On the Trade-Off between Actionable Explanations and the Right to be Forgotten

Martin Pawelczyk
University of Tübingen
first.last@uni-tuebingen.de

Tobias Leemann
University of Tübingen
first.last@uni-tuebingen.de

Asia Biega *
Max Planck Institute for Security and Privacy
asia.biega.de

Gjergji Kasneci *
University of Tübingen
first.last@uni-tuebingen.de

Abstract

As machine learning (ML) models are increasingly being deployed in high-stakes applications, policymakers have suggested tighter data protection regulations (e.g., GDPR, CCPA). One key principle is the "right to be forgotten" which gives users the right to have their data deleted. Another key principle is the right to an actionable explanation, also known as algorithmic recourse, allowing users to reverse unfavorable decisions. To date, it is unknown whether these two principles can be operationalized simultaneously. Therefore, we introduce and study the problem of recourse invalidation in the context of data deletion requests. More specifically, we theoretically and empirically analyze the behavior of popular state-of-the-art algorithms and demonstrate that the recourses generated by these algorithms are likely to be invalidated if a small number of data deletion requests (e.g., 1 or 2) warrant updates of the predictive model. For the setting of linear models and overparameterized neural networks – studied through the lens of neural tangent kernels (NTKs) – we suggest a framework to identify a minimal subset of critical training points which, when removed, maximize the fraction of invalidated recourses. Using our framework, we empirically show that the removal of as little as 2 data instances from the training set can invalidate up to 95 percent of all recourses output by popular state-of-the-art algorithms. Thus, our work raises fundamental questions about the compatibility of "the right to an actionable explanation" in the context of the "right to be forgotten" while also providing constructive insights on the determining factors of recourse robustness.

1 Introduction

Machine learning (ML) models make a variety of consequential decisions in domains such as finance, healthcare, and policy. To protect users, laws such as the European Union's General Data Protection Regulation (GDPR) [21] or the California Consumer Privacy Act (CCPA) [40] constrain the usage of personal data and ML model deployments. For example, individuals who have been adversely impacted by the predictions of these models have the right to *recourse* [58], i.e., a constructive instruction on how to act to arrive at a more desirable outcome (e.g., change a model prediction from "loan denied" to "approved"). Several approaches in recent literature tackled the problem of providing recourses by generating instance level counterfactual explanations [59, 55, 34, 43].

Complementarily, data protection laws provide users with greater authority over their personal data. For instance, users are granted the right to *withdraw consent to the usage of their data* at any time [5]. These regulations affect technology platforms that train their ML models on personal user data under

2022 Trustworthy and Socially Responsible Machine Learning (TSRML 2022) co-located with NeurIPS 2022.

the respective legal regime. Law scholars have argued that the continued use of ML models relying on deleted data instances could be deemed illegal [57].

Irrespective of the underlying mandate, data deletion has raised a number of algorithmic research questions. In particular, recent literature has focused on the efficiency of deletion (i.e., how to delete individual data points without retraining the model [24, 27]) and model accuracy aspects of data deletion (i.e., how to remove data without compromising model accuracy [6, 29]). An aspect of data deletion which has not been examined before is *whether and how data deletion may impact model explanation frameworks*. Thus, there is a need to understand and systematically characterize the limitations of recourse algorithms when personal user data may need to be deleted from trained ML models. Indeed, deletion of certain data instances might invalidate actionable model explanations – both for the deleting user and, critically, unsuspecting other users. Such invalidations can be especially problematic in cases where users have already started to take costly actions to change their model outcomes based on previously received explanations.

In this paper, we formally examine the problem of algorithmic recourse in the context of data deletion requests. We consider the setting where a small set of individuals has decided to withdraw their data and, as a consequence of the deletion request, the model needs to be updated [24]. In particular, this work tackles the subsequent pressing question:

What is the worst impact that a deleted data instance can have on the recourse validity?

We approach this question by considering two distinct scenarios. The first setting considers to what extent the outdated recourses still lead to a desirable prediction (e.g., loan approval) on the updated model. For this scenario, we suggest a robustness measure called *recourse outcome instability* to quantify the fragility of recourse methods. Second, we consider the setting where the recourse action is being updated as a consequence of the prediction model update. In this case, we study what maximal change in recourse will be required to maintain the desirable prediction. To quantify the extent of this second problem, we suggest the notion of *recourse action instability*.

Given these robustness measures, we derive and analyze theoretical worst-case guarantees of the maximal instability induced for linear models and neural networks in the overparameterized regime, which we study through the lens of neural tangent kernels. We furthermore define an optimization problem for empirically quantifying recourse instability under data deletion. For a given trained ML model, we identify small sets of data points that maximize the proposed instability measures when deleted. Since the resulting brute-force approach (i.e., retraining models for every possible removal set) is NP-hard, we propose two relaxations for recourse instability maximization that can be optimized using (i) end-to-end gradient descent or (ii) via a greedy approximation algorithm. To summarize, in this work we make the following key contributions:

- **Novel recourse robustness problem.** We introduce the problem of *recourse invalidation under the right to be forgotten* by defining two new recourse instability measures.
- **Tractable algorithms.** Using our instability measures, we present an optimization framework to identify a small set of critical training data points which, when removed, invalidates most of the issued recourses.
- **Comprehensive experiments.** We conduct extensive experiments on multiple real-world data sets for both regression and classification tasks with our proposed algorithms, showing that the removal of even one point from the training set can invalidate up to 95 percent of all recourses output by state-of-the-art methods

Our results also have practical implications for system designers. First, our analysis and algorithms help identify parameters and model classes leading to higher stability when a trained ML model is subjected to deletion requests. Furthermore, our proposed methods can provide an informed way towards practical implementations of data minimization [20], as one could argue that data points contributing to recourse instability could be minimized out. Hence, our methods could increase designer's awareness and the compliance of their trained models.

2 Related Work

Algorithmic Approaches to Recourse. Several approaches in recent literature have been suggested to generate recourse for users who have been negatively impacted by model predictions [53, 36, 13,

59, 55, 56, 43, 37, 39, 34, 47, 12, 3, 51, 1]. These approaches generate recourses assuming a static environment without data deletion requests, where both the model and the recourse remain stable.

A related line of work has focused on determining the extent to which recourses remain invariant to the model choice [44, 7], to data distribution shifts [46, 54], perturbations to the input instances [4, 14, 50], or perturbations to the recourses [45].

Sample Deletion in Predictive Models. Since according to EU's GDPR individuals can request to have their data deleted, several approaches in recent literature have been focusing on updating a machine learning model without the need of retraining the entire model from scratch [61, 24, 31, 27, 28, 9]. A related line of work considers the problem of data valuation [22, 23]. Finally, removing subsets of training data is an ingredient used for model debugging [15] or the evaluation of explanation techniques [30, 48].

Contribution. While we do not suggest a new recourse algorithm, our work addresses the problem of recourse fragility in the presence of data deletion requests, which has previously not been studied. To expose this fragility, we suggest effective algorithms to delete a minimal subset of critical training points so that the fraction of invalidated recourses due to a required model update is maximized. Moreover, while prior research in the data deletion literature has primarily focused on effective data removal strategies for predictive models, there is no prior work that studies to what extent recourses output by state-of-the-art methods are affected by data deletion requests. Our work is the first to tackle these important problems and thereby paves the way for recourse providers to evaluate and rethink their recourse strategies in light of the right to be forgotten.

3 Preliminaries

The Predictive Model and the Data Deletion Mechanism. We consider prediction problems from some input space $\mathbb{R}^d$ to an output space $\mathcal{Y}$, where d is the number of input dimensions. We denote a sample by $\mathbf{z} = (\mathbf{x}, y)$, and denote the training data set by $\mathcal{D} = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$. Consider the weighted empirical risk minimization problem (ERM), which gives rise to the optimal model parameters:

$$\mathbf{w}_{\boldsymbol{\omega}} = \arg\min_{\mathbf{w}'} \sum_{i=1}^{n} \omega_i \cdot \ell\big(y_i, f_{\mathbf{w}'}(\mathbf{x}_i)\big), \tag{1}$$

where $\ell(\cdot, \cdot)$ is an instance-wise loss function (e.g., binary cross-entropy, mean-squared-error (MSE) loss, etc.) and $\boldsymbol{\omega} \in \{0, 1\}^n$ are data weights that *are fixed at training time*. If $\omega_i = 1$, then the point $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ is part of the training data set, otherwise it is not. During model training, we set $\omega_i = 1\ \forall i$, that is, the decision maker uses all available training instances at training time. In the optimization expressed in (1), the model parameters $\mathbf{w}$ are usually an implicit function of the data weight vector $\boldsymbol{\omega}$ and we write $\mathbf{w}_{\boldsymbol{\omega}}$ to highlight this fact; in particular, when all training instances are used we write $\mathbf{w}_{\mathbf{1}}$, where $\mathbf{1} \in \mathbb{R}^n$ is a vector of 1s. In summary, we have introduced the *weighted* ERM problem since it allows us to understand the impact of arbitrary data deletion patterns on actionable explanations as we allow users to withdraw their entire input $\mathbf{z}_i = (y_i, \mathbf{x}_i)$ from the training set used to train the model $f_{\mathbf{w}_{\mathbf{1}}}$. Next, we present the recourse model we consider.

The Recourse Problem in the Context of the Data Deletion Mechanism. We follow an established definition of counterfactual explanations originally proposed by [59]. For a given model $f_{\mathbf{w}_{\boldsymbol{\omega}}} : \mathbb{R}^d \to \mathbb{R}$ parameterized by $\mathbf{w}$ and a distance function $d(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, the problem of finding a recourse $\check{\mathbf{x}} = \mathbf{x} + \boldsymbol{\delta}$ for a factual instance $\mathbf{x}$ is given by:

$$\boldsymbol{\delta}_{\boldsymbol{\omega},\mathbf{x}} \in \arg\min_{\boldsymbol{\delta}' \in \mathcal{A}_d} (f_{\mathbf{w}_{\boldsymbol{\omega}}}(\mathbf{x} + \boldsymbol{\delta}') - s)^2 + \lambda \cdot d(\mathbf{x}, \mathbf{x} + \boldsymbol{\delta}'), \tag{2}$$

where $\lambda \geq 0$ is a scalar tradeoff parameter and s denotes the target score. In the optimization from (2), the optimal recourse action $\boldsymbol{\delta}$ usually depends on the model parameters and since the model parameters themselves depend on the exact data weights configuration we write $\boldsymbol{\delta}_{\boldsymbol{\omega},\mathbf{x}}$ to highlight this fact. The first term in the objective on the right-hand-side of (2) encourages the outcome $f_{\mathbf{w}_{\boldsymbol{\omega}}}(\check{\mathbf{x}})$ to become close to the user-defined target score s, while the second term encourages the distance between the factual instance $\mathbf{x}$ and the recourse $\check{\mathbf{x}}_{\boldsymbol{\omega}} := \mathbf{x} + \boldsymbol{\delta}_{\boldsymbol{\omega},\mathbf{x}}$ to be low. The set of constraints $\mathcal{A}_d$ ensures that only admissible changes are made to the factual $\mathbf{x}$.

Recourse Robustness Through the Lens of the Right to be Forgotten. We first introduce several key terms, namely, *prescribed recourses* and *recourse outcomes*. A prescribed recourse $\check{\mathbf{x}}$ refers to a

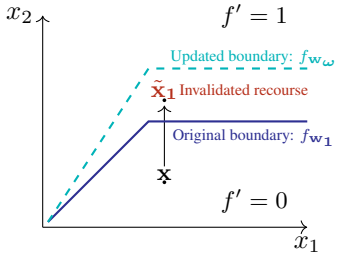
recourse that was provided to an end user by a recourse method (e.g., salary was increased by \$500). The recourse outcome $f(\check{\mathbf{x}})$ is the model's prediction evaluated at the recourse. With these concepts in place, we develop two recourse instability definitions.

Definition 1. *(Recourse outcome instability) The recourse outcome instability with respect to a factual instance* $\mathbf{x}$*, where at least one data weight is set to* 0*, is defined as follows:*

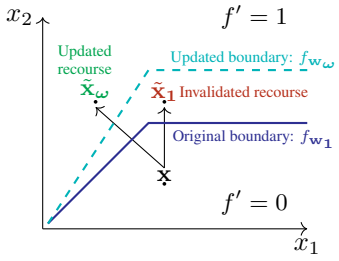
$$\Delta_{\mathbf{x}}(\boldsymbol{\omega}) = \left| f_{\mathbf{w}_1}\left(\check{\mathbf{x}}_1\right) - f_{\mathbf{w}_\omega}\left(\check{\mathbf{x}}_1\right) \right|, \tag{3}$$

where $f_{\mathbf{w}_1}(\check{\mathbf{x}}_1)$ *is the prediction at the prescribed recourse* $\check{\mathbf{x}}_1$ *based on the model that uses the full training set (i.e.,* $f_{\mathbf{w}_1}$*) and* $f_{\mathbf{w}_\omega}(\check{\mathbf{x}}_1)$ *is the prediction at the prescribed recourse for an updated model and data deletion requests have been incorporated into the predictive model (i.e.,* $f_{\mathbf{w}_\omega}$*).*

The above definition concisely describes the effect of applying "outdated" recourses to the updated model. We assume that only the model parameters are being updated while the prescribed recourses remain unchanged. For a discrete model with $\mathcal{Y} = \{0, 1\}$, Definition 1 captures whether the prescribed recourses will be invalid ($\Delta_{\mathbf{x}} = 1$) after deletion of training instances (see Fig. 1a). To obtain invalidation rates of recourses for a continuous-score model with target value s, we can also apply Definition 1 with a discretized $f'(\mathbf{x}) = \mathbb{I}\left[f(\mathbf{x}) > s\right]$, where $\mathbb{I}$ denotes the indicator function.



(a) Recourse Outcome Instability



(b) Recourse Action Instability

Figure 1: Visualizing the two key robustness notions. In Fig. 1a, recourse $\tilde{\mathbf{x}}_1$ for an input $\mathbf{x}$ is invalidated due to a model update. In Fig. 1b, recourse is additionally recomputed (i.e., $\check{\mathbf{x}}_{\boldsymbol{\omega}}$) to avoid recourse invalidation.

In Definition 2, consistent with related work (e.g., [59]), the distance function d is specified to be a p-norm and the recourse is allowed to change due to model parameter updates.

Definition 2. *(Recourse action instability) The Recourse action instability with respect to a factual input* $\mathbf{x}$*, where at least one data weight is set to* 0*, is defined as follows:*

$$\Phi_{\mathbf{x}}^{(p)}(\boldsymbol{\omega}) = \left\| \check{\mathbf{x}}_1 - \check{\mathbf{x}}_{\boldsymbol{\omega}} \right\|_p, \tag{4}$$

where $p \in [1, \infty)$*, and* $\check{\mathbf{x}}_{\boldsymbol{\omega}}$ *is the recourse obtained for the model trained on the data instances that remain present in the data set after the deletion request.*

Definition 2 quantifies the extent to which the prescribed recourses would have to additionally change to still achieve the desired recourse outcome *after data deletion* requests (i.e., $\check{\mathbf{x}}_{\boldsymbol{\omega}}$, see Fig. 1b). Note that we are interested in how the optimal low cost recourse changes even if the outdated recourse would remain valid. Using our invalidation measures defined above, in the next section, we formally study the trade-offs between actionable explanations and the right to be forgotten. To do so, we provide data dependent upper bounds on the invalidation measures from Definitions 1 and 2, which practitioners can use to probe the worst-case vulnerability of their algorithmic recourse to data deletion requests.

4 Finding the Set of Most Critical Data Points

The Objective Function. In this section, we present optimization procedures that can be readily used to assess recourses' vulnerability to deletion requests. On this way, we start by formulating our optimization objective. We denote by $m \in \{\Delta, \Phi^{(2)}\}$ the measure we want to optimize for. We consider the summed instability of over the data set by omitting the subscript $\mathbf{x}$, e.g., $\Delta = \sum_{\mathbf{x} \in \mathcal{D}_{test}} \Delta_{\mathbf{x}}$. Our goal is to find the smallest number of deletion requests that leads to a maximum impact on the instability measure m. To formalize this objective, we define the set of possible data weight configurations:

$$\Gamma_\alpha := \{\boldsymbol{\omega} : \text{Maximally } \lfloor \alpha \cdot n \rfloor \text{ entries of } \boldsymbol{\omega} \text{ are 0 and the remainder is 1.}\}. \tag{5}$$

In (5), the parameter α controls the fraction of instances that are being removed from the training set. For a fixed fraction α, our problem of interest becomes:

$$\boldsymbol{\omega}^* = \arg\max_{\boldsymbol{\omega} \in \Gamma_\alpha} m(\boldsymbol{\omega}). \tag{6}$$

Fundamental Problems. When optimizing the above objective we face two fundamental problems: (i) *evaluating* $m(\boldsymbol{\omega})$ for many weight configurations $\boldsymbol{\omega}$ can be prohibitively expensive as the objective is defined implicitly through solutions of several non-linear optimization problems (i.e., model fitting and finding recourses). Further, (ii) even for an objective $m(\boldsymbol{\omega})$ which can be computed in constant or polynomial time *optimizing* this objective can still be NP-hard (a proof is given in Appendix A.3).

Practical Algorithms. We devise two practical algorithms which approach the problem in (6) in different ways. As for the problem of computing $m(\boldsymbol{\omega})$ in (i), we can either solve this by (a) using a closed-form expression indicating the dependency of m on $\boldsymbol{\omega}$ or (b) by using an approximation of m that is differentiable with respect to $\boldsymbol{\omega}$. As for the optimization in (ii), once we have established the dependency of m on $\boldsymbol{\omega}$ we can either (a) use a gradient descent approach or (b) we use a greedy method. Below we explain the individual steps required for the construction of our algorithms.

4.1 Computing the Objective

In the objective $m(\boldsymbol{\omega})$, notice the dependencies $\Delta_{\mathbf{x}}(\boldsymbol{\omega}) = \Delta_{\mathbf{x}}\left(f(\mathbf{w}(\boldsymbol{\omega}), \check{\mathbf{x}})\right)$ for the recourse outcome instability, and $\Phi^{(2)}_{\mathbf{x}}(\boldsymbol{\omega}) = \Phi^{(2)}_{\mathbf{x}}\left(\boldsymbol{\delta}(\mathbf{w}(\boldsymbol{\omega}), \mathbf{x})\right)$ for the recourse action instability. In the following, we briefly discuss how we efficiently compute each of these functions without numerical optimization.

Model parameters from data weights $\mathbf{w}(\boldsymbol{\omega})$. For the linear model, an analytical solution can be obtained, $\mathbf{w}_{\mathrm{L}}(\boldsymbol{\omega}) = \left(\mathbf{X}^\top \boldsymbol{\Omega} \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{\Omega} \mathbf{Y}$, where $\boldsymbol{\Omega} = \operatorname{diag}(\boldsymbol{\omega})$. The same goes for the NTK model where $\mathbf{w}_{\mathrm{NTK}}(\boldsymbol{\omega}) = \boldsymbol{\Omega}^{\frac{1}{2}} \left(\boldsymbol{\Omega}^{\frac{1}{2}} \mathbf{K}^{\infty}(\mathbf{X}, \mathbf{X}) \boldsymbol{\Omega}^{\frac{1}{2}} + \beta \mathbf{I}\right)^{-1} \boldsymbol{\Omega}^{\frac{1}{2}} \mathbf{Y}$ [8, Eqn. 3]. When no closed-form expressions for the model parameters exist, we can resort to the infinitesimal jackknife (IJ) [33, 19, 26, 25], that can be seen as a linear approximation to this implicit function. We refer to Appendix C for additional details on this matter.

Model prediction from model parameters $f(\mathbf{w}, \check{\mathbf{x}})$. Having established the model parameters, evaluating the prediction at a given point can be quickly done even in a differentiable manner with respect to $\mathbf{w}$ for the models we consider in this work.

Recourse action from model parameters $\boldsymbol{\delta}(\mathbf{w}, \check{\mathbf{x}})$. Estimating the recourse action is more challenging as it requires solving (2). However, a differentiable solution exists for linear models, where the optimal recourse action is given by $\boldsymbol{\delta}_{\mathrm{L}} = \frac{s - \mathbf{w}_{\mathrm{L}}(\boldsymbol{\omega})^\top \mathbf{x}}{\lambda + \|\mathbf{w}_{\mathrm{L}}(\boldsymbol{\omega})\|_2^2} \mathbf{w}_{\mathrm{L}}(\boldsymbol{\omega})$. When the underlying predictor is a wide neural network we can approximate the recourse expression of the corresponding NTK, $\boldsymbol{\delta}_{\mathrm{NTK}} \approx \frac{s - f_{\boldsymbol{\omega},\mathrm{NTK}}(\mathbf{x})}{\lambda + \|\bar{\mathbf{w}}_{\mathrm{NTK}}(\bar{\boldsymbol{\omega}})\|_2^2} \bar{\mathbf{w}}_{\mathrm{NTK}}(\boldsymbol{\omega})$, which stems from the first-order taylor expansion $f_{\boldsymbol{\omega},\mathrm{NTK}}(\mathbf{x} + \boldsymbol{\delta}) \approx f_{\boldsymbol{\omega},\mathrm{NTK}}(\mathbf{x}) + \boldsymbol{\delta}^\top \bar{\mathbf{w}}_{\mathrm{NTK}}(\boldsymbol{\omega})$ with $\bar{\mathbf{w}}_{\mathrm{NTK}}(\boldsymbol{\omega}) = \nabla_{\mathbf{x}} K(\mathbf{x}, \mathbf{X}) \mathbf{w}_{\mathrm{NTK}}(\boldsymbol{\omega})$.

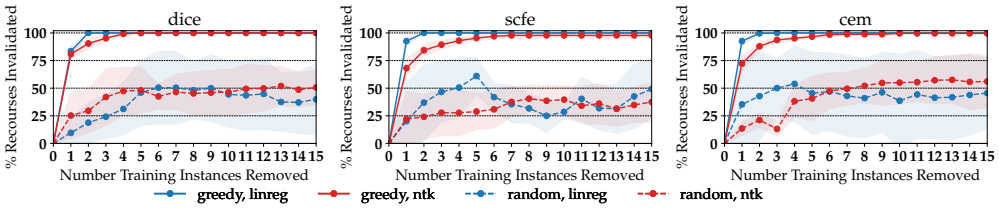
4.2 Optimizing the Objective Function

The Greedy Algorithm. We consider the model on the full data set and compute the objective function $m(\boldsymbol{\omega})$ under deletion of every instance (alone). We then select the instance that leads to the highest increase in the objective. We add this instance to the set of deleted points. Subsequently, we refit the model and compute the impact of deletion for every second instance, when deleted in combination with the first one. Again, we add the instance that results in the largest increase to the set. Iteratively repeating these steps, we identify more instances to be deleted. Computational complexity depends on the implementation of the model weight recomputation, which is required $\mathcal{O}(\alpha n^2)$ times.

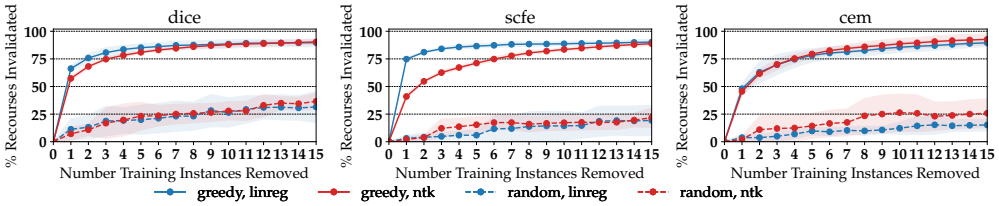
The Gradient Descent Algorithm. Because our developed computation of $m(\boldsymbol{\omega})$ can be made differentiable, we also propose a *gradient-based optimization* framework. We consider the relaxation of the problem in (6),

$$\boldsymbol{\omega}^* = \underset{\boldsymbol{\omega} \in \{0,1\}^n}{\arg\max} \; m(\boldsymbol{\omega}) - \|\mathbf{1} - \boldsymbol{\omega}\|_0, \tag{7}$$

where the ℓ_0 norm encourages to change as few data weights from 1 to 0 as possible while few removals of training instances should have maximum impact on the robustness measure. The problem in (7) can be further relaxed to a continuous and unconstrained optimization problem. To do so we use a recently suggested stochastic surrogate loss for the ℓ_0 term [63]. Using this technique, a surrogate loss for (7) can be optimized using stochastic gradient descent (SGD). We refer to Appendix C for more details and pseudo-code of the two algorithms.



(a) Admission



(b) Heloc

Figure 2: Measuring the tradeoff between recourse outcome instability and the number of deletion requests for both the Admission and the Heloc data sets for regression and NTK models and various recourse methods. Results were obtained by greedy optimization; see Appendix B for SGD results.

5 Experimental Evaluation

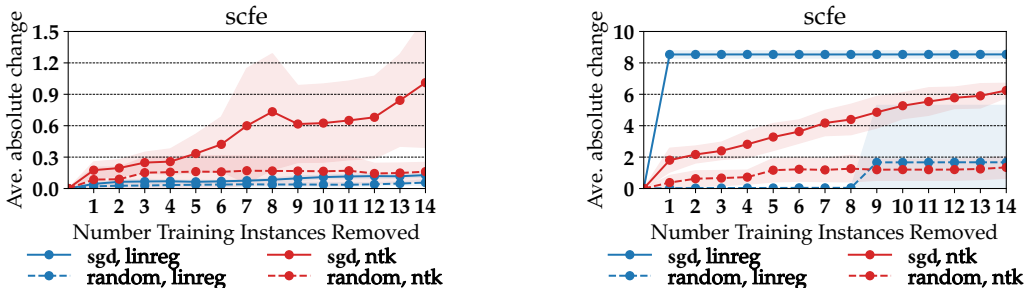
We experimentally evaluate our framework in terms of its ability to find significant recourse invalidations using the instability measures presented in Section 3.

Data Sets. For our experiments on regression tasks we use two real-world data sets. In addition, we provide results for two classification datasets in the Appendix B. First, we use law school data from the Law School Admission Council (**Admission**). The council carried out surveys across 163 law schools in the US, in which they collected information from 21,790 law students across the US [60]. The data contains information on the students' prior performances. The task is to predict the students' first-year law-school average grades. Second, we use the *Home Equity Line of Credit* **(Heloc)** data set. Here, the target variable is a score indicating whether individuals will repay the Heloc account within a fixed time window. Across both tasks we consider individuals in need of recourse if their scores lie below the median score across the data set.

Recourse Methods. We apply our techniques to four different methods which aim to generate low-cost recourses using different principles: `SCFE` was suggested by Wachter et al. [59] and uses a gradient-based objective to find recourses, `DICE` [39] uses a gradient-based objective to find recourses subject to a diversity constraint, and `CEM` [13] uses a generative model to encourage recourses to lie on the data manifold. For all methods, we used the recourse method implementations from the `CARLA` library [42] and specify the ℓ_1 cost constraint. Further details on these algorithms are provided in App. C.

Evaluation Measures. For the purpose of our evaluation, we use both the recourse outcome instability measure and the recourse action instability measure presented in Definitions 1 and 2. We evaluate the efficacy of our framework to destabilize a large fraction of recourses using a small number of deletion requests (up to 14). To find critical instances, we use the greedy and the gradient-based algorithms described in Sec. 4. After having established a set of critical points, we recompute the metrics with the refitted models and recourses to obtain a ground truth result.

For the *recourse outcome instability*, our metric Δ counts the number of invalidated recourses. We use the median as the target score s, i.e., if the recourse outcome flips back from a positive leaning prediction (above median) to a negative one (below median) it is considered invalidated. When evaluating *recourse action instability*, we identify a set of critical points, delete these points from the train set and refit the predictive model. In this case, we also have to recompute the recourses to evaluate Φ_p. We then measure the recourse instability using Definition 2 with $p = 2$. Additionally, we compare with a random baseline, which deletes points uniformly at random from the train set. We compute these measures for all individuals from the test set who require algorithmic recourse. To obtain standard errors, we split the test set into 5 folds and report averaged results over these 5 folds.



(a) Admission (b) Heloc

Figure 3: Quantifying the tradeoff between recourse action instability as measured in Definition 2 and the number of deletion requests for both the Admission and the Heloc data sets for the SCFE method when the underlying model is linear or an NTK (results by SGD optimization).

Results. In Figure 2, we measure the tradeoff between *recourse outcome instability* and the number of deletion requests. We plot the number of deletion requests against the fraction of all recourses that become invalidated when up to $k \in \{1, \ldots, 14\}$ training points are removed from the training set of the predictive model. When the underlying model is linear, we observe that the removal of as few as 5 training points induces invalidation rates of all recourses that are as high as 95 $\%$ percent – we observe a similar trend across all recourse methods. Note that a similar trend is present for the NTK model; however, a larger number of deletion requests (roughly 9) is required to achieve similar invalidation rates. Finally, also note that our approach is always much more effective at deleting instances than the random baseline. In Figure 3, we measure the tradeoff between *recourse action instability* and the number of deletion requests with respect to the SCFE recourse method when the underlying predictive model is linear or an NTK model. For this complex objective, we use the more efficient SGD optimization. Again, we observe that our optimization method significantly outperforms the random baselines at finding the most influential points to be removed.

6 Conclusion

In this work, we made the first step towards understanding the tradeoffs between actionable model explanations and the right to be forgotten. We theoretically analyzed the robustness of state-of-the-art recourse methods under data deletion requests and suggested (i) a greedy and (ii) a gradient-based algorithm to efficiently identify a small subset of individuals, whose data, when removed, would lead to invalidation of a large number of recourses for unsuspecting other users. Our experimental evaluation with multiple real-world data sets on both regression and classification tasks demonstrates that the right to be forgotten presents a significant challenge to the reliability of actionable explanations.

Furthermore, our findings raise compelling questions on the deployment of counterfactual explanations in practice. First of all, *Are the two requirements of actionable explanations and the right to be forgotten fundamentally at odds with one another?* The theoretical and empirical results in this work indicate that for many model and recourse method pairs, this might indeed be the case. This finding leads to the pressing follow-up question: *How can practitioners make sure that their recourses stay valid under deletion requests?* A first take might be to implement the principle of data minimization [6, 5, 49] in the first place, i.e., exclude the k most critical data points from model training. In addition to the increase in recourse robustness some deletion requests would then go totally unheeded as the data might not be part of the trained ML models.

Finally, our theoretical results suggest that the robustness to deletion increases when the model parameter changes under data deletion remain small. This formulation closely resembles the definition of *Differential Privacy* (DP) [17, 10, 18]. We therefore conjecture that the reliability of actionable recourse could benefit from models that have been trained under DP constraints As the field of AI rapidly evolves, data protection authorities will further refine the precise interpretations of general principles in regulations such as GDPR. The present paper contributes towards this goal theoretically, algorithmically, and empirically by providing evidence of tensions between different data protection principles.

References

[1] E. Albini, J. Long, D. Dervovic, and D. Magazzeni. Counterfactual shapley additive explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FACCT)*, 2022.

[2] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks, 2016.

[3] J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. In *International Conference on Learning Representations (ICLR)*, 2021.

[4] A. Artelt, V. Vaquet, R. Velioglu, F. Hinder, J. Brinkrolf, M. Schilling, and B. Hammer. Evaluating robustness of counterfactual explanations. *arXiv:2103.02354*, 2021.

[5] A. J. Biega and M. Finck. Reviving purpose limitation and data minimisation in data-driven systems. *Technology and Regulation*, 2021.

[6] A. J. Biega, P. Potash, H. Daumé, F. Diaz, and M. Finck. Operationalizing the legal principle of data minimization for personalization. In *ACM(43) SIGIR '20*, page 399–408, 2020.

[7] E. Black, Z. Wang, M. Fredrikson, and A. Datta. Consistent counterfactuals for deep models. *arXiv:2110.03109*, 2021.

[8] S. Busuttil and Y. Kalnishkan. Weighted kernel regression for predicting changing dependencies. In *European Conference on Machine Learning*, pages 535–542. Springer, 2007.

[9] G. C. Cawley and N. L. Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural networks*, 17(10):1467–1475, 2004.

[10] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[11] Y. Cho and L. Saul. Kernel methods for deep learning. *Advances in neural information processing systems (NeurIPS)*, 22, 2009.

[12] S. Dandl, C. Molnar, M. Binder, and B. Bischl. Multi-objective counterfactual explanations. In *International Conference on Parallel Problem Solving from Nature*, pages 448–469. Springer, 2020.

[13] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[14] R. Dominguez-Olmedo, A.-H. Karimi, and B. Schölkopf. On the adversarial robustness of causal algorithmic recourse. In *International Conference on Machine Learning (ICML)*, 2022.

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

[16] S. S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

[17] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[18] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[19] B. Efron. *The jackknife, the bootstrap and other resampling plans*. SIAM, 1982.

[20] M. Finck and A. J. Biega. Reviving purpose limitation and data minimisation in data-driven systems. *Technology and Regulation*, 2021:44–61, 2021.

[21] GDPR. Regulation (eu) 2016/679 of the european parliament and of the council. *Official Journal of the European Union*, 2016.

[22] A. Ghorbani, M. Kim, and J. Zou. A distributional framework for data valuation. In *International Conference on Machine Learning (ICML)*, pages 3535–3544. PMLR, 2020.

[23] A. Ghorbani and J. Zou. Data shapley: Equitable valuation of data for machine learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 2242–2251. PMLR, 2019.

[24] A. Ginart, M. Y. Guan, G. Valiant, and J. Zou. Making ai forget you: Data deletion in machine learning. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, 2019.

[25] R. Giordano, M. I. Jordan, and T. Broderick. A higher-order swiss army infinitesimal jackknife. *ArXiv*, abs/1907.12116, 2019.

[26] R. Giordano, W. Stephenson, R. Liu, M. Jordan, and T. Broderick. A swiss army infinitesimal jackknife. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

[27] A. Golatkar, A. Achille, and S. Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[28] A. Golatkar, A. Achille, and S. Soatto. Forgetting outside the box: Scrubbing deep networks of information accessible from input-output observations. *arXiv:2003.02960*, 2020.

[29] A. Goldsteen, G. Ezov, R. Shmelkin, M. Moffie, and A. Farkash. Data minimization for gdpr compliance in machine learning models. *AI and Ethics*, pages 1–15, 2021.

[30] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[31] Z. Izzo, M. Anne Smart, K. Chaudhuri, and J. Zou. Approximate data deletion from machine learning models. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 130. PMLR, 2021.

[32] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems (NeurIPS)*, 31, 2018.

[33] L. Jaeckel. The infinitesimal jackknife. memorandum. Technical report, MM 72-1215-11, Bell Lab. Murray Hill, NJ, 1972.

[34] A.-H. Karimi, G. Barthe, B. Balle, and I. Valera. Model-agnostic counterfactual explanations for consequential decisions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[35] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[36] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, and M. Detyniecki. Inverse classification for comparison-based interpretability in machine learning. *arXiv preprint arXiv:1712.08443*, 2017.

[37] D. Mahajan, C. Tan, and A. Sharma. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv preprint arXiv:1912.03277*, 2019.

[38] R. G. Miller Jr. An unbalanced jackknife. *The Annals of Statistics*, pages 880–891, 1974.

[39] R. K. Mothilal, A. Sharma, and C. Tan. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT*)*, 2020.

[40] C. OAG. Ccpa regulations: Final regulation text. *Office of the Attorney General, California Department of Justice*, 2021.

[41] M. Pawelczyk, C. Agarwal, S. Joshi, S. Upadhyay, and H. Lakkaraju. Exploring counterfactual explanations through the lens ofadversarial examples: A theoretical and empirical analysis. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[42] M. Pawelczyk, S. Bielawski, J. Van den Heuvel, T. Richter, and G. Kasneci. Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms. In *Advances in Neural Information Processing Systems (NeurIPS) (Benchmark and Datasets Track)*, 2021.

[43] M. Pawelczyk, K. Broelemann, and G. Kasneci. Learning model-agnostic counterfactual explanations for tabular data. In *Proceedings of The Web Conference 2020 (WWW)*. ACM, 2020.

[44] M. Pawelczyk, K. Broelemann, and G. Kasneci. On counterfactual explanations under predictive multiplicity. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 809–818. PMLR, 2020.

[45] M. Pawelczyk, T. Datta, J. van-den Heuvel, G. Kasneci, and H. Lakkaraju. Algorithmic recourse in the face of noisy human responses. *arXiv preprint arXiv:2203.06768*, 2022.

[46] K. Rawal, E. Kamar, and H. Lakkaraju. Algorithmic recourse in the wild: Understanding the impact of data and model shifts. *arXiv:2012.11788*, 2021.

[47] K. Rawal and H. Lakkaraju. Interpretable and interactive summaries ofactionable recourses. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[48] Y. Rong, T. Leemann, V. Borisov, G. Kasneci, and E. Kasneci. A consistent and efficient evaluation strategy for attribution methods. In *International Conference on Machine Learning (ICML)*, 2022.

[49] D. Shanmugam, F. Diaz, S. Shabanian, M. Finck, and A. J. Biega. Learning to limit data collection via scaling laws: A computational interpretation for the legal principle of data minimization. In *ACM FAccT '22*, 2022.

[50] D. Slack, S. Hilgard, H. Lakkaraju, and S. Singh. Counterfactual explanations can be manipulated. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

[51] T. Spooner, D. Dervovic, J. Long, J. Shepard, J. Chen, and D. Magazzeni. Counterfactual explanations for arbitrary regression models. *arXiv preprint arXiv:2106.15212*, 2021.

[52] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014, 2014.

[53] G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. ACM, 2017.

[54] S. Upadhyay, S. Joshi, and H. Lakkaraju. Towards robust and reliable algorithmic recourse. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

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

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

[57] E. F. Villaronga, P. Kieseberg, and T. Li. Humans forget, machines remember: Artificial intelligence and the right to be forgotten. *Computer Law & Security Review*, 34(2):304–313, 2018.

[58] P. Voigt and A. Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

[59] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2), 2018.

[60] L. F. Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

[61] Y. Wu, E. Dobriban, and S. Davidson. Deltagrad: Rapid retraining of machine learning models. In *International Conference on Machine Learning (ICML)*, pages 10355–10366. PMLR, 2020.

[62] B. Xie, Y. Liang, and L. Song. Diverse neural network learns true target functions. In *Artificial Intelligence and Statistics (AISTATS)*, pages 1216–1224. PMLR, 2017.

[63] Y. Yamada, O. Lindenbaum, S. Negahban, and Y. Kluger. Feature selection using stochastic gates. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, 13–18 Jul 2020.

[64] R. Zhang and S. Zhang. Rethinking influence functions of neural networks in the over-parameterized regime. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

Appendix

A Theoretical Results

A.1 Upper Bounds on Recourse Outcome Instability

Proposition 1 (Upper Bound on Output Robustness for Linear Models)**.** *For the linear regression model $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ with weights given by $\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y}$, an upper bound for the output robustness by removing an instance $(\mathbf{x}, y)$ from the training set is given by:*

$$\Delta_{\mathbf{x}} \leq \max_{i \in [n]} \|\mathbf{d}_i\|_2 \cdot \|\check{\mathbf{x}}_{\mathbf{1}}\|_2, \tag{8}$$

where $\mathbf{d}_i = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{x}_i \cdot \frac{r_i}{1-h_{ii}}$, $r_i = y_i - \mathbf{w}^\top \mathbf{x}_i$ and $h_{ii} = \mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{x}_i$.

Proof. By Definition 1, we have:

$$\begin{aligned}
\Delta_{\mathbf{x}} &= \left|\mathbf{w}_{\mathbf{1}}^\top \check{\mathbf{x}}_{\mathbf{1}} - \mathbf{w}_{-i}^\top \check{\mathbf{x}}_{\mathbf{1}}\right| && (9)\\
&= \left|\left(\mathbf{w}_{\mathbf{1}} - \mathbf{w}_{-i}\right)^\top \check{\mathbf{x}}_{\mathbf{1}}\right| \\
&= \left|\left((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i \frac{(y_i - \mathbf{w}^T\mathbf{x}_i)}{1-h_{ii}}\right)^\top \check{\mathbf{x}}_{\mathbf{1}}\right| && \text{(by Theorem 1)} \quad (10)\\
&\leq \|\mathbf{d}_i\|_2 \cdot \|\check{\mathbf{x}}_{\mathbf{1}}\|_2 && \text{(by Cauchy-Schwartz)} \quad (11)\\
&\leq \|\check{\mathbf{x}}_{\mathbf{1}}\|_2 \cdot \max_{i \in [n]} \|\mathbf{d}_i\|_2 \cdot, && (12)
\end{aligned}$$

where $\mathbf{d}_i = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i \frac{(y_i - \mathbf{w}^T\mathbf{x}_i)}{1-h_{ii}}$. This completes our proof. □

Proposition 2 (Upper Bound on Output Robustness for NTK)**.** *For the NTK model with $\mathbf{w}_{NTK} = \left(\mathbf{K}^\infty(\mathbf{X}, \mathbf{X}) + \lambda \mathbf{I}_n\right)^{-1}\mathbf{Y}$, an upper bound for the output robustness by removing an instance $(\mathbf{x}, y)$ from the training set is given by:*

$$\Delta_{\mathbf{x}} \leq \|\mathbf{K}^\infty(\check{\mathbf{x}}_{\mathbf{1}}, \mathbf{X})\|_2 \cdot \max_{i \in [n]} \|\mathbf{d}_i\|_2, \tag{13}$$

where $\mathbf{d}_i = \frac{1}{k_{ii}}\mathbf{k}_i\mathbf{k}_i^\top \mathbf{Y}$, where $\mathbf{k}_i$ is the i-th column of the matrix $\left(\mathbf{K}^\infty(\mathbf{X}, \mathbf{X}) + \beta \mathbf{I}_n\right)^{-1}$, and k_{ii} is its i-th diagonal element.

Proof. By Definition 1, and the assumption of the over-parameterized regime, we have:

$$\begin{aligned}
\Delta_{\mathbf{x}} &= \left|f_{\text{NTK}}(\check{\mathbf{x}}_{\mathbf{1}}) - f_{\text{NTK}}^{-i}(\check{\mathbf{x}}_{\mathbf{1}})\right| \\
&= \left|\left(\mathbf{K}^\infty(\check{\mathbf{x}}_{\mathbf{1}}, \mathbf{X})\right)^\top \mathbf{w}_{\text{NTK}} - \left(\mathbf{K}^\infty(\check{\mathbf{x}}_{\mathbf{1}}, \mathbf{X})\right)^\top \left(\left(\mathbf{K}^\infty(\mathbf{X}, \mathbf{X}) + \beta \mathbf{I}_n\right)^{-1} - \frac{1}{k_{ii}}\mathbf{k}_i\mathbf{k}_i^\top\right)\mathbf{Y}\right| && (14)\\
&= \left|\left(\mathbf{K}^\infty(\check{\mathbf{x}}_{\mathbf{1}}, \mathbf{X})\right)^\top \frac{1}{k_{ii}}\mathbf{k}_i\mathbf{k}_i^\top \mathbf{Y}\right| && (15)\\
&\leq \|\mathbf{d}_i\|_2 \cdot \|\mathbf{K}^\infty(\check{\mathbf{x}}_{\mathbf{1}}, \mathbf{X})\|_2 \quad \text{(by Cauchy-Schwartz)} \\
&\leq \|\check{\mathbf{x}}_{\mathbf{1}}\|_2 \cdot \max_{i \in [n]} \|\mathbf{d}_i\|_2, && (16)
\end{aligned}$$

where $\mathbf{d}_i = \frac{1}{k_{ii}}\mathbf{k}_i\mathbf{k}_i^\top \mathbf{Y}$ which completes our proof. □

A.2 Upper Bounds on Recourse Action Instability

Proposition 3 (Upper Bound on Input Robustness)**.** *For the linear regression model $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ with weights given by $\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{Y}$, an upper bound for the input robustness in the setting $s = 0, \lambda = 0$ by removing the i-th instance $(\mathbf{x}_i, y_i)$ from the training set is given by:*

$$\Phi_{\mathbf{x}}^{(2)} \leq \|\mathbf{d}_i\|_2 \frac{4\sqrt{2}\|\mathbf{x}\|_2}{\min(\|\mathbf{w}\|_2, \|\mathbf{w}_{-i}\|_2)}, \tag{17}$$

under the condition that $\mathbf{w}^\top \mathbf{w}_{-i} \leq 0$ (no diametrical weight changes), where $\mathbf{w}_{-i} = \mathbf{w} - \mathbf{d}_i$ is the weight after removal of training instance i and $\mathbf{d}_i = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{x}_i \frac{(y_i - \mathbf{w}^\top \mathbf{x}_i)}{1-h_{ii}}$.

Proof. For a linear scoring function $f(\mathbf{x}) = \mathbf{w}'^{\top}\mathbf{x}$ with given parameters $\mathbf{w}'$, under the squared ℓ_2 norm constraint with balance parameter λ, the optimal recourse action is given by [41]:

$$\boldsymbol{\delta}(\mathbf{w}') = \frac{s - \mathbf{w}'^{\top}\mathbf{x}}{\|\mathbf{w}'\|_2^2 + \lambda} \cdot \mathbf{w}'. \tag{18}$$

Using Definition 2, we can express the total change in $\boldsymbol{\delta}$ as a path integral over changes in $\mathbf{w}$, times the change $\frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}}$ they entail:

$$\Phi_{\mathbf{x}}^{(2)} = \left\|\boldsymbol{\delta}_{\mathbf{1}} - \boldsymbol{\delta}_{\boldsymbol{\omega}}\right\|_2 = \left\|\boldsymbol{\delta}(\mathbf{w}) - \boldsymbol{\delta}(\mathbf{w}_{-i})\right\|_2 \tag{19}$$

$$\leq \int_0^1 \left\| \frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}}(\gamma\mathbf{w} + (1-\gamma)\mathbf{w}_{-i}) \right\| \|\mathbf{w} - \mathbf{w}_{-i}\|_2 d\gamma, \tag{20}$$

where $\frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}}$ denotes the Jacobian, with the corresponding operator matrix norm. Defining $\tilde{\mathbf{w}} \coloneqq \gamma\mathbf{w} + (1-\gamma)\,\mathbf{w}_{-i}$ and using $\|\mathbf{w} - \mathbf{w}_{-i}\|_2 = \|\mathbf{d}_i\|_2$, we obtain

$$\Phi_{\mathbf{x}}^{(2)} \leq \|\mathbf{d}_i\|_2 \int_0^1 \left\| \frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}}(\tilde{\mathbf{w}}) \right\|_2 d\gamma. \tag{21}$$

Because of the form $\boldsymbol{\delta}(\mathbf{w}') = f(\mathbf{w}')\mathbf{w}'$, where $f(\mathbf{w}') \coloneqq \frac{s - \mathbf{w}'^{\top}\mathbf{x}}{\|\mathbf{w}'\|_2^2 + \lambda}$ is a scalar function, its Jacobian has the form $\frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}'} = \mathbf{w}'\left(\nabla f(\mathbf{w}')\right)^{\top} + f(\mathbf{w}')\mathbf{I}$. We will now derive a bound on the Jacobian's operator norm:

$$\left\| \frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}'}(\tilde{\mathbf{w}}) \right\|_2 = \max_{\|\mathbf{a}\|=1} \left\| \frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}'}\mathbf{a} \right\|_2 = \max_{\|\mathbf{a}\|=1} \left\| \mathbf{w}'\left(\nabla f(\tilde{\mathbf{w}})\right)^{\top}\mathbf{a} + f(\tilde{\mathbf{w}})\mathbf{a} \right\|_2 \tag{22}$$

$$\leq \|\nabla f(\tilde{\mathbf{w}})\|_2 \|\tilde{\mathbf{w}}\|_2 + |f(\tilde{\mathbf{w}})|. \tag{23}$$

Additionally, we know that for $s = 0$, $|f(\mathbf{w}')| \leq \frac{\|\mathbf{x}\|_2\|\tilde{\mathbf{w}}\|_2}{\|\tilde{\mathbf{w}}\|_2^2} = \frac{\|\mathbf{x}\|_2}{\|\tilde{\mathbf{w}}\|_2}$. The gradient is given by

$$\|\nabla f(\tilde{\mathbf{w}})\|_2 = \left\| \frac{-(\|\tilde{\mathbf{w}}\|_2^2 + \lambda)\mathbf{x} - 2(s - \tilde{\mathbf{w}}^{\top}\mathbf{x})\tilde{\mathbf{w}}}{(\|\tilde{\mathbf{w}}\|_2^2 + \lambda)^2} \right\|_2 \tag{24}$$

$$\leq \frac{(\|\tilde{\mathbf{w}}\|_2^2 + \lambda)\|\mathbf{x}\|_2 + 2(s + \|\tilde{\mathbf{w}}\|_2\|\mathbf{x}\|_2)\|_2\tilde{\mathbf{w}}\|_2}{\|\tilde{\mathbf{w}}\|_2^4} \tag{25}$$

$$= \frac{3\|\mathbf{x}\|_2}{\|\tilde{\mathbf{w}}\|_2^2} \qquad \text{(Using } \lambda \to 0,\ s = 0\text{)}. \tag{26}$$

In summary,

$$\left\| \frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}'}(\tilde{\mathbf{w}}) \right\|_2 \leq \frac{3\|\mathbf{x}\|_2}{\|\tilde{\mathbf{w}}\|_2^2}\|\tilde{\mathbf{w}}\|_2 + \frac{\|\mathbf{x}\|_2}{\|\tilde{\mathbf{w}}\|_2} = \frac{4\|\mathbf{x}\|_2}{\|\tilde{\mathbf{w}}\|_2}. \tag{27}$$

Because $\tilde{\mathbf{w}}$ is a line between $\mathbf{w}$ and $\mathbf{w}_{-i}$, its norm is bounded from below by $\|\tilde{\mathbf{w}}\|_2 \geq \frac{1}{\sqrt{2}}\min(\|\mathbf{w}\|_2, \|\mathbf{w}_{-i}\|_2) \geq \frac{1}{\sqrt{2}}\left(\|\mathbf{w}\|_2 - \|\mathbf{w} - \mathbf{w}_{-i}\|_2\right) = \frac{1}{\sqrt{2}}\left(\|\mathbf{w}\|_2 - \|\mathbf{d}_i\|_2\right)$. We can thus uniformly bound the integral and plug in the bound because of its positivity,

$$\Phi_{\mathbf{x}}^{(2)} \leq \|\mathbf{d}_i\|_2 \int_0^1 \left\| \frac{\mathbf{D}\boldsymbol{\delta}}{\mathbf{D}\mathbf{w}}(\tilde{\mathbf{w}}) \right\|_2 d\gamma \tag{28}$$

$$\leq \|\mathbf{d}_i\|_2 \int_0^1 \frac{4\sqrt{2}\|\mathbf{x}\|_2}{\min(\|\mathbf{w}\|_2, \|\mathbf{w}_{-i}\|_2)} d\gamma \tag{29}$$

$$= \|\mathbf{d}_i\|_2 \frac{4\sqrt{2}\|\mathbf{x}\|_2}{\min(\|\mathbf{w}\|_2, \|\mathbf{w}_{-i}\|_2)}, \tag{30}$$

which completes the proof. □

A.3 Calculating Recourse Outcome Instability for k Deletions is NP-hard

We can show that, for a general scoring function f, the problem defined in (6) is NP-hard. We make this proof by providing a function f for which solving the recourse outcome invalidity problem is as hard as solving the well-known Knapsack problem, that has been shown to be NP-hard [35]. The knapsack problem is defined as follows:

$$\max_{q_i \in \{0,1\}} \sum_{i=1}^{n} v_i q_i \text{ s.t. } \sum_{i=1}^{n} y_i q_i \leq W, \tag{31}$$

where the problem considers n fixed items $(v_i, y_i)_{i=1\ldots n}$ with a value v_i and knapsack weight $y_i > 0$, and W is a fixed weight budget. The optimization problem consists of choosing the items that maximize the summed values but have a weight lower than W. To solve this problem through the recourse outcome invalidation problem, we suppose there is a data point for each item. We can choose any $k > \frac{W}{\min y_i}$ of points to be deleted, where this condition ensures that we can remove the number of samples maximally required to solve the corresponding knapsack problem. Note that we can always add a number of dummy points that have no effect such that the total number of data points is at least k. Suppose there is a classifier function:

$$f_{\boldsymbol{\omega}}(\mathbf{x}) \coloneqq \begin{cases} \sum_{i=1}^{n} v_i(1-\omega_i), & \sum_{i=1}^{n} y_i(1-\omega_i) \leq W \\ 0, & \text{else} \end{cases}. \tag{32}$$

In this case, solving Eqn. 6 comes down to finding the set of items (i.e., removing the data points) that have maximum value, but stay under the threshold W. Thus, if we can solve Eqn. 6, the solution to the equivalent knapsack problem is given by $\mathbf{q} = (\mathbf{1} - \boldsymbol{\omega})$.

A.4 Auxiliary Theoretical Results

We state the following classic result by [38] without proof.

Theorem 1. *(Leave-One-Out Estimator, [38]) Define $(\mathbf{x}_i, y_i)$ as the point to be removed from the training set. Given the optimal weight vector $\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}$ which solves for a linear model under mean-squared-error loss, the leave-one-out estimator is given by:*

$$\mathbf{w} - \mathbf{w}_{-i} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_i \frac{(y_i - \mathbf{w}^T \mathbf{x}_i)}{1 - \mathbf{x}_i^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_i} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_i \frac{(y_i - \mathbf{w}^T \mathbf{x}_i)}{1 - h_{ii}} =: \mathbf{d}_i.$$

A.5 An Analytical NTK Kernel

In this section, we provide theoretical results that allow deriving the closed form solution of the NTK for the two-layer ReLU network. First, see the paper by Jacot et al. [32] for the original derivation of the neural tangent kernel.

A closed-form solution for two-layer ReLU networks. From [64, 16, Assumption 3.1] we obtain the definition of the Kernel matrix $\mathbf{K}^\infty$ (termed Gram matrix in the paper [16]) for ReLU networks:

$$\begin{aligned} \mathbf{K}_{ij}^\infty = \mathbf{K}^\infty(\mathbf{x}_i, \mathbf{x}_j) &= \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})} \left[\mathbf{x}_i^\top \mathbf{x}_j \mathbb{I} \left\{ \mathbf{w}^\top \mathbf{x}_i \geq 0, \mathbf{w}^\top \mathbf{x}_j \geq 0 \right\} \right] \\ &= \mathbf{x}_i^\top \mathbf{x}_j \mathbb{E}_{\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})} \left[\mathbb{I} \left\{ \mathbf{w}^\top \mathbf{x}_i \geq 0, \mathbf{w}^\top \mathbf{x}_j \geq 0 \right\} \right] \\ &= \mathbf{x}_i^\top \mathbf{x}_j \frac{\pi - \arccos\left(\frac{\mathbf{x}_i^\top \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right)}{2\pi}. \end{aligned}$$

The last reformulation uses an analytical result by [11]. The derived result matches the one by [62], which however does not provide a comprehensive derivation.

B Additional Experimental Results

Data sets for the Classification Tasks When considering classification tasks on the *heloc* and *admission* data sets, we threshold the scores based on the median to obtain binary target labels. On the Admission data set (in the classification setting), a counterfactual is found when the predicted first-year average score switches from 'below median' to 'above median'. We then count an invalidation if,

after the model update, the score of a counterfactual switches back to 'below median'. In addition to the aforementioned data sets, we use both the *Diabetes* and the *Compas* data sets. The *Diabetes* data set which contains information on diabetic patients from 130 different US hospitals [52]. The patients are described using administrative (e.g., length of stay) and medical records (e.g., test results), and the prediction task is concerned with identifying whether a patient will be readmitted within the next 30 days. We sub sampled a smaller data sets of 10000 points from this dataset. 8000 points are left to train the model, while 2000 points are left for the test set. The *Compas* data set [2] contains data for more than 10,000 criminal defendants in Florida. It is used by the jurisdiction to score defendant's likelihood of reoffending. We kept a small part of the raw data as features like *name*, *id*, *casenumbers* or *date-time* were dropped. The classification task consists of classifying an instance into high risk of recidivism. Across all data sets, we dropped duplicate instances.

Discussing the Results As suggested in Section 5, here we are discussing the remaining recourse outcome invalidation results. We show these results for two settings. In Figure 4, we demonstrate the efficacy of our greedy deletion algorithm across 4 data sets on the classification tasks using different classification models (ANN, logistic regression, Kernel-SVM). For the logistic regression and the ANN model, we use the infinitesimal jackknife approximation to calculate the probitively expensive retraining step as described in Section 4. We observe that our method well outperforms random guessing. The results also highlight that while the NTK theory allows to study the deletion effects from a theoretical point of view, if one is interested in empirical worst-case approximations, the infinitesimal jackknife can be a method of choice. As we observe this pattern across all recourse methods, we hypothesize that this is related to the instability of the trained ANN models, and we leave an investigation of this interesting phenomenon for future work.

Additionally, in Figure 5, we compare our SGD-based deletion algorithm to the greedy algorithm. For the SGD-based deletion results, we observe inverse-u-shaped curves on some method-data-model combinations. The reason for this phenomenon can be explained as follows: when the ℓ_0 regularization strength (i.e., η) is not strong enough, then the importance weights for the k-th removal with $k > 5$ become more variable (i.e., SGD does not always select the most important data weight for larger k). This drop in performance can be mitigated by increasing the strength of the ℓ_0 regularizer within our SGD-based deletion algorithm.

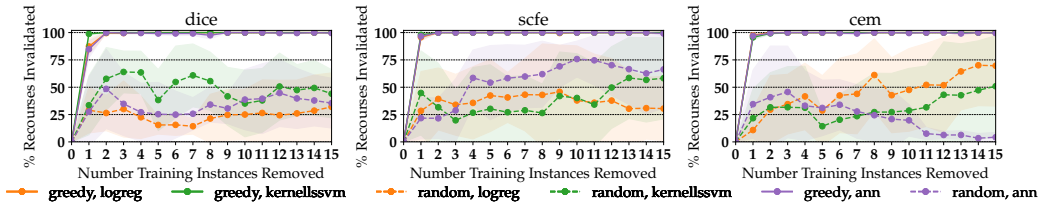
In Figure 6, we study a simple removal strategy aimed at increasing the stability of algorithmic recourse. To this end, we identified the 15 points that lead to the highest invalidation on the NTK and linear regression models when the underlying recourse method is `SCFE`. Using our greedy method, we remove these 15 points from the training data set, and we then then rerun our proposed greedy removal algorithm. This strategy leads to an improvement of up to 6 percentage points over the initial model where the 15 most critical points were included, suggesting that the removal of these critical points can be used to alleviate the recourse instability issue. In future work, we plan to investigate strategies that increase the robustness of algorithmic recourse even further.

Finally, in Figure 7 we study how well the critical points identified for the NTK model would invalidate a wide 2-layer ReLU network with 10000 hidden nodes. To study that question, we identified the points that lead to the highest invalidation on the NTK using our greedy method, and we then use these identified training points to invalidate the recourses suggested by the wide ANN. As before, we are running these experiments on the full data set across 5 folds. Figure 7 demonstrates the results of this strategy for the `SCFE` recourse method. We see that this strategy increases the robustness of up to 30 percentage points over the random baseline, suggesting that critical points under NTK can be used to estimate recourse invalidation for wide ANN models.

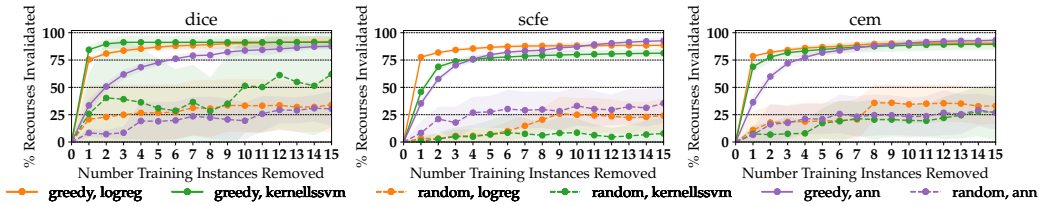
C Implementation Details

C.1 Details on Model Training

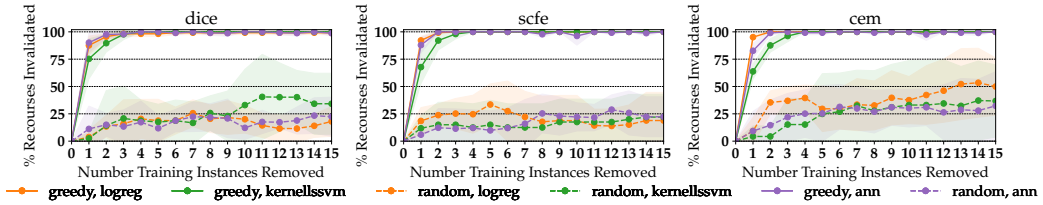
We train the classification models using the hyperparameters given in Table 1. The ANN and the Logistic regression models are fit using the quasi-newton `lbgfs` solver. We add L2-regularization to the ANN weights. The other methods are trained via their analytical solutions. Below, in Algorithms 1 and 2, we show pseudocodes for both our greedy and sgd-based deletion methods to invalidate the recourse outcome. In order to do the optimization with respect to the recourse action stability measure, we slightly adjust Algorithm 2 to optimize the right metric from Definition 2.



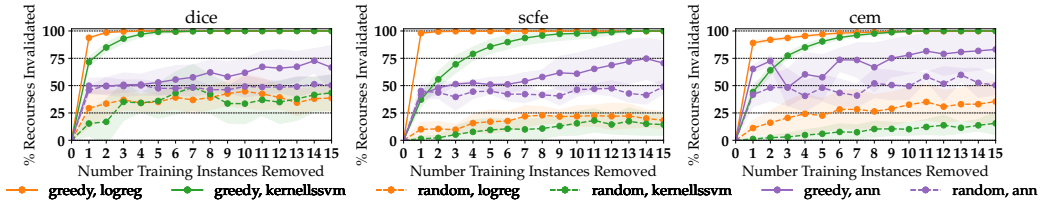
(a) Admission



(b) Heloc



(c) Compas



(d) Diabetes

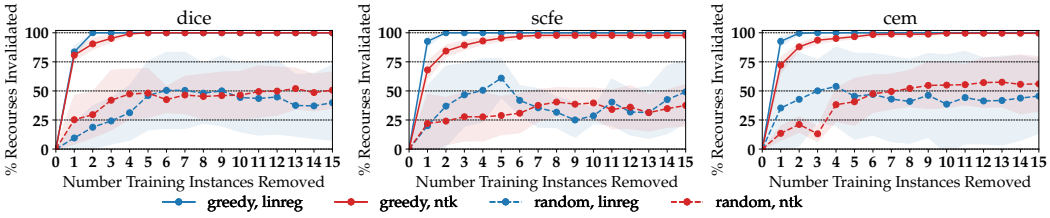
Figure 4: Measuring the tradeoff between recourse outcome instability and the number of deletion requests for the Admission, Heloc, Diabetes and Compas data sets for logistic regression, kernel svm, and ANN models across recourse methods on classification tasks. Results were obtained by greedy optimization. The dotted lines indicate the random baselines.

Model	Parameters
Linear Regression	OLS, no hyperparameters.
NTK Regression	$\beta = 2$ (Admission), $\beta = 5$ (other data sets)
Logistic Regression	L2-Regularization with $C = 1.0$
Kernel-LSSVM	Gaussian Kernel with $\gamma = 1.0$ (see [9])
ANN	2-Layer, 30 Hidden units, Sigmoid, $\alpha = 10$ (L2-Regularization)

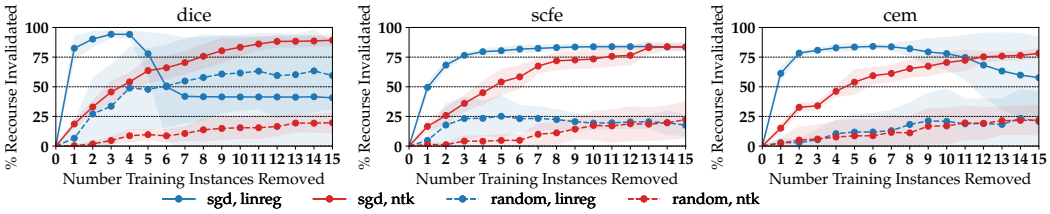
Table 1: Model hyperparameters used in this work

C.2 Details on Generating the Counterfactuals

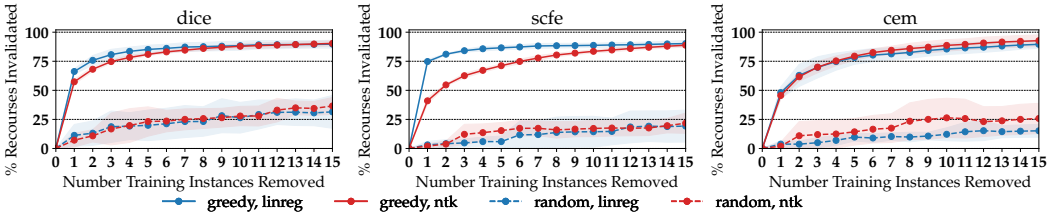
For `DICE`, for every test input, we generate two different counterfactual explanations. Then we randomly pick either the first or second counterfactual to be the counterfactual assigned to the given input. Across all recourse methods the success rates lie above 95%, i.e., for 95% of recourse seeking individuals the algorithms can identify recourses. The only exception is admission data set for the NTK model, where the success rate lies at 60%. Across all recourse methods we set $\lambda \to 0$. Note that the default implementations use early stopping once a feasible recourse has been identified.



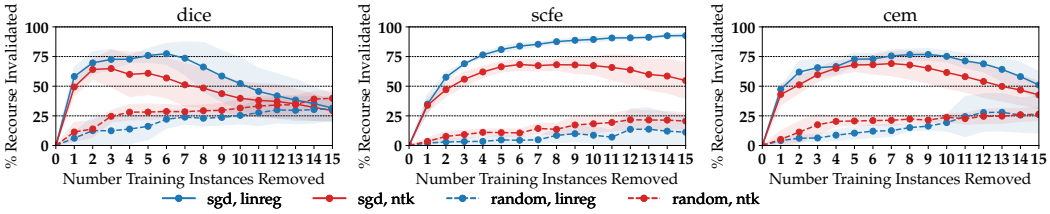
(a) Admission (Greedy)



(b) Admission (SGD)



(c) Heloc (Greedy)



(d) Heloc (SGD)

Figure 5: Measuring the tradeoff between recourse outcome instability and the number of deletion requests for the Admission and Heloc data sets for linear regression and NTK models across recourse methods on regression tasks. Results were obtained by both SGD and Greedy optimization. The dotted lines indicate the random baselines.

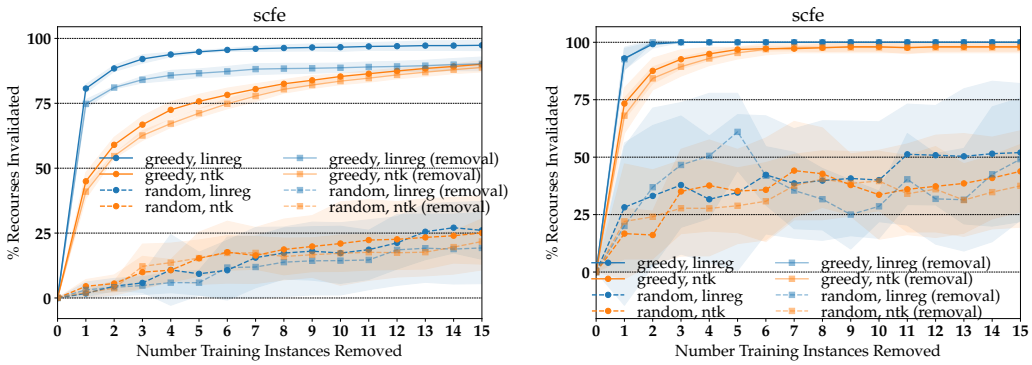
C.3 Details on the ℓ_0 Regularizer

Since an ℓ_0 regularizer is computationally intractable for high-dimensional optimization problems, we have to resort to approximations. One such approximation approach was recently suggested by **(author?)** [63]. The underlying idea consists of converting the combinatorial search problem to a continuous search problem over distribution parameters. To this end, recall our optimization problem from the main text:

$$\boldsymbol{\omega}^* = \underset{\boldsymbol{\omega} \in \{0,1\}^n}{\arg\max} \; m(\boldsymbol{\omega}) - \eta \cdot \|\mathbf{1} - \boldsymbol{\omega}\|_0. \tag{33}$$

We will now introduce Bernoulli random variables $Z_i \in \{0, 1\}$ with corresponding parameters π_i to model the individual ω_i. Instead of optimizing the objective above with respect to $\boldsymbol{\omega}$ we will optimize with respect to distribution parameters $\boldsymbol{\pi}$:

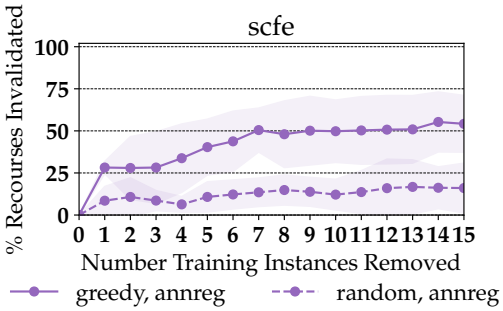
$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi}}{\arg\max} \; m(\mathbf{Z}(\boldsymbol{\pi})) - \eta \cdot \|\mathbf{1} - \mathbf{Z}(\boldsymbol{\pi})\|_0. \tag{34}$$



(a) Heloc (Greedy)

(b) Admission (Greedy)

Figure 6: Measuring the efficacy of a simple removal strategy on the Heloc and Admission data set for linear and NTK regression models. We removed the 15 critical points identified for the linear and NTK models when the underlying recourse method is SCFE and reran the removal algorithm on the remaining training set. Results were obtained by Greedy optimization. The dotted lines indicate the random baselines.



(a) Admission (Greedy)

Figure 7: Measuring the tradeoff between recourse outcome instability and the number of deletion requests for the Admission data set for a neural network regression model. We used the critical points identified for the NTK model to invalidate the recourses identified by a wide 2-layer ReLU network with 10000 hidden nodes. Results were obtained by Greedy optimization. The dotted lines indicate the random baselines.

Since the above optimization problem is known to suffer from high-variance solutions, [63] suggest to use a Gaussian-based continuous relaxation of the Bernoulli variables:

$$\tilde{Z}_i = \max(0, \min(1, \mu_i + \epsilon_i)), \tag{35}$$

where $\epsilon_i = \mathcal{N}(0, \sigma^2)$, resulting in the following optimization problem:

$$\boldsymbol{\mu}^* = \arg\max_{\boldsymbol{\mu}} \; m(\tilde{\mathbf{Z}}(\boldsymbol{\mu})) - \eta \cdot \|\mathbf{1} - \tilde{\mathbf{Z}}(\boldsymbol{\mu})\|_0. \tag{36}$$

At inference time, the optimal weights are then given by $\tilde{Z}_i^* = \max(0, \min(1, \mu_i^*)) \; \forall i \in [n]$. To obtain discrete weights, we take the argmax over each individual $\tilde{Z}_i$.

C.4 Details on the Jackknife Approximation

When the model parameters $\mathbf{w}$ are a function of the data weights by solving (1) we can approximate $\mathbf{w}(\boldsymbol{\omega})$ using the infinitesimal Jackknife (IJ) without having to optimize (1) repeatedly [33, 19, 26, 25]:

$$\mathbf{w}_{\text{IJ}}(\boldsymbol{\omega}) = \mathbf{w}_\mathbf{1} - \mathbf{H}_\mathbf{1}^{-1} \mathbf{G}_{\boldsymbol{\omega}-\mathbf{1}}, \tag{37}$$

Algorithm 1 Greedy recourse outcome invalidation

Required: Model: $f_{\mathbf{w}(\mathbf{1})}$; Matrix of Recourses: $\check{\mathbf{X}}_{\mathbf{1}} \in \mathbb{R}^{q \times d}$; d: input dimension; q number of recourse points on test set; n: # train points; M: max # deleted train points; s: invalidation target

$\boldsymbol{\omega}^{(0)} = \mathbf{1}_n$ ▷ All training instances present

for $m = 1 : M$ **do**

  $\boldsymbol{\omega}^{(m)} \leftarrow \boldsymbol{\omega}^{(m-1)}$

  $\tilde{\mathbf{Y}} = \mathbf{0}_{n \times q}$ ▷ Recourse outcomes

  $\boldsymbol{J} = \mathbf{0}_{n \times q}$ ▷ Invalidations present

  $S^{(m)} \leftarrow \left\{ i \;\middle|\; \boldsymbol{\omega}_i^{(m)} \neq 0 \right\}$ ▷ Set of train instances present at iteration m

  for $i \in S^{(m)}$ **do**

    $\mathbf{w}_{\text{new}}^{(i)} = \texttt{update\_w}(\boldsymbol{\omega}_{-i}^{(m)})$ ▷ $\boldsymbol{\omega}_{-i}^{(m)}$ has additionally set weight i to 0.

    ▷ Use analytical or IJ solution for $\mathbf{w}(\boldsymbol{\omega})$

    $\tilde{\mathbf{Y}}[i, :] = f_{\mathbf{w}_{\text{new}}^{(i)}}(\check{\mathbf{X}}_{\mathbf{1}})$ ▷ New recourse outcomes

    $\boldsymbol{J}[i, :] = \mathbb{I}(\tilde{\mathbf{Y}}[i, :] < s)$ ▷ Invalidation present

  end for

  index $\leftarrow \arg\max_{i \in S^{(m)}} \|\mathbf{J}[i, :]\|_1$ ▷ Find point that leads to highest invalidation

  $\boldsymbol{\omega}^{(m)}[\text{index}] = 0$ ▷ Remove training point

end for

return: $\boldsymbol{\omega}^{(M)}$ ▷ data weights indicating M removals

where $\mathbf{G}$ and $\mathbf{H_1}$ are the Jacobian and the Hessian matrices of the loss function with respect to the data weights evaluated at the optimal model parameters $\mathbf{w}$, i.e., $\mathbf{G}_{\boldsymbol{\omega}-\mathbf{1}} = \frac{1}{n} \sum_{i=1}^{n} (\omega_i - 1) \cdot \frac{\partial \ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)}{\partial \mathbf{w}}$ and $\mathbf{H_1} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial^2 \ell(f_{\mathbf{w}}(\mathbf{x}_i), y_i)}{\partial \mathbf{w} \partial \mathbf{w}^\top}$. Note that this technique computes the Hessian matrix $\mathbf{H_1}$ only once. Using this Jackknife approximation, the Jacobian term $\mathbf{G}_{\boldsymbol{\omega}-1}$ becomes an explicit function of the data weights which makes the Jackknife approximation amenable to optimization.

Algorithm 2 SGD recourse outcome invalidation

Required: Model: $f_{\mathbf{w}(\mathbf{1})}$; Matrix of Recourses: $\check{\mathbf{X}}_{\mathbf{1}} \in \mathbb{R}^{q \times d}$; d: input dimension; q number of recourse points on test set; n: # train points; M: max # deleted train points; s: invalidation target

$\boldsymbol{\mu}^{(1)} = \mathbf{1}_n$ ▷ Mu are soft data weights that are opimized.

for $m = 1$: Step **do** ▷ Perform $Step$ number of updates.

  $\delta-$loss=0.0

  for $k = 1 : K$ **do** ▷ Use K Monte-Carlo Samples for the approximation

    Sample $\boldsymbol{\epsilon}_k^{(m)} \sim \mathcal{N}\left(0, \sigma^2 \mathbf{I}_n\right)$

    $\boldsymbol{\omega}_k^{(m)} = \max\left(0, \min\left(1, \boldsymbol{\mu}^{(m)} + \boldsymbol{\epsilon}_k^{(m)}\right)\right)$ ▷ Sample (soft) data weights as in [63]

    $\mathbf{w}_{k,\text{new}}^{(m)} = \texttt{update\_w}(\boldsymbol{\omega}_k^{(m)})$ ▷ Compute model weights from data weights either analytically or with IJ

    $l_k^{(m)} = \text{sigmoid}\left(f_{\mathbf{w}_{k,\text{new}}^{(m)}}(\check{\mathbf{X}}_{\mathbf{1}}) - s\right)$ ▷ Predict with new weights and compute soft invalidation.

    $\delta-\text{loss} = \delta-\text{loss} + \|l_k^{(m)}\|_1$ ▷ Add up soft inval. loss

  end for

  $r^{(m)} = \sum_{i=1}^{n} \Phi\left(\frac{1-(\boldsymbol{\mu}^{(m)})_i}{\sigma}\right)$ ▷ Sparsity Regularizer from [63]

  $\boldsymbol{\mu}^{(m+1)} = \boldsymbol{\mu}^{(m)} + \gamma \nabla_{\boldsymbol{\mu}^{(m)}}\left(\frac{\delta-\text{loss}}{D} + \lambda r^{(m)}\right)$ ▷ Grad. Descent with lr. γ

end for

removed_ind = $\texttt{argsort}(\boldsymbol{\mu}^{(\text{Step}+1)})$ ▷ Sort indices ascendingly

$j = 0$

$\boldsymbol{\omega} = \mathbf{1}_n$

while $j < M$ **do** ▷ Binarize and fulfil max number M

  $\boldsymbol{\omega}$[removed_ind[j]]= 0

  $j = j + 1$

end while

return: $\boldsymbol{\omega}$ ▷ data weights indicating M removals