, editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 1488–1497. PMLR, Aug 2022.
- [53] Martin Pawelczyk, Klaus Broelemann, and Gjergji Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020 (WWW ’20)*, pages 3126–3132. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380087.
- [54] Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijn De Bie, and Peter Flach. FACE: Feasible and actionable counterfactual explanations. *CoRR*, abs/1909.09369, 2019. doi: 10.48550/arXiv.1909.09369. Presented at AAAI/ACM Conference on AI, Ethics, and Society 2020.
- [55] Ross Quinlan. Statlog (Australian Credit Approval). UCI Machine Learning Repository, 1987. DOI: <https://doi.org/10.24432/C59012>.
- [56] Cynthia Rudin, Chudi Zhong, Lesia Semenova, Margo Seltzer, Ronald Parr, Jiachang Liu, Srikanth Katta, Jon Donnelly, Harry Chen, and Zachery Boner. Amazing things come from having many good models. *arXiv preprint arXiv:2407.04846*, 2024.
- [57] Chris Russell. Efficient search for diverse coherent explanations. *CoRR*, abs/1901.04909, 2019. doi: 10.48550/arXiv.1901.04909.
- [58] Lesia Semenova, Cynthia Rudin, and Ronald Parr. On the existence of simpler machine learning models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1827–1858, 2022.
- [59] Lesia Semenova, Harry Chen, Ronald Parr, and Cynthia Rudin. A path to simpler models starts with noise. In *Advances in Neural Information Processing Systems*, 2023.
- [60] John W. Smith, William A. Everhart, William C. Dickson, William C. Knowler, and Richard S. Johannes. Pima Indians Diabetes Database, 1988.
- [61] Yiyang Sun, Zhi Chen, Vittorio Orlandi, Tong Wang, and Cynthia Rudin. Sparse and faithful explanations without sparse models. *CoRR*, abs/2402.09702, 2024. doi: 10.48550/arXiv.2402.09702. Accepted in AISTATS 2024.
- [62] Athanasios Tsanas and Max Little. Parkinsons Telemonitoring. UCI Machine Learning Repository, 2009. DOI: <https://doi.org/10.24432/C5ZS3N>.

- [63] Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. Towards robust and reliable algorithmic recourse. *CoRR*, abs/2102.13620, 2021. doi: 10.48550/arXiv.2102.13620. Last revised July 13, 2021.
- [64] Sohini Upadhyay, Shalmali Joshi, and Himabindu Lakkaraju. Towards robust and reliable algorithmic recourse. *arXiv preprint arXiv:2102.13620*, 2021.
- [65] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. *CoRR*, abs/1809.06514, 2018. doi: 10.48550/arXiv.1809.06514. Extended version. ACM Conference on Fairness, Accountability and Transparency (FAT* 2019).
- [66] Sandra Wachter, Brent D. Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *CoRR*, abs/1711.00399, 2017. doi: 10.48550/arXiv.1711.00399.
- [67] Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the whole Rashomon set of sparse decision trees. In *Neural Information Processing Systems (NeurIPS)*, volume 35, pages 14071–14084, 2022.
- [68] Rui Xin, Chudi Zhong, Zhi Chen, Takuya Takagi, Margo Seltzer, and Cynthia Rudin. Exploring the whole rashomon set of sparse decision trees. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- [69] Jayanth Yetukuri, Ian Hardy, Yevgeniy Vorobeychik, Berk Ustun, and Yang Liu. Providing fair recourse over plausible groups. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21753–21760, Mar 2024. doi: 10.1609/aaai.v38i19.30175.
- [70] Chudi Zhong, Zhi Chen, Jiachang Liu, Margo Seltzer, and Cynthia Rudin. Exploring and interacting with the set of good sparse generalized additive models. In *Advances in Neural Information Processing Systems*, 2023.

NeurIPS Paper Checklist

1. Claims

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

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

Justification: The claims in the abstract and introduction are consistent with the paper’s scope.

Guidelines:

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

2. Limitations

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

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

Justification: We discussed the limitation in the conclusion section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate “Limitations” section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren’t acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

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

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

Justification: We provide the theorems in the main paper and proofs in the Appendix.

Guidelines:

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

4. Experimental result reproducibility

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

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

Justification: Experimental setups are detailed in the Experimental Section and the Appendix. The code is available in the supplement.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

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

Answer: [Yes]

Justification: All datasets used are publicly available (Australian Credit, COMPAS, Diabetes, FICO, German Credit). The paper provides sufficient algorithmic details and experimental settings to reproduce results. Code is available in the supplement.

Guidelines:

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

6. Experimental setting/details

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

Answer: [Yes]

Justification: See Section 6.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [Yes]

Justification: We report mean and standard deviation over multiple runs.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [Yes]

Justification: Please see Appendix.

Guidelines:

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

9. Code of ethics

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

Answer: [Yes]

Justification: Our work adheres to the NeurIPS Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [Yes]

Justification: We have mentioned the broader impact in the introduction and conclusion.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: Our paper doesn't release models that have the potential to cause harm.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: We use open access datasets and baselines and cite the sources of all the datasets and baselines we used in the paper.⁸³⁵

Guidelines:

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

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New assets**

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

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

Justification: We provide the code for this paper.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

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

Justification: The paper does not involve crowd-sourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

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

Justification: The paper does not involve crowd-sourcing nor research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: We used LLM for editing and improving the clarity of wording.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

Appendix

Contents

A	Proofs for Theoretical Results in Sections 4 and 5	24
A.1	Proof for Theorem 1	24
A.2	Proof for Theorem 2	24
A.3	Proof for Theorem 3	25
A.4	Proof for Theorem 4	25
A.5	Proof for Theorem 5	26
B	Additional Experiments	27
B.1	Datasets	27
B.2	Empirical Rashomon Set Construction	27
B.3	Computation Resources	28
B.4	Optimization	28
B.5	Hyperparameter Tuning	28
B.6	The Choice of ϵ Parameter for ElliCE	31
B.7	Experiments for Data-Supported Counterfactual Generation	33
B.8	Robustness & length tradeoff	35
B.9	Experiments for Non-data Supported Counterfactual Generation	35
B.10	Data shift	36
C	Delta-Robustness Works in the Last Layer under Reparameterization	36
C.1	Problem Setting	37
C.2	Necessary Conditions for Delta-Robustness	37
C.3	The Impact of Model Reparameterization	42
D	Enhanced Algorithmic Recourse: Actionability, Sparsity, and Extended Applications	44
D.1	Plausibility Evaluation	45
D.2	Actionability Enhancements	45
D.3	Sparsity Control	46
D.4	Multi-Class Extension	46
D.5	Mixed Feature Types with Gumbel-Softmax	47

A Proofs for Theoretical Results in Sections 4 and 5

In this appendix we provide the proof of theoretical results provided in Sections 4 and 5.

A.1 Proof for Theorem 1

We state and prove Theorem 1 below.

Theorem 1 (Closed-form solution). *For positive-definite Hessian H , the inner minimization problem over the ellipsoid-approximated Rashomon set $\hat{\mathcal{R}}(\epsilon)$ has the closed-form solution $\min_{\theta \in \hat{\mathcal{R}}(\epsilon)} \theta^\top \mathbf{x}_c = \hat{\theta}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$. Moreover, for a given \mathbf{x}_c , the worst-case model $\theta_{\text{worst}}(\mathbf{x}_c)$ that achieves this minimum is: $\theta_{\text{worst}}(\mathbf{x}_c) = \hat{\theta} - \sqrt{2\epsilon} \frac{H^{-1} \mathbf{x}_c}{\sqrt{\mathbf{x}_c^\top H^{-1} \mathbf{x}_c}}$.*

Proof. Recall that H is positive definite, therefore $H^{1/2}$ is well-defined, symmetric, and invertible. Let $\mathbf{v} = H^{1/2}(\theta - \hat{\theta})$ and $\xi = H^{-1/2} \mathbf{x}_c$. Then, we can reformulate our inner optimization problem using \mathbf{v} and ξ as variables.

First, denote $\theta - \hat{\theta} = H^{-1/2} \mathbf{v}$, then we can substitute $\theta - \hat{\theta} = H^{-1/2} \mathbf{v}$ into the constraint $\frac{1}{2}(\theta - \hat{\theta})^\top H(\theta - \hat{\theta}) \leq \epsilon$ to get:

$$\begin{aligned} (H^{-1/2} \mathbf{v})^\top H (H^{-1/2} \mathbf{v}) &= \mathbf{v}^\top (H^{-1/2})^\top H H^{-1/2} \mathbf{v} \\ &= \mathbf{v}^\top H^{-1/2} H H^{-1/2} \mathbf{v} \\ &= \mathbf{v}^\top I \mathbf{v} = \mathbf{v}^\top \mathbf{v} = \|\mathbf{v}\|_2^2. \end{aligned}$$

Second, recall that $\xi = H^{-1/2} \mathbf{x}_c$, then we have that:

$$\theta^\top \mathbf{x}_c = (\hat{\theta} + H^{-1/2} \mathbf{v})^\top \mathbf{x}_c = \hat{\theta}^\top \mathbf{x}_c + \xi^\top \mathbf{v}.$$

The original optimization problem is now entirely in terms of \mathbf{v} . Since $\hat{\theta}^\top \mathbf{x}_c$ is a constant with respect to \mathbf{v} , the optimization problem becomes:

$$\hat{\theta}^\top \mathbf{x}_c + \min_{\|\mathbf{v}\|_2 \leq \sqrt{2\epsilon}} \mathbf{v}^\top \xi.$$

Note that we are looking for a minimum of a linear function over an Euclidean ball in terms of \mathbf{v} . This problem of minimizing a linear function $\mathbf{a}^\top \mathbf{x}$ subject to an ℓ_2 -norm constraint $\|\mathbf{x}\|_2 \leq B$ has a well-known closed-form solution [7]. The optimal value is $-B\|\mathbf{a}\|_2$, achieved at $\hat{\mathbf{x}} = -B \frac{\mathbf{a}}{\|\mathbf{a}\|_2}$ (for $\mathbf{a} \neq \mathbf{0}$). Translating to \mathbf{v} and ξ , we get that the optimal value of the \mathbf{v} -minimization is $-\sqrt{2\epsilon}\|\xi\|_2$, achieved at $\hat{\mathbf{v}} = -\sqrt{2\epsilon} \frac{\xi}{\|\xi\|_2}$ (for $\xi \neq \mathbf{0}$).

And translating to the original problem formulation, we get that the overall optimal value is $\hat{\theta}^\top \mathbf{x}_c - \sqrt{2\epsilon}\|H^{-1/2} \mathbf{x}_c\|_2$. The optimal θ_{worst} is achieved at:

$$\theta_{\text{worst}} = \hat{\theta} - \sqrt{2\epsilon} \left(\frac{H^{-1} \mathbf{x}_c}{\|H^{-1/2} \mathbf{x}_c\|_2} \right).$$

Note that if $\mathbf{x}_c = \mathbf{0}$, then the solutions is $\theta_{\text{worst}} = \hat{\theta}$. □

The corollary 1 follows immediately from the above theorem. It allows us to easily check if the counterfactual explanation is robust given the ellipsoid-based Rashomon set, even if this explanation was generated by another method.

A.2 Proof for Theorem 2

We state and prove Theorem 2 below.

Theorem 2 (Uniqueness). *If a solution \mathbf{x}_c to the optimization problem (2) exists, then \mathbf{x}_c is unique.*

Proof. The objective function $\|\mathbf{x}_c - \mathbf{x}_0\|_2^2$ is strictly convex as a squared Euclidean norm.

Consider the constraint function $g_c(\mathbf{x}_c) = \hat{\boldsymbol{\theta}}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$. The first term, $\hat{\boldsymbol{\theta}}^\top \mathbf{x}_c$, is linear and thus concave. The second term is convex, since $\sqrt{\mathbf{x}_c^\top H^{-1} \mathbf{x}_c} = \|H^{-1/2} \mathbf{x}_c\|_2$ is a norm, which is convex given that H and H^{-1} is positive definite. Given that $\epsilon \geq 0$, $g_c(\mathbf{x}_c)$ is the sum of a concave function ($\hat{\boldsymbol{\theta}}^\top \mathbf{x}_c$) and the concave function ($-\sqrt{2\epsilon} \|H^{-1/2} \mathbf{x}_c\|_2$), which means $g_c(\mathbf{x}_c)$ is concave.

Therefore, we get that the feasible set $S = \{\mathbf{x}_c \mid g_c(\mathbf{x}_c) \geq t\}$ is a convex set for a threshold t . Minimizing a strictly convex function over a convex set guarantees that if a solution exists, it is unique. \square

A.3 Proof for Theorem 3

We prove Theorem 3 after first proving a helping Lemma about the set of feasible solutions S defined in the proof of Theorem 2 above.

Lemma 1. *For $H \succ 0$ and $\epsilon \geq 0$, the feasible set $S = \{\mathbf{x}_c \mid \hat{\boldsymbol{\theta}}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c} \geq t\}$ is closed and convex. Furthermore, if a solution to the optimization problem (2) exists, S is non-empty.*

Proof. Let $g_c(\mathbf{x}_c) = \hat{\boldsymbol{\theta}}^\top \mathbf{x}_c - \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$, then the feasible set is $S = \{\mathbf{x}_c \mid g_c(\mathbf{x}_c) \geq t\}$. We already showed in the proof of Theorem 2 that S is a convex set. It must be non-empty, so that a solution to the optimization problem (2) exists. Therefore, we next focus on showing that S is closed.

Notice that $g_c(\mathbf{x}_c)$ is continuous, as it is being composed of differences and compositions of continuous functions. Since $g_c(\mathbf{x}_c)$ is continuous, the set $S = \{\mathbf{x}_c \mid g_c(\mathbf{x}_c) \geq t\}$ is closed. \square

Now, we state and prove Theorem 3

Theorem 3 (Stability). *Given an input \mathbf{x}_0 , let \mathbf{x}_c be the robust counterfactual solution for \mathbf{x}_0 . If the input is perturbed to $\mathbf{x}'_0 = \mathbf{x}_0 + \boldsymbol{\delta}$, where $\boldsymbol{\delta} \in \mathbb{R}^d$, and \mathbf{x}'_c is the robust counterfactual solution for \mathbf{x}'_0 , then $\|\mathbf{x}_c - \mathbf{x}'_c\|_2 \leq \|\boldsymbol{\delta}\|_2$.*

Proof. Let $S = \{\mathbf{x} \mid \hat{\boldsymbol{\theta}}^\top \mathbf{x} - \sqrt{2\epsilon \mathbf{x}^\top H^{-1} \mathbf{x}} \geq t\}$ denote the feasible set for the robust counterfactual solutions. According to Lemma 1, S is a non-empty, closed, and convex set.

The robust counterfactual solution \mathbf{x}_c corresponding to an input \mathbf{x}_0 minimizes $\|\mathbf{x} - \mathbf{x}_0\|_2^2$ for $\mathbf{x} \in S$. Thus, \mathbf{x}_c is the Euclidean projection of \mathbf{x}_0 onto S . Let $P_S(\cdot)$ denote this projection operator. Then, we have:

$$\mathbf{x}_c = P_S(\mathbf{x}_0), \quad \mathbf{x}'_c = P_S(\mathbf{x}'_0).$$

The Euclidean projection P_S onto a non-empty, closed, convex set S is 1-Lipschitz continuous. Given that $\|\mathbf{x}_0 - \mathbf{x}'_0\|_2 = \|\boldsymbol{\delta}\|_2$ since $\mathbf{x}'_0 = \mathbf{x}_0 + \boldsymbol{\delta}$, we get:

$$\begin{aligned} \|\mathbf{x}_c - \mathbf{x}'_c\|_2 &= \|P_S(\mathbf{x}_0) - P_S(\mathbf{x}'_0)\|_2 \\ &\leq \|\mathbf{x}_0 - \mathbf{x}'_0\|_2 = \|\boldsymbol{\delta}\|_2. \end{aligned}$$

Therefore, we received that: $\|\mathbf{x}_c - \mathbf{x}'_c\|_2 \leq \|\boldsymbol{\delta}\|_2$. \square

Next we focus on proving that our counterfactual solutions are aligned with directions of the important features.

A.4 Proof for Theorem 4

We state and prove Theorem 4 below.

Theorem 4 (Alignment with Important Feature Directions.). *Define the robustness penalty as $C_{rob}(\epsilon, \mathbf{x}_c) = \sqrt{2\epsilon \mathbf{x}_c^\top H^{-1} \mathbf{x}_c}$ for a symmetric positive definite Hessian H . Let λ_1 be the largest eigenvalue of H with corresponding eigenvector \mathbf{q}_1 , and assume that λ_1 is unique. Then the robustness penalty term $C_{rob}(\epsilon, \mathbf{x}_c)$ is minimized when the counterfactual vector \mathbf{x}_c is aligned with the eigenvector \mathbf{q}_1 .*

Proof. When $\mathbf{x}_c = \mathbf{0}$, the conclusion of the theorem are satisfied in the trivial sense. Assume that $\mathbf{x}_c \neq \mathbf{0}$. Minimizing $C_{rob}(\epsilon, \mathbf{x}_c)$ for $\epsilon > 0$ is equivalent to minimizing $\mathbf{x}_c^\top H^{-1} \mathbf{x}_c$.

Since H is symmetric positive definite, its decomposition is $H = U\Lambda U^\top$, where U is an orthogonal matrix, such that its columns $(\mathbf{q}_1, \dots, \mathbf{q}_d)$ are the eigenvectors of H , and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d)$ is the matrix with eigenvalues on the diagonal. Then $H^{-1} = U\Lambda^{-1}U^\top$, where $\Lambda^{-1} = \text{diag}(1/\lambda_1, 1/\lambda_2, \dots, 1/\lambda_d)$. The smallest eigenvalue of H^{-1} is $1/\lambda_1$, since by assumption of the theorem $\lambda_1 > \lambda_2 \geq \dots \geq \lambda_d > 0$.

Let $\boldsymbol{\xi} = U^\top \mathbf{x}_c$. Then $\mathbf{x}_c = U\boldsymbol{\xi}$, and $\|\boldsymbol{\xi}\|_2 = \|U^\top \mathbf{x}_c\|_2 = \|\mathbf{x}_c\|_2$ since U is orthogonal. We are minimizing the following quadratic form:

$$\mathbf{x}_c^\top H^{-1} \mathbf{x}_c = (U\boldsymbol{\xi})^\top (U\Lambda^{-1}U^\top)(U\boldsymbol{\xi}) = \boldsymbol{\xi}^\top \Lambda^{-1} \boldsymbol{\xi} = \sum_{j=1}^d \frac{\xi_j^2}{\lambda_j}.$$

To find the optimal for $\boldsymbol{\xi}$, we analyze which direction minimizes this sum for any given positive norm. Thus, we add constraints that $\|\boldsymbol{\xi}\|_2^2$ is fixed and positive, $\|\boldsymbol{\xi}\|_2^2 = a^2 > 0$, and search for minimizing direction among all vectors with the fixed norm a . Since $1/\lambda_1 < 1/\lambda_j$ for $j \neq 1$, the sum $\sum_{j=1}^d \xi_j^2/\lambda_j$ is minimized when the whole mass of the squared norm $\|\boldsymbol{\xi}\|_2^2$ is placed on the component that corresponds to the smallest coefficient $1/\lambda_1$. Therefore, the minimizing $\boldsymbol{\xi}$ is $(\pm\|\boldsymbol{\xi}\|_2, 0, \dots, 0)^\top$ in the basis of U . Transforming back to \mathbf{x}_c :

$$\mathbf{x}_c = U\boldsymbol{\xi} = \xi_1 \mathbf{q}_1 + \sum_{i=2}^d \xi_i \mathbf{q}_i = (\pm\|\boldsymbol{\xi}\|_2) \mathbf{q}_1 = (\pm\|\mathbf{x}_c\|_2) \mathbf{q}_1.$$

Thus, for any fixed non-zero norm, $\mathbf{x}_c^\top H^{-1} \mathbf{x}_c$ is minimized when \mathbf{x}_c is aligned with the eigenvector \mathbf{q}_1 . □

A.5 Proof for Theorem 5

The last property that we state and prove below is that the counterfactuals generated by ElliCE have robustness-proximity trade-off. This property has been observed by other works as well [22] and is a natural for counterfactual generations. ElliCE allows us to find the smallest length counterfactuals that are robust to all models in the Rashomon set.

We next state corrected Theorem 5 and then prove it.

Theorem 5 (Robustness-Proximity Trade-off.). *For an input \mathbf{x}_0 such that $\hat{\boldsymbol{\theta}}^\top \mathbf{x}_0 \leq t$, where $\hat{\boldsymbol{\theta}}$ is ERM, let $\mathbf{x}_c^*(\epsilon)$ be the optimal robust counterfactual for a given robustness level $\epsilon > 0$, and let $\nu(\epsilon) = \|\mathbf{x}_c^*(\epsilon) - \mathbf{x}_0\|_2^2$ be its ℓ_2 distance from \mathbf{x}_0 . If $\nu(\epsilon_1) > 0$ and $\mathbf{x}_c^*(\epsilon_1) \neq \mathbf{0}$, then for any two robustness levels $0 < \epsilon_1 < \epsilon_2$, $\nu(\epsilon_1) < \nu(\epsilon_2)$.*

Proof. For every robustness level $\epsilon > 0$ define the feasible region

$$\mathcal{C}(\epsilon) := \left\{ \mathbf{x} \in \mathbb{R}^d \mid \hat{\boldsymbol{\theta}}^\top \mathbf{x} - \sqrt{2\epsilon \mathbf{x}^\top H^{-1} \mathbf{x}} \geq t \right\}.$$

Recall that $\mathbf{x}_c^*(\epsilon)$ is the *unique* minimiser of the strictly-convex objective $\text{Obj}(\mathbf{x}) = \min_{\mathbf{x}} \|\mathbf{x} - \mathbf{x}_0\|_2^2$ over $\mathcal{C}(\epsilon)$ as we proved in Theorem 2. Based on the theorem notations, $\nu(\epsilon) = \|\mathbf{x}_c^*(\epsilon) - \mathbf{x}_0\|_2^2$.

If $0 < \epsilon_1 < \epsilon_2$ then for every $\mathbf{x} \in \mathbb{R}^d$: $-\sqrt{2\epsilon_1 \mathbf{x}^\top H^{-1} \mathbf{x}} \geq -\sqrt{2\epsilon_2 \mathbf{x}^\top H^{-1} \mathbf{x}}$. Therefore

$$\hat{\boldsymbol{\theta}}^\top \mathbf{x} - \sqrt{2\epsilon_1 \mathbf{x}^\top H^{-1} \mathbf{x}} \geq \hat{\boldsymbol{\theta}}^\top \mathbf{x} - \sqrt{2\epsilon_2 \mathbf{x}^\top H^{-1} \mathbf{x}} \geq t.$$

This means that sets $\mathcal{C}(\epsilon_2)$ and $\mathcal{C}(\epsilon_1)$ are enclosed:

$$\mathcal{C}(\epsilon_2) \subseteq \mathcal{C}(\epsilon_1).$$

Then for $\mathbf{x}_c^*(\epsilon_2) \in \mathcal{C}(\epsilon_1)$ and $\mathbf{x}_c^*(\epsilon_1)$ in $\mathcal{C}(\epsilon_1)$ we get that ν is non-decreasing:

$$\nu(\epsilon_1) = \|\mathbf{x}_0 - \mathbf{x}_c^*(\epsilon_1)\|_2^2 \leq \|\mathbf{x}_0 - \mathbf{x}_c^*(\epsilon_2)\|_2^2 = \nu(\epsilon_2).$$

To show strict increase we will assume contradiction. More specifically, we assume that $\nu(\epsilon_1) = \nu(\epsilon_2)$. Then both $\mathbf{x}_c^*(\epsilon_1)$ and $\mathbf{x}_c^*(\epsilon_2)$ minimize $Obj(\mathbf{x})$ over $\mathcal{C}(\epsilon_1)$. Uniqueness of the solution means that it is possible only when:

$$\mathbf{x}_c^*(\epsilon_1) = \mathbf{x}_c^*(\epsilon_2) =: \mathbf{x}^* \neq \mathbf{0}.$$

Because the optimum of $Obj(\mathbf{x})$ on the convex set $\mathcal{C}(\epsilon)$ is obtained on its boundary, \mathbf{x}^* must satisfy the boundary equation for both $\mathcal{C}(\epsilon_1)$ and $\mathcal{C}(\epsilon_2)$:

$$\hat{\boldsymbol{\theta}}^\top \mathbf{x}^* - \sqrt{2\epsilon_1 \mathbf{x}^{*\top} H^{-1} \mathbf{x}^*} = t = \hat{\boldsymbol{\theta}}^\top \mathbf{x}^* - \sqrt{2\epsilon_2 \mathbf{x}^{*\top} H^{-1} \mathbf{x}^*}.$$

Subtracting the two equalities yields

$$\sqrt{2\epsilon_2 \mathbf{x}^{*\top} H^{-1} \mathbf{x}^*} = \sqrt{2\epsilon_1 \mathbf{x}^{*\top} H^{-1} \mathbf{x}^*},$$

which means that $\epsilon_2 = \epsilon_1$, which contradicts our assumption that $0 < \epsilon_1 < \epsilon_2$.

Therefore $\nu(\epsilon_1) < \nu(\epsilon_2)$ whenever $0 < \epsilon_1 < \epsilon_2$, completing the proof. \square

B Additional Experiments

Now, that we have proved the theoretical properties of our method, in this Appendix, we provide more experimental results and show how ElliCE performs under different optimization schema and as compared to different baselines.

B.1 Datasets

Please see Table 3 for the summary of all datasets used in the paper and pre-processing notes, if any. For each dataset and each fold, we fit a StandardScaler on the training set, computing the per-feature mean and variance, and then apply those parameters to both the validation and test splits. This ensures that each continuous feature is centered at zero mean and scaled to unit variance based on the training data. One-hot encoded and binary features are left untouched.

B.2 Empirical Rashomon Set Construction

Each evaluator set of models was constructed as follows. **Retrain models** were obtained by independently retraining the base model from random initialization using different random seeds on the same training and validation data. Models whose training loss remained within the Rashomon bound were retained to form the retrained Rashomon set. **Dropout models** were generated by applying Gaussian dropout noise directly to the weights of the trained base model. The dropout variance hyperparameter was first tuned to approximate the target Rashomon bound by finding the maximum dropout value such that at least 5% of the sampled models under it remained within the Rashomon set (which in practice, due to the discreteness of the dropout search space, typically ranged between 10% and 15%). During model sampling, if a specific perturbed model exceeded the Rashomon bound, its dropout variance was reduced by a factor of 2; if the resulting model satisfied the Rashomon constraint, it was retained. **AWP models** were produced using Adversarial Weight Perturbation (AWP), where the base model's parameters were iteratively updated with adversarial perturbations computed on the X_{train} , using the training loss. Perturbations were halted once the training loss exceeded the Rashomon threshold, taking model weights from previous step, ensuring that all generated models remained valid members of the Rashomon set.

Table 3: Datasets description and pre-processing notes. In Comments, we provide original dimension of the dataset before any edits (#samples \times #features).

Dataset Name	# Samples	# Features	Preprocessing Notes	Source	Comments
Parkinson’s	5872	18	<i>motor_UPDRS</i> is considered as the target, that was binarized by cutting at the median value. Dropped subject#, test_time, raw UPDRS columns. Under-sampled to balance the dataset. Shuffled data.	[62]	Orig. 5875×21
Wine Quality	2554	11	Binarized target <i>quality</i> at the median value. Dropped <i>type</i> and raw <i>quality</i> . Under-sampled to balance dataset, Shuffled data.	[12]	Orig. 6497×12
German Credit	600	61	Performed one-hot encoding for all categorical features. Under-sampled to balance the dataset. Shuffled data.	[28]	Orig. 1000×20
Extended Iris	800	4	Converted to binary classification: 1, if Iris-setosa, 0 otherwise. Under-sampled to balance the dataset. Shuffled data. Only kept features present in the original Iris dataset.	[3]	Orig. 1200×67
Banknote	1220	4	Selected variance, skewness, curtosis, entropy as features. Under-sampled to balance the dataset. Shuffled data.	[47]	Orig. 1372×4
Australian Credit	530	18	Dropped the first column (unique identifier). Under-sampled to balance the dataset. Shuffled data.	[55]	Orig. 640×19
FICO	9470	20	Dropped 3 features with the high amount of missing values (MSinceMostRecentDelq, NetFractionInstallBurden, MSinceMostRecentInqexcl7days) and removed rows with $>40\%$ missingness among remaining features. Filled remaining missing values with -1. Under-sampled to balance the dataset. Shuffled data.	[19]	Orig. 10459×23
COMPAS	6392	7	Under-sampled to balance the dataset. Shuffled data.	[2]	Orig. 6907×7
Diabetes	536	8	Under-sampled to balance the dataset. Shuffled data.	[60]	Orig. 768×8

B.3 Computation Resources

Our experiments were conducted on Apple M2/M2 Max MacBooks (16-32 GB RAM) for development and Google Cloud Platform C4 VMs (4-16 vCPUs, 16-64 GB RAM) for production runs. We performed parallelization using multiprocessing for cross-validation folds. Our complete pipeline (5 datasets, 2 model types) required ~ 64 hours sequential execution on a single cpu. We used fixed random seeds for reproducibility.

B.4 Optimization

For scenarios where counterfactuals need not lie on the data manifold, we implemented a continuous optimization approach described in Algorithm 1 and Algorithm 2. This gradient-based method directly optimizes in the input space by iteratively computing the worst-case model using Theorem 1 and updating the counterfactual through gradient descent on a multi-objective loss combining prediction, robustness, proximity, and sparsity terms.

B.5 Hyperparameter Tuning

For hyperparameter tuning, we chose the best hyperparameters for each method, targeting validity first and then robustness. We observed that ElliCE, TRex, and ROAR behaved more consistently across different model types and datasets, not requiring significant adjustments to their hyperparameter search ranges. In contrast, Delta Robustness and PROPLACE necessitated disproportionately

Algorithm 1 Continuous ElliCE (Preperation & CE generation)

```
1: Input:  $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ , baseline model  $f_{\theta}$ , Rashomon budget  $\epsilon$ , regularization  $\lambda$ , initial  $\mathbf{x}_0$ 
   Preparation:
2:  $E_i \leftarrow \phi(\mathbf{x}_i)$ ,  $\tilde{H} \leftarrow [E, \mathbf{1}]$ 
3: Optionally refit last layer  $\theta^* \leftarrow \text{LR}(\tilde{H}, y)$ 
4:  $\omega_c \leftarrow \theta^*$ ,  $s \leftarrow \tilde{H} \omega_c$ ,  $p \leftarrow \sigma(s)$ 
5:  $W \leftarrow \text{diag}(p \odot (1 - p))$ 
6:  $H \leftarrow \frac{1}{n} \tilde{H}^\top W \tilde{H} + \lambda I$ 
   Generation:
7: Initialise:  $\mathbf{x}_c \leftarrow \mathbf{x}_0$ ,  $\text{steps} \leftarrow 0$ 
8: while  $\text{steps} < T$  do
9:    $h \leftarrow \phi(\mathbf{x}_c)$ ,  $\tilde{h} \leftarrow [h^\top, 1]^\top$ 
10:  Worst-case model (Theorem 1):
11:     $\theta_w = \omega_c - \sqrt{2\epsilon} \frac{H^{-1} \tilde{h}}{\sqrt{\tilde{h}^\top H^{-1} \tilde{h}}}$ 
12:    Prediction loss:  $\ell_{\text{pred}} = \text{BCE}(\tilde{h}^\top \omega_c, 1)$ 
13:    Robustness loss:  $\ell_{\text{rob}} = \text{BCE}(\tilde{h}^\top \theta_w, 1)$ 
14:    Proximity loss:  $\ell_{\text{prox}} = \|\mathbf{x}_c - \mathbf{x}_0\|_2^2$ 
15:    Sparsity loss:  $\ell_{\text{sparse}} = \|\mathbf{x}_c - \mathbf{x}_0\|_1$ 
16:    Total objective:  $\mathcal{L} = \alpha \ell_{\text{pred}} + \beta \ell_{\text{rob}} + \lambda \ell_{\text{prox}} + \gamma \ell_{\text{sparse}}$ 
17:    Gradient update:  $\mathbf{x}_c \leftarrow \mathbf{x}_c - \eta \nabla_{\mathbf{x}_c} \mathcal{L}$ 
18:    if  $\tilde{h}^\top \theta_w \geq 0$  then
19:      break
20:     $\text{steps} \leftarrow \text{steps} + 1$ 
21: Output: robust counterfactual  $\mathbf{x}_c$ 
```

Algorithm 2 Continuous ElliCE (CE generation)

```
1: Input: factual  $\mathbf{x}_0$ , central weights  $\omega_c$ , Hessian  $H$ , Rashomon radius  $\epsilon$ , learning-rate  $\eta$ , max_steps  $T$ ,
   coefficients  $(\alpha, \beta, \lambda, \gamma)$ 
2: Initialise:  $\mathbf{x}_c \leftarrow \mathbf{x}_0$ ,  $\text{steps} \leftarrow 0$ 
3: while  $\text{steps} < T$  do
4:   Compute penultimate features  $h \leftarrow \phi(\mathbf{x}_c)$ , augmented  $\tilde{h} \leftarrow [h^\top, 1]^\top$ 
5:   Worst-case model (Thm.1):
6:      $\theta_w = \omega_c - \sqrt{2\epsilon} \frac{H^{-1} \tilde{h}}{\sqrt{\tilde{h}^\top H^{-1} \tilde{h}}}$ 
7:     Prediction loss  $\ell_{\text{pred}} = \text{BCE}(\tilde{h}^\top \omega_c, 1)$ 
8:     Robustness loss  $\ell_{\text{rob}} = \text{BCE}(\tilde{h}^\top \theta_w, 1)$ 
9:     Proximity loss  $\ell_{\text{prox}} = \|\mathbf{x}_c - \mathbf{x}_0\|_2^2$ 
10:    Sparsity loss  $\ell_{\text{sparse}} = \|\mathbf{x}_c - \mathbf{x}_0\|_1$ 
11:    Total objective  $\mathcal{L} = \alpha \ell_{\text{pred}} + \beta \ell_{\text{rob}} + \lambda \ell_{\text{prox}} + \gamma \ell_{\text{sparse}}$ 
12:    Gradient update  $\mathbf{x}_c \leftarrow \mathbf{x}_c - \eta \nabla_{\mathbf{x}_c} \mathcal{L}$ 
13:    if  $\tilde{h}^\top \theta_w \geq 0$  then
14:      break
15:     $\text{steps} \leftarrow \text{steps} + 1$ 
16: Output: robust counterfactual  $\mathbf{x}_c$ 
```

different search ranges depending on the model, primarily because different model architectures (linear vs. neural networks) operate with vastly different parameter scales and amplitudes, requiring delta-based perturbation methods to adjust their sensitivity accordingly.

For linear models, the search range for Delta Robustness was $[0.2, 3.0]$ with a step size of 0.1 to 0.2 (dataset-dependent), and for PROPLACE, the range was $[0.2, 6.5]$ with a step size of 0.2. Neural Networks (NNs) required much smaller amplitudes: Delta Robustness used a range of $[0.002, 0.1]$ with a step of 0.004, and PROPLACE used $[0.02, 0.5]$ with a step of 0.02. Interestingly, for an MLP with hidden layers [32, 32] on the Parkinsons dataset, PROPLACE began to lose validity and failed to find counterfactuals (CFs) for $\delta > 0.001$. Consequently, its search range was reduced to $[0.0001, 0.001]$ with a step of 0.0004. Even within this adjusted range, PROPLACE sometimes

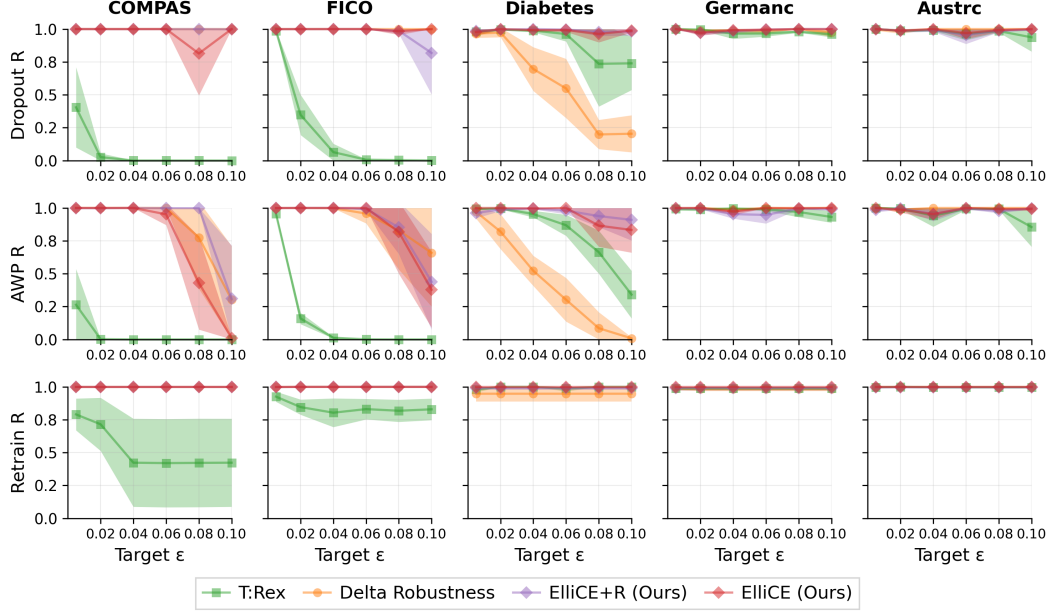


Figure 3: Robustness evaluation of ElliCE against baselines for MLPs for all datasets for data-supported generation.

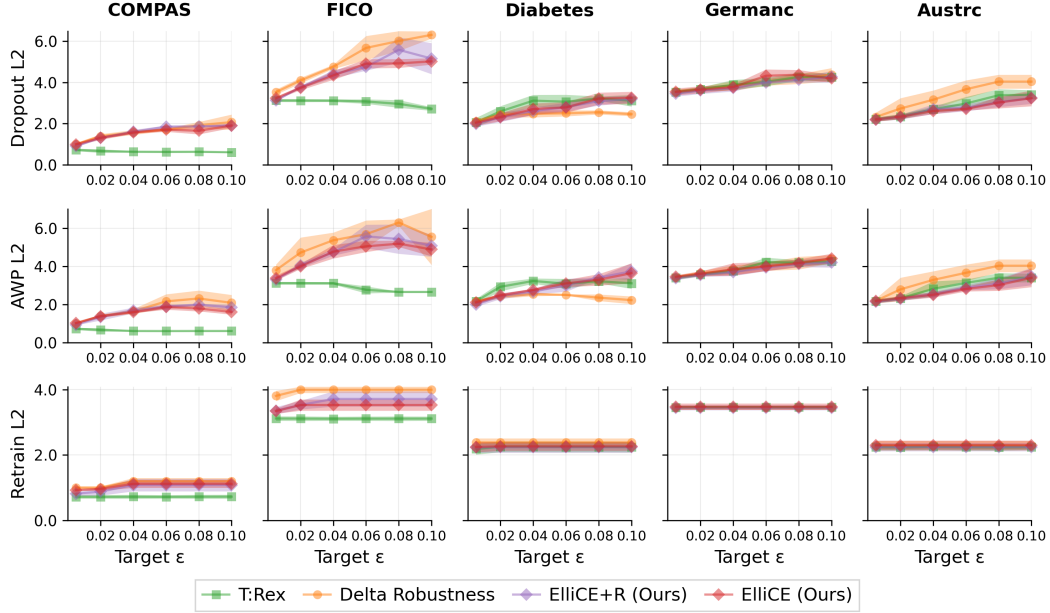


Figure 4: Length evaluation of ElliCE against baselines for MLPs for all datasets for data-supported generation.

failed to converge in a reasonable time when generating CFs for certain inputs x . Thus, we imposed a hard time limit of 30 seconds per CF generation.

For both TRex and TRexI, we tuned the threshold τ in the range $[0.375, 1.0]$ with a step of 0.05 to 0.1. For ElliCE and Continuous ElliCE, the respective search ranges were $[0.00125-0.0025, 0.15-0.2]$ with step sizes of 0.0025–0.01, and $[0.0025, 0.2]$ with step sizes of 0.0025–0.01. For ROAR, we used ranges of $[0.01-0.6]$ with a step of 0.06.

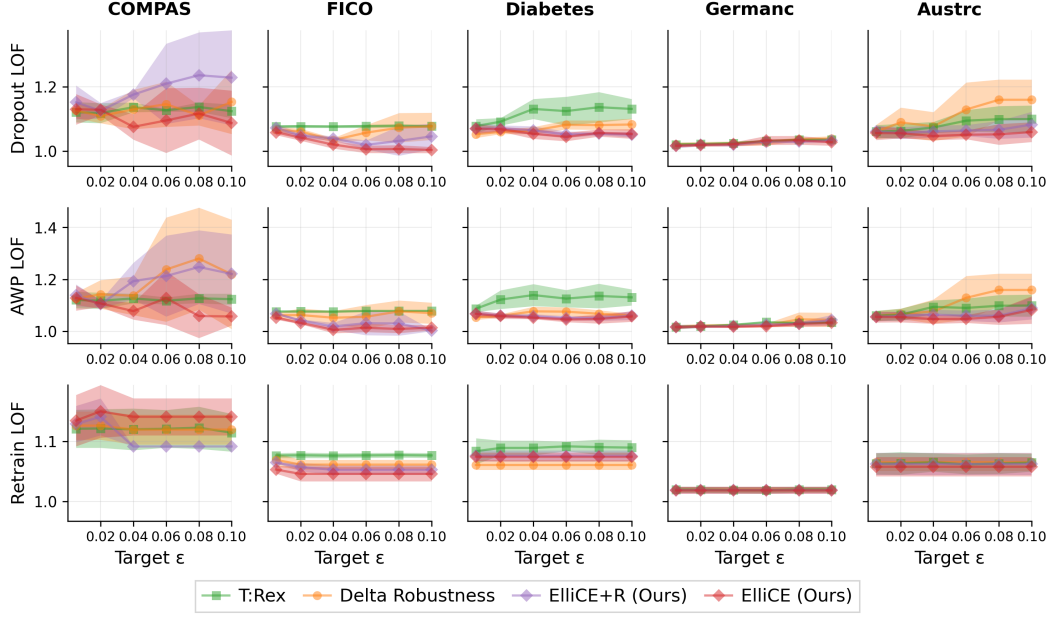


Figure 5: LOF evaluation of ElliCE against baselines for MLPs for all datasets for data-supported generation.

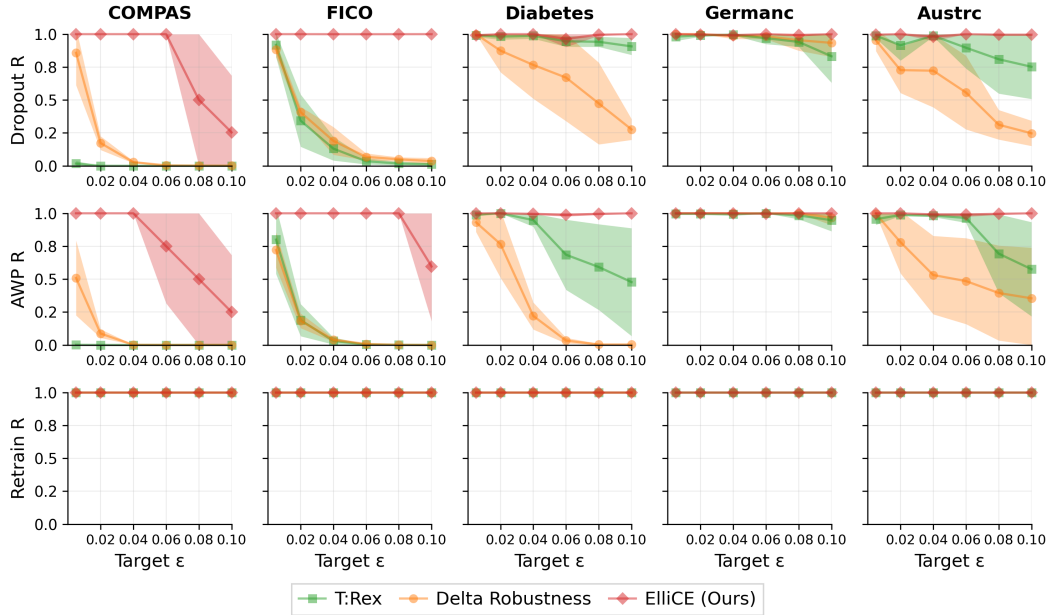


Figure 6: Robustness evaluation of ElliCE against baselines for linear models for all datasets for data-supported generation.

Additionally, we applied early stopping for all methods upon achieving a robustness score of 1, which significantly reduced running times for some methods.

B.6 The Choice of ϵ Parameter for ElliCE

The choice of the robustness parameter ϵ is a key consideration when generating recourse. ElliCE is designed to produce explanations robust to the Rashomon set, offering a basis for selecting ϵ .

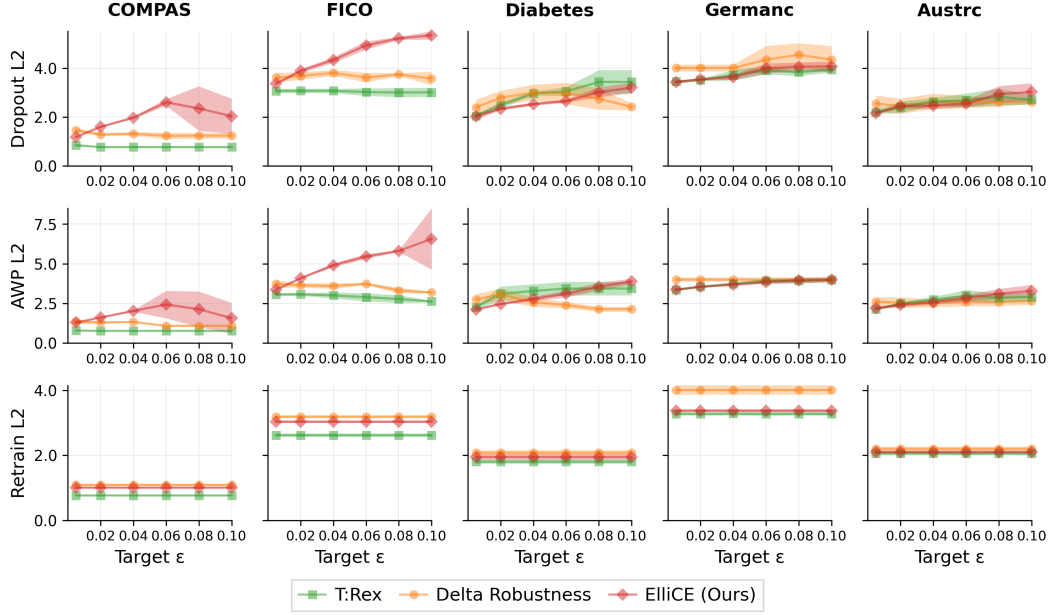


Figure 7: Length evaluation of ElliCE against baselines for linear models for all datasets for data-supported generation.

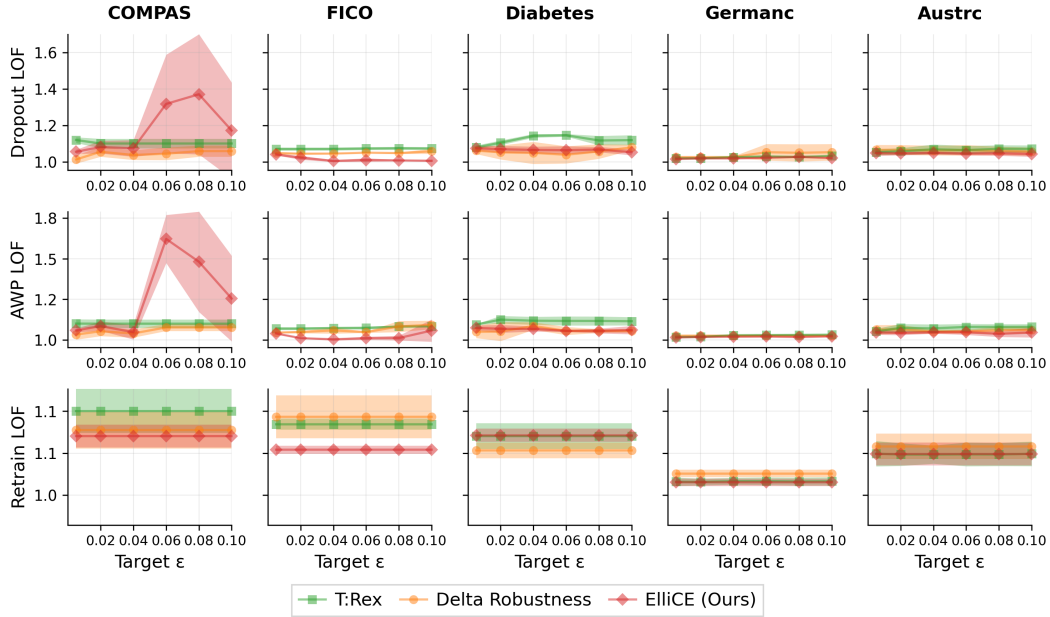


Figure 8: LOF evaluation of ElliCE against baselines for linear models for all datasets for data-supported generation.

Empirical evidence (as seen in Figures 6, 3, 13, and 14) indicates that models produced by random retraining often represent a smaller, less diverse subset of the true Rashomon set, making them easier to be robust against. ElliCE frequently achieves perfect robustness to such retrained models with relatively small ϵ values.

Ideally, if ϵ_{target} (the Rashomon parameter of the evaluator) is known, we recommend choosing $\epsilon \geq \epsilon_{target}$. Selecting a value slightly higher, such as $\epsilon = \epsilon_{target} + 0.01$, can provide additional

Table 4: Counterfactual explanations generated by ElliCE based on German Credit dataset. We provide the original input, closest counterfactual, and actionable counterfactual respecting immutable features (denoted by †). $\epsilon_{target} = 0.01\hat{L}_{train}(f_{baseline})$

Feature	Original	Robust CE	Actionable and Robust CE
Checking Account Status	No account	No account	No account
Duration (months)	48	37.09	34.7
Credit History†	Critical account	Critical account	Critical account
Purpose†	Car (new)	Car (new)	Car (new)
Credit Amount (DM)	10 127	7 584	7 073
Savings Account Status	500–1000 DM	500–1000 DM	500–1000 DM
Employment Duration	1–4 yrs	1–4 yrs	1–4 yrs
Installment Rate (%)	2	2	2
Personal Status & Sex†	Male, single	Male, single	Male, single
Other Debtors/Guarantors	None	None	None
Residence Since	2	2	2
Property†	No property	No property	No property
Age †	44	54.06	44
Other Installment Plans	Bank	Bank	Bank
Housing†	Free	Free	Free
Number of Credits†	0	0.29	0
Job Level†	Skilled employee	Skilled employee	Skilled employee
Number of Dependents†	0	0	0
Telephone	None	None	Registered
Foreign Worker†	Yes	Yes	Yes

confidence by accounting for broader model multiplicity beyond the local estimates ElliCE focuses on. This helps prevent under-exploration of the Rashomon set.

If a precise ϵ_{target} value is unavailable from an evaluator or prior analysis, we recommend using model performance metrics as a practical way to gauge an appropriate ϵ . For example, one might set a target based on acceptable performance deviations. Empirically, rules of thumb such as choosing ϵ that corresponds to a 10% increase above the original model’s loss, or an ϵ that captures models within a 5% accuracy deviation, often work well as starting points. The goal is to select an ϵ that reflects a meaningful degree of model variability relevant to the specific application.

B.7 Experiments for Data-Supported Counterfactual Generation

This section extends results from Section 6. Here, we provide plots for all datasets and hypothesis spaces. More specifically, robustness and length results for linear models are shown in Figure 6 and Figure 7 and for multi-layer perceptrons in Figure 3 and Figure 4. Table 7 contains a snapshot of figures when target epsilon is 10% of the optimal loss on train data. Robustness-Proximity tradeoff plots are in Figure 9 for linear models and Figure 10 for MLPs.

B.7.1 Examples of Counterfactuals Generated by ElliCE

Based on the German Credit dataset, we demonstrate ElliCE’s ability to generate meaningful and actionable recourse. More specifically, when dealing with immutable features (those that cannot be changed), ElliCE can impose immutable feature constraints by restricting candidate selection or gradient flow, thereby generating realistic alternatives.

In Table 4, we present the original input, the closest output that is robust to model changes (over the Rashomon set), and then the actionable output. Fixed features are marked with †. Features such as “Age,” “Personal status & Sex,” and whether someone is a “Foreign Worker” were held fixed.

As input (second column), consider an applicant who was denied credit. The closest robust counterfactual requires waiting 10 years and dropping a dependant, which is not feasible for the applicant. We allow changes in other features such as “Installment Rate” but fix “Number of Dependents” and other features of choice, which can be chosen by user depending on what is actionable for them at the time.

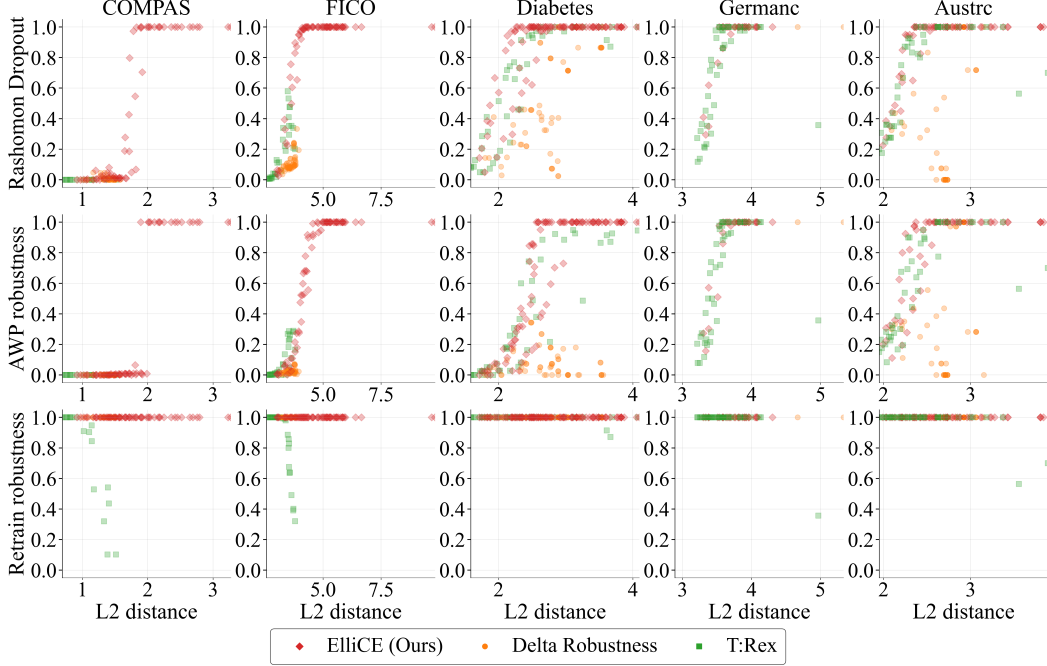


Figure 9: Robustness vs length tradeoff for all datasets and methods. Linear models. Data-supported generation. $\epsilon_{\text{target}} = 0.04$

B.7.2 Analysis of Robustness and Length

As shown in Figure 6, across all five datasets (COMPAS, FICO, Diabetes, German and Australian) as well as all metrics, ElliCE keeps high robustness score almost everywhere, showing its ability to adapt to different multiplicity levels. All three methods are robust to Retrain Robustness. Note, that Retrain Robustness for linear models is not influenced by ϵ_{target} in any way, due to a determinism of linear solvers, thus the plots stay constant. Delta Robustness starts strong, but quickly drops with increase of ϵ_{target} . This shows that Delta Robustness is able to capture slight model perturbations, but might not be able to extend to more uncertainty. T:Rex performs on par or better than Delta Robustness on all datasets except for COMPAS, showing overall similar behavior.

Figure 7 depicts, for each dataset, how the average ℓ_2 distance of counterfactuals increases with increasing ϵ . For random retrain, all methods give approximately the same length. ElliCE tends to be in the middle, but has the longest counterfactuals for FICO and COMPAS datasets.

For FICO dataset we observe a clear trade-off between robustness and length: while other methods struggle to find robust counterfactuals on this dataset, ElliCE does it by increasing the length.

For MLPs, Figure 3 shows the robustness of ElliCE, Delta Robustness, and T:Rex across different metrics (Retrain, AWP, and Rashomon dropout) and multiplicity levels (ϵ_{target}).

Figure 4 shows that, on average, both the counterfactual lengths themselves and their increasing trends with ϵ_{target} are similar across methods for MLPs.

A method’s ability to maintain robustness for large ϵ_{target} values often depends on its capacity to explore more distant counterfactuals as its internal hyperparameters are varied. Delta Robustness exhibits this property of exploring further points. T:Rex, however, not always does. If T:Rex considers a point robust for a given threshold τ , increasing τ will not necessarily change that specific counterfactual, if it remains robust under new τ . ElliCE is designed to navigate the trade-off between counterfactual length and robustness. It increases counterfactual length strategically, only if doing so leads to an improvement in robustness.

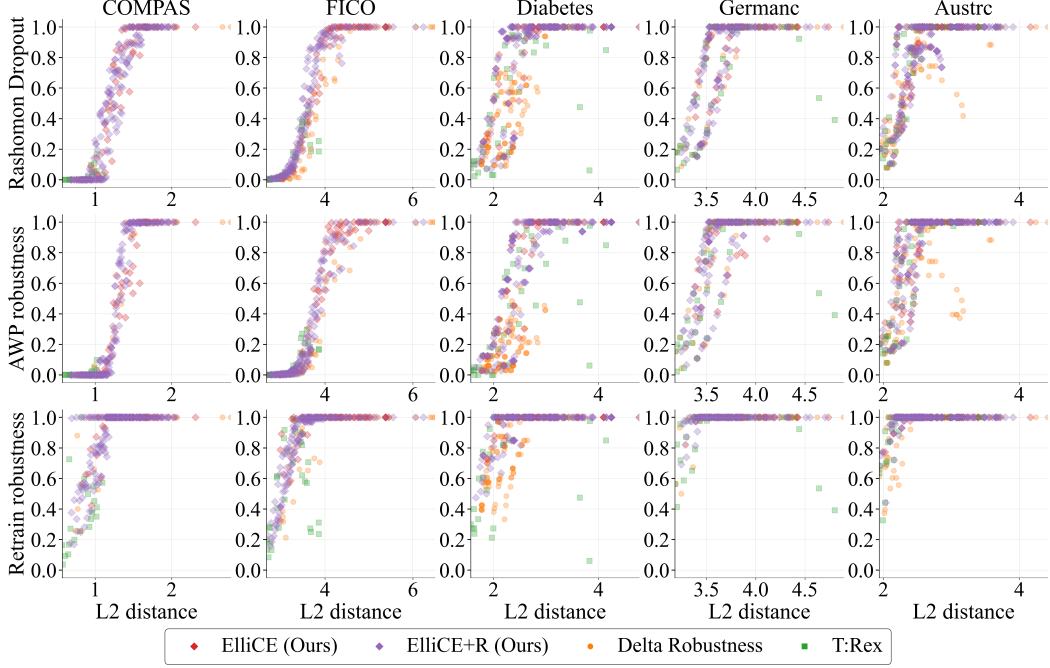


Figure 10: Robustness vs length tradeoff for all datasets and methods. MLPs. Data-supported generation. $\varepsilon_{\text{target}} = 0.04$.

B.8 Robustness & length tradeoff

Figure 9 presents the robustness–length trade-off for linear models, computed as the average achieved robustness on the validation dataset compared to the used length. It is noteworthy that for the Retrain Robustness metric, all methods achieve high robustness scores, which can be attributed to the lower variability typically observed in linear models under retraining. ElliCE demonstrates a consistent trade-off, producing more robust examples as the length budget is increased. For Delta Robustness, on the Diabetes and Australian datasets, increased counterfactual length does not always translate to proportionally higher robustness compared to other methods.

Figure 10 illustrates, for five benchmark datasets (with $\varepsilon_{\text{target}} = 0.04$), how counterfactual length (ℓ_2 distance) trades off against three robustness metrics: Dropout Robustness (top row), AWP robustness (middle row), and Retrain robustness (bottom row).

All methods demonstrate a clear trade-off. On some datasets T:Rex explores a range of counterfactual lengths that are, in some instances, shorter than those achieved by ElliCE or Delta Robustness when ElliCE achieves comparable or higher robustness.

Additional figures for Continuous CE methods are presented in Figures 11 and 12.

B.9 Experiments for Non-data Supported Counterfactual Generation

Figures 13 and 14 illustrate robustness across varying multiplicity levels (ϵ) for linear models on five benchmark tabular datasets: Diabetes, Iris, Parkinson’s, Wine Quality, and Banknotes. We compare ElliCE with PROPLACE, ROAR, and T:RexI, using Rashomon Dropout Robustness, AWP Robustness, and Retrain Robustness metrics. In Table 8 we present all robustness and length results for Non-data Supported Counterfactual Generation for LMs and MLPs.

For Random Retrain evaluator, all methods achieve perfect scores. This is likely because linear models tend to not change upon retraining. Consequently, this also indicates that all methods achieve perfect validity (i.e., a score of 1) for the retrain method.

Figures 15 and 16 present validity results. Most methods generally find valid counterfactuals.

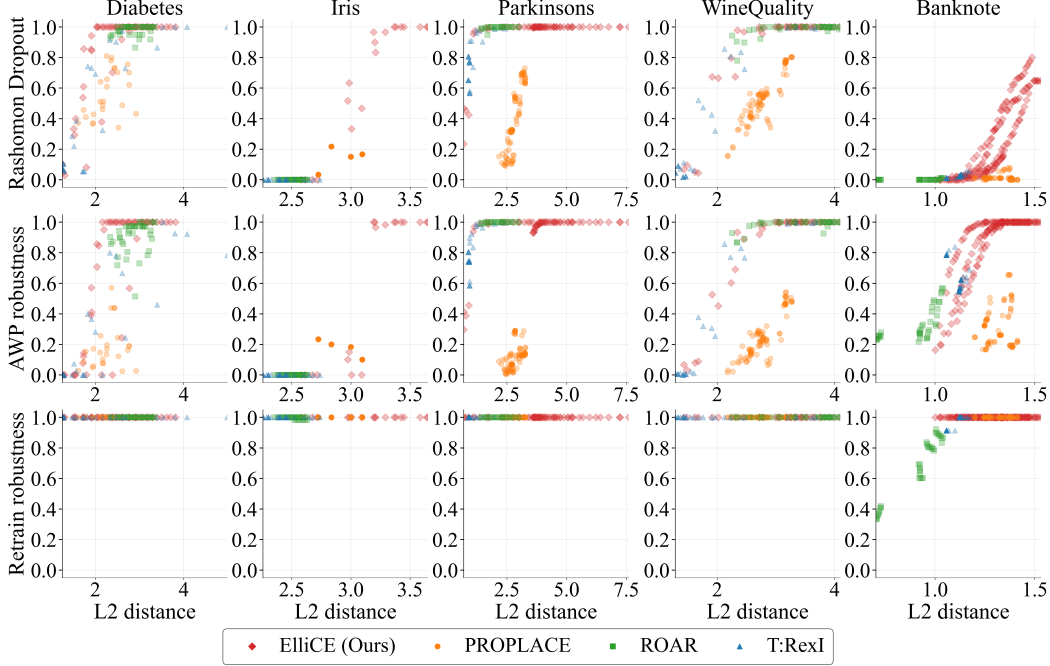


Figure 11: Robustness vs length tradeoff for all datasets and methods. Linear. Continuous generation. $\epsilon_{\text{target}} = 0.04$.

Figures 17 and 18 illustrate the relationship between counterfactual length and robustness. While all methods often start with similar counterfactual lengths for small multiplicity values, ElliCE distinctively increases the length of counterfactuals to enhance robustness, especially as ϵ_{target} grows. Thus, when robust counterfactuals are located nearby, other methods also tend to find them. However, if achieving robustness at larger ϵ values necessitates exploring more distant points (i.e., increasing the length budget), ElliCE demonstrates a superior capability to do so effectively.

B.10 Data shift

To further support that the ElliCE effectively handles re-training scenarios (model shifts), we evaluate our methods and baselines under the data shift scenario. Specifically, we split each dataset into two parts: the first part was used to train our method (ElliCE) and baseline models (STCE, RNCE), while the second part was used exclusively to train the Random Retraining evaluator. This setup follows a setup similar to that used by T:Rex and Argumentative Ensembling and ensures that the methods under evaluation and the evaluator are trained on disjoint data, capturing a realistic data-shift scenario. For counterfactual evaluation, we use the same data split that was used to train evaluators model (the second data part described above). Under the data shift evaluation pipeline, ElliCE continues to outperform or perform on par with the baselines across both continuous and data-supported settings. Please see Figures 21 and 22 for more details.

C Delta-Robustness Works in the Last Layer under Reparameterization

Delta-Robustness, a leading baseline for generating robust counterfactual explanations, perturbs all network parameters within an ℓ_∞ -norm ball of a given radius δ . Our analysis demonstrates a key finding: under a specific model reparameterization Delta-Robustness works in the final layer of a MLP, an approach similar to that of ElliCE.

This reparameterization implies that Delta-Robustness then approximates the model uncertainty set in the last layer using an ℓ_∞ -norm constraint (a hypercube), whereas ElliCE uses an ℓ_2 -norm constraint (an ellipsoid). While ellipsoidal bounds are often presumed to offer a tighter fit for convex

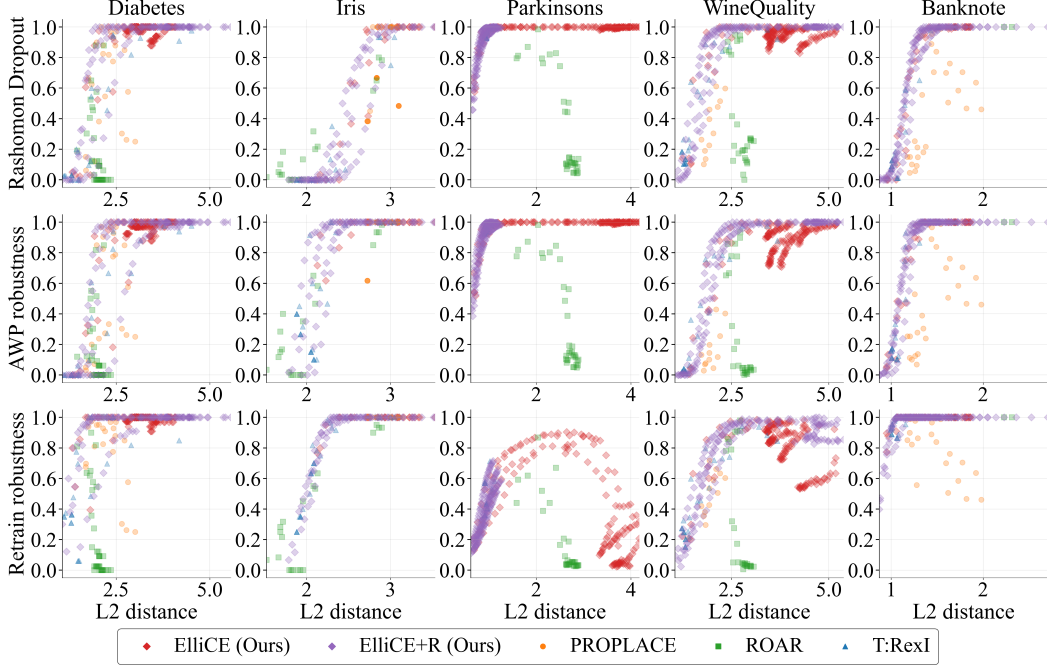


Figure 12: Robustness vs length tradeoff for all datasets and methods. MLPs. Continuous generation. $\varepsilon_{\text{target}} = 0.04$.

losses, a hypercubic approximation might, in some cases, be more accurate. This geometric difference could potentially explain cases where Delta-Robustness performs better than ElliCE.

In this section, we establish theoretical limitations for delta interval approaches to robust counterfactuals under model multiplicity. We prove that for certain model reparameterizations, establishing robust counterfactuals becomes either too restrictive or computationally inefficient.

C.1 Problem Setting

Consider a two-layer perceptron with input vector $x \in \mathbb{R}^d$, weight matrices $W_1 \in \mathbb{R}^{h \times d}$ and $W_2 \in \mathbb{R}^{c \times h}$, and bias vectors $b_1 \in \mathbb{R}^h$, $b_2 \in \mathbb{R}^c$. The network output can be expressed as:

$$y = W_2 \cdot \sigma(W_1 x + b_1) + b_2,$$

where σ is a 1-Lipschitz non-linear activation function (e.g., ReLU). Throughout the following theorems we adopt a threshold of 0.

Let $\Theta := \{(W_i, b_i)\}_{i=1}^L$ denote the network parameters of an L-layer network. One approach to ensure robustness under model multiplicity is to establish a δ -interval around these parameters. For a fixed radius $\delta > 0$ we define ℓ_∞ -ball:

$$\mathcal{M}_\delta(\Theta) = \left\{ \Theta' = \{(W'_i, b'_i)\}_{i=1}^L \mid \max(\|W_i - W'_i\|_\infty, \|b_i - b'_i\|_\infty) \leq \delta \right\}.$$

A counterfactual \mathbf{x}_c is considered δ -robust if it maintains the desired prediction across all models in $\mathcal{M}_\delta(\Theta)$.

C.2 Necessary Conditions for Delta-Robustness

We first establish necessary conditions for a meaningful delta-robustness analysis.

Proposition 1. Consider an L-layer network whose last affine layer is $h_L(x) = W_L h_{L-1}(x) + b_L$ with decision rule $\hat{y}(x) = \mathbf{1}[h_L(x) > 0]$. Assume the final-layer bias satisfies $b_L < 0$.

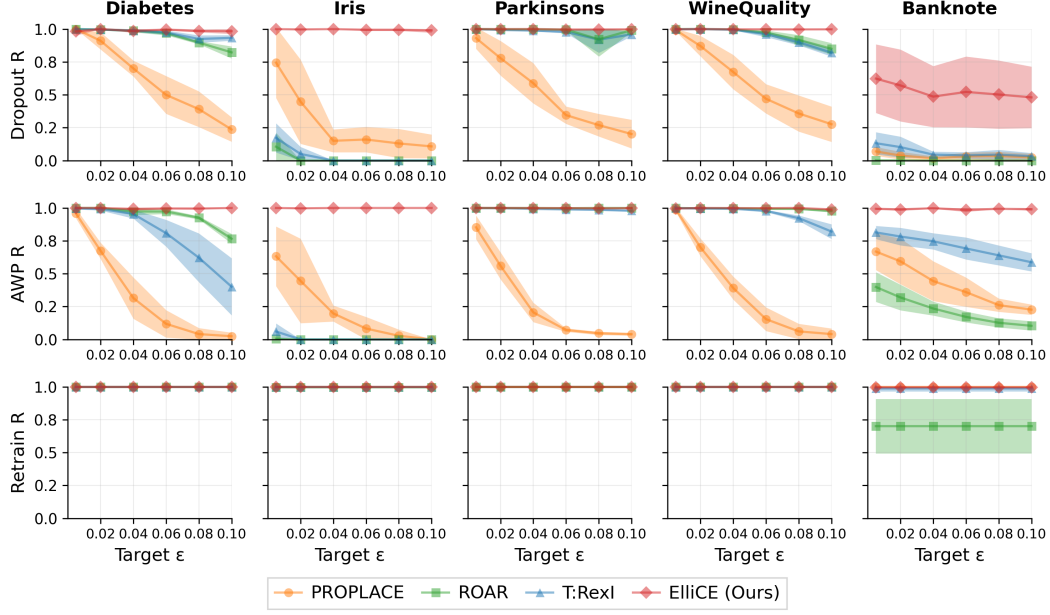


Figure 13: Robustness evaluation of Continuous ElliCE against baselines for linear models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved robustness score.

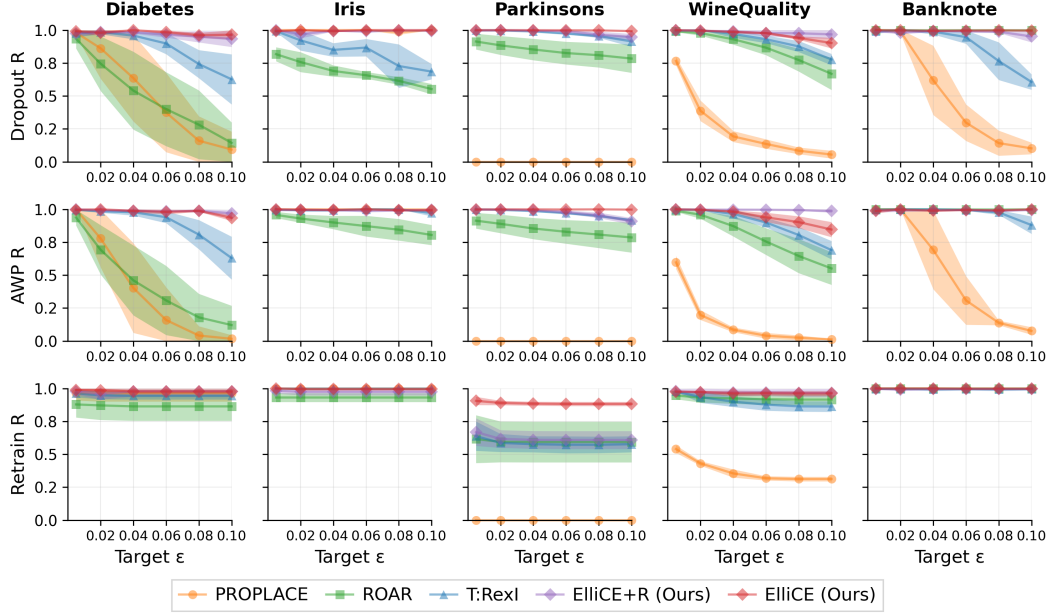


Figure 14: Robustness evaluation of Continuous ElliCE against baselines for NN models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved robustness score.

Robust $0 \rightarrow 1$ counterfactuals (i.e., classified as 0 under the reference parameters Θ but classified as 1 under every $\Theta' \in M_\delta(\Theta)$) exist within ℓ_∞ -ball $M_\delta(\Theta)$ only if the perturbation radius $\delta > 0$ satisfies the following:

$$\delta < \|W_L\|_\infty$$

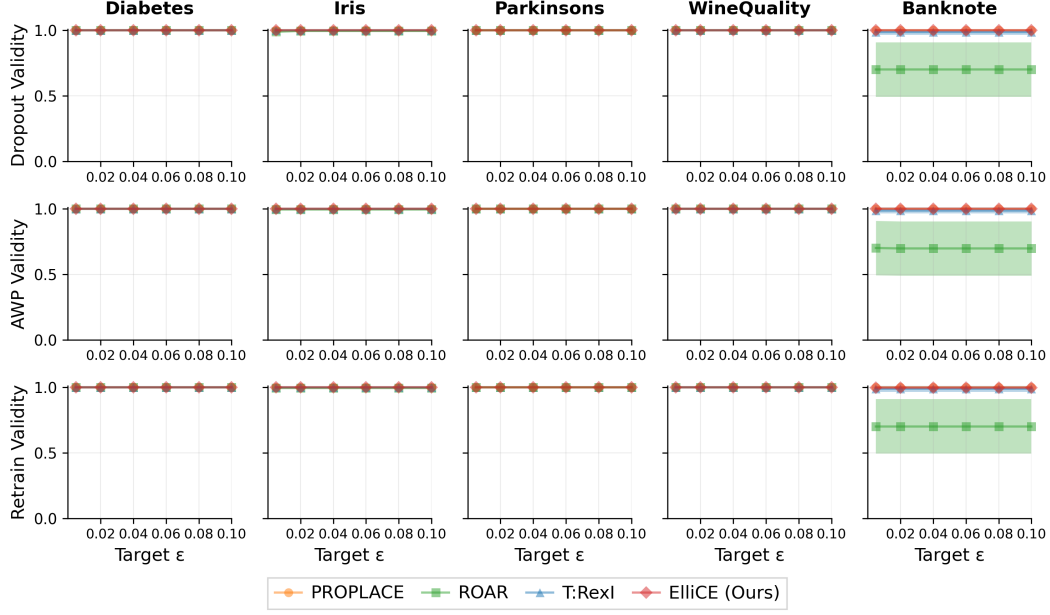


Figure 15: Validity evaluation of Continuous ElliCE against baselines for linear models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved robustness score.

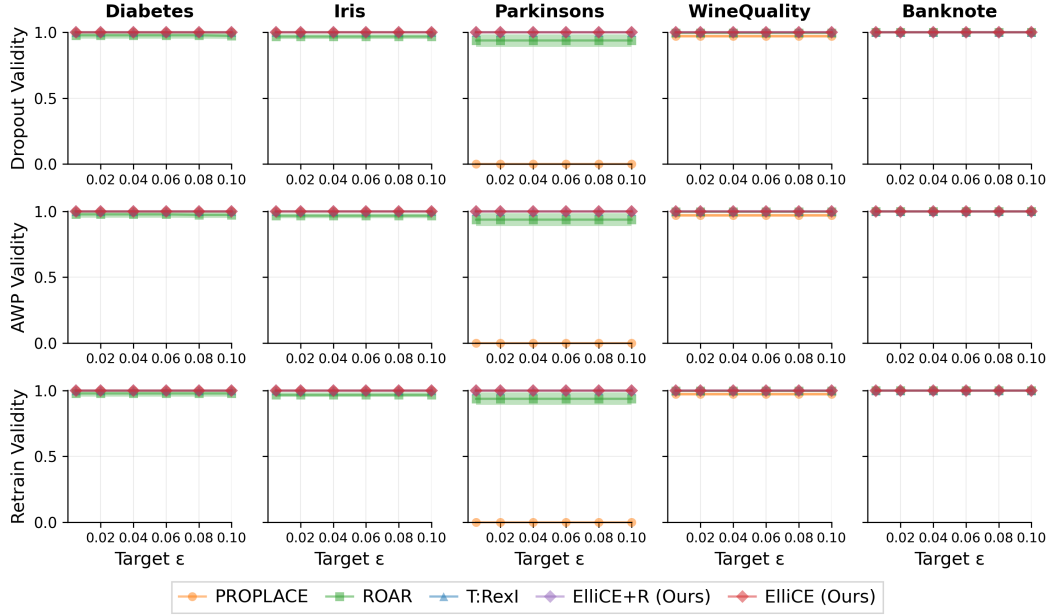


Figure 16: Validity evaluation of Continuous ElliCE against baselines for NN models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved robustness score.

Proof. Assume that the perturbation radius satisfies $\delta \geq \|W_L\|_\infty$. Consider a parameter set $\Theta' = \{(W'_i, b'_i)\}_{i=1}^L$, where

$$W'_i = W_i, \quad b'_i = b_i, \quad \forall i = 1..L-1, \quad W'_L = \mathbf{0}, \quad b'_L = b_L$$

As all layers except the last one are unchanged, we can verify that $\Theta' \in \mathcal{M}_\delta(\Theta)$:

$$\begin{aligned} \|W_L - W'_L\|_\infty &= \|W_L - \mathbf{0}\|_\infty = \|W_L\|_\infty \leq \delta \\ \|b_L - b'_L\|_\infty &= 0 \end{aligned}$$

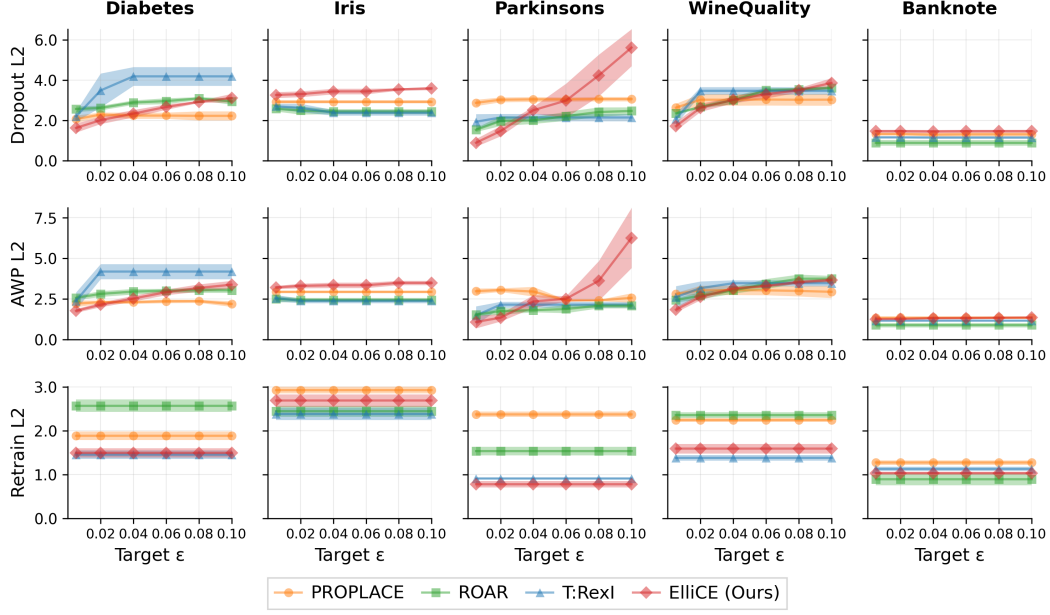


Figure 17: Length evaluation of Continuous ElliCE against baselines for linear models.

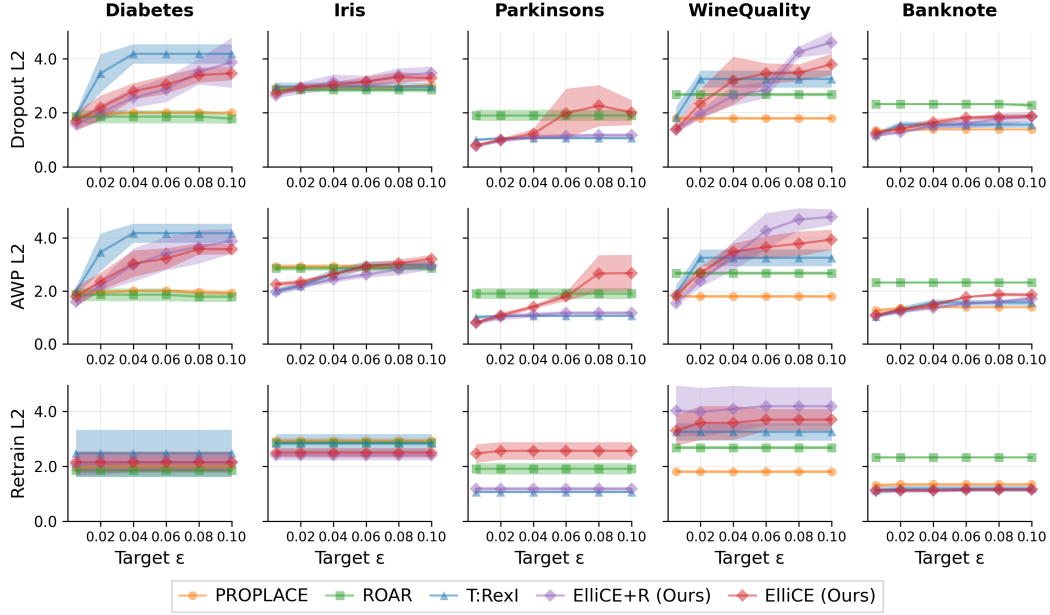


Figure 18: Length evaluation of Continuous ElliCE against baselines for NN models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved l_2 length.

For any input x in the last layer we have:

$$h'_L(x) = W'_L h_{L-1}(x) + b'_L = \mathbf{0} h_{L-1}(x) + b_L = b_L < 0,$$

The model Θ' has a degenerate last layer that produces a constant output $b_L < 0$ and predicts class 0 regardless of the input, making the model equivalent to a constant function. Since this degenerate model belongs to $\mathcal{M}_\delta(\Theta)$, no counterfactual (0 to 1) can be robust with respect to all models in the interval. Therefore, to avoid this case, we must have $\delta < \|W_L\|_\infty$. \square

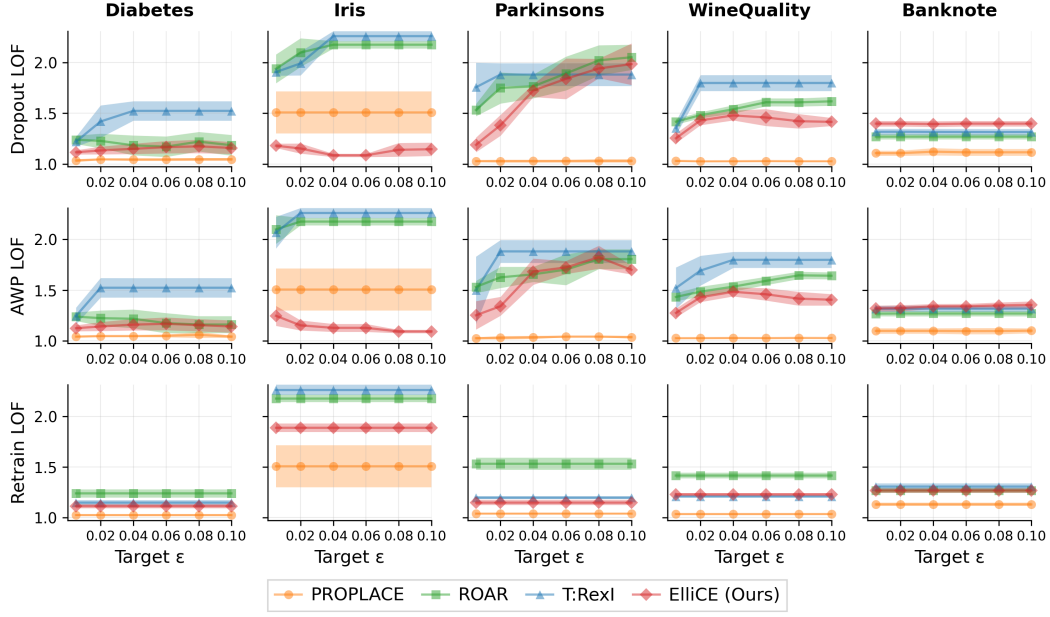


Figure 19: LOF evaluation of Continuous ElliCE against baselines for linear models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved LOF score.

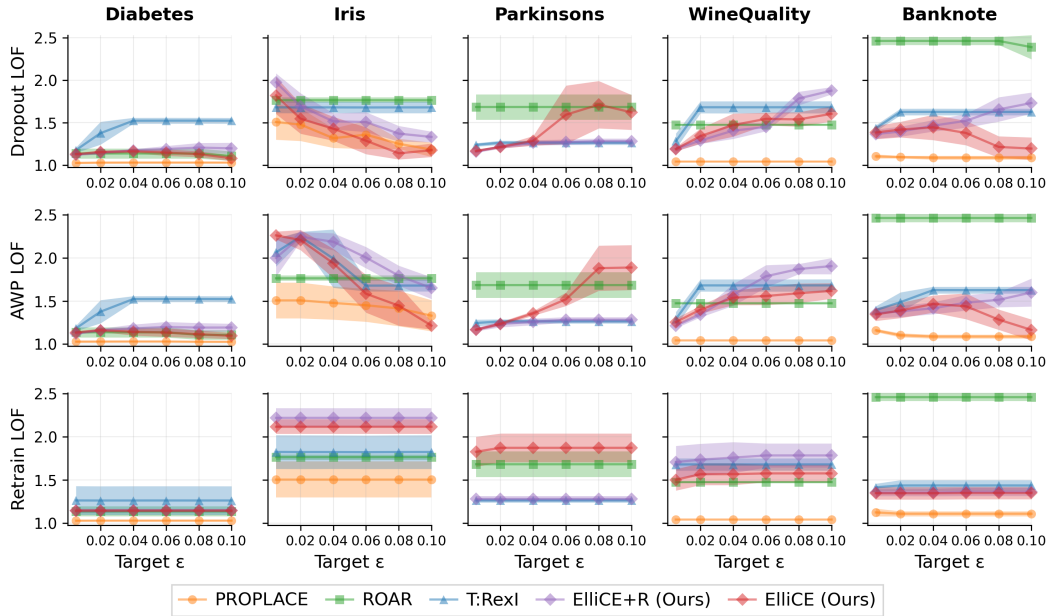


Figure 20: LOF evaluation of Continuous ElliCE against baselines for NN models. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved LOF score.

This proposition establishes our first condition (A) and provides a clear upper bound on the permissible magnitude of δ .

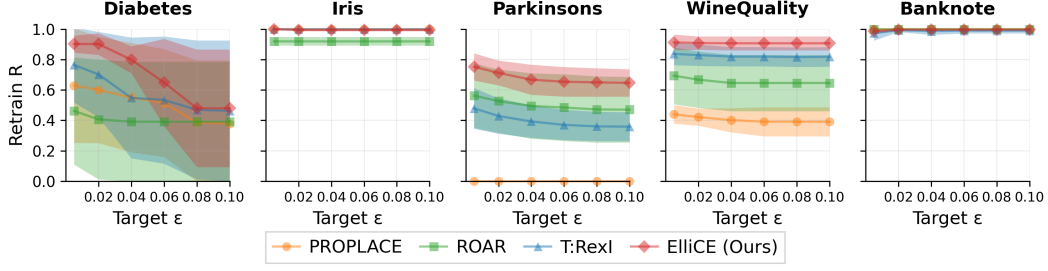


Figure 21: Robustness evaluation of Continuous ElliCE against baselines for MLP for Data Shift. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved Robustness score.

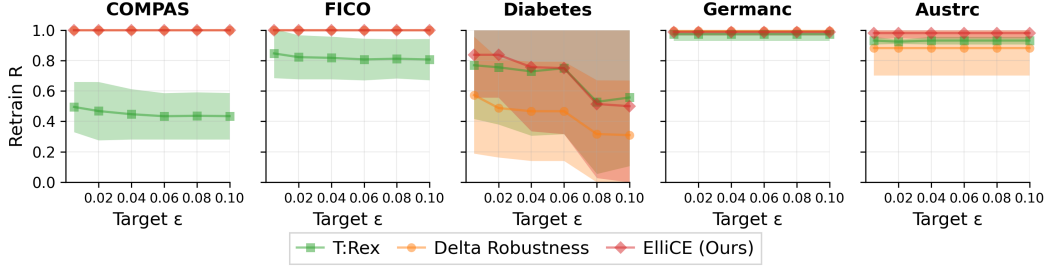


Figure 22: Robustness evaluation of Data supported ElliCE against baselines for MLP for Data Shift. On x-axis we have the target robustness level $\varepsilon_{\text{target}}$ and on y-axis the achieved Robustness score.

C.3 The Impact of Model Reparameterization

Neural networks exhibit reparameterization invariance, where different parameter configurations can represent functionally equivalent models. We now demonstrate that this property creates fundamental challenges for the delta interval approach.

Let a two-layer perceptron be given by

$$y = W_2 \sigma(W_1 x + b_1) + b_2,$$

where W_1, W_2 are weight matrices, b_1, b_2 are bias vectors and σ is the ReLU activation.

Reparameterization can be achieved by scaling. For any $\alpha > 0$ we define

$$W_1^{(\alpha)} = \alpha W_1, \quad b_1^{(\alpha)} = \alpha b_1, \quad W_2^{(\alpha)} = \frac{1}{\alpha} W_2, \quad b_2^{(\alpha)} = \frac{1}{\alpha} b_2$$

Because ReLU is positively homogeneous, which means that $\sigma(\alpha z) = \alpha \sigma(z), \forall \alpha > 0$, the output is unchanged:

$$y' = \frac{W_2}{\alpha} \sigma(\alpha(W_1 x + b_1)) + b_2 = \frac{W_2}{\alpha} \alpha \sigma(W_1 x + b_1) + b_2 = y$$

Theorem 6. *Let*

$$f_{\Theta}(x) = W_2 \sigma(W_1 x + b_1) + b_2, \quad \Theta = (W_1, b_1, W_2, b_2)$$

be a two-layer ReLU (1-Lipschitz) perceptron. Suppose the decision threshold is 0, input data is $X = [-1, 1]^d$ and the negative final bias $b_2 < 0$.

Fix $\delta > 0$ and $\gamma > 0$. Suppose the following two conditions for Θ hold:

1. (A) *Final-layer robustness bound*

$$\delta < \|W_2\|_{\infty}$$

(see Proposition 1)

2. (B) Layer-wise tangibility

For each layer $i \in \{1, 2\}$, there exists a parameter perturbation in that layer that is no larger than δ in ℓ_∞ norm what changes the output by at least γ for some $x \in X$.

For any pair (W, b) and radius $\delta > 0$ set:

$$\mathcal{B}_\infty := \left\{ (W', b') \mid \|W' - W\|_\infty \leq \delta, \|b' - b\|_\infty \leq \delta \right\}$$

Now layer-wise tangibility condition for a particular pair W, b can be defined as as:

$$\sup_{\substack{x \in \mathcal{X} \\ (W', b') \in \mathcal{B}_\infty((W, b), \delta)}} |f(W', b') - f(W'_{-i} \cup W_i, b'_{-i} \cup b_i)| > \gamma \quad (3)$$

where $\gamma > 0$ is a tangibility threshold. For binary classification, γ can be set to:

$$\gamma = \min_{x \in \mathcal{D}} |f(x)| \quad (4)$$

representing the minimum distance to the decision boundary ($f(x) = 0$) across all examples in the training dataset \mathcal{D} . This condition ensures that perturbations to each layer can potentially change the prediction for at least some examples in the dataset, making each layer's contribution meaningful to the model's decision-making process.

Then there exists a reparameterization M' with weights $W'_1 = \alpha W_1$, $b'_1 = \alpha b_1$, $W'_2 = W_2/\alpha$, and $b'_2 = b_2$, for which no choice of δ' can simultaneously satisfy both equivalent conditions specified for the new model.

Proof. . After reparameterization, equivalent condition to (A) for new model requires:

$$\delta < \|W_2/\alpha\|_\infty.$$

We now establish a key lemma:

Lemma 2. Consider the first layer of a two-layer neural network with inputs $x \in [-1, 1]^d$ and 1-Lipschitz activation function σ . Let a perturbation W'_1 and b'_1 of the weight matrix W_1 and bias vector b_1 be such that $\|W'_1 - W_1\|_\infty \leq \delta$ and $\|b'_1 - b_1\|_\infty \leq \delta$.

Then, the resulting change in the network's logit value (the output before the final decision) is at most

$$2\delta \cdot \|W_2\|_\infty.$$

Proof of Lemma. Let $x \in [-1, 1]^d$ be an input, W_1 and W'_1 be weight matrices such that $\|W_1 - W'_1\|_\infty \leq \delta$, and σ be the activation function. The change in the pre-activation of the second layer is:

$$\begin{aligned} \|\sigma(W_1 x + b_1) - \sigma(W'_1 x + b'_1)\|_\infty &\leq \|W_1 x - W'_1 x + b_1 - b'_1\|_\infty \\ &\leq \|(W_1 - W'_1)x + (b_1 - b'_1)\|_\infty \\ &\leq \|W_1 - W'_1\|_\infty \|x\|_\infty + \|b_1 - b'_1\|_\infty \\ &\leq \delta \cdot 1 + \delta = 2\delta. \end{aligned}$$

where we use the 1-Lipschitz property of common activation functions and the fact that $\|x\|_\infty \leq 1$.

The maximum change in the logit value is thus:

$$\|W_2 \sigma(W_1 x) - W_2 \sigma(W'_1 x) + b_2 - b'_2\|_\infty \leq \|W_2\|_\infty \cdot \|\sigma(W_1 x) - \sigma(W'_1 x)\|_\infty \leq \|W_2\|_\infty \cdot 2\delta.$$

□

Returning to our main proof, we must now also account for perturbations in the final-layer weights. First, by the lemma, any change in the first layer alone can shift the logit by at most

$$\|W_2\sigma(W_1x + b_1) - W_2\sigma(W'_1x + b'_1)\|_\infty \leq 2\delta \|W_2\|_\infty.$$

Next, for any perturbed second-layer weight W'_2 with $\|W_2 - W'_2\|_\infty \leq \delta$,

$$\|W'_2\sigma(W_1x + b_1) - W'_2\sigma(W'_1x + b'_1)\|_\infty \leq 2\delta \|W'_2\|_\infty \leq 2\delta(\|W_2\|_\infty + \delta) = 2\delta\|W_2\|_\infty + 2\delta^2$$

Putting both effects together, for this δ perturbation to be tangible, maximum possible perturbation should satisfy:

$$2\delta\|W_2\|_\infty + 2\delta^2 > \gamma.$$

Under the reparameterization $W'_2 = W_2/\alpha$, this becomes

$$2\delta \frac{\|W_2\|_\infty}{\alpha} + 2\delta^2 > \gamma \implies 2\delta^2 + 2\delta \frac{\|W_2\|_\infty}{\alpha} - \gamma > 0 \quad (5)$$

Meanwhile condition (A) still forces

$$\delta < \frac{\|W_2\|_\infty}{\alpha}. \quad (6)$$

Thus we need a δ satisfying

$$2\delta^2 + 2\delta \frac{\|W_2\|_\infty}{\alpha} - \gamma > 0, \quad \delta < \frac{\|W_2\|_\infty}{\alpha}.$$

For any fixed $\gamma > 0$, as $\alpha \rightarrow \infty$ the linear term $\frac{2\|W_2\|_\infty}{\alpha} \delta$ vanishes, so (4) forces

$$\delta > \sqrt{\frac{\gamma}{2}} - \epsilon > 0.$$

For some small constant ϵ . But for sufficiently large α we have

$$\sqrt{\frac{\gamma}{2}} - \epsilon > \frac{\|W_2\|_\infty}{\alpha},$$

so no δ can satisfy both (5) and (6). This completes the proof. \square

Implications

Under the hypotheses of the theorem, and more broadly whenever a *single, layer-agnostic* radius δ is used, two tendencies emerge:

1. If δ is chosen comparatively large, the resulting δ -interval may admit models in which some layers could shrink to (near-)zero. Such potentially degenerate members of the interval might predict the same undesired class for every input, so otherwise plausible counterfactuals could cease to be robust.
2. Because neural networks admit reparameterisations that preserve their input–output behaviour, one may be able to construct an equivalent parameter set for which *no* positive radius satisfies both the "non-degeneracy" and "tangibility" requirements. In other words, a single global δ may fail to accommodate all members of a functionally equivalent family.

These observations suggest that — without additional machinery such as layer-specific or parameter-specific radii, its ability to capture meaningful model multiplicity may be limited.

D Enhanced Algorithmic Recourse: Actionability, Sparsity, and Extended Applications

This section presents additional contributions beyond the core ElliCE framework, encompassing enhancements for actionability, sparsity control, plausibility evaluation, and extensions to broader model classes and feature types. These improvements address practical deployment challenges and extend the applicability of robust counterfactual generation to more diverse real-world scenarios.

D.1 Plausibility Evaluation

Plausibility is an essential part of counterfactual explanations, ensuring that generated recourse lies in realistic regions of the feature space.

To quantify plausibility, we adopt the *Local Outlier Factor* (LOF) score, a widely used measure of local density deviation in outlier detection [35]. Let S denote the training dataset, $x \in \mathbb{R}^d$ a query point, and $L_k(x)$ the set of its k nearest neighbors in S under an ℓ_p distance $\delta(\cdot, \cdot)$. For any neighbor $\mathbf{x}_c \in L_k(x)$, we define the k -distance of \mathbf{x}_c as $d_k(\mathbf{x}_c)$, and the *reachability distance* of x with respect to \mathbf{x}_c as

$$r_{d_k}(x, \mathbf{x}_c) := \max\{\delta(x, \mathbf{x}_c), d_k(\mathbf{x}_c)\}.$$

The *local reachability density* of x is then

$$\text{lrd}_k(x) := \frac{|L_k(x)|}{\sum_{\mathbf{x}_c \in L_k(x)} r_{d_k}(x, \mathbf{x}_c)}.$$

Finally, the LOF score of x with respect to S is defined as

$$\text{LOF}_{k,S}(x) := \frac{1}{|L_k(x)|} \sum_{\mathbf{x}_c \in L_k(x)} \frac{\text{lrd}_k(\mathbf{x}_c)}{\text{lrd}_k(x)}.$$

By construction, $\text{LOF}_{k,S}(x) \approx 1$ indicates that x lies in a region of comparable density to its neighbors, and is therefore considered plausible. Values $\text{LOF}_{k,S}(x) > 1$ suggest that x lies in a relatively sparse region (i.e., more outlier-like), while values < 1 indicate higher-than-average local density. For a set of counterfactuals, we report the mean LOF score across all points. This measure provides a principled way to compare the plausibility of counterfactuals generated by ElliCE and competing baselines.

For data-supported counterfactuals, ElliCE achieves plausibility comparable to baseline methods since explanations lie on the data manifold by construction. For non-data-supported counterfactuals, ElliCE performs competitively with methods not specifically designed for plausibility, outperforming TRexI and ROAR on the diabetes dataset while maintaining superior robustness guarantees. PROPLACE, which is explicitly designed for plausibility optimization, achieves better LOF scores but at the cost of reduced robustness as demonstrated in our main experimental results.

LOF metrics results are presented in Figures 8, 5, 19 and 20.

The inherent robustness-proximity trade-off in ElliCE, explored in section B, naturally contributes to improved plausibility. By requiring counterfactuals to remain valid across multiple models in the Rashomon set, our method pushes explanations away from unstable regions near decision boundaries, often resulting in more realistic feature combinations that better align with the underlying data distribution.

D.2 Actionability Enhancements

In practical deployment, counterfactual explanations must respect application-specific constraints that restrict which features can be modified and how these modifications may occur. To incorporate such requirements, we extend ElliCE with actionability-aware mechanisms that ensure generated recourse recommendations remain feasible.

D.2.1 Immutable Features and Directional Constraints

We distinguish two primary classes of actionability constraints. (1) *Immutable features*: attributes such as demographic characteristics or historical records that cannot be modified. These are enforced by gradient masking in continuous optimization and by pre-filtering in data-supported search. (2) *Range-constrained features*: attributes restricted to remain within predefined feasible intervals. This general class includes monotonic one-directional changes as a special case, e.g., non-decreasing education level or non-increasing debt. Range constraints are enforced by projection operators during optimization and by pre-filtering infeasible candidates in search-based methods.

Table 4 reports an illustration using the German Credit dataset. The example contrasts the closest counterfactual with an actionable alternative that respects immutability (e.g., age, foreign worker,

Algorithm 3 Continuous Sparsity Optimization

Require: original instance x , model parameters θ , constraint set \mathcal{C}

Ensure: sparse robust counterfactual \mathbf{x}_c

```
1:  $\mathbf{x}_c \leftarrow x$ 
2: Run full optimization to accumulate gradient magnitudes  $g_i$  for each feature  $i$ 
3: Rank features by decreasing  $|g_i|$  to obtain list  $\mathcal{L}$ 
4:  $\mathcal{A} \leftarrow \emptyset$ 
5: for feature  $i \in \mathcal{L}$  do
6:    $\mathcal{A} \leftarrow \mathcal{A} \cup \{i\}$ 
7:   Optimize  $\mathbf{x}_c$  with gradients masked outside  $\mathcal{A}$  and  $\mathbf{x}_c$  projected onto  $\mathcal{C}$  (range and im-
     mutability constraints) during each step
8:   if robustness criterion is satisfied then
9:     break
10: return  $\mathbf{x}_c$ 
```

gender). The results demonstrate the trade-off between minimal proximity and constraint satisfaction: actionable counterfactuals require larger modifications but remain valid and feasible. The example in the Table 4 was generated using Continuous ElliCE method.

D.3 Sparsity Control

Sparse counterfactuals, which alter fewer features, are generally more interpretable and actionable. ElliCE incorporates possibility of sparsity control for both data-supported and continuous optimization settings.

D.3.1 Data-Supported Sparsity Optimization

For data-supported methods, sparsity is promoted either by criteria-based filtering or by a modified BallTree distance:

$$d(x, y) = C \cdot \text{Hamming}(x, y) + \|x - y\|_1 = C \cdot \|x, y\|_0 + \|x - y\|_1,$$

where $C > 0$ controls the relative importance of sparsity. Larger values of C emphasize reducing the number of modified features, while smaller values prioritize proximity. This formulation enables efficient sublinear search while balancing sparsity and distance objectives.

Moreover, ElliCE is method-agnostic with respect to nearest neighbor search and supports any available algorithm, including standard KDTree search with ℓ_1 or ℓ_2 distances.

D.3.2 Continuous Sparsity Optimization

For gradient-based methods, sparsity is enforced through an iterative procedure that integrates feature importance ranking, greedy selection, and coordinate masking. The procedure is summarized in Algorithm 3.

This procedure produces counterfactuals that are both robust and sparse by modifying only the most influential features while preventing unnecessary changes.

D.4 Multi-Class Extension

Although our primary analysis considers binary classification, ElliCE naturally extends to multi-class settings while preserving convexity. For K classes and target y^* , we impose robustness constraints against all competitors:

$$\min_{\theta \in \tilde{\mathcal{R}}(\epsilon)} [\theta_{y^*}^\top \mathbf{x}_c - \theta_c^\top \mathbf{x}_c] \geq \tau, \quad \forall c \neq y^*,$$

where $\tau \geq 0$ is a robustness margin. This produces $K - 1$ second-order cone constraints, maintaining computational tractability. For neural architectures, we apply the ellipsoidal Rashomon approximation in embedding space, with optimization via gradient descent. All theoretical guarantees (uniqueness, stability, alignment) remain intact under this extension.

Table 5: Runtime performance. Data supported CE for MLP. Time required for method-dependent preprocessing and for computing counterfactuals on the validation set for each subfold.

Dataset	ELIICE	ELIICE+R	T:Rex	Delta Rob
<i>Linear</i>				
FICO	1.521 ± 0.019	—	5.447 ± 0.062	10.734 ± 0.070
COMPAS	0.816 ± 0.016	—	3.037 ± 0.056	4.640 ± 0.053
Australian	0.055 ± 0.007	—	0.288 ± 0.010	0.544 ± 0.008
Diabetes	0.051 ± 0.000	—	0.266 ± 0.004	0.413 ± 0.005
German	0.089 ± 0.008	—	0.433 ± 0.008	1.214 ± 0.023
<i>MLP</i>				
FICO	1.792 ± 0.123	1.283 ± 0.076	7.006 ± 0.058	242.035 ± 1.161
COMPAS	0.526 ± 0.011	0.443 ± 0.014	3.534 ± 0.128	360.480 ± 6.701
Australian	0.057 ± 0.011	0.036 ± 0.003	0.281 ± 0.006	2.783 ± 0.032
Diabetes	0.053 ± 0.001	0.033 ± 0.001	0.296 ± 0.006	1.922 ± 0.032
German	0.101 ± 0.001	0.037 ± 0.001	0.432 ± 0.013	9.905 ± 0.068

D.5 Mixed Feature Types with Gumbel-Softmax

Many real-world datasets contain a mixture of continuous and categorical variables, which complicates joint optimization. We adopt a unified relaxation strategy that treats the two types differently but within a single optimization framework.

Continuous features. Continuous variables are optimized directly in \mathbb{R}^d , with projection operators enforcing domain-specific constraints such as feature bounds and immutability.

Categorical features. Each categorical variable represented by a one-hot group is relaxed into a probability vector using the Gumbel-Softmax distribution. Given a logit vector $z \in \mathbb{R}^K$ for a categorical group with K possible categories, we draw Gumbel noise $g_i \sim \text{Gumbel}(0, 1)$ and compute

$$y_i = \frac{\exp((z_i + g_i)/\tau)}{\sum_{j=1}^K \exp((z_j + g_j)/\tau)}, \quad i = 1, \dots, K,$$

where $\tau > 0$ is a fixed temperature parameter. As $\tau \rightarrow 0$, y approaches a one-hot vector, while for larger τ the distribution is smoother. In practice, we keep τ fixed during optimization, yielding a differentiable approximation that allows backpropagation through categorical choices while preserving the simplex constraint $\sum_i y_i = 1$.

Unified optimization. The counterfactual candidate x_c is obtained by combining continuous variables with the relaxed categorical vectors y . Gradient-based updates are applied jointly to both parts, with projection steps ensuring immutability and range constraints. During optimization, we sample discrete counterfactuals by fixing continuous values and drawing one-hot categorical assignments. If at least one sampled counterfactual satisfies the robustness condition, gradient optimization is terminated. A fixed number of additional samples are then evaluated to explore whether a better robust counterfactual can be identified.

Table 6: Runtime performance. Continuous CE for MLP. Time required for method-dependent preprocessing and for computing counterfactuals on the validation set for each subfold.

Dataset	Ellice	Ellice+R	T:Rex	ROAR	PROPLACE
<i>Linear</i>					
Diabetes	0.250 ± 0.031	—	19.476 ± 1.116	15.282 ± 1.533	0.836 ± 0.038
Iris	0.563 ± 0.023	—	3.219 ± 0.354	15.603 ± 0.242	0.639 ± 0.008
Parkinsons	12.339 ± 4.898	—	81.553 ± 13.906	73.409 ± 11.057	11.157 ± 0.256
Wine Quality	1.407 ± 0.061	—	54.151 ± 2.269	58.821 ± 1.432	4.876 ± 0.158
Banknote	0.493 ± 0.010	—	6.579 ± 9.541	19.914 ± 11.567	1.088 ± 0.017
<i>MLP</i>					
Diabetes	4.861 ± 1.539	1.122 ± 0.578	56.104 ± 12.937	9.448 ± 0.579	78.323 ± 85.695
Iris	0.562 ± 0.025	0.476 ± 0.008	20.984 ± 3.801	17.134 ± 1.816	3.703 ± 0.383
Parkinsons	112.323 ± 6.253	1.409 ± 0.037	25.771 ± 10.461	104.660 ± 14.314	—
Wine Quality	26.961 ± 4.745	17.199 ± 4.981	160.531 ± 37.765	39.688 ± 5.101	869.119 ± 794.885
Banknote	0.631 ± 0.030	0.526 ± 0.062	14.770 ± 3.422	22.985 ± 3.777	10.622 ± 2.580

Table 7: Performance comparison of Data-Supported counterfactual methods at target $\varepsilon_{target} = 0.1\hat{L}_{train}(f_{baseline})$

Dataset	Method	Evaluation Metrics					
		Retrain		Dropout Rashomon		AWP	
		R \uparrow	L2 \downarrow	R \uparrow	L2 \downarrow	R \uparrow	L2 \downarrow
Linear models							
Australian	Ellice	—	—	0.981 (0.03)	2.46 (0.12)	0.991 (0.01)	2.57 (0.15)
	DeltaRob	—	—	0.722 (0.28)	2.64 (0.32)	0.531 (0.30)	2.51 (0.25)
	T:Rex	—	—	0.988 (0.01)	2.62 (0.16)	0.983 (0.01)	2.73 (0.18)
COMPAS	Ellice	—	—	0.501 (0.50)	2.35 (0.91)	0.500 (0.50)	2.13 (1.12)
	DeltaRob	—	—	0.001 (0.00)	1.24 (0.11)	0.000 (0.00)	1.09 (0.04)
	T:Rex	—	—	0.000 (0.00)	0.77 (0.02)	0.000 (0.00)	0.77 (0.02)
Diabetes	Ellice	—	—	0.964 (0.06)	2.65 (0.06)	0.989 (0.01)	3.10 (0.22)
	DeltaRob	—	—	0.670 (0.33)	2.97 (0.44)	0.033 (0.02)	2.39 (0.20)
	T:Rex	—	—	0.946 (0.04)	3.04 (0.18)	0.685 (0.27)	3.43 (0.43)
FICO	Ellice	—	—	1.000 (0.00)	4.93 (0.17)	1.000 (0.00)	5.46 (0.14)
	DeltaRob	—	—	0.068 (0.03)	3.61 (0.19)	0.008 (0.00)	3.74 (0.06)
	T:Rex	—	—	0.035 (0.02)	3.02 (0.15)	0.003 (0.00)	2.89 (0.25)
German	Ellice	—	—	1.000 (0.00)	3.99 (0.20)	0.997 (0.01)	3.85 (0.11)
	DeltaRob	—	—	0.983 (0.02)	4.34 (0.57)	1.000 (0.00)	4.01 (0.14)
	T:Rex	—	—	0.969 (0.04)	3.91 (0.16)	1.000 (0.00)	3.92 (0.16)
Multi-layer perceptron							
Australian	Ellice	1.000 (0.00)	2.30 (0.14)	1.000 (0.00)	2.61 (0.14)	0.954 (0.04)	2.51 (0.15)
	Ellice+R	0.996 (0.01)	2.24 (0.13)	0.993 (0.01)	2.70 (0.23)	0.954 (0.05)	2.53 (0.13)
	DeltaRob	1.000 (0.00)	2.28 (0.15)	1.000 (0.00)	3.16 (0.45)	1.000 (0.00)	3.29 (0.45)
	T:Rex	0.996 (0.01)	2.25 (0.09)	0.992 (0.01)	2.69 (0.13)	0.942 (0.08)	2.82 (0.40)
COMPAS	Ellice	1.000 (0.00)	1.12 (0.12)	0.814 (0.32)	1.65 (0.17)	0.430 (0.35)	1.79 (0.15)
	Ellice+R	0.999 (0.00)	1.08 (0.19)	1.000 (0.00)	1.86 (0.04)	0.998 (0.00)	1.99 (0.05)
	DeltaRob	1.000 (0.00)	1.20 (0.10)	0.999 (0.00)	1.92 (0.16)	0.775 (0.39)	2.31 (0.42)
	T:Rex	0.422 (0.34)	0.72 (0.06)	0.000 (0.00)	0.63 (0.02)	0.000 (0.00)	0.60 (0.02)
Diabetes	Ellice	1.000 (0.00)	2.27 (0.13)	0.993 (0.01)	2.78 (0.26)	1.000 (0.00)	3.11 (0.18)
	Ellice+R	0.992 (0.01)	2.24 (0.17)	0.996 (0.01)	2.81 (0.21)	0.980 (0.03)	3.01 (0.34)
	DeltaRob	0.947 (0.06)	2.39 (0.13)	0.549 (0.23)	2.50 (0.15)	0.302 (0.17)	2.50 (0.03)
	T:Rex	0.988 (0.02)	2.25 (0.16)	0.962 (0.06)	3.06 (0.34)	0.869 (0.08)	3.09 (0.26)
FICO	Ellice	1.000 (0.00)	3.53 (0.17)	1.000 (0.00)	4.91 (0.22)	1.000 (0.00)	5.06 (0.29)
	Ellice+R	1.000 (0.00)	3.72 (0.23)	1.000 (0.00)	4.75 (0.17)	0.993 (0.01)	5.58 (0.62)
	DeltaRob	1.000 (0.00)	4.00 (0.10)	1.000 (0.00)	5.67 (0.58)	0.957 (0.07)	5.70 (0.72)
	T:Rex	0.830 (0.08)	3.12 (0.07)	0.006 (0.00)	3.07 (0.11)	0.001 (0.00)	2.77 (0.19)
German	Ellice	1.000 (0.00)	3.48 (0.10)	0.996 (0.01)	4.32 (0.31)	1.000 (0.00)	4.00 (0.24)
	Ellice+R	0.990 (0.02)	3.44 (0.08)	0.996 (0.01)	4.00 (0.06)	0.947 (0.06)	3.94 (0.17)
	DeltaRob	0.982 (0.01)	3.45 (0.06)	0.988 (0.02)	4.00 (0.15)	1.000 (0.00)	3.99 (0.22)
	T:Rex	0.989 (0.01)	3.47 (0.04)	0.967 (0.02)	4.03 (0.20)	0.989 (0.01)	4.23 (0.24)

Table 8: Performance comparison of Continuous counterfactual methods at target $\varepsilon_{target} = 0.1\hat{L}_{train}(f_{baseline})$ (continuous datasets), using L2 distance.

Dataset	Method	Evaluation Metrics					
		Retrain		Dropout Rashomon		AWP	
		R \uparrow	L2 \downarrow	R \uparrow	L2 \downarrow	R \uparrow	L2 \downarrow
Linear models							
Banknote	Ellice	—	—	0.623 (0.26)	1.470 (0.07)	0.993 (0.00)	1.259 (0.04)
	PROPLACE	—	—	0.069 (0.03)	1.329 (0.07)	0.668 (0.14)	1.348 (0.07)
	ROAR	—	—	0.002 (0.00)	0.890 (0.13)	0.399 (0.11)	0.890 (0.13)
	T:Rex	—	—	0.132 (0.08)	1.163 (0.05)	0.814 (0.05)	1.163 (0.05)
Diabetes	Ellice	—	—	0.996 (0.01)	2.669 (0.13)	0.996 (0.01)	2.914 (0.17)
	PROPLACE	—	—	0.499 (0.14)	2.242 (0.15)	0.118 (0.10)	2.357 (0.11)
	ROAR	—	—	0.970 (0.02)	2.962 (0.15)	0.973 (0.02)	3.010 (0.12)
	T:Rex	—	—	0.974 (0.02)	4.188 (0.46)	0.808 (0.10)	4.188 (0.46)
Iris	Ellice	—	—	1.000 (0.00)	3.265 (0.15)	1.000 (0.00)	3.213 (0.10)
	PROPLACE	—	—	0.745 (0.27)	2.929 (0.08)	0.633 (0.23)	2.929 (0.08)
	ROAR	—	—	0.105 (0.11)	2.578 (0.13)	0.005 (0.01)	2.492 (0.19)
	T:Rex	—	—	0.173 (0.11)	2.701 (0.14)	0.060 (0.06)	2.552 (0.20)
Parkinsons	Ellice	—	—	0.997 (0.00)	4.236 (1.03)	0.997 (0.00)	3.621 (1.22)
	PROPLACE	—	—	0.269 (0.09)	3.062 (0.11)	0.046 (0.01)	2.414 (0.04)
	ROAR	—	—	0.923 (0.13)	2.419 (0.23)	1.000 (0.00)	2.080 (0.15)
	T:Rex	—	—	0.919 (0.10)	2.148 (0.18)	0.986 (0.01)	2.148 (0.18)
Wine Quality	Ellice	—	—	0.998 (0.00)	3.300 (0.20)	0.998 (0.00)	3.306 (0.07)
	PROPLACE	—	—	0.470 (0.11)	3.037 (0.29)	0.154 (0.09)	3.030 (0.29)
	ROAR	—	—	0.973 (0.02)	3.502 (0.13)	0.994 (0.01)	3.431 (0.31)
	T:Rex	—	—	0.961 (0.02)	3.470 (0.19)	0.978 (0.01)	3.470 (0.19)
Multi-layer perceptron							
Banknote	Ellice	1.000 (0.00)	1.125 (0.06)	0.995 (0.01)	1.644 (0.13)	0.990 (0.01)	1.475 (0.09)
	Ellice+R	0.995 (0.01)	1.116 (0.08)	0.993 (0.01)	1.496 (0.09)	0.993 (0.01)	1.365 (0.06)
	PROPLACE	1.000 (0.00)	1.339 (0.05)	0.619 (0.26)	1.386 (0.02)	0.692 (0.30)	1.386 (0.02)
	ROAR	1.000 (0.00)	2.323 (0.03)	1.000 (0.00)	2.323 (0.03)	1.000 (0.00)	2.323 (0.03)
	T:Rex	0.993 (0.01)	1.188 (0.11)	0.988 (0.02)	1.563 (0.12)	1.000 (0.00)	1.563 (0.12)
Diabetes	Ellice	0.977 (0.01)	2.146 (0.39)	0.985 (0.02)	3.050 (0.34)	0.980 (0.02)	3.221 (0.40)
	Ellice+R	0.978 (0.04)	2.133 (0.33)	0.965 (0.02)	2.864 (0.46)	0.985 (0.00)	3.414 (0.63)
	PROPLACE	0.482 (0.48)	2.007 (0.05)	0.187 (0.28)	2.007 (0.05)	0.079 (0.19)	2.007 (0.05)
	ROAR	0.864 (0.11)	1.858 (0.24)	0.400 (0.28)	1.858 (0.24)	0.308 (0.26)	1.858 (0.24)
	T:Rex	0.944 (0.03)	2.471 (0.86)	0.899 (0.08)	4.181 (0.36)	0.940 (0.04)	4.181 (0.36)
Iris	Ellice	0.995 (0.01)	2.492 (0.17)	1.000 (0.00)	2.936 (0.12)	0.998 (0.00)	2.333 (0.10)
	Ellice+R	0.975 (0.03)	2.401 (0.18)	0.970 (0.03)	2.916 (0.21)	0.995 (0.01)	2.193 (0.10)
	PROPLACE	1.000 (0.00)	2.929 (0.08)	1.000 (0.00)	2.933 (0.08)	1.000 (0.00)	2.929 (0.08)
	ROAR	0.930 (0.03)	2.851 (0.09)	0.758 (0.08)	2.851 (0.09)	0.930 (0.03)	2.851 (0.09)
	T:Rex	1.000 (0.00)	2.838 (0.34)	0.922 (0.08)	2.973 (0.15)	0.992 (0.01)	2.201 (0.11)
Parkinsons	Ellice	0.885 (0.02)	2.560 (0.32)	0.999 (0.00)	1.226 (0.18)	0.999 (0.00)	1.403 (0.07)
	Ellice+R	0.610 (0.07)	1.170 (0.07)	0.996 (0.00)	1.112 (0.07)	0.993 (0.00)	1.100 (0.10)
	PROPLACE	—	—	—	—	—	—
	ROAR	0.593 (0.15)	1.902 (0.21)	0.853 (0.08)	1.902 (0.21)	0.854 (0.08)	1.902 (0.21)
	T:Rex	0.577 (0.06)	1.062 (0.04)	0.990 (0.00)	1.062 (0.04)	0.987 (0.00)	1.062 (0.04)
Wine Quality	Ellice	0.964 (0.02)	3.695 (0.39)	0.978 (0.01)	3.457 (0.38)	0.939 (0.04)	3.653 (0.42)
	Ellice+R	0.969 (0.03)	4.185 (0.70)	0.978 (0.02)	2.848 (0.30)	0.997 (0.00)	4.266 (0.67)
	PROPLACE	0.159 (0.16)	1.796 (0.05)	0.068 (0.07)	1.796 (0.05)	0.020 (0.03)	1.796 (0.05)
	ROAR	0.918 (0.04)	2.674 (0.06)	0.867 (0.06)	2.674 (0.06)	0.755 (0.10)	2.674 (0.06)
	T:Rex	0.877 (0.05)	3.252 (0.32)	0.931 (0.03)	3.252 (0.32)	0.900 (0.04)	3.252 (0.32)