Distributionally Robust Algorithmic Recourse for Tree Ensembles

Kentaro Kanamori
Artificial Intelligence Laboratory
Fujitsu Limited
k.kanamori@fujitsu.com

Ken Kobayashi
School of Engineering
Institute of Science Tokyo
kobayashi.k@iee.eng.isct.ac.jp

Takuya Takagi
Artificial Intelligence Laboratory
Fujitsu Limited
takagi.takuya@fujitsu.com

Abstract

Algorithmic recourse (AR) aims to provide a recourse action that alters an undesired prediction made by a machine learning model. While standard AR methods assume the model does not change over time, this assumption is often violated in practice due to distribution shifts or data updates, rendering the suggested actions invalid. To address this issue, recent studies have proposed several methods that provide actions robust to model changes. Most of them construct an uncertainty set of models by perturbing continuous model parameters, such as the weight vectors of neural networks, and then seek actions that remain valid for all models in the set. However, such an approach cannot be directly applied to tree ensembles because their model parameters include discrete tree structures, which cannot be perturbed in the same way as neural networks. In this paper, we introduce a new robust AR framework for tree ensembles through the lens of distributionally robust optimization (DRO). Our key idea is to express the model change in a tree ensemble as a perturbation to the distribution of predictions made by decision trees in the ensemble, rather than to the model parameters themselves. This formulation allows us to naturally cast the robust AR problem for tree ensembles as a DRO problem. We then show that, by choosing a specific discrepancy measure between distributions, our robust validity constraint can be reformulated into a tractable one that can be evaluated as efficiently as the standard validity constraint. Building on this reformulation, we propose an efficient optimization algorithm by extending the existing search-based method for tree ensembles. Experimental results demonstrated that our method achieved a better robustness-cost trade-off than existing robust methods applicable to tree ensembles, while maintaining comparable runtime to the standard method.

1 Introduction

In high-stakes decision-making with machine learning models, there is a growing demand for explaining how users can alter undesired predictions [47, 70]. *Algorithmic Recourse (AR)* aims to provide such information [67]. For a model $h\colon \mathcal{X} \to \mathcal{Y}$, a desired class $y^* \in \mathcal{Y}$, and an instance $\boldsymbol{x} \in \mathcal{X}$ such that $h(\boldsymbol{x}) \neq y^*$, AR provides a perturbation $\boldsymbol{a}$ that changes the prediction to the desired class, i.e., $h(\boldsymbol{x} + \boldsymbol{a}) = y^*$. Users can regard the perturbation $\boldsymbol{a}$ as a *recourse action* that explains how to change their current status $\boldsymbol{x}$ for obtaining the desired outcome y^* [36]. For example, let us consider a situation where a bank deploys a model h for predicting whether a loan applicant will

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

repay the loan or default, and a user $\boldsymbol{x}$ gets the loan application rejected. To help the user $\boldsymbol{x}$ get the loan approved, AR suggests an action $\boldsymbol{a}$ that alters the prediction of h from "default" to "repayment."

While standard AR methods assume that the model h does not change over time, this assumption is often violated in practice [2, 15]. For example, economic conditions may change over time, leading to *distribution shifts* in the population of loan applicants [60]. Another example is the *right to be forgotten*, in which some applicants may request the bank to delete their data from the training dataset [57]. These situations require the bank to update the model h on the shifted data. Such a shift in the data can result in an equally accurate but different model h' [45, 56, 61, 65]. This can cause a critical issue that the actions optimized for the current model h may not be valid for the updated model h', i.e., $h'(\boldsymbol{x} + \boldsymbol{a}) \neq y^*$, which can undermine the reliability of AR [2, 15].

To alleviate this issue, several approaches have been proposed to provide actions that are *robust* against the model change [28, 29]. Most of these approaches are based on *robust optimization* [3], which seeks an action that is valid for all the models in an uncertainty set around the current model h [66]. The uncertainty set is usually defined by perturbing the continuous model parameters of h, e.g., the weight vector of a linear model or neural network, to express model changes that can potentially occur in the future. Such a formulation allows us to design efficient algorithms for obtaining robust actions and to provide theoretical guarantees of their robustness to model changes [9, 27, 52, 65]. We provide a detailed review of existing work related to robust AR in Appendix A.

In this paper, we focus on robust AR for *tree ensembles*, such as gradient boosted decision trees [13, 20, 37, 59] and random forests [7, 21]. Given their high predictive performance and computational efficiency, they are established as one of the state-of-the-art models for tabular datasets [17, 22, 25, 46], which are prevalent in domains requiring AR, such as finance or justice [31, 34, 68]. Despite their popularity and significance in AR, most existing robust AR methods cannot be directly applied to tree ensembles. A fundamental reason is that tree ensembles include discrete structures, and thus their model parameters cannot be perturbed in the same way as those of linear models or neural networks, which lie in continuous spaces. Furthermore, tree ensembles are neither differentiable nor smooth, which makes it difficult to exploit gradient-based methods or local linear approximations [16]. These facts suggest that formulating the robust AR problem for tree ensembles is non-trivial and motivate us to design a new optimization framework tailored to tree ensembles.

Our contributions In this paper, we propose *Distributionally Robust Algorithmic Recourse for Tree Ensembles (DiRARTE)*, a new framework for providing robust actions for tree ensembles. We formulate the robust AR problem for tree ensembles as a *distributionally robust optimization (DRO)* problem [40], derive tractable reformulations for specific distributional discrepancies, and propose an efficient algorithm by extending the existing method. Our contributions are summarized as follows:

- We propose a new formulation of the robust AR problem for tree ensembles via DRO. To characterize model changes in tree ensembles, we exploit the notion of ensemble distribution, which is a probabilistic interpretation of the ensemble prediction [44]. We express the model change in a tree ensemble as a perturbation of its ensemble distribution, which enables us to naturally formulate the robust AR problem for tree ensembles as a DRO problem.
- We propose an efficient algorithm for solving our robust AR problem for tree ensembles. We first show that the robust validity constraint in our formulation can be reformulated as a tractable constraint by choosing the Wasserstein distance or the χ^2-divergence as the discrepancy measure between distributions. Then, we propose an efficient algorithm by extending and generalizing the existing search-based algorithm for tree ensembles [64].
- We conducted experiments on real datasets under three model-change scenarios and demonstrated that our method achieved a better trade-off between the robustness and cost of actions than existing methods applicable to tree ensembles. We also confirmed that our method improved robustness while maintaining a comparable runtime to the standard AR method.

2 Preliminaries

For a positive integer $n \in \mathbb{N}$, we write $[n] \coloneqq \{1, \ldots, n\}$. Following the standard setting [36, 68], we consider a binary classification task between undesired and desired classes. Let $\mathcal{X} \subseteq \mathbb{R}^D$ and $\mathcal{Y} = \{\pm 1\}$ be input and output domains, respectively. We assume that all the categorical features are encoded by one-hot encoding. We call a vector $\boldsymbol{x} = (x_1, \ldots, x_D) \in \mathcal{X}$ an *instance*. We assume

that a *classifier* $h_\theta \colon \mathcal{X} \to \mathcal{Y}$ with model parameter θ makes a prediction according to $h_\theta(\boldsymbol{x}) = +1$ if $f_\theta(\boldsymbol{x}) \geq 0$ and $h_\theta(\boldsymbol{x}) = -1$ otherwise, where $f_\theta \colon \mathcal{X} \to \mathbb{R}$ is a *scoring function* of h_θ. We sometimes refer to f_θ as the classifier itself, and call its real-valued output $\hat{y} = f_\theta(\boldsymbol{x})$ a *prediction score*.

2.1 Tree ensembles

A *tree ensemble* f_θ consists of T *decision trees* $f_{\theta_1}, \ldots, f_{\theta_T} \colon \mathcal{X} \to \mathbb{R}$, where $\theta = \{\theta_1, \ldots, \theta_T\}$. It makes a prediction by combining the outputs of all the trees. There exist two major frameworks for learning tree ensembles: *bagging* [6] and *gradient boosting* [20]. The former, including random forests [7], trains each tree independently on different bootstrap samples of the training data. The latter, including XGBoost [13], trains each tree sequentially by fitting the pseudo residuals from previous trees. Without loss of generality, a prediction score of a learned tree ensemble f_θ can be expressed as the average of the outputs of the trees in the ensemble, i.e., $f_\theta(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} f_{\theta_t}(\boldsymbol{x})$ [75].

Each decision tree f_{θ_t} has a binary tree structure that represents a piecewise constant function [8]. Each internal node of the tree corresponds to the *split condition* of the form $x_d \leq b$ for some feature $d \in [D]$ and threshold $b \in \mathbb{R}$. Each leaf node corresponds to a constant value $w \in \mathbb{R}$ called the *leaf weight*. For a given input instance $\boldsymbol{x} \in \mathcal{X}$, a decision tree outputs the leaf weight of the corresponding leaf node. The corresponding leaf is determined by traversing the tree from the root, depending on whether the split condition of each internal node is true or not. Let I_t be the total number of leaves in f_{θ_t} and $w_{t,i}$ be the weight of the leaf $i \in [I_t]$. Then, we can express $f_{\theta_t}(\boldsymbol{x}) = \sum_{i=1}^{I_t} w_{t,i} \cdot \mathbb{I}\left[\boldsymbol{x} \in r_{t,i}\right]$, where $r_{t,i} \subseteq \mathcal{X}$ is the region corresponding to i. Each region $r_{t,i}$ is an axis-aligned rectangle and is determined by the split conditions on the path from the root to the leaf i. The set of regions $\{r_{t,1}, \ldots, r_{t,I_t}\}$ forms a partition of $\mathcal{X}$, i.e., $\cup_{i=1}^{I_t} r_{t,i} = \mathcal{X}$ and $r_{t,i} \cap r_{t,j} = \emptyset$ for all $i \neq j$ [24, 69].

2.2 Algorithmic recourse

For an instance $\boldsymbol{x} \in \mathcal{X}$, we define a *recourse action* as a perturbation vector $\boldsymbol{a} \in \mathbb{R}^D$ such that $\boldsymbol{x} + \boldsymbol{a} \in \mathcal{X}$. Let $\mathcal{A}(\boldsymbol{x})$ be a set of feasible actions for $\boldsymbol{x}$ such that $\boldsymbol{0} \in \mathcal{A}(\boldsymbol{x})$ and $\mathcal{A}(\boldsymbol{x}) \subseteq \{\boldsymbol{a} \in \mathbb{R}^D \mid \boldsymbol{x} + \boldsymbol{a} \in \mathcal{X}\}$. It often includes some constraints on the action $\boldsymbol{a}$, such as the immutability of certain features (e.g., gender or race). For a classifier h_θ and desired class $y^* \in \mathcal{Y}$, an action $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$ is *valid* for $\boldsymbol{x}$ if $h_\theta(\boldsymbol{x} + \boldsymbol{a}) = y^*$ holds. For an instance $\boldsymbol{x} \in \mathcal{X}$ and its feasible action $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$, a *cost function* $c \colon \mathcal{A}(\boldsymbol{x}) \to \mathbb{R}_{\geq 0}$ measures the required effort of $\boldsymbol{a}$ with respect to $\boldsymbol{x}$.

Given an instance $\boldsymbol{x}$, the aim of *Algorithmic Recourse (AR)* [67] is to find a feasible action $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$ that is valid for $\boldsymbol{x}$ with respect to h_θ and minimizes its cost $c(\boldsymbol{a} \mid \boldsymbol{x})$. For later formulation, we consider a margin-based formulation of AR that is equivalent to the standard one [36]. For a desired class $y^* \in \mathcal{Y}$ and a prediction score $\hat{y} \in \mathbb{R}$, let $\rho(y^*, \hat{y}) \coloneqq y^* \cdot \hat{y}$ be the *margin* [49]. Without loss of generality, we assume $y^* = +1$. Then, $\rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a})) \geq 0$ holds if and only if $h_\theta(\boldsymbol{x} + \boldsymbol{a}) = y^*$ holds, i.e., $\boldsymbol{a}$ is valid for $\boldsymbol{x}$. It allows us to formulate the standard AR problem as follows:

$$\text{minimize}_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})} \; c(\boldsymbol{a} \mid \boldsymbol{x}) \quad \text{subject to} \;\; \rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a})) \geq 0. \tag{1}$$

This paper focuses on the case where the classifier f_θ is a tree ensemble. In contrast to linear models or neural networks, tree ensembles are neither differentiable nor smooth, which makes it difficult to solve the above optimization problem via gradient-based methods or local linear approximations [16]. To address this issue, previous studies have proposed several techniques, including mixed-integer linear optimization [14, 32, 55], search-based algorithms [11, 64], and continuous relaxations [43].

3 Problem formulation

In this paper, we aim to provide actions that are *robust* against the model change of tree ensembles. More precisely, our goal is to find an action $\boldsymbol{a}$ that is valid not only for the current tree ensemble model f_θ, but also for models that are slightly changed from f_θ [28]. One major approach is to extend the standard AR problem in (1) to a *robust optimization* problem [3]. This approach first formulates an uncertainty set of models around f_θ by perturbing the model parameter θ. Then, it seeks a minimum-cost action $\boldsymbol{a}$ that satisfies the *robust validity constraint*, which requires that $\boldsymbol{a}$ is valid for all the models in the uncertainty set [66]. While this approach has been widely used, it cannot be directly applied to tree ensembles because of their discrete nature [16]. Unlike those of

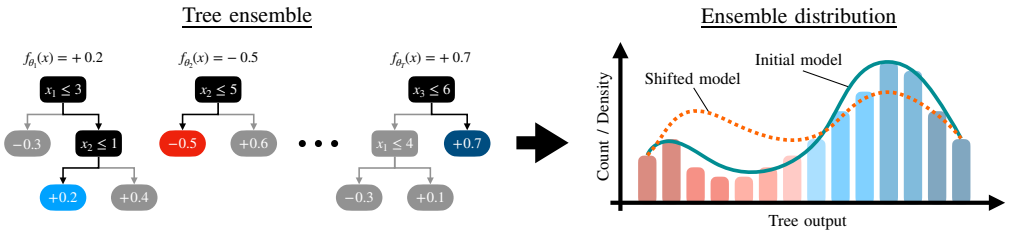


Figure 1: An illustrative example of the ensemble distribution. Given a tree ensemble f_θ and an instance $\boldsymbol{x}$, we define the ensemble distribution $q_\theta(\hat{y} \mid \boldsymbol{x})$ as the distribution of outputs of the trees $f_{\theta_1}, \ldots, f_{\theta_T}$ in the ensemble. We can regard the ensemble prediction $f_\theta(\boldsymbol{x})$ as the expectation over $q_\theta(\hat{y} \mid \boldsymbol{x})$, i.e., $f_\theta(\boldsymbol{x}) = \mathbb{E}_{\hat{y} \sim q_\theta(\cdot \mid \boldsymbol{x})}[\hat{y}]$. Our key idea is to characterize the model change of a tree ensemble f_θ by a perturbation to its ensemble distribution $q_\theta(\hat{y} \mid \boldsymbol{x})$, instead of the model parameter θ. It allows us to formulate the robust AR problem for tree ensembles as a DRO problem.

linear models or neural networks, the model parameter θ of a tree ensemble encodes discrete tree structures, for which continuous perturbations are not well-defined. To address this issue, we need to design a robust validity constraint suited to tree ensembles from a different perspective.

In this section, we propose a new formulation of the robust AR problem for a tree ensemble through the lens of *distributionally robust optimization (DRO)* [40]. Our core idea is to express the model change in a tree ensemble as a perturbation of its *ensemble distribution*, which is a probabilistic interpretation of the ensemble prediction [44]. Then, we define the uncertainty set of a tree ensemble f_θ by perturbing its ensemble distribution rather than the model parameter θ. This allows us to naturally express the robust validity constraint and thus formulate our task as a DRO problem; that is, the ensemble distribution serves as an interface through which tractable DRO techniques become applicable to tree ensembles.

3.1 Ensemble distribution and uncertainty set

To characterize the model change in tree ensembles, we introduce the notion of *ensemble distribution* [44]. We interpret a tree ensemble f_θ as making its prediction by taking the expectation over the distribution of outputs of the decision trees $f_{\theta_1}, \ldots, f_{\theta_T}$ in the ensemble. For a learned tree ensemble f_θ and a test instance $\boldsymbol{x}$, we define its ensemble distribution by

$$q_\theta(\hat{y} \mid \boldsymbol{x}) \coloneqq \frac{1}{T} \sum_{t=1}^{T} \delta(\hat{y} - f_{\theta_t}(\boldsymbol{x})),$$

where $\delta(\cdot)$ is the Dirac delta function. Equivalently, $q_\theta(\cdot \mid \boldsymbol{x})$ is the empirical distribution that assigns mass $\frac{1}{T}$ to the output of each tree $f_{\theta_t}(\boldsymbol{x})$. By definition, we have $f_\theta(\boldsymbol{x}) = \mathbb{E}_{\hat{y} \sim q_\theta(\cdot \mid \boldsymbol{x})}[\hat{y}]$; that is, the ensemble prediction $f_\theta(\boldsymbol{x})$ is equal to the expectation over the ensemble distribution $q_\theta(\hat{y} \mid \boldsymbol{x})$. An illustrative example of the ensemble distribution is shown in Figure 1.

Our key idea is to characterize the model change in a tree ensemble via a perturbation of its ensemble distribution. If each tree in the ensemble is changed, its ensemble distribution also changes, leading to a change in the ensemble prediction. Thus, we define the uncertainty set of a tree ensemble as the set of distributions perturbed from the ensemble distribution of the current model. For a given radius $\varepsilon \geq 0$, we define our uncertainty set of a current tree ensemble f_θ at a point $\boldsymbol{x}$ as follows:

$$\mathcal{Q}_\varepsilon(\theta \mid \boldsymbol{x}) \coloneqq \{q \in Q \mid \Delta(q, q_\theta(\cdot \mid \boldsymbol{x})) \leq \varepsilon\},$$

where Q is the set of Borel probability distributions over $\mathbb{R}$ and Δ is an arbitrary discrepancy measure between distributions. Our uncertainty set is intended to contain the ensemble distributions of models shifted from the current model f_θ. The radius ε controls the size of the uncertainty set; a larger ε yields a larger set that includes more diverse shifted models, allowing a more drastic change from f_θ. Note that our uncertainty set may also contain distributions not induced by any finite tree ensemble, which makes our formulation conservative. In our experiments, however, we empirically show that this conservativeness did not materially affect practical performance.

3.2 Problem statement

Now, we formulate our robust AR problem for a tree ensemble. Our aim here is to obtain a minimum-cost action $\boldsymbol{a}$ that is valid for all the models in our uncertainty set of f_θ at $\boldsymbol{x}+\boldsymbol{a}$. Recall that the validity constraint requires the margin $\rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a}))$ to be non-negative and that the ensemble prediction $f_\theta(\boldsymbol{x}+\boldsymbol{a})$ is equal to the expectation over the ensemble distribution $q_\theta(\cdot \mid \boldsymbol{x}+\boldsymbol{a})$. Based on these facts, we define our robust validity constraint by requiring that the expected margin $\mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})]$ is non-negative for all the ensemble distributions q in our uncertainty set $\mathcal{Q}_\varepsilon(\theta \mid \boldsymbol{x}+\boldsymbol{a})$. It leads to the following formulation of our robust AR problem for tree ensembles:

$$\text{minimize}_{\boldsymbol{a}\in\mathcal{A}(\boldsymbol{x})}\ c(\boldsymbol{a} \mid \boldsymbol{x}) \quad \text{subject to} \quad \min_{q\in\mathcal{Q}_\varepsilon(\theta\mid\boldsymbol{x}+\boldsymbol{a})} \mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] \geq 0. \tag{2}$$

By solving this problem, we can obtain a minimum-cost action that is robust to the model change expressed by our uncertainty set. More precisely, the obtained action $\boldsymbol{a}$ is valid for a shifted tree ensemble if the discrepancy Δ between its ensemble distribution and that of the current model f_θ at $\boldsymbol{x}+\boldsymbol{a}$ is at most ε. Note that we can recover the standard AR problem (1) by setting $\varepsilon = 0$ because $\mathcal{Q}_0(\theta \mid \boldsymbol{x}+\boldsymbol{a}) = \{q_\theta(\cdot \mid \boldsymbol{x}+\boldsymbol{a})\}$ and $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot\mid\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*, \hat{y})] = \rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a}))$.

A key benefit of our formulation is that it can express the robust validity constraint without explicitly constructing shifted models. It enables us to avoid the combinatorial nature of tree ensembles and design an efficient algorithm via tractable reformulation, as we will show in the next section.

4 Optimization framework

In this section, we propose an efficient algorithm to solve our robust AR problem for tree ensembles formulated in (2). A main obstacle to solving the problem (2) is that its constraint involves a minimization problem over the uncertainty set $\mathcal{Q}_\varepsilon$, which is generally intractable. To address this issue, we show that, if we choose a specific discrepancy measure Δ, our constraint can be reformulated into a tractable one that can be computed in the same time complexity as that in the standard AR problem. Then, we propose an optimization algorithm for the problem (2) by extending the existing search-based method for the standard AR [64]. All the proofs of our results are given in Appendix B.

4.1 Tractable reformulation

First, we show that our robust validity constraint in (2) can be reformulated into a tractable one if we choose a specific discrepancy measure Δ. We consider the *Wasserstein distance* Δ_{W_p} and the χ^2*-divergence* Δ_{χ^2}, which are two major discrepancy measures in the DRO literature [26, 48, 50, 51], respectively defined as follows:

$$\Delta_{W_p}(q, q') \coloneqq \left(\inf_{\pi\in\Pi(q,q')} \int_{\mathbb{R}\times\mathbb{R}} |\hat{y}-\hat{y}'|^p d\pi(\hat{y}, \hat{y}')\right)^{\frac{1}{p}}, \quad \Delta_{\chi^2}(q \mid\mid q') \coloneqq \mathbb{E}_{\hat{y}\sim q'}\left[\left(\frac{dq}{dq'}(\hat{y}) - 1\right)^2\right],$$

where $p \geq 1$, $\Pi(q, q')$ is the set of all joint distributions π over $\mathbb{R}\times\mathbb{R}$ with marginals $q, q' \in Q$, and dq/dq' is the Radon–Nikodym derivative of q with respect to q'.

4.1.1 Wasserstein distance

In Proposition 1, we show that our robust validity constraint in (2) can be reformulated into an intuitive extension of the standard validity constraint in (1) when we use the Wasserstein distance.

Proposition 1. *For the uncertainty set $\mathcal{Q}_\varepsilon$, we assume $\Delta = \Delta_{W_p}$ with $p \geq 1$. Then, we have*

$$\min_{q\in\mathcal{Q}_\varepsilon(\theta\mid\boldsymbol{x}+\boldsymbol{a})} \mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] \geq 0 \iff \rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a})) \geq \varepsilon.$$

Proposition 1 indicates that our robust validity constraint with the Wasserstein distance can be reduced to a simple one that requires the margin $\rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a}))$ to be at least ε. This is a natural extension of the standard validity constraint in (1) and intuitive in that a larger margin ensures the stability of predictions [5]. That is, if the margin of f_θ at $\boldsymbol{x}+\boldsymbol{a}$ is sufficiently large, the corresponding prediction is less likely to be changed by a small change in the model, implying the robustness of the action $\boldsymbol{a}$.

We remark that Forel et al. [19] showed a similar result by assuming that the output of each tree follows a Bernoulli distribution and is aggregated by hard voting. Their assumptions require each tree

Algorithm 1 *FT++*: our generalized feature tweaking algorithm for solving the problem (2).

Input: Instance $\boldsymbol{x}$, desired class y^*, tree ensemble model f_θ with T decision trees $f_{\theta_1}, \ldots, f_{\theta_T}$, cost function c, feasible actions $\mathcal{A}(\boldsymbol{x})$, merging level L, discrepancy measure Δ, radius ε

$R_t^{(1)} \leftarrow \{r_{t,1}, \ldots, r_{t,I_t}\}$ for all $t \in [T]$; ▷ Enumerate all leaf regions in ensemble
for $l = 2, \ldots, L$ **do**
  for $k = 1, \ldots, \lceil T/2^{l-1} \rceil$ **do**
    $R_k^{(l)} \leftarrow \{r \cap r' \mid (r, r') \in R_{2k-1}^{(l-1)} \times R_{2k}^{(l-1)}, r \cap r' \neq \emptyset\}$; ▷ Merge two sets of leaf regions
  end for
end for
$\mathcal{R} \leftarrow R_1^{(L)} \cup \cdots \cup R_{\lceil T/2^{L-1} \rceil}^{(L)}$; $\boldsymbol{a}^* \leftarrow \mathbf{0}$; $c^* \leftarrow \infty$;
for $r \in \mathcal{R}$ **do** ▷ Search for minimum-cost robust action with merged leaf regions
  $\hat{\boldsymbol{a}} \leftarrow \arg\min_{\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})} c(\boldsymbol{a} \mid \boldsymbol{x})$ s.t. $\boldsymbol{x} + \boldsymbol{a} \in r$;
  If $\hat{\boldsymbol{a}}$ satisfies the constraint in (2) and $c(\hat{\boldsymbol{a}} \mid \boldsymbol{x}) < c^*$ **then** $\boldsymbol{a}^* \leftarrow \hat{\boldsymbol{a}}$; $c^* \leftarrow c(\hat{\boldsymbol{a}} \mid \boldsymbol{x})$;
end for
return $\boldsymbol{a}^*$;

to output a binary-valued prediction. In contrast, we consider a setting where the output of each tree follows an arbitrary distribution and can be aggregated via soft voting, allowing each tree to output a real-valued prediction score. Thus, our result provides a theoretical justification of the large-margin strategy for a more general class of tree ensembles from the perspective of DRO.

4.1.2 χ^2-divergence

In Proposition 2, we show that our robust validity constraint in (2) can be reformulated into a different form if we employ the χ^2-divergence as the discrepancy measure Δ.

Proposition 2. *Let $\hat{\sigma}_\theta^2(\boldsymbol{x} + \boldsymbol{a}) := \frac{1}{T} \sum_{t=1}^{T} (f_\theta(\boldsymbol{x} + \boldsymbol{a}) - f_{\theta_t}(\boldsymbol{x} + \boldsymbol{a}))^2$ be the empirical variance of the outputs over a tree ensemble f_θ at $\boldsymbol{x} + \boldsymbol{a}$. For the uncertainty set $\mathcal{Q}_\varepsilon$, we assume $\Delta = \Delta_{\chi^2}$ and $\sqrt{\varepsilon} \leq \hat{\sigma}_\theta(\boldsymbol{x} + \boldsymbol{a}) / \max_{t \in [T]} |f_\theta(\boldsymbol{x} + \boldsymbol{a}) - f_{\theta_t}(\boldsymbol{x} + \boldsymbol{a})|$ for any $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$. Then, we have*

$$\min_{q \in \mathcal{Q}_\varepsilon(\theta \mid \boldsymbol{x} + \boldsymbol{a})} \mathbb{E}_{\hat{y} \sim q}[\rho(y^*, \hat{y})] \geq 0 \iff \rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a})) \geq \sqrt{\varepsilon} \cdot \hat{\sigma}_\theta(\boldsymbol{x} + \boldsymbol{a}).$$

Proposition 2 indicates that our robust validity constraint with the χ^2-divergence can be expressed as a penalized version of the margin-based constraint with the standard deviation over the ensemble distribution at $\boldsymbol{x} + \boldsymbol{a}$. This constraint requires the margin $\rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a}))$ to be sufficiently large compared to the variability of the tree outputs. In terms of computational efficiency, the term $\hat{\sigma}_\theta(\boldsymbol{x} + \boldsymbol{a})$ can be computed in the same time complexity as the ensemble prediction $f_\theta(\boldsymbol{x} + \boldsymbol{a})$. This implies that the above constraint can be evaluated with the same time complexity as that in (1).

One interesting note is that the term $\hat{\sigma}_\theta^2(\boldsymbol{x})$ is known as the *diversity* or *ambiguity* in the ensemble learning literature, which is a key factor in improving the generalization performance of ensemble learners [41, 71, 75]. Proposition 2 implies that our constraint can become more stringent for a tree ensemble with higher diversity, suggesting a connection between the robustness of recourse actions and the generalization performance of tree ensembles, which is a new insight in the context of AR.

Note that the assumption on ε is a technical condition, which is satisfied by setting ε to be sufficiently small in practice. Even without this assumption, our constraint remains a sufficient condition for robust validity. We also note that, since the χ^2-divergence requires absolute continuity, our uncertainty set with $\Delta = \Delta_{\chi^2}$ only contains reweightings of the current tree outputs. While it does not formally cover model changes that create new tree-output values, we observed that it could capture actual model changes well despite these limitations, which will be demonstrated in our experiments.

4.2 Optimization algorithm

Based on the above results, we design an optimization algorithm for the problem (2). Propositions 1 and 2 show that our robust validity constraint is exactly equivalent to, rather than merely implied by, a tractable one that tightly characterizes the worst case over our uncertainty set and can be evaluated in a similar way to the standard validity constraint in (1). This allows us to directly integrate our robust validity constraint into the existing standard AR algorithms for tree ensembles [11, 14, 32, 43, 55, 64]. In this paper, we focus on the *feature tweaking (FT)* algorithm [64], which is a search-based heuristic

algorithm. It first enumerates all regions $r_{t,i}$ corresponding to the leaves i in each tree f_{θ_t}. Then, it seeks a minimum-cost feasible action $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$ that moves $\boldsymbol{x}$ into each region $r_{t,i}$, which can be computed easily due to the axis-aligned structure of the leaf regions [11]. Finally, the algorithm returns the minimum-cost action among the generated ones that satisfy the validity constraint. Compared to other methods [14, 32, 43, 55], FT has two advantages: (i) scalability due to its efficient search strategy; (ii) flexibility to handle broad classes of cost functions and constraints.

One main limitation of the FT algorithm is its limited search space, which only includes the region corresponding to each leaf in the ensemble. In general, a tree ensemble partitions the input space into more fine-grained regions expressed by the intersections of the leaf regions across the trees [24, 39, 69]. Thus, the original FT algorithm may fail to find feasible solutions or yield suboptimal solutions, even when feasible or better solutions exist in fine-grained regions. Moreover, Propositions 1 and 2 imply that our robust validity constraint is more stringent than the standard validity constraint, which makes it more difficult to find feasible solutions to the problem (2). These facts motivate us to extend the original FT algorithm by enlarging the search space while maintaining its efficiency.

To this end, we propose *FT++*, a generalized version of the original FT algorithm. Algorithm 1 presents our FT++ algorithm for solving the problem (2). Our key idea is to expand the search space by merging the leaf regions across the trees. To efficiently merge the leaf regions, we adopt a multi-level merging strategy originally proposed for robustness verification of tree ensembles [12]. We iteratively merge two sets of regions until a predefined merging level L is reached. After generating the fine-grained regions via multi-level merging, we search for a minimum-cost feasible action that satisfies our constraint, as in the original FT algorithm. In Proposition 3, we show that our FT++ can exactly solve the problem (2) if we set the merging level L to be sufficiently large.

Proposition 3. *Algorithm 1 exactly solves the problem* (2) *if* $L = \lceil \log_2 T \rceil + 1$.

Let I and H be the maximum number of leaves and depth of each tree, respectively. Algorithm 1 searches over $O(TI^{2^{L-1}}/2^{L-1})$ regions and, for each region, computes the optimal action in $O(D)$ time for separable cost functions [11] and checks our constraint in $O(TH)$ time. Thus, its time and space complexities are $O(TI^{2^{L-1}}(D + TH)/2^{L-1})$ and $O(TI^{2^{L-1}}D/2^{L-1})$, which can be exponential in T for $L = \lceil \log_2 T \rceil + 1$. Fortunately, we empirically observed that $L = 2$, i.e., $O(TI^2(D + TH))$ time and $O(TI^2D)$ space, was sufficient to obtain feasible solutions while keeping the runtime below 0.2 seconds per instance, and was drastically faster than an exact mixed-integer optimization method [14, 32]. We report an experimental comparison between existing algorithms and Algorithm 1 in Appendix E. Furthermore, we also propose a heuristic method to accelerate Algorithm 1 by leveraging the *ensemble pruning* technique [41, 74] in Appendix C.

4.3 Theoretical connection to actual model changes

Finally, we connect our uncertainty set based on the ensemble distribution to actual model changes occurring in tree ensembles. Following the prior studies [16, 23], we consider a type of model change that focuses on changes in the model output. In Proposition 4, we show that our formulation with the Wasserstein distance can guarantee the robustness to a certain type of model change in tree ensembles.

Proposition 4. *For a tree ensemble* f_θ *and* $\varepsilon \geq 0$*, let* $f_{\theta'}$ *be a shifted model of* f_θ *with* T *decision trees* $f_{\theta'_1}, \ldots, f_{\theta'_T}$ *such that* $\max_{\boldsymbol{x} \in \mathcal{X}} \min_{\sigma \in \Sigma} \frac{1}{T} \sum_{t=1}^{T} |f_{\theta_t}(\boldsymbol{x}) - f_{\theta'_{\sigma(t)}}(\boldsymbol{x})| \leq \varepsilon$*, where* Σ *is the set of all one-to-one mappings* $\sigma \colon [T] \to [T]$ *between the trees in* f_θ *and those in* $f_{\theta'}$*. Then, any feasible solution* $\boldsymbol{a}$ *to the problem* (2) *with* $\Delta = \Delta_{W_1}$ *is valid for* $f_{\theta'}$*, i.e.,* $h_{\theta'}(\boldsymbol{x} + \boldsymbol{a}) = y^*$.

Retraining on a slightly different dataset may change the trees in the ensemble and thus their outputs. Proposition 4 guarantees that our actions remain valid for such a shifted model as long as, at every input, its tree outputs differ from those of f_θ by at most ε on average under the best one-to-one matching. Thus, ε can be interpreted as a bound on the average change in the tree outputs, which gives a guideline for choosing it. Because the ensemble prediction is invariant to the order of the trees, two ensembles trained on different datasets have no natural correspondence between their trees, and we can adopt the best matching. Since the inner minimum in our assumption equals the 1-Wasserstein distance $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ [58], Proposition 4 is a conditional guarantee for shifted models whose ensemble distributions lie in our uncertainty set with $\Delta = \Delta_{W_1}$ at every input. Its role is to translate this membership condition over distributions into a condition on the outputs of individual trees. Note that since $|f_\theta(\boldsymbol{x}) - f_{\theta'}(\boldsymbol{x})| \leq \Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ always holds, our assumption is

sufficient but not necessary for bounding the change in the ensemble prediction; that is, Proposition 4 does not cover all the model changes under which our actions remain valid. We empirically examine the relationship between our uncertainty set and actual model changes in Appendix E.

5 Experiments

To investigate the effectiveness of our method, we conducted experiments on real datasets. All the code was implemented in Python 3.10 and is available at `https://github.com/kelicht/dirarte`. All the experiments were conducted on macOS Sequoia with Apple M2 Ultra CPU and 128 GB memory. Our experiments aim to answer the following questions: (i) Can our method effectively provide robust recourse actions while maintaining reasonable costs? (ii) How do the scalability of algorithms and the quality of obtained actions differ between our method and the baselines? (iii) Does the choice of the discrepancy measure Δ affect the performance of our method? Due to page limitations, the implementation details and the complete results are given in Appendices D and E, respectively.

5.1 Experimental setup

We used five real datasets commonly used in the AR literature: *FICO* ($D = 23$) [18], *Credit* ($D = 13$) [73], *German* ($D = 11$) [38], *COMPAS* ($D = 15$) [1], and *Bail* ($D = 16$) [62]. We imposed actionability constraints on certain features in each dataset (e.g., gender). For tree ensembles, we employed *XGBoost* [13] and *random forest* [7] with $T = 100$ trees and a maximum depth of 6. As the cost function c, we used the *total log percentile shift (TLPS)* [67] and the *weighted ℓ_1-norm* [70]. We show the results with XGBoost and TLPS here, and the other results are given in Appendix E.

Protocol For each dataset, we randomly split the data into training and test sets at an $80 : 20$ ratio. To simulate the model change, we consider the following three types of scenarios:

- *DataSplit*: Randomly split the training set into two same-sized subsets, and train a tree ensemble on each subset, following the previous study [16];
- *DataDeletion*: Train a tree ensemble on the full training set, and then train another tree ensemble on the training set after randomly deleting 10% of the training data;
- *LabelShift*: Randomly sample two subsets from the training set, each with a label ratio that differs by $\pm 10\%$ from the original ratio, and train a tree ensemble on each subset.

After training an initial model f_θ and a shifted model $f_{\theta'}$, we generated an action $\boldsymbol{a}$ for each test instance $\boldsymbol{x}$ that receives the undesired prediction from f_θ. We measured the average values of (1) *robust validity*, i.e., the fraction of the test instances $\boldsymbol{x}$ for which $\boldsymbol{x} + \boldsymbol{a}$ is classified into the desired class by $f_{\theta'}$; (2) *cost* $c(\boldsymbol{a} \mid \boldsymbol{x})$; (3) *sparsity* $\|\boldsymbol{a}\|_0$; (4) *plausibility*, which is evaluated by the outlier score of the isolation forest [42] at $\boldsymbol{x} + \boldsymbol{a}$, following previous work [55]; (5) *runtime* per instance. For each dataset and scenario, we repeated the above procedure with 10 different data splits and report the average results.

Baselines We compare our methods (*DiRARTE-W* and *DiRARTE-χ^2*) to the two existing methods that can be applied to tree ensembles. One competitor is *RobX* [16], which is the first robust AR method designed for tree ensembles. Given an initial action obtained by a standard AR method, it iteratively updates the action by interpolating with the nearest valid counterfactuals until its robustness is verified via the stability score. The stability score evaluates the mean and standard deviation of predictions for perturbed instances around $\boldsymbol{x} + \boldsymbol{a}$, simulating changes in split conditions in the trees. Another competitor is *BetaRobust* [63], which is a model-agnostic robust AR method. It first constructs a set of perturbed models by training models with different hyperparameters from the initial one (e.g., number of trees and maximum depth). Then, it evaluates the validity of actions with respect to each model and verifies their robustness at a given confidence level, assuming they follow a beta distribution. Because RobX and BetaRobust need to be combined with a standard AR method, we employed Algorithm 1 with $\varepsilon = 0$ so that differences reflect their robustness criteria rather than their underlying algorithms. All methods have a hyperparameter that controls the robustness of the obtained actions: the radius ε for DiRARTE, the stability threshold τ for RobX, and the threshold δ for BetaRobust. We varied these values within a reasonable range and investigated the robustness-cost trade-off of each method. We report the results of each configuration only if it found feasible solutions for at least 90% of the test instances, and average the cost, sparsity, and plausibility

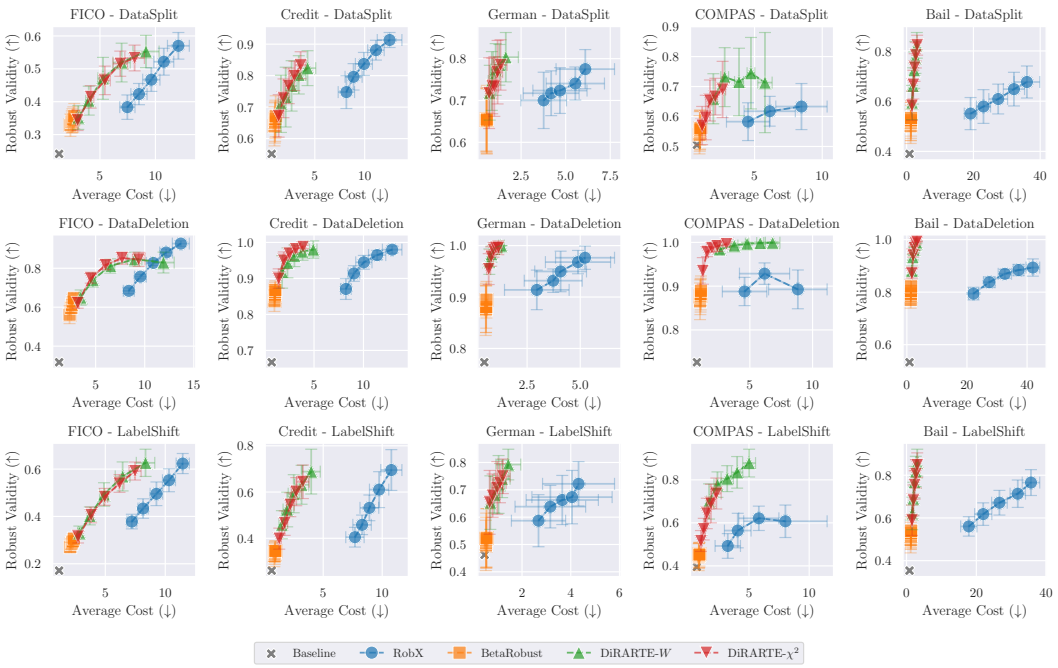


Figure 2: Experimental results on the robust validity (higher is better) and the average cost (lower is better). Each point shows the average over 10 splits for one hyperparameter value of each method, and error bars indicate the standard deviation. Our DiRARTE achieved higher robust validity than BetaRobust and attained a better trade-off between robustness and cost than RobX in most cases.

over those instances; thus, any two configurations share at least 80% of the instances. As a reference, we also report the results of Algorithm 1 with $\varepsilon = 0$ alone (*Baseline*), corresponding to a standard AR method without robustness considerations.

5.2 Experimental results

Robustness-cost trade-off Figure 2 shows the results for robust validity and average cost. From Figure 2, we observed two findings. First, RobX and DiRARTE consistently achieved higher robust validity than BetaRobust, suggesting that this model-agnostic retraining-based method was insufficient to capture model changes in tree ensembles. Second, DiRARTE attained substantially lower average cost than RobX in most cases, while achieving comparable robust validity. This indicates that RobX was overly conservative in identifying robust actions, resulting in unnecessarily high costs. In contrast, our DiRARTE achieved a better trade-off between the robustness and cost of obtained actions than RobX. From these results, we confirmed that *our DiRARTE could provide robust recourse actions for more instances while maintaining reasonable costs compared to the existing methods*.

Scalability and recourse quality Figure 3 presents the average runtime of each method. RobX incurred significant computational overhead due to its iterative search strategy, and BetaRobust also took a moderate amount of time because it checks the validity across multiple perturbed models. Compared to them, our DiRARTE was consistently faster, and its time was comparable to that of Baseline, i.e., a non-robust AR method. Table 1 presents the average sparsity and plausibility of the obtained actions. We can see that RobX often failed to find sparse actions, whereas the other methods, including DiRARTE, could find actions that change only a few features. We also observed that there was only a small difference in plausibility between the methods. These results indicate that our DiRARTE could maintain the practical quality of the obtained actions in terms of sparsity and plausibility. By combining these results with Figure 2, we can conclude that *our DiRARTE achieved robustness without sacrificing the computational efficiency and practical quality of actions*.

Comparison between DiRARTE-W and DiRARTE-χ^2 In terms of the robustness-cost trade-off, DiRARTE-χ^2 dominated DiRARTE-W in some cases (e.g., Credit dataset under the DataDeletion scenario in Figure 2). In addition, DiRARTE-χ^2 often attained slightly better sparsity and plausibility

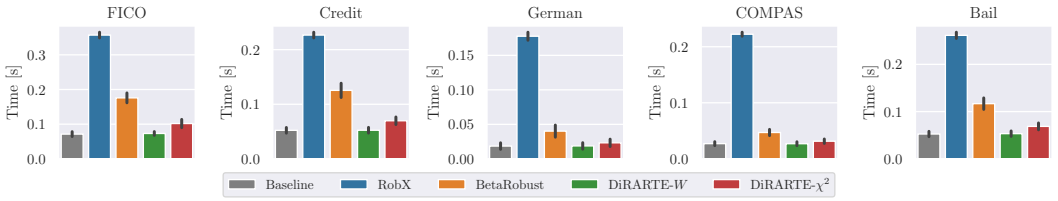


Figure 3: Average runtime per instance of each method (lower is better). Our DiRARTE was stably faster than RobX and BetaRobust, and its runtime was comparable to that of the non-robust Baseline.

Table 1: Average sparsity and plausibility of each method (lower is better). We observed that our DiRARTE did not significantly sacrifice sparsity and plausibility compared to the baselines.

	Method	FICO	Credit	German	COMPAS	Bail
Sparsity	Baseline	1.779 ± 0.05	1.122 ± 0.04	1.049 ± 0.03	1.118 ± 0.03	1.330 ± 0.03
	RobX	15.20 ± 0.13	6.527 ± 0.26	3.859 ± 0.18	2.238 ± 0.23	5.574 ± 0.40
	BetaRobust	2.398 ± 0.11	1.247 ± 0.07	1.090 ± 0.05	1.209 ± 0.04	1.508 ± 0.04
	DiRARTE-W	3.536 ± 0.55	2.176 ± 0.40	1.434 ± 0.21	1.678 ± 0.12	1.976 ± 0.19
	DiRARTE-χ^2	3.485 ± 0.53	1.962 ± 0.34	1.392 ± 0.18	1.460 ± 0.15	2.002 ± 0.22
Plausibility	Baseline	0.453 ± 0.00	0.509 ± 0.01	0.499 ± 0.01	0.452 ± 0.00	0.494 ± 0.00
	RobX	0.438 ± 0.00	0.482 ± 0.01	0.495 ± 0.01	0.447 ± 0.01	0.497 ± 0.01
	BetaRobust	0.452 ± 0.01	0.512 ± 0.01	0.499 ± 0.01	0.451 ± 0.01	0.496 ± 0.00
	DiRARTE-W	0.458 ± 0.01	0.524 ± 0.01	0.499 ± 0.01	0.475 ± 0.02	0.502 ± 0.00
	DiRARTE-χ^2	0.453 ± 0.00	0.514 ± 0.01	0.498 ± 0.01	0.452 ± 0.01	0.501 ± 0.00

than DiRARTE-W, as shown in Table 1. However, we did not observe a remarkable difference between them. These results suggest that *much of the empirical gain in our DiRARTE may come from our DRO formulation itself, rather than from a particular choice of the discrepancy measure* Δ. Note that our uncertainty set with both discrepancies ensures only limited formal guarantees with respect to actual model changes. Additional empirical analyses on the relationship between our uncertainty set and actual model changes are given in Appendix E.

6 Conclusion

This paper proposed distributionally robust algorithmic recourse for tree ensembles (DiRARTE), a new framework for providing robust recourse actions against model changes in tree ensembles. We formulated the robust AR problem for tree ensembles as a distributionally robust optimization problem and showed that the robust validity constraint can be reformulated as tractable constraints with the Wasserstein distance and the χ^2-divergence. We also proposed an efficient algorithm for solving the reformulated problem by extending the existing search-based algorithm. Experimental results demonstrated that DiRARTE achieved favorable robustness-cost trade-offs relative to applicable baselines without sacrificing the computational efficiency and practical quality of the obtained actions.

Limitations Our work has several limitations that should be addressed in future research. First, Propositions 1 and 2 yield reformulations analogous to the existing results of DRO in other contexts [26, 48, 51], rather than new DRO duality results. Other discrepancy measures may provide different implications; however, how to handle them in a tractable way is non-trivial (e.g., the KL divergence requires numerically solving a one-dimensional convex dual problem). Second, despite its empirical performance, the formal guarantee of our method is limited: Proposition 4 gives only a sufficient condition for the Wasserstein distance, and the χ^2-divergence set does not formally cover support-changing updates. While we demonstrate calibration of the radius ε in Appendix E, it is empirical and assumes a retraining mechanism comparable to that used for calibration. Since it is theoretically unclear why our uncertainty set can capture actual model changes well, bridging this gap would be an important future direction. Finally, although our formulation applies to other ensembles (e.g., AdaBoost and deep ensembles), our algorithm is tailored to tree ensembles. While our primary contribution is to propose the first DRO-based framework of robust AR for tree ensembles, addressing these limitations would be important to make our framework more effective.

Acknowledgments

We wish to thank the anonymous reviewers for their insightful comments. This work was supported in part by JST ACT-X JPMJAX23C6 and JSPS KAKENHI Grant-in-Aid for Early-Career Scientists JP24K17465.

References

[1] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. Machine Bias. *ProPublica*, 2016.

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

[3] A. Ben-Tal, A. Nemirovski, and L. El Ghaoui. *Robust Optimization*. Princeton University Press, 2009.

[4] D. Bertsimas, V. Gupta, and N. Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018.

[5] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.

[6] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[7] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[8] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[9] N. Bui, D. Nguyen, and V. A. Nguyen. Counterfactual plans under distributional ambiguity. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[10] N. Bui, D. Nguyen, M.-C. Yue, and V. A. Nguyen. Coverage-validity-aware algorithmic recourse. *Operations Research*, 73(6):3294–3310, 2025.

[11] M. A. Carreira-Perpiñán and S. S. Hada. Very fast, approximate counterfactual explanations for decision forests. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 6935–6943, 2023.

[12] H. Chen, H. Zhang, S. Si, Y. Li, D. Boning, and C.-J. Hsieh. Robustness verification of tree-based models. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*, pages 12327–12337, 2019.

[13] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

[14] Z. Cui, W. Chen, Y. He, and Y. Chen. Optimal action extraction for random forests and boosted trees. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 179–188, 2015.

[15] G. De Toni, S. Teso, B. Lepri, and A. Passerini. Time can invalidate algorithmic recourse. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, pages 89–107, 2025.

[16] S. Dutta, J. Long, S. Mishra, C. Tilli, and D. Magazzeni. Robust counterfactual explanations for tree-based ensembles. In *Proceedings of the 39th International Conference on Machine Learning*, pages 5742–5756, 2022.

[17] N. Erickson, L. Purucker, A. Tschalzev, D. Holzmüller, P. M. Desai, D. Salinas, and F. Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, 2025.

[18] FICO, Google, Imperial College London, MIT, University of Oxford, UC Irvine, and UC Berkeley. Explainable Machine Learning Challenge. Available via OpenML (ID: 45554), `https://www.openml.org/d/45554`, 2018. Accessed: 2026-11-05.

[19] A. Forel, A. Parmentier, and T. Vidal. Don't explain noise: Robust counterfactuals for randomized ensembles. In *Proceedings of the 21st International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 293–309, 2024.

[20] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2001.

[21] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[22] L. Grinsztajn, E. Oyallon, and G. Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems*, pages 507–520, 2022.

[23] F. Hamman, E. Noorani, S. Mishra, D. Magazzeni, and S. Dutta. Robust counterfactual explanations for neural networks with probabilistic guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, pages 12351–12367, 2023.

[24] S. Hara and K. Hayashi. Making tree ensembles interpretable: A Bayesian model selection approach. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, pages 77–85, 2018.

[25] D. Holzmüller, L. Grinsztajn, and I. Steinwart. Better by default: Strong pre-tuned MLPs and boosted trees on tabular data. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems*, pages 26577–26658, 2024.

[26] H. Husain, V. Nguyen, and A. van den Hengel. Distributionally robust Bayesian optimization with ϕ-divergences. In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*, pages 20133–20145, 2023.

[27] J. Jiang, F. Leofante, A. Rago, and F. Toni. Formalising the robustness of counterfactual explanations for neural networks. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*, pages 14901–14909, 2023.

[28] J. Jiang, F. Leofante, A. Rago, and F. Toni. Robust counterfactual explanations in machine learning: a survey. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, pages 8086–8094, 2024.

[29] J. Jiang, L. Marzari, A. Purohit, and F. Leofante. RobustX: robust counterfactual explanations made easy. In *Proceedings of the 34th International Joint Conference on Artificial Intelligence*, pages 11067–11071, 2025.

[30] K. Kanamori, K. Kobayashi, S. Hara, and T. Takagi. Algorithmic recourse for long-term improvement. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 28849–28877, 2025.

[31] K. Kanamori, K. Kobayashi, and T. Takagi. Learning gradient boosted decision trees with algorithmic recourse. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, pages 76509–76542, 2025.

[32] K. Kanamori, T. Takagi, K. Kobayashi, and H. Arimura. DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 2855–2862, 2020.

[33] K. Kanamori, T. Takagi, K. Kobayashi, and Y. Ike. Counterfactual explanation trees: Transparent and consistent actionable recourse with decision trees. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, pages 1846–1870, 2022.

[34] K. Kanamori, T. Takagi, K. Kobayashi, and Y. Ike. Learning decision trees and forests with algorithmic recourse. In *Proceedings of the 41st International Conference on Machine Learning*, pages 22936–22962, 2024.

[35] K. Kanamori, T. Takagi, K. Kobayashi, Y. Ike, K. Uemura, and H. Arimura. Ordered counterfactual explanation by mixed-integer linear optimization. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 11564–11574, 2021.

[36] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera. A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Computing Surveys*, 55(5):1–29, 2022.

[37] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 3149–3157, 2017.

[38] M. Kelly, R. Longjohn, and K. Nottingham. The UCI machine learning repository. `https://archive.ics.uci.edu`. Accessed: 2026-11-05.

[39] A. Khouna, J. Ferry, and T. Vidal. Counterfactual maps: What they are and how to find them. *arXiv,* arXiv:2602.09128, 2026.

[40] D. Kuhn, S. Shafiee, and W. Wiesemann. Distributionally robust optimization. *Acta Numerica*, 34:579–804, 2025.

[41] N. Li, Y. Yu, and Z.-H. Zhou. Diversity regularized ensemble pruning. In *Proceedings of the 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 330–345, 2012.

[42] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *Proceedings of the 2008 IEEE International Conference on Data Mining*, pages 413–422, 2008.

[43] A. Lucic, H. Oosterhuis, H. Haned, and M. de Rijke. FOCUS: Flexible optimizable counterfactual explanations for tree ensembles. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*, pages 5313–5322, 2022.

[44] A. Malinin, B. Mlodozeniec, and M. Gales. Ensemble distribution distillation. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[45] C. Marx, F. Calmon, and B. Ustun. Predictive multiplicity in classification. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6765–6774, 2020.

[46] D. McElfresh, S. Khandagale, J. Valverde, V. P. C., G. Ramakrishnan, M. Goldblum, and C. White. When do neural nets outperform boosted trees on tabular data? In *Proceedings of the 37th Annual Conference on Neural Information Processing Systems*, pages 76336–76369, 2023.

[47] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[48] P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

[49] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2012.

[50] H. Namkoong and J. C. Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, pages 2216–2224, 2016.

[51] H. Namkoong and J. C. Duchi. Variance-based regularization with convex objectives. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 2975–2984, 2017.

[52] D. Nguyen, N. Bui, and V. A. Nguyen. Distributionally robust recourse action. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

[53] T.-D. H. Nguyen, N. Bui, D. Nguyen, M.-C. Yue, and V. A. Nguyen. Robust Bayesian recourse. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*, pages 1498–1508, 2022.

[54] E. Noorani, P. Dissanayake, F. Hamman, and S. Dutta. Counterfactual explanations for model ensembles using entropic risk measures. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pages 1566–1575, 2025.

[55] A. Parmentier and T. Vidal. Optimal counterfactual explanations in tree ensembles. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8422–8431, 2021.

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

[57] M. Pawelczyk, T. Leemann, A. Biega, and G. Kasneci. On the trade-off between actionable explanations and the right to be forgotten. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

[58] G. Peyré and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

[59] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. CatBoost: unbiased boosting with categorical features. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, pages 6639–6649, 2018.

[60] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[61] C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

[62] P. Schmidt and A. D. Witte. Predicting recidivism in North Carolina, 1978 and 1980. *Inter-university Consortium for Political and Social Research,*, 1988.

[63] I. Stępka, J. Stefanowski, and M. Lango. Counterfactual explanations with probabilistic guarantees on their robustness to model change. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1277–1288, 2025.

[64] G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 465–474, 2017.

[65] B. Turbal, I. Voitsitska, and L. Semenova. ElliCE: Efficient and provably robust algorithmic recourse via the Rashomon sets. In *Proceedings of the 39th Annual Conference on Neural Information Processing Systems*, pages 123872–123893, 2025.

[66] S. Upadhyay, S. Joshi, and H. Lakkaraju. Towards robust and reliable algorithmic recourse. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems*, pages 16926–16937, 2021.

[67] B. Ustun, A. Spangher, and Y. Liu. Actionable recourse in linear classification. In *Proceedings of the 2019 Conference on Fairness, Accountability, and Transparency*, pages 10–19, 2019.

[68] S. Verma, V. Boonsanong, M. Hoang, K. Hines, J. Dickerson, and C. Shah. Counterfactual explanations and algorithmic recourses for machine learning: A review. *ACM Computing Surveys*, 56(12), 2024.

[69] T. Vidal and M. Schiffer. Born-again tree ensembles. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9743–9753, 2020.

[70] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31:841–887, 2018.

[71] D. Wood, T. Mu, A. M. Webb, H. W. J. Reeve, M. Luján, and G. Brown. A unified theory of diversity in ensemble learning. *Journal of Machine Learning Research*, 24(359):1–49, 2023.

[72] S. Yamao, K. Kobayashi, K. Kanamori, T. Takagi, Y. Ike, and K. Nakata. Distribution-aligned sequential counterfactual explanation with local outlier factor. In *Proceedings of the 21st Pacific Rim International Conference on Artificial Intelligence*, pages 243–256, 2024.

[73] I.-C. Yeh and C.-H. Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2, Part 1):2473–2480, 2009.

[74] Y. Zhang, S. Burer, and W. N. Street. Ensemble pruning via semi-definite programming. *Journal of Machine Learning Research*, 7(48):1315–1338, 2006.

[75] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

A Related work

Algorithmic Recourse (AR) [67], also referred to as *Counterfactual Explanation* [70], has attracted increasing attention in recent years. Previous studies often assume that the predictive model is differentiable and smooth, and rely on the gradient-based methods or local linear approximations [36, 68]. However, these methods are not suitable for tree ensembles, as they are neither differentiable nor smooth. To address this issue, several techniques tailored to tree ensembles have been proposed, including mixed-integer linear optimization [14, 32, 55], search-based algorithms [11, 64], continuous relaxations [43], and so on [39]. We present an experimental comparison of these methods in Appendix E.3 and find that a search-based method, the feature tweaking (FT) algorithm [64], yields the best trade-off between scalability and action quality. Thus, we adopted the FT algorithm as the base algorithm for our method and extended it to handle our robust validity constraint. Note that our formulation is also compatible with other existing methods.

Among several extended formulations of AR in the prior studies [30, 33, 35, 72], one of the emerging topics is the *robustness* to model changes [28, 29]. While standard AR methods assume that the model does not change over time, this assumption is often violated in practice [2, 15], e.g., due to retraining on new data or changes in the data distribution [57, 60]. To maintain the validity of obtained actions under model changes, several robust AR methods have been proposed [9, 23, 27, 52, 65, 66]. One major approach is to extend the standard AR formulation via *robust optimization* [3], which optimizes the worst-case validity of actions over an uncertainty set that expresses the possible model changes. For example, ROAR [66] first obtains a linear approximation of the model around a given instance and then formulates the robust AR problem as a robust optimization problem with an uncertainty set defined by the ℓ_p-norm ball around the approximated linear model. It then solves the robust optimization problem by alternately updating the worst-case model within the uncertainty set and optimizing the action against the worst-case model. However, since most existing robust AR methods, including ROAR, assume that model changes can be characterized by perturbing model parameters, they cannot be directly applied to tree ensembles. This is because tree ensembles include discrete structures as parameters; thus, model changes cannot be expressed in a continuous parameter space like linear models or neural networks.

One exception is RobX [16], which focuses on tree ensembles and proposes a heuristic stability score to measure the robustness of actions. Given an initial action obtained by a standard AR method, it iteratively updates the action by interpolating with the nearest valid counterfactuals until its robustness is verified via the stability score. Their stability score evaluates the mean and standard deviation of predictions for perturbed instances around $\boldsymbol{x} + \boldsymbol{a}$, simulating changes in split conditions in the trees. However, as we observed in our experiments, RobX often yields overly conservative actions and results in significantly higher costs. In addition, it incurs large computational overhead and fails to find sparse actions, which may limit its practical applicability. Another exception is the work by Forel et al. [19], which focuses on the algorithmic uncertainty of randomized ensembles. It proposes verifying robustness by treating the prediction made by each classifier in the ensemble as a Bernoulli random variable. While the idea of modeling the prediction as a random variable is similar to our method, their method is applicable only to hard-voting ensembles of classification trees that follow a Bernoulli distribution. In contrast, our method can be applied to a broader class of tree ensembles, including soft-voting ensembles of regression trees, without assuming any specific distribution. Noorani et al. [54] also proposed a method for robust AR for ensemble models, but it is based on the gradient-based method and thus cannot be applied to tree ensembles. Note that there exist a few model-agnostic methods [10, 53], including BetaRobust [63]. As shown in our experiments, BetaRobust often failed to provide robust actions compared to methods designed for tree ensembles.

Our main contribution is to propose a new tractable framework of robust AR for tree ensembles from the perspective of *distributionally robust optimization (DRO)* [26, 40, 48, 50, 51]. To the best of our knowledge, Nguyen et al. [52] is the first work that applies DRO to robust AR. They assume that the model parameters follow some mixture of distributions and formulate the uncertainty set as a set of mixture distributions with bounded Gelbrich distance from the initial distribution. Then, they formulate the robust AR problem as a DRO problem that maximizes the worst-case expected validity of actions over the uncertainty set, and propose an efficient algorithm based on projected gradient descent. However, their method relies on the differentiability of the model and thus cannot be applied to tree ensembles. Compared to their work, one novelty of our work is to introduce a

new DRO formulation of the robust AR problem for tree ensembles by leveraging the *ensemble distribution* [44], which allows us to express the uncertainty set of tree ensembles in a tractable way.

B Omitted proofs

B.1 Proof of Proposition 1

Proof of Proposition 1. We consider the following optimization problem:

$$\min_{q \in \mathcal{Q}_\varepsilon(\theta|\boldsymbol{x}+\boldsymbol{a})} \mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})],$$

which is the minimization problem in the robust validity constraint in (2). By the definition of the Wasserstein distance, we can reformulate the above problem as follows:

$$\min_{q \in Q} \mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] \text{ s.t. } \mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[|\hat{y} - \hat{y}'|^p] \leq \varepsilon^p,$$

where π is a joint distribution of $\hat{y}$ and $\hat{y}'$ such that the marginal distributions of $\hat{y}$ and $\hat{y}'$ are $q_\theta(\cdot \mid \boldsymbol{x} + \boldsymbol{a})$ and q, respectively. Recall that the margin ρ is defined by $\rho(y^*, \hat{y}) = y^* \cdot \hat{y}$, and that $y^* \cdot \hat{y}' = y^* \cdot \hat{y} + y^* \cdot (\hat{y}' - \hat{y})$ holds. By the linearity of the expectation, we have

$$\mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] = \mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[y^* \cdot \hat{y}'] = \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[y^* \cdot \hat{y}] + \mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[y^* \cdot (\hat{y}' - \hat{y})].$$

While the first term is constant, the second term can be bounded as follows:

$$\mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[y^* \cdot (\hat{y}' - \hat{y})] \geq -\mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[|\hat{y} - \hat{y}'|],$$

where we used the fact that $y^* \cdot (\hat{y}' - \hat{y}) \geq -|y^*| \cdot |\hat{y}' - \hat{y}| = -|\hat{y} - \hat{y}'|$. From Jensen's inequality, we have

$$(\mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[|\hat{y} - \hat{y}'|])^p \leq \mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[|\hat{y} - \hat{y}'|^p] \leq \varepsilon^p,$$

which implies that $\mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[|\hat{y} - \hat{y}'|] \leq \varepsilon$ and thus $\mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[y^* \cdot (\hat{y}' - \hat{y})] \geq -\varepsilon$. Therefore, we have a lower bound on the objective value of the above optimization problem as follows:

$$\mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] \geq \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[y^* \cdot \hat{y}] - \varepsilon.$$

Finally, we show that the above lower bound is tight by considering a deterministic shift q such that $\hat{y}' = \hat{y} - y^* \cdot \varepsilon$ for $\hat{y} \sim q_\theta(\cdot \mid \boldsymbol{x} + \boldsymbol{a})$. We have $|\hat{y} - \hat{y}'|^p = |y^* \cdot \varepsilon|^p = \varepsilon^p$ and

$$\mathbb{E}_{\hat{y}'\sim q}[\rho(y^*, \hat{y}')] = \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[y^* \cdot (\hat{y} - y^* \cdot \varepsilon)] = \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[y^* \cdot \hat{y}] - \varepsilon,$$

which equals the above lower bound. Note that $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[y^* \cdot \hat{y}] = \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*, \hat{y})] = \rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a}))$ holds by the definition of q_θ. Combining the above results, we have

$$\min_{q \in \mathcal{Q}_\varepsilon(\theta|\boldsymbol{x}+\boldsymbol{a})} \mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] = \rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a})) - \varepsilon,$$

which completes the proof. □

B.2 Proof of Proposition 2

Proof of Proposition 2. As with the proof of Proposition 1, we consider the following optimization problem:

$$\min_{q \in \mathcal{Q}_\varepsilon(\theta|\boldsymbol{x}+\boldsymbol{a})} \mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})],$$

which is the minimization problem in the robust validity constraint in (2). Let $L(\hat{y}) = \frac{q(\hat{y})}{q_\theta(\hat{y}|\boldsymbol{x}+\boldsymbol{a})}$ be the likelihood ratio between q and $q_\theta(\cdot \mid \boldsymbol{x} + \boldsymbol{a})$. Then, we can rewrite the objective value of the above optimization problem as follows:

$$\mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] = \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[L(\hat{y}) \cdot \rho(y^*, \hat{y})],$$

and the constraints $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[L(\hat{y})] = 1$ and $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[(L(\hat{y})-1)^2] \le \varepsilon$ are imposed. Let $D(\hat{y}) = L(\hat{y})-1$ be the shifted likelihood ratio. From the constraints, we have $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})] = 0$ and $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})^2] \le \varepsilon$. Then, we have

$$
\begin{aligned}
&\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[L(\hat{y})\cdot\rho(y^*,\hat{y})] \\
&= \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})] + \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})\cdot\rho(y^*,\hat{y})] \\
&= \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})] + \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})\cdot(\rho(y^*,\hat{y}) - \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})])].
\end{aligned}
$$

The last equality holds by the fact that adding a constant does not change the value of the second term due to $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})] = 0$. By the Cauchy–Schwarz inequality, we obtain a lower bound on the second term as follows:

$$
\begin{aligned}
&\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})\cdot(\rho(y^*,\hat{y}) - \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})])] \\
&\ge -\sqrt{\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[D(\hat{y})^2]\cdot\sqrt{\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[(\rho(y^*,\hat{y}) - \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})])^2]} \\
&\ge -\sqrt{\varepsilon}\cdot\sqrt{\mathrm{Var}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})]},
\end{aligned}
$$

and the equality holds by setting $q(\hat{y}) = q_\theta(\hat{y} \mid \boldsymbol{x}+\boldsymbol{a})\cdot\left(1 - \sqrt{\varepsilon}\cdot\frac{\rho(y^*,\hat{y}) - \mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})]}{\sqrt{\mathrm{Var}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})]}}\right)$. From the assumption of ε, we can verify that q is non-negative and thus a valid probability distribution. Recall that $\mathrm{Var}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})] = {y^*}^2\cdot\mathrm{Var}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\hat{y}] = \hat{\sigma}_\theta^2(\boldsymbol{x}+\boldsymbol{a})$ and $\mathbb{E}_{\hat{y}\sim q_\theta(\cdot|\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*,\hat{y})] = \rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a}))$ hold by the definition of q_θ. Combining the above results, we have

$$
\min_{q\in\mathcal{Q}_\varepsilon(\theta|\boldsymbol{x}+\boldsymbol{a})}\ \mathbb{E}_{\hat{y}\sim q}[\rho(y^*,\hat{y})] = \rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a})) - \sqrt{\varepsilon}\cdot\hat{\sigma}_\theta(\boldsymbol{x}+\boldsymbol{a}),
$$

which completes the proof. □

B.3 Proof of Proposition 3

Proof of Proposition 3. By definition, a tree ensemble f_θ partitions the input space into a finite number of fine-grained regions, where each region is expressed as an intersection of the leaf region of each tree and corresponds to a constant weight. Let M be the number of fine-grained regions and $\phi_t\colon [M] \to [I_t]$ be the mapping from the fine-grained regions to the corresponding leaf nodes of the t-th tree. Then, a tree ensemble f_θ can be expressed as $f_\theta(\boldsymbol{x}) = \sum_{m=1}^M \frac{w_m}{T}\cdot\mathbb{I}[\boldsymbol{x}\in r_m]$, where $r_m = \bigcap_{t=1}^T r_{t,\phi_t(m)}$ is the m-th fine-grained region and $w_m = \sum_{t=1}^T w_{t,\phi_t(m)}$ is the corresponding weight. Note that since the number of trees and the maximum depth of trees are finite, the number of fine-grained regions M is also finite. From the above expression of f_θ, we can see that the prediction of f_θ is constant within each region r_m; that is, $f_\theta(\boldsymbol{x}) = \frac{w_m}{T}$ holds for any $\boldsymbol{x}\in r_m$. This fact implies that $\rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a})) = \rho(y^*, f_\theta(\boldsymbol{x}+\boldsymbol{a}'))$ and $\hat{\sigma}_\theta(\boldsymbol{x}+\boldsymbol{a}) = \hat{\sigma}_\theta(\boldsymbol{x}+\boldsymbol{a}')$ hold for any $\boldsymbol{a},\boldsymbol{a}'\in\mathbb{R}^D$ such that $\boldsymbol{x}+\boldsymbol{a}, \boldsymbol{x}+\boldsymbol{a}'\in r_m$. Combining these results, the problem (2) can be reformulated as follows:

$$
\begin{array}{ll}
\min_{m\in[M]} & \min_{\boldsymbol{a}\in\mathcal{A}(\boldsymbol{x}):\boldsymbol{x}+\boldsymbol{a}\in r_m} c(\boldsymbol{a}\mid\boldsymbol{x}) \\
\text{subject to} & \begin{cases}\rho(y^*,\hat{y}_m)\ge\varepsilon & \text{if } \Delta=\Delta_{W_p} \\ \rho(y^*,\hat{y}_m)\ge\sqrt{\varepsilon}\cdot\hat{\sigma}_m & \text{if } \Delta=\Delta_{\chi^2},\end{cases}
\end{array}
$$

where $\hat{y}_m = \frac{w_m}{T}$ and $\hat{\sigma}_m = \sqrt{\frac{1}{T}\sum_{t=1}^T (w_{t,\phi_t(m)} - \hat{y}_m)^2}$. The above problem can be solved by enumerating all the fine-grained regions r_m (in lines 1–7 of Algorithm 1 with $L = \lceil\log_2 T\rceil + 1$) and solving the inner minimization problem $\min_{\boldsymbol{a}\in\mathcal{A}(\boldsymbol{x}):\boldsymbol{x}+\boldsymbol{a}\in r_m} c(\boldsymbol{a}\mid\boldsymbol{x})$ for each region r_m (in lines 8–11 of Algorithm 1). This completes the proof. □

We remark that the inner minimization problem $\min_{\boldsymbol{a}\in\mathcal{A}(\boldsymbol{x}):\boldsymbol{x}+\boldsymbol{a}\in r_m} c(\boldsymbol{a}\mid\boldsymbol{x})$ can be easily solved for the commonly used cost functions c and constraints $\mathcal{A}$ [11]. For example, let c be the cost function that satisfies $c(\boldsymbol{a}\mid\boldsymbol{x}) = \sum_{d=1}^D c_d(a_d\mid x_d)$ and $c_d(a_d\mid x_d)\le c_d(\alpha\cdot a_d\mid x_d)$ for $\alpha\ge 1$, which includes the weighted ℓ_p-norm and TLPS. Assuming $\mathcal{A}(\boldsymbol{x}) = \mathbb{R}^D$, the inner minimization problem can be decomposed into D independent one-dimensional optimization problems

$\min_{x_d+a_d\in[l_{m,d},u_{m,d}]} c_d(a_d \mid x_d)$ for $d \in [D]$, where $l_{m,d}$ and $u_{m,d}$ are the lower and upper bounds of the d-th feature in the region r_m, respectively. This problem can be solved by the following closed-form expression: $a_d^* = \mathrm{Median}\{l_{m,d}, x_d, u_{m,d}\} - x_d$ [11]. This approach can be easily extended to handle other types of constraints, such as the bounded action set $\mathcal{A}(\boldsymbol{x}) = [l_1, u_1] \times \cdots \times [l_D, u_D]$ and sparsity constraint $\|\boldsymbol{a}\|_0 \leq k$. Note that while the inner minimization problem under such constraints may have no feasible solution for some regions, it can be easily detected by checking whether the closed-form solution to the unconstrained problem satisfies the constraints.

B.4 Proof of Proposition 4

Proof of Proposition 4. To prove the statement, we first show that the ensemble distribution of $f_{\theta'}$ at $\boldsymbol{x} + \boldsymbol{a}$ is within the uncertainty set $\mathcal{Q}_\varepsilon(\theta \mid \boldsymbol{x} + \boldsymbol{a})$ with $\Delta = \Delta_{W_1}$. Recall that Σ is the set of all one-to-one mappings $\sigma\colon [T] \to [T]$ between the trees in f_θ and those in $f_{\theta'}$. Note that the one-dimensional 1-Wasserstein distance between two empirical distributions with the same number of equally weighted points equals the average absolute difference between the points of the two distributions under the optimal mapping [58]. Thus, for any $\boldsymbol{x}' \in \mathcal{X}$, we have

$$\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}'), q_{\theta'}(\cdot \mid \boldsymbol{x}')) = \min_{\sigma\in\Sigma} \frac{1}{T} \sum_{t=1}^{T} \left| f_{\theta_t}(\boldsymbol{x}') - f_{\theta'_{\sigma(t)}}(\boldsymbol{x}') \right|.$$

Since $\boldsymbol{x} + \boldsymbol{a} \in \mathcal{X}$ holds for any $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$, our assumption implies that

$$\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x} + \boldsymbol{a}), q_{\theta'}(\cdot \mid \boldsymbol{x} + \boldsymbol{a})) \leq \max_{\boldsymbol{x}'\in\mathcal{X}} \min_{\sigma\in\Sigma} \frac{1}{T} \sum_{t=1}^{T} \left| f_{\theta_t}(\boldsymbol{x}') - f_{\theta'_{\sigma(t)}}(\boldsymbol{x}') \right| \leq \varepsilon,$$

which implies that $q_{\theta'}(\cdot \mid \boldsymbol{x} + \boldsymbol{a}) \in \mathcal{Q}_\varepsilon(\theta \mid \boldsymbol{x} + \boldsymbol{a})$ with $\Delta = \Delta_{W_1}$.

Next, we show that any feasible solution $\boldsymbol{a}$ to the problem (2) with $\Delta = \Delta_{W_p}$ for some $p \geq 1$ is valid for $f_{\theta'}$. From Proposition 1, the robust validity constraint in (2) with $\Delta = \Delta_{W_p}$ is equivalent to $\rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a})) \geq \varepsilon$, which does not depend on p. Hence, $\boldsymbol{a}$ is also a feasible solution to the problem (2) with $\Delta = \Delta_{W_1}$; that is, $\boldsymbol{a}$ satisfies $\mathbb{E}_{\hat{y}\sim q}[\rho(y^*, \hat{y})] \geq 0$ for any $q \in \mathcal{Q}_\varepsilon(\theta \mid \boldsymbol{x} + \boldsymbol{a})$ with $\Delta = \Delta_{W_1}$. Combining these facts, we have $\rho(y^*, f_{\theta'}(\boldsymbol{x} + \boldsymbol{a})) = \mathbb{E}_{\hat{y}\sim q_{\theta'}(\cdot\mid\boldsymbol{x}+\boldsymbol{a})}[\rho(y^*, \hat{y})] \geq 0$, which implies that $\boldsymbol{a}$ is valid for $f_{\theta'}$, i.e., $h_{\theta'}(\boldsymbol{x} + \boldsymbol{a}) = y^*$. This completes the proof. □

Note that the assumption of Proposition 4 is strictly weaker than the per-tree bound under the fixed matching, i.e., $\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\theta_t}(\boldsymbol{x}) - f_{\theta'_t}(\boldsymbol{x})| \leq \varepsilon$ for any $t \in [T]$. Indeed, the per-tree bound implies $\frac{1}{T}\sum_{t=1}^{T} |f_{\theta_t}(\boldsymbol{x}) - f_{\theta'_t}(\boldsymbol{x})| \leq \varepsilon$ for any $\boldsymbol{x} \in \mathcal{X}$, and thus our assumption holds since the identity mapping is in Σ. The converse does not hold in general; for example, if $\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\theta_1}(\boldsymbol{x}) - f_{\theta'_1}(\boldsymbol{x})| = T \cdot \varepsilon$ and $f_{\theta_t} = f_{\theta'_t}$ for $t \geq 2$, our assumption holds, whereas the per-tree bound is violated for $T \geq 2$ and $\varepsilon > 0$.

On the other hand, our assumption is a sufficient but not necessary condition for bounding the change in the ensemble prediction by ε. For any $\boldsymbol{x} \in \mathcal{X}$ and any joint distribution $\pi \in \Pi(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$, we have $|f_\theta(\boldsymbol{x}) - f_{\theta'}(\boldsymbol{x})| = |\mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[\hat{y} - \hat{y}']| \leq \mathbb{E}_{(\hat{y},\hat{y}')\sim\pi}[|\hat{y} - \hat{y}'|]$, which implies $|f_\theta(\boldsymbol{x}) - f_{\theta'}(\boldsymbol{x})| \leq \Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ by taking the infimum over π. The converse does not hold in general; for example, if $T = 2$, $(f_{\theta_1}(\boldsymbol{x}), f_{\theta_2}(\boldsymbol{x})) = (-1, 1)$, and $(f_{\theta'_1}(\boldsymbol{x}), f_{\theta'_2}(\boldsymbol{x})) = (0, 0)$, we have $f_\theta(\boldsymbol{x}) = f_{\theta'}(\boldsymbol{x}) = 0$ but $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x})) = 1$. Thus, Proposition 4 does not cover all the model changes under which our actions remain valid, and characterizing them more tightly is left for future work.

C Acceleration via ensemble pruning

To further accelerate our FT++, we design an ensemble pruning method that selects a subset of trees from the original ensemble $\{f_{\theta_1}, \ldots, f_{\theta_T}\}$ [41, 74]. By selecting a subset of trees, we can reduce the number of fine-grained regions and, in turn, the search space of our FT++, thereby significantly reducing runtime. Existing ensemble pruning methods often aim to select a subset of trees that can maintain the predictive performance of the original ensemble or maximize the diversity of predictions among the selected trees [41, 74]. However, such methods are not suitable for our purpose, as we need

to select a subset of trees while ensuring that the search space of the fine-grained regions extracted from those trees remains similar to that of the original ensemble. That is, we need to select a subset of diverse trees such that the search space induced by them covers the original search space as much as possible.

To this end, we design a diversity score that measures the dissimilarity between trees based on how they partition the input space. Because computing such a diversity score exactly is computationally expensive, we propose a proxy score that can be efficiently computed using the training data. Let $\Phi_t \colon \mathcal{X} \to [I_t]$ be the function that maps an input instance to the corresponding leaf node that the instance falls into. It enables us to interpret each tree f_{θ_t} as partitioning the training data into I_t clusters, where each cluster corresponds to a leaf node. Thus, we can measure the dissimilarity between two trees f_{θ_t} and $f_{\theta_{t'}}$ by measuring the dissimilarity between their clustering results. Given a training sample $X \subseteq \mathcal{X}$, let $\mathrm{dissim}(\Phi_t, \Phi_{t'} \mid X)$ be a dissimilarity score between the clustering results of f_{θ_t} and $f_{\theta_{t'}}$ on the training sample X. While we can use any clustering dissimilarity score as dissim, we used the adjusted Rand index in our experiments. Because the adjusted Rand index is a similarity score, we define the dissimilarity score as one minus the adjusted Rand index. Then, we define the diversity score of a subset of trees $\mathcal{S} \subseteq [T]$ as follows:

$$\mathrm{div}(\mathcal{S} \mid X) = \frac{1}{|\mathcal{S}| \cdot (|\mathcal{S}| - 1)} \sum_{t,t' \in \mathcal{S} : t \neq t'} \mathrm{dissim}(\Phi_t, \Phi_{t'} \mid X).$$

Given a target number of trees $T' < T$, we can select a subset of trees $\mathcal{S} \subseteq [T]$ with $|\mathcal{S}| = T'$ by maximizing the diversity score $\mathrm{div}(\mathcal{S} \mid X)$. Because this task is computationally expensive due to the combinatorial nature of the problem, we employ the following greedy algorithm to approximately solve the problem:

1. Select two trees that have the highest dissimilarity score:
$$(t_1, t_2) = \operatorname*{arg\,max}_{t,t' \in [T] : t \neq t'} \mathrm{dissim}(\Phi_t, \Phi_{t'} \mid X).$$
2. Initialize the selected subset of trees as $\mathcal{S} = \{t_1, t_2\}$.
3. Select a tree that maximizes our diversity score by adding it to $\mathcal{S}$:
$$t^* = \operatorname*{arg\,max}_{t \in [T] \setminus \mathcal{S}} \mathrm{div}(\mathcal{S} \cup \{t\} \mid X).$$
4. Update $\mathcal{S} \leftarrow \mathcal{S} \cup \{t^*\}$ and repeat the above step until $|\mathcal{S}| = T'$.

In our experiments, we applied the above pruning method only when using random forests as classifiers, as we observed that the runtime of our FT++ was higher with random forests than with XGBoost, even when the number of trees was the same. This is because XGBoost often results in an ensemble of shallow trees due to the regularization of the boosting process, while random forests often result in an ensemble of deep trees with a predefined maximum depth. Note that our pruning method can be applied to any tree ensemble and is effective even for XGBoost, which will be demonstrated by our experimental results (see Table 9).

D Implementation details

D.1 RobX

RobX is a robust AR method for tree ensembles [16]. A key idea of RobX is to design a stability score that measures the robustness of an action to model changes. For an action $\boldsymbol{a}$ for $\boldsymbol{x}$, let $N_K(\boldsymbol{x} + \boldsymbol{a})$ be a set of K randomly perturbed instances around $\boldsymbol{x} + \boldsymbol{a}$. Then, the stability score is defined as

$$R(\boldsymbol{x} + \boldsymbol{a}) = \frac{1}{K} \sum_{\boldsymbol{x}' \in N_K(\boldsymbol{x}+\boldsymbol{a})} f_\theta(\boldsymbol{x}') - \sqrt{\frac{1}{K} \sum_{\boldsymbol{x}' \in N_K(\boldsymbol{x}+\boldsymbol{a})} \left(f_\theta(\boldsymbol{x}') - \frac{1}{K} \sum_{\boldsymbol{x}' \in N_K(\boldsymbol{x}+\boldsymbol{a})} f_\theta(\boldsymbol{x}') \right)^2}.$$

It evaluates the mean and standard deviation of the predictions for the perturbed instances $N_K(\boldsymbol{x}+\boldsymbol{a})$. Such perturbations can be interpreted as the changes in the branching rules of each tree, which

simulate the changes in the tree ensemble f_θ. RobX aims to find an action $\boldsymbol{a}$ that minimizes the cost $c(\boldsymbol{a} \mid \boldsymbol{x})$ while ensuring that the stability score $R(\boldsymbol{x} + \boldsymbol{a})$ is above a certain threshold τ.

To obtain an action $\boldsymbol{a}$ that satisfies the above constraint, RobX assumes access to an initial solution $\boldsymbol{a}_0$ obtained by the standard AR method and a set of "conservative counterfactuals" $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_C\}$ such that $R(\boldsymbol{x}_c) \geq \tau$ holds for any $c \in [C]$. After obtaining an initial action $\boldsymbol{a}_0$, RobX generates a set of interpolated instances $\boldsymbol{x}'$ between $\boldsymbol{x} + \boldsymbol{a}_0$ and each conservative counterfactual $\boldsymbol{x}_c$ and verifies the stability score of each interpolated instance. It then selects the interpolated instance with the minimum cost among those with $R(\boldsymbol{x}') \geq \tau$ and returns the action $\boldsymbol{a} = \boldsymbol{x}' - \boldsymbol{x}$ as the final solution. Note that the original paper [16] does not take into account the actionability constraints in the interpolation step, which may lead to infeasible actions. In our implementation, we modified the interpolation step to ensure that the resulting action satisfies the actionability constraints (e.g., immutable features are not changed during interpolation).

Following the original paper [16], we generated $K = 1000$ perturbed instances by adding Gaussian noise with a standard deviation of $0.1 \cdot \sigma_d$, where σ_d is the standard deviation of the d-th feature in the training data. We set $C = 10$ and generated 10 interpolated instances for each conservative counterfactual. Based on the results of our preliminary experiments, we set the range of the threshold τ to be $\{0.5, 0.55, 0.6, 0.65, 0.7\}$ for XGBoost and $\{0.5, 0.525, 0.55, 0.575, 0.6\}$ for random forests. These ranges were determined such that RobX could find feasible solutions for at least some of the test instances. Note that when we set δ larger than the above range, RobX often failed to find feasible solutions for most of the test instances.

D.2 BetaRobust

BetaRobust is a model-agnostic robust AR method based on the beta distribution [63]. As with RobX, BetaRobust also introduces a verification method that assesses the robustness of an action to model changes. It assumes access to the space of "admissible models" $\mathcal{M}$, which is the probabilistic distribution over all models that are the results of a complete retraining of the original model f_θ using arbitrary settings from the predefined set of model changes. For an action $\boldsymbol{a}$ for $\boldsymbol{x}$, BetaRobust samples a set of classifiers $\{h_1, \ldots, h_K\}$ from $\mathcal{M}$ and computes whether the prediction of each sampled classifier h_k is valid (i.e., $\mathbb{I}[h_k(\boldsymbol{x} + \boldsymbol{a}) = y^*]$). Then, it fits a beta distribution $B(a, b)$ to the indicators of the validity over the sampled classifiers and verifies the robustness of $\boldsymbol{a}$ by checking whether $F^{-1}_{B(a,b)}(1 - \alpha) \geq \delta$ holds, where $F^{-1}_{B(a,b)}$ is the inverse cumulative distribution function of the fitted beta distribution, and α is a confidence level. This constraint ensures that the probability of the action $\boldsymbol{a}$ being valid with respect to the admissible models is at least δ with probability α. BetaRobust aims to find an action $\boldsymbol{a}$ that minimizes the cost $c(\boldsymbol{a} \mid \boldsymbol{x})$ while ensuring that the above constraint is satisfied.

We implemented BetaRobust by directly integrating the robustness constraint into Algorithm 1. Following the original paper [63], we set $K = 30$ and $\alpha = 0.95$. Based on our preliminary experiments, we set the threshold δ to $\{0.7, 0.75, 0.8, 0.85, 0.9\}$ so BetaRobust could find feasible solutions for at least some test instances. Note that when we set δ outside the above ranges, BetaRobust often failed to find feasible solutions for most test instances. To generate the set of admissible models, we retrained the original model by randomly varying the number of trees and the maximum tree depth, as the original paper reports that this strategy performed the best in the experiments [63].

D.3 DiRARTE

Both DiRARTE-W and DiRARTE-χ^2 have the same hyperparameter ε that controls the radius of the uncertainty set. For DiRARTE-W, we set ε to be $\{1.5, 2.0, 2.5, 3.0, 3.5\}$ for XGBoost and $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ for random forests. For DiRARTE-χ^2, we set ε to be $\{0.02, 0.04, 0.06, 0.08, 0.1\}$ for XGBoost and $\{0.04, 0.08, 0.12, 0.16, 0.2\}$ for random forests. These ranges were determined based on the results of our preliminary experiments, and we observed that they performed well across all datasets.

In our implementation, each tree f_{θ_t} of a random forest outputs the predicted probability of the positive class (i.e., soft voting), and the ensemble prediction $f_\theta(\boldsymbol{x})$ is obtained by averaging the predicted probabilities $f_{\theta_t}(\boldsymbol{x})$ of each tree and applying a threshold of 0.5. To align with our formulation, for DiRARTE-W, we transform the averaged probability into log-odds via the logit function and use it as a prediction score to compute the margin. For DiRARTE-χ^2, we add a constant

term -0.5 when computing the margin; that is, $\rho(y^*, f_\theta(\boldsymbol{x} + \boldsymbol{a}) - 0.5)$. Note that this modification does not change the prediction of the model and thus does not affect the validity of actions.

E Additional experimental results

We present the complete results of our experiments. Tables 2 to 6 summarize the details of the datasets used in our experiments. We also present the following additional results: (a) ablation studies on our FT++ algorithm; (b) experimental comparisons between AR methods for tree ensembles; (c) empirical analyses of the relationship between our uncertainty set and actual model changes; (d) empirical analyses of a data-driven calibration of the radius ε, respectively.

E.1 Complete results on the baseline comparison

Figures 4 and 5 show the complete results on the trade-off between the robust validity and the average cost. Tables 7 and 8 show the complete results on the average sparsity, plausibility, and runtime of each method. Note that we applied our ensemble pruning method only when the classifier was a random forest, as mentioned in Appendix C.

E.2 Ablation study on FT++ algorithm

To evaluate the efficacy of our algorithmic design for FT++, we conducted an ablation study to investigate the effects of the multi-level merging strategy and the ensemble pruning method.

Multi-level merging First, we investigated the effect of the merging level L in our FT++ algorithm for solving the problem (2). We varied L within $\{1, 2\}$ and measured the ratio of test instances for which we could find a feasible solution to the problem (2). Figures 6 and 7 show the results. We observed that the original FT algorithm with $L = 1$ sometimes failed to find feasible solutions, especially on the FICO dataset, which includes many more numerical features than the other datasets. On the other hand, our FT++ with $L = 2$ could find feasible solutions for almost all the instances across all datasets, which indicates that the multi-level merging strategy is effective in enlarging the search space and finding feasible solutions to the problem (2). Note that increasing the merging level to $L = 3$ did not further improve the results, while it significantly increased the runtime. Thus, we set $L = 2$ for all the methods in our main experiments.

Ensemble pruning Next, we investigated the effect of ensemble pruning on the performance of our FT++. We set the merging level L to 2, and applied our ensemble pruning method to reduce the number of trees in the ensemble to $T' = 50$. Figures 8 and 9 present the results on the trade-off between the robust validity and the average cost. We can see that the performance of each method did not change significantly after ensemble pruning. Table 9 shows the results on the average runtime. We observed that ensemble pruning significantly reduced the runtime of our FT++, especially for random forests. These results indicate that our ensemble pruning method effectively accelerated FT++ without significantly sacrificing performance.

E.3 Comparison between standard AR methods for tree ensembles

To investigate the behavior of existing AR methods for tree ensembles, we compared standard AR methods. For that purpose, we implemented the original FT algorithm [64], optimal action extraction (OAE) via mixed-integer linear programming [14, 32], and FOCUS based on continuous relaxation [43]. We split each dataset into training and test sets at an $80 : 20$ ratio, and trained XGBoost and random forest classifiers on the training sets. We set $T = 100$ and limited the tree depth to 6. Then, we obtained recourse actions for 20 randomly selected test instances with each method and measured the average validity, cost, and runtime of the obtained actions. We implemented OAE with the SCIP solver, a state-of-the-art non-commercial mixed-integer programming solver, and imposed a time limit of 180 seconds to improve computational efficiency. Because FOCUS has the trade-off parameter β that controls the balance between the cost and the validity of actions, we varied β within $\{0.001, 0.01, 0.1\}$, and fixed the maximum number of iterations in gradient descent to 100. As the cost function c, we used the weighted ℓ_1-norm for this experiment because FOCUS does not support TLPS. We repeated the above procedure with 5 different splits and report the average results.

Table 10 shows the results. We observed that OAE achieved the lowest cost in the German and COMPAS datasets (and Bail with XGBoost), whereas it yielded much higher costs than FT and FT++ in the FICO and Credit datasets. This result indicates that OAE could not scale well to large datasets, especially those with many numerical features. In addition, FOCUS often failed to balance the costs and validity of actions, even by varying the trade-off parameter β. Note that increasing the maximum number of iterations in FOCUS could not improve the results, while it significantly increased the runtime. In contrast, FT and FT++ could stably find low-cost valid actions with the fastest runtime among the methods, across datasets. For these reasons, we chose FT as the base algorithm.

E.4 Relationship between our uncertainty set and actual model changes

To examine whether our uncertainty set captures the model changes arising from realistic retraining, we investigated the relationship between the discrepancy in the ensemble distributions and the actual change in the model output. For an initial model f_θ and a shifted model $f_{\theta'}$ obtained in each scenario described in Section 5, we measured the absolute difference in their ensemble outputs $|f_\theta(\boldsymbol{x}) - f_{\theta'}(\boldsymbol{x})|$ and the 1-Wasserstein distance between their ensemble distributions $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ for each test instance $\boldsymbol{x}$. While the former quantifies the degree of the actual model change [23], the latter is our proxy for measuring the model change, which determines whether the ensemble distribution of $f_{\theta'}$ lies within our uncertainty set (see Proposition 4). We used XGBoost and random forests, the five datasets, and the three scenarios, and repeated the procedure with 10 different data splits for each setting, resulting in a total of $2 \times 5 \times 3 \times 10 = 300$ settings. To evaluate the change from different perspectives, we also computed the average and the maximum of each quantity over the test instances for each setting, and measured the correlation coefficient between the two quantities per instance as well as for the average and the maximum.

We observed that the correlation coefficients between the two quantities were 0.89 per test instance, 0.98 for the average, and 0.9996 for the maximum; that is, they were highly correlated at both the per-instance and the aggregate levels. These results indicate that the discrepancy in the ensemble distributions closely tracks the actual change in the model output, and thus the ensemble distribution of a shifted model $f_{\theta'}$ suggest that the ensemble distribution of a shifted model can be captured by our uncertainty set with a radius ε proportional to the actual change in the model output. In particular, the high correlation for the maximum is relevant to the uniform condition over $\mathcal{X}$ assumed in Proposition 4. While these results do not provide a formal guarantee, they suggest that our uncertainty set can capture the model changes arising from realistic retraining.

Under the same setup, we also measured the maximum per-tree change under the fixed matching, i.e., $\max_{t \in [T]} |f_{\theta_t}(\boldsymbol{x}) - f_{\theta'_t}(\boldsymbol{x})|$, which must be bounded by ε if we assume the per-tree bound discussed in Appendix B instead of our assumption. We observed that this quantity was, on average, 6.4 times larger than the maximum difference in the ensemble outputs over the test instances, i.e., $\max_{\boldsymbol{x}} |f_\theta(\boldsymbol{x}) - f_{\theta'}(\boldsymbol{x})|$. This result indicates that the per-tree bound under the fixed matching forces the radius ε to be far larger than the actual change in the model output, i.e., it is overly pessimistic, which supports our assumption based on the best matching in Proposition 4.

E.5 Data-driven calibration of the radius

In practice, we need to determine the radius ε before a future shifted model is observed. A general approach to determining ε in a data-driven manner is bootstrap-based calibration [4]. Specifically, we first obtain several pairs of the current model f_θ and a shifted model $f_{\theta'}$, either from (1) the past history of model updates or (2) simulating distribution shifts over the current dataset, as we did in our experiments. For each pair, we compute the discrepancy $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ for each instance $\boldsymbol{x}$ in the dataset, which yields an empirical distribution of the discrepancies, i.e., the estimated degree of the model change over the input space. Then, we set ε to the α-quantile of this empirical distribution, where $\alpha \in (0, 1)$ is a target coverage, i.e., the fraction of users for whom the model owner wants to ensure robustness; for example, to ensure robustness for 50% of users, we can set ε to the 50% quantile. This strategy allows practitioners to specify an interpretable target coverage α instead of tuning ε directly.

To verify whether the uncertainty set calibrated in this way actually includes unseen shifted models, we conducted a held-out coverage analysis. We used the XGBoost models, five datasets, and the

three scenarios as in Appendix E.4, and performed the following procedure for each pair of a dataset and a scenario:

1. We fixed an initial model f_θ and generated 20 shifted models $f_{\theta'}$ by retraining under the scenario with different random seeds.
2. For the first 10 shifted models (calibration models), we computed $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ for each test instance $\boldsymbol{x}$ and set ε_α to the α-quantile of these discrepancies.
3. For the remaining 10 shifted models (held-out models), we computed the same discrepancy for each test instance $\boldsymbol{x}$ and measured the coverage of ε_α, i.e., the fraction of the pairs of test instances and held-out models with $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x})) \leq \varepsilon_\alpha$.

We repeated this procedure 5 times for each setting and averaged the results for each calibration level $\alpha \in \{0.1, 0.2, \ldots, 0.9\}$. Since the three scenarios showed similar trends, we report the results averaged over them.

Note that the above analysis evaluates the marginal coverage over the pairs of instances and models, whereas Proposition 4 requires the uniform condition over the input space, i.e., $\max_{\boldsymbol{x} \in \mathcal{X}} \Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x})) \leq \varepsilon$. To evaluate the coverage in this uniform form, we also conducted a variant of the above analysis. For each shifted model, we computed the maximum discrepancy $\max_{\boldsymbol{x}} \Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x}))$ over the test instances, set ε_α to the α-quantile of the maximum discrepancies of the 10 calibration models, and measured the coverage as the fraction of the held-out models whose maximum discrepancy is at most ε_α.

Table 11 shows the results. In the marginal form, the coverage closely matched the corresponding calibration level α, with deviations within 0.01 for all the datasets. That is, the radius ε_α, which was calibrated without access to the held-out models, captured the unseen shifted models at the target rate α. In the uniform form, the coverage also tracked α, though less tightly than in the marginal form; the deviation from α was within about 0.03 in either direction for all the datasets except German, our smallest dataset, where the coverage was consistently lower than α (e.g., 0.832 for $\alpha = 0.9$). One possible reason for this looser fit is that ε_α is estimated from only ten per-model maximum discrepancies, which makes the quantile coarse.

We acknowledge that this validation is empirical and has several limitations. First, it assumes that the shifted models used for calibration and evaluation are generated by a comparable retraining mechanism, and the coverage may not transfer to model changes caused by a different mechanism. Second, the maximum discrepancy over finitely many test instances is only a lower bound on the supremum over the input space $\mathcal{X}$ required by Proposition 4, and thus the calibrated radius may be optimistic. One way to alleviate the latter is data augmentation via Algorithm 1: for each test instance $\boldsymbol{x}$, Algorithm 1 generates feasible actions $\boldsymbol{a} \in \mathcal{A}(\boldsymbol{x})$ in its search procedure, and every $\boldsymbol{x} + \boldsymbol{a}$ lies in $\mathcal{X}$ by definition, so the discrepancy can also be measured at these points. However, it remains an empirical procedure without a formal guarantee.

F Additional comments on existing assets

scikit-learn[1] 1.7.2 is publicly available under the BSD-3-Clause license. XGBoost[2] 1.7.4 is publicly available under the Apache-2.0 license. Numba[3] 0.64.0 is publicly available under the BSD-2-Clause license. PySCIPOpt[4] 6.1.0 is publicly available under the MIT license. All the scripts and datasets used in our experiments are available in the GitHub repository at `https://github.com/kelicht/dirarte`.

All the datasets used in Section 5 are publicly available and do not contain any identifiable information or offensive content. As they are accompanied by appropriate citations in the main body, see the corresponding references for more details.

[1] `https://scikit-learn.org/`
[2] `https://xgboost.readthedocs.io/`
[3] `https://numba.pydata.org/`
[4] `https://pyscipopt.readthedocs.io/`

G Discussion on potential societal impacts

Our proposed method, named DiRARTE, is a new framework for robust algorithmic recourse (AR) that can be applied to tree ensembles. As demonstrated in our experiments, our method can identify low-cost, valid actions that are robust to model changes, which can be beneficial for individuals affected by machine learning model decisions. For example, in the context of credit scoring, our method can provide individuals with actionable recommendations to improve their credit scores, while remaining robust to changes in the credit scoring model, helping them access better financial opportunities. It can also be applied to other domains, such as healthcare and criminal justice, where algorithmic decisions can have significant impacts on individuals' lives. Our method has the potential to contribute to the development of fairer and more transparent machine learning systems that can provide actionable recommendations to individuals.

On the other hand, our framework also has potential societal impacts that need careful consideration. For instance, malicious users may exploit our method to find actions that manipulate model predictions in unintended ways, leading to negative consequences. Note that this risk is not specific to our method but is a common issue for all AR methods. To mitigate this risk, it is important to implement appropriate safeguards during the deployment of our method, such as monitoring the actions it generates. Considering the plausibility of actions can also help mitigate this risk by preventing the generation of unrealistic actions. Our framework can incorporate plausibility constraints, and we encourage practitioners to use them when deploying our method in real-world applications.

In addition, the robustness of our actions is formally guaranteed only for model changes covered by our uncertainty set (Proposition 4), and our calibration of the radius ε is empirical (Appendix E.5). Thus, our actions may become invalid if an actual model change is not covered, e.g., when it is caused by a mechanism different from the one assumed for calibration. Presenting such actions as guaranteed to remain valid may mislead individuals; we encourage practitioners to communicate this limitation to them and to monitor the validity of the provided actions after model updates.

Overall, the proposed method has the potential to significantly improve the trustworthiness of the decision-making process, but we need careful consideration of its risks before incorporating it into the actual decision-making process.

Table 2: Summary of the FICO dataset ($N = 9871, D = 23$).

Feature	Type	Min	Max	Immutable	Constraint
ExternalRiskEstimate	Integer	0.0	94.0	No	Nothing
MSinceOldestTradeOpen	Integer	0.0	803.0	No	Nothing
MSinceMostRecentTradeOpen	Integer	0.0	383.0	No	Nothing
AverageMInFile	Integer	4.0	383.0	No	Nothing
NumSatisfactoryTrades	Integer	0.0	79.0	No	Nothing
NumTrades60Ever2DerogPubRec	Integer	0.0	19.0	Yes	Fix
NumTrades90Ever2DerogPubRec	Integer	0.0	19.0	Yes	Fix
PercentTradesNeverDelq	Integer	0.0	100.0	No	Nothing
MSinceMostRecentDelq	Integer	0.0	83.0	No	Nothing
MaxDelq2PublicRecLast12M	Integer	0.0	9.0	Yes	Fix
MaxDelqEver	Integer	2.0	8.0	Yes	Fix
NumTotalTrades	Integer	0.0	104.0	No	Nothing
NumTradesOpeninLast12M	Integer	0.0	19.0	No	Nothing
PercentInstallTrades	Integer	0.0	100.0	No	Nothing
MSinceMostRecentInqexcl7days	Integer	0.0	24.0	No	Nothing
NumInqLast6M	Integer	0.0	66.0	No	Nothing
NumInqLast6Mexcl7days	Integer	0.0	66.0	No	Nothing
NetFractionRevolvingBurden	Integer	0.0	232.0	No	Nothing
NetFractionInstallBurden	Integer	0.0	471.0	No	Nothing
NumRevolvingTradesWBalance	Integer	0.0	32.0	No	Nothing
NumInstallTradesWBalance	Integer	0.0	23.0	No	Nothing
NumBank2NatlTradesWHighUtilization	Integer	0.0	18.0	No	Nothing
PercentTradesWBalance	Integer	0.0	100.0	No	Nothing

Table 3: Summary of the Credit dataset ($N = 30000, D = 13$).

Feature	Type	Min	Max	Immutable	Constraint
Married	Binary	0.0	1.0	Yes	Fix
Single	Binary	0.0	1.0	Yes	Fix
Age	Integer	21.0	79.0	No	Irreducible
EducationLevel	Integer	0.0	3.0	No	Nothing
MaxBillAmountOverLast6Months	Integer	0.0	50810.0	No	Nothing
MaxPaymentAmountOverLast6Months	Integer	0.0	51430.0	No	Nothing
MonthsWithZeroBalanceOverLast6Months	Integer	0.0	6.0	No	Nothing
MonthsWithLowSpendingOverLast6Months	Integer	0.0	6.0	No	Nothing
MonthsWithHighSpendingOverLast6Months	Integer	0.0	6.0	No	Nothing
MostRecentBillAmount	Integer	0.0	29450.0	No	Nothing
MostRecentPaymentAmount	Integer	0.0	26670.0	No	Nothing
TotalOverdueCounts	Integer	0.0	3.0	No	Nothing
TotalMonthsOverdue	Integer	0.0	36.0	No	Nothing

Table 4: Summary of the German dataset ($N = 1000, D = 11$).

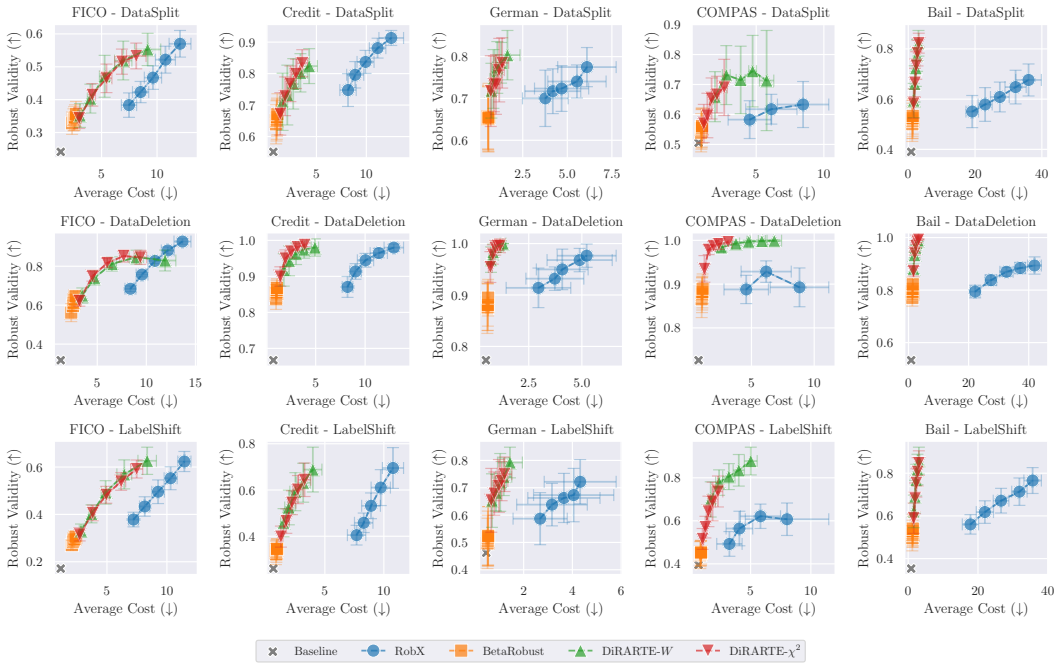
Feature	Type	Min	Max	Immutable	Constraint
Age	Integer	19.0	75.0	No	Irreducible
Sex	Binary	0.0	1.0	Yes	Fix
Job	Integer	0.0	3.0	No	Nothing
Housing	Binary	0.0	1.0	Yes	Fix
Saving accounts	Integer	0.0	4.0	No	Nothing
Checking account	Integer	0.0	3.0	No	Nothing
Credit amount	Integer	250.0	18424.0	No	Nothing
Duration	Integer	4.0	72.0	No	Nothing
Purpose:investment	Binary	0.0	1.0	No	Nothing
Purpose:leisure	Binary	0.0	1.0	No	Nothing
Purpose:personal	Binary	0.0	1.0	No	Nothing

Table 5: Summary of the COMPAS dataset ($N = 7214, D = 15$).

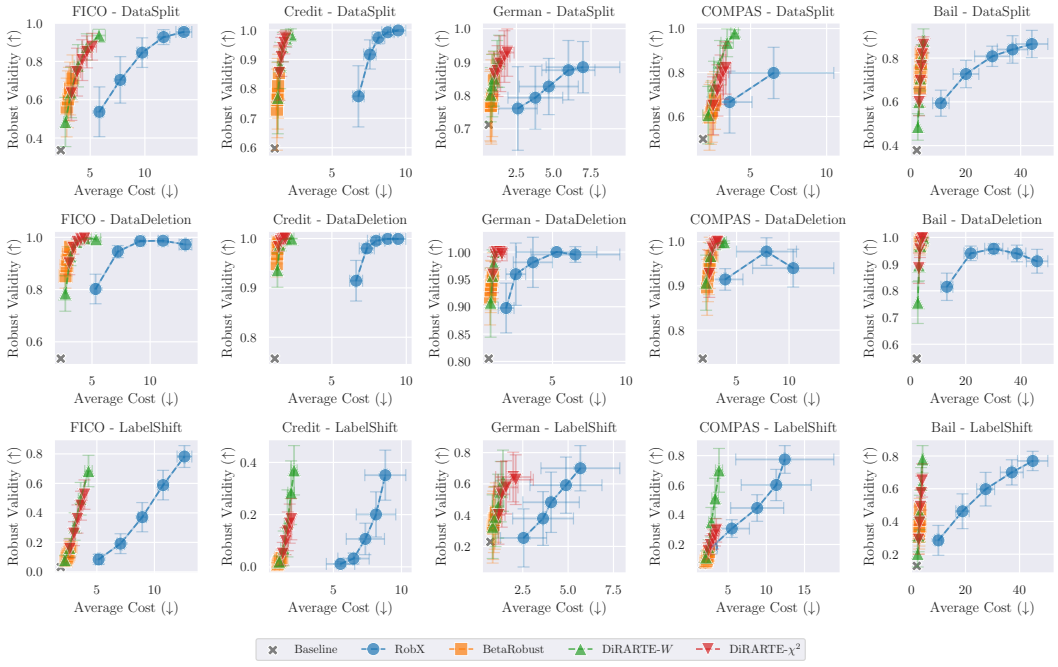
Feature	Type	Min	Max	Immutable	Constraint
age	Integer	18.0	96.0	No	Irreducible
juv_fel_count	Integer	0.0	20.0	No	Nothing
juv_misd_count	Integer	0.0	13.0	No	Nothing
juv_other_count	Integer	0.0	17.0	No	Nothing
priors_count	Integer	0.0	38.0	No	Nothing
sex:Female	Binary	0.0	1.0	Yes	Fix
sex:Male	Binary	0.0	1.0	Yes	Fix
race:African-American	Binary	0.0	1.0	Yes	Fix
race:Asian	Binary	0.0	1.0	Yes	Fix
race:Caucasian	Binary	0.0	1.0	Yes	Fix
race:Hispanic	Binary	0.0	1.0	Yes	Fix
race:Native-American	Binary	0.0	1.0	Yes	Fix
race:Other	Binary	0.0	1.0	Yes	Fix
c_charge_degree:F	Binary	0.0	1.0	No	Nothing
c_charge_degree:M	Binary	0.0	1.0	No	Nothing

Table 6: Summary of the Bail dataset ($N = 8923, D = 16$).

Feature	Type	Min	Max	Immutable	Constraint
White	Binary	0.0	1.0	Yes	Fix
Alchy	Binary	0.0	1.0	No	Nothing
Junky	Binary	0.0	1.0	No	Nothing
Super	Binary	0.0	1.0	No	Nothing
Married	Binary	0.0	1.0	Yes	Fix
Felon	Binary	0.0	1.0	No	Nothing
Workrel	Binary	0.0	1.0	No	Nothing
Propty	Binary	0.0	1.0	No	Nothing
Person	Binary	0.0	1.0	No	Nothing
Male	Binary	0.0	1.0	Yes	Fix
Priors	Integer	0.0	40.0	No	Nothing
School	Integer	1.0	19.0	No	Nothing
Rule	Integer	0.0	39.0	No	Nothing
Age	Integer	15.0	72.0	No	Nothing
Tservd	Integer	1.0	287.0	No	Nothing
Follow	Integer	46.0	57.0	No	Nothing

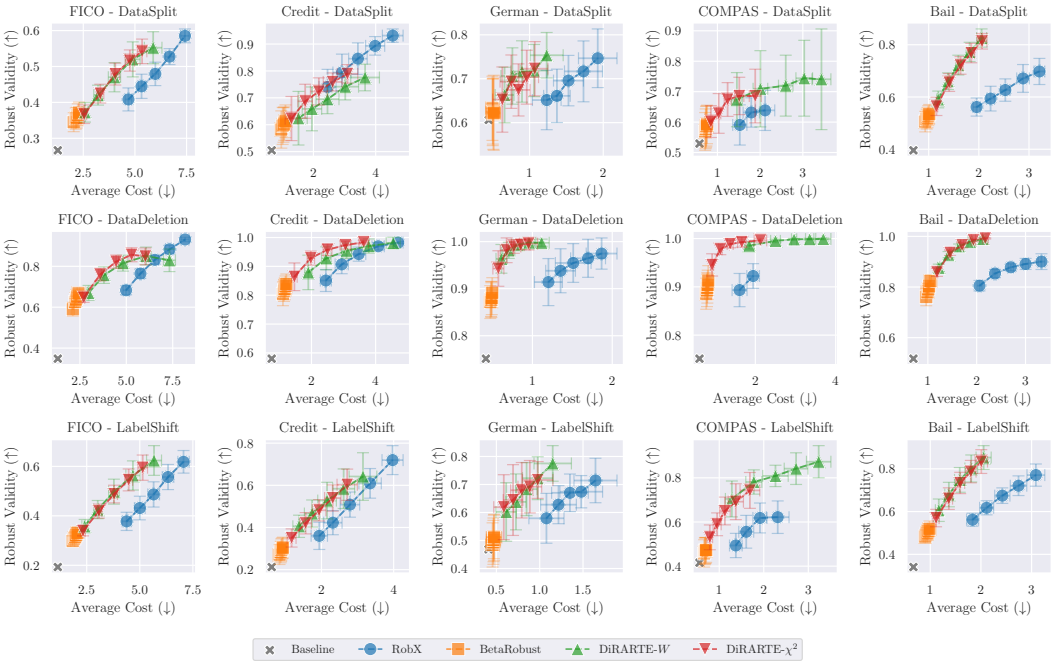


(a) XGBoost

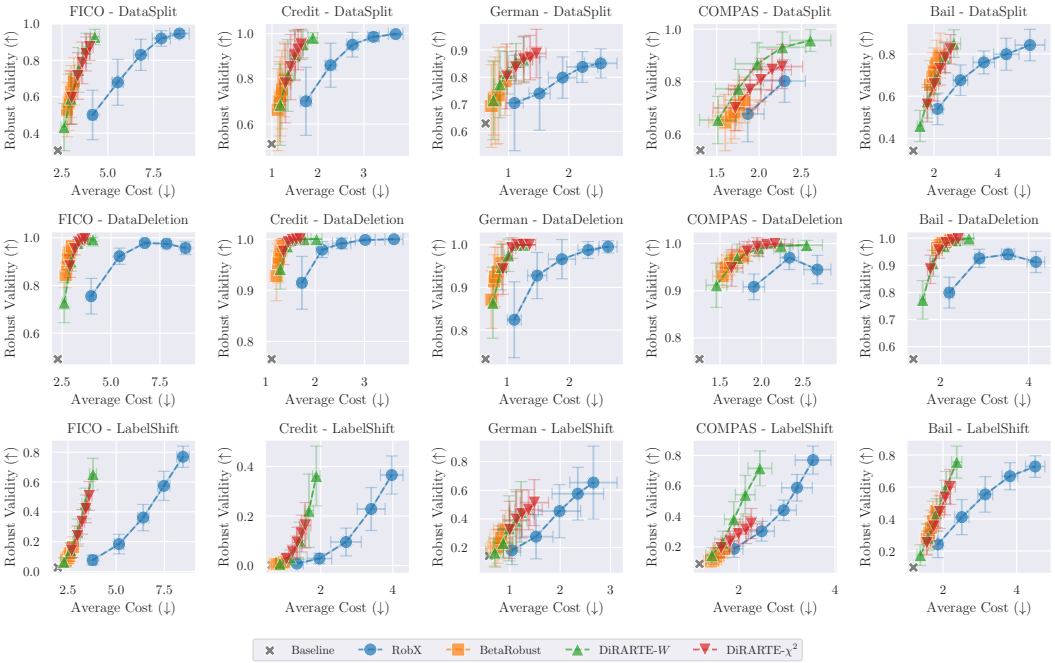


(b) Random Forest

Figure 4: Experimental results on TLPS with respect to the robust validity (higher is better) and the average cost (lower is better). Error bars indicate the standard deviation across 10 different splits.



(a) XGBoost



(b) Random Forest

Figure 5: Experimental results on weighted ℓ_1-norm with respect to the robust validity (higher is better) and the average cost (lower is better). Error bars indicate the standard deviation across 10 different splits.

Table 7: Average sparsity, plausibility, and runtime [s] of each method with TLPS (lower is better).

(a) XGBoost

	Method	FICO	Credit	German	COMPAS	Bail
Sparsity	Baseline	1.779 ± 0.05	1.122 ± 0.04	1.049 ± 0.03	1.118 ± 0.03	1.330 ± 0.03
	RobX	15.20 ± 0.13	6.527 ± 0.26	3.859 ± 0.18	2.238 ± 0.23	5.574 ± 0.40
	BetaRobust	2.398 ± 0.11	1.247 ± 0.07	1.090 ± 0.05	1.209 ± 0.04	1.508 ± 0.04
	DiRARTE-W	3.536 ± 0.55	2.176 ± 0.40	1.434 ± 0.21	1.678 ± 0.12	1.976 ± 0.19
	DiRARTE-χ^2	3.485 ± 0.53	1.962 ± 0.34	1.392 ± 0.18	1.460 ± 0.15	2.002 ± 0.22
Plausibility	Baseline	0.453 ± 0.00	0.509 ± 0.01	0.499 ± 0.01	0.452 ± 0.00	0.494 ± 0.00
	RobX	0.438 ± 0.00	0.482 ± 0.01	0.495 ± 0.01	0.447 ± 0.01	0.497 ± 0.01
	BetaRobust	0.452 ± 0.01	0.512 ± 0.01	0.499 ± 0.01	0.451 ± 0.01	0.496 ± 0.00
	DiRARTE-W	0.458 ± 0.01	0.524 ± 0.01	0.499 ± 0.01	0.475 ± 0.02	0.502 ± 0.00
	DiRARTE-χ^2	0.453 ± 0.00	0.514 ± 0.01	0.498 ± 0.01	0.452 ± 0.01	0.501 ± 0.00
Time [s]	Baseline	0.071 ± 0.01	0.052 ± 0.01	0.019 ± 0.00	0.027 ± 0.00	0.053 ± 0.01
	RobX	0.357 ± 0.01	0.227 ± 0.01	0.178 ± 0.01	0.222 ± 0.00	0.261 ± 0.01
	BetaRobust	0.176 ± 0.01	0.125 ± 0.01	0.040 ± 0.01	0.047 ± 0.01	0.117 ± 0.01
	DiRARTE-W	0.073 ± 0.01	0.052 ± 0.01	0.019 ± 0.00	0.027 ± 0.00	0.053 ± 0.01
	DiRARTE-χ^2	0.102 ± 0.01	0.070 ± 0.01	0.023 ± 0.01	0.031 ± 0.00	0.069 ± 0.01

(b) Random Forest

	Method	FICO	Credit	German	COMPAS	Bail
Sparsity	Baseline	2.133 ± 0.15	1.221 ± 0.08	1.069 ± 0.06	1.300 ± 0.04	1.569 ± 0.06
	RobX	15.14 ± 0.16	6.142 ± 0.24	3.957 ± 0.29	2.198 ± 0.27	5.857 ± 0.63
	BetaRobust	2.475 ± 0.18	1.226 ± 0.09	1.161 ± 0.11	1.402 ± 0.06	2.106 ± 0.18
	DiRARTE-W	2.893 ± 0.45	1.377 ± 0.17	1.243 ± 0.15	1.530 ± 0.16	2.077 ± 0.31
	DiRARTE-χ^2	2.935 ± 0.28	1.319 ± 0.12	1.408 ± 0.20	1.490 ± 0.08	2.197 ± 0.25
Plausibility	Baseline	0.447 ± 0.00	0.511 ± 0.02	0.537 ± 0.02	0.444 ± 0.01	0.500 ± 0.00
	RobX	0.436 ± 0.01	0.488 ± 0.02	0.533 ± 0.02	0.426 ± 0.01	0.500 ± 0.01
	BetaRobust	0.447 ± 0.00	0.503 ± 0.02	0.538 ± 0.02	0.436 ± 0.01	0.504 ± 0.00
	DiRARTE-W	0.445 ± 0.00	0.500 ± 0.01	0.540 ± 0.02	0.430 ± 0.01	0.505 ± 0.01
	DiRARTE-χ^2	0.444 ± 0.01	0.496 ± 0.01	0.538 ± 0.02	0.428 ± 0.01	0.506 ± 0.00
Time [s]	Baseline	0.094 ± 0.01	0.069 ± 0.00	0.042 ± 0.01	0.083 ± 0.01	0.062 ± 0.01
	RobX	0.839 ± 0.09	0.410 ± 0.03	0.366 ± 0.02	0.423 ± 0.01	1.066 ± 0.05
	BetaRobust	0.554 ± 0.03	0.385 ± 0.02	0.233 ± 0.04	0.271 ± 0.02	0.327 ± 0.02
	DiRARTE-W	0.097 ± 0.00	0.070 ± 0.00	0.043 ± 0.00	0.082 ± 0.01	0.064 ± 0.01
	DiRARTE-χ^2	0.110 ± 0.01	0.076 ± 0.00	0.048 ± 0.01	0.086 ± 0.01	0.069 ± 0.01

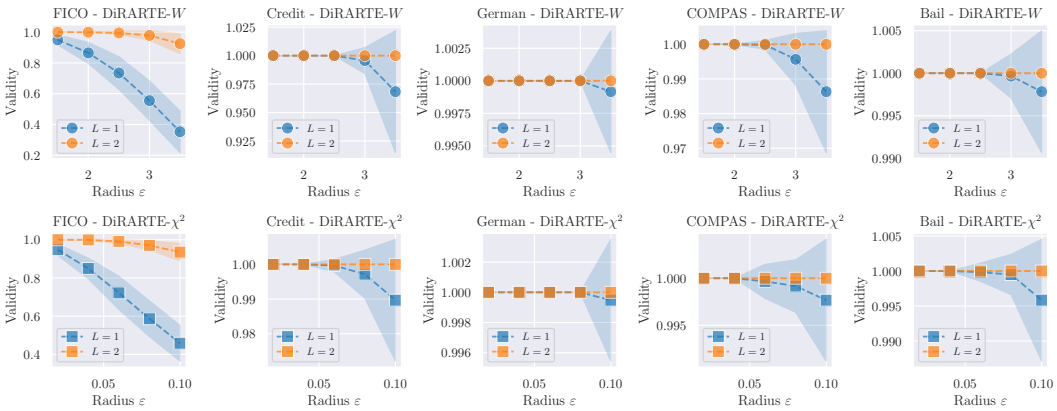
Table 8: Average sparsity, plausibility, and runtime [s] of each method with weighted ℓ_1-norm (lower is better).

(a) XGBoost

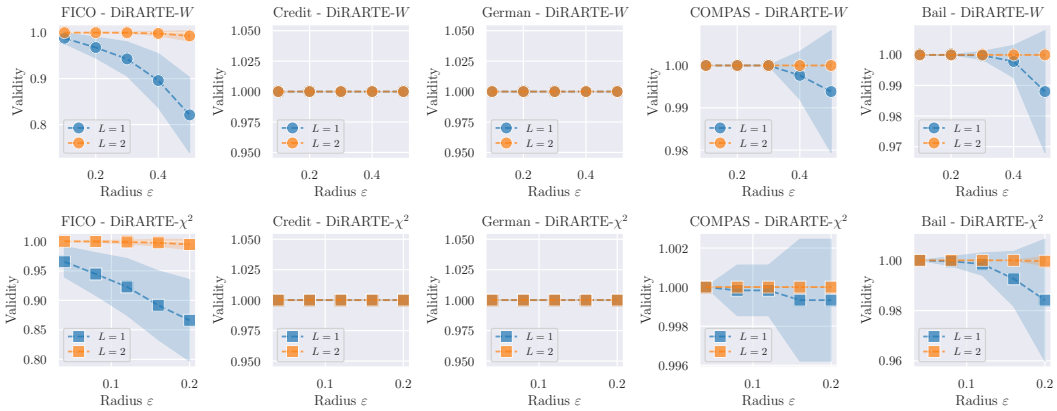
	Method	FICO	Credit	German	COMPAS	Bail
Sparsity	Baseline	1.701 ± 0.05	1.078 ± 0.03	1.058 ± 0.04	1.121 ± 0.02	1.233 ± 0.02
	RobX	15.33 ± 0.11	6.671 ± 0.24	3.881 ± 0.21	2.239 ± 0.23	5.706 ± 0.41
	BetaRobust	2.312 ± 0.11	1.209 ± 0.04	1.092 ± 0.05	1.190 ± 0.04	1.418 ± 0.03
	DiRARTE-W	3.362 ± 0.53	2.015 ± 0.35	1.443 ± 0.20	1.623 ± 0.14	1.914 ± 0.22
	DiRARTE-χ^2	3.355 ± 0.51	1.829 ± 0.31	1.393 ± 0.17	1.467 ± 0.17	1.950 ± 0.24
Plausibility	Baseline	0.453 ± 0.00	0.502 ± 0.01	0.496 ± 0.01	0.452 ± 0.01	0.494 ± 0.00
	RobX	0.437 ± 0.00	0.473 ± 0.01	0.493 ± 0.01	0.447 ± 0.01	0.501 ± 0.01
	BetaRobust	0.452 ± 0.00	0.500 ± 0.01	0.497 ± 0.01	0.451 ± 0.01	0.497 ± 0.00
	DiRARTE-W	0.458 ± 0.01	0.518 ± 0.01	0.496 ± 0.01	0.479 ± 0.02	0.505 ± 0.00
	DiRARTE-χ^2	0.453 ± 0.00	0.506 ± 0.01	0.496 ± 0.01	0.454 ± 0.01	0.504 ± 0.00
Time [s]	Baseline	0.062 ± 0.01	0.049 ± 0.01	0.018 ± 0.00	0.026 ± 0.00	0.048 ± 0.00
	RobX	0.346 ± 0.01	0.226 ± 0.01	0.178 ± 0.00	0.224 ± 0.00	0.256 ± 0.01
	BetaRobust	0.166 ± 0.02	0.122 ± 0.01	0.040 ± 0.01	0.046 ± 0.01	0.112 ± 0.01
	DiRARTE-W	0.063 ± 0.01	0.049 ± 0.01	0.018 ± 0.00	0.026 ± 0.00	0.049 ± 0.01
	DiRARTE-χ^2	0.092 ± 0.01	0.067 ± 0.01	0.023 ± 0.00	0.031 ± 0.00	0.064 ± 0.01

(b) Random Forest

	Method	FICO	Credit	German	COMPAS	Bail
Sparsity	Baseline	2.162 ± 0.16	1.206 ± 0.10	1.097 ± 0.09	1.316 ± 0.06	1.643 ± 0.08
	RobX	15.36 ± 0.11	6.399 ± 0.31	4.061 ± 0.35	2.186 ± 0.26	6.152 ± 0.70
	BetaRobust	2.514 ± 0.18	1.220 ± 0.09	1.195 ± 0.12	1.423 ± 0.08	2.179 ± 0.19
	DiRARTE-W	2.896 ± 0.41	1.337 ± 0.16	1.284 ± 0.16	1.558 ± 0.18	2.192 ± 0.33
	DiRARTE-χ^2	2.940 ± 0.26	1.309 ± 0.13	1.443 ± 0.21	1.526 ± 0.11	2.296 ± 0.25
Plausibility	Baseline	0.452 ± 0.00	0.510 ± 0.01	0.534 ± 0.02	0.449 ± 0.01	0.504 ± 0.00
	RobX	0.436 ± 0.01	0.488 ± 0.02	0.528 ± 0.02	0.432 ± 0.01	0.505 ± 0.01
	BetaRobust	0.450 ± 0.00	0.504 ± 0.01	0.536 ± 0.02	0.440 ± 0.01	0.507 ± 0.00
	DiRARTE-W	0.447 ± 0.00	0.501 ± 0.01	0.537 ± 0.02	0.438 ± 0.01	0.509 ± 0.01
	DiRARTE-χ^2	0.446 ± 0.01	0.498 ± 0.01	0.535 ± 0.02	0.436 ± 0.01	0.510 ± 0.00
Time [s]	Baseline	0.081 ± 0.01	0.064 ± 0.00	0.041 ± 0.01	0.080 ± 0.01	0.058 ± 0.00
	RobX	0.822 ± 0.09	0.405 ± 0.03	0.367 ± 0.02	0.420 ± 0.01	1.059 ± 0.05
	BetaRobust	0.537 ± 0.04	0.381 ± 0.03	0.238 ± 0.04	0.267 ± 0.02	0.324 ± 0.02
	DiRARTE-W	0.085 ± 0.01	0.066 ± 0.00	0.043 ± 0.01	0.080 ± 0.01	0.059 ± 0.00
	DiRARTE-χ^2	0.095 ± 0.01	0.072 ± 0.00	0.048 ± 0.01	0.084 ± 0.01	0.065 ± 0.00

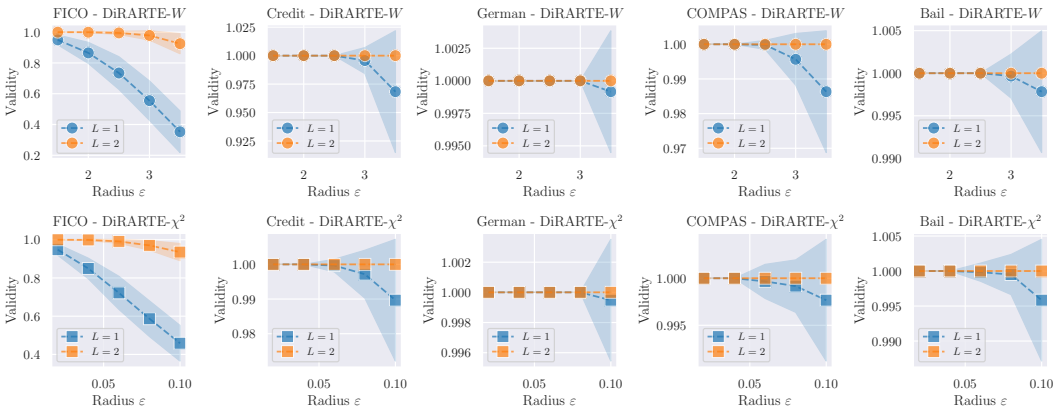


(a) XGBoost

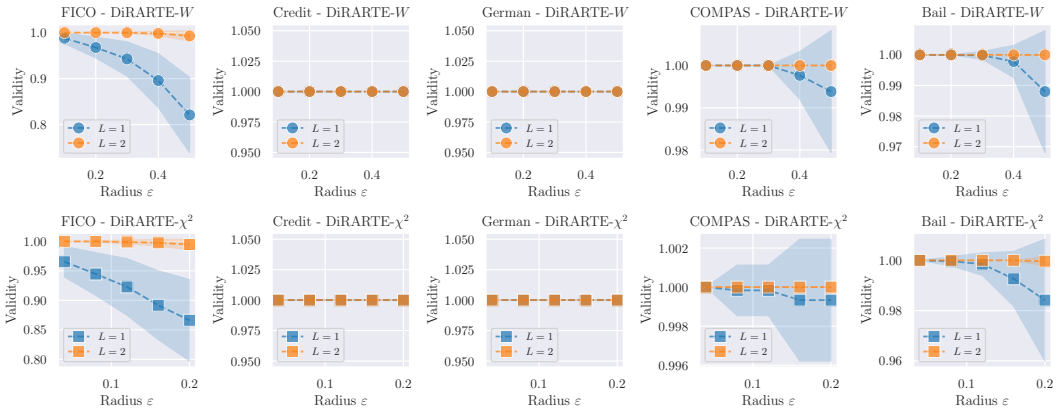


(b) Random Forest

Figure 6: Experimental results on TLPS for the ablation study of the merging level L with respect to the standard validity. The shaded area indicates the standard deviation across 10 different splits. We observed that while the validity of $L = 1$ was often degraded when the radius ε was large, the validity of $L = 2$ was stably high regardless of ε. These results indicate that expanding the search space by setting $L = 2$ enables us to obtain feasible solutions to our robust AR problem, even when the radius ε is large.

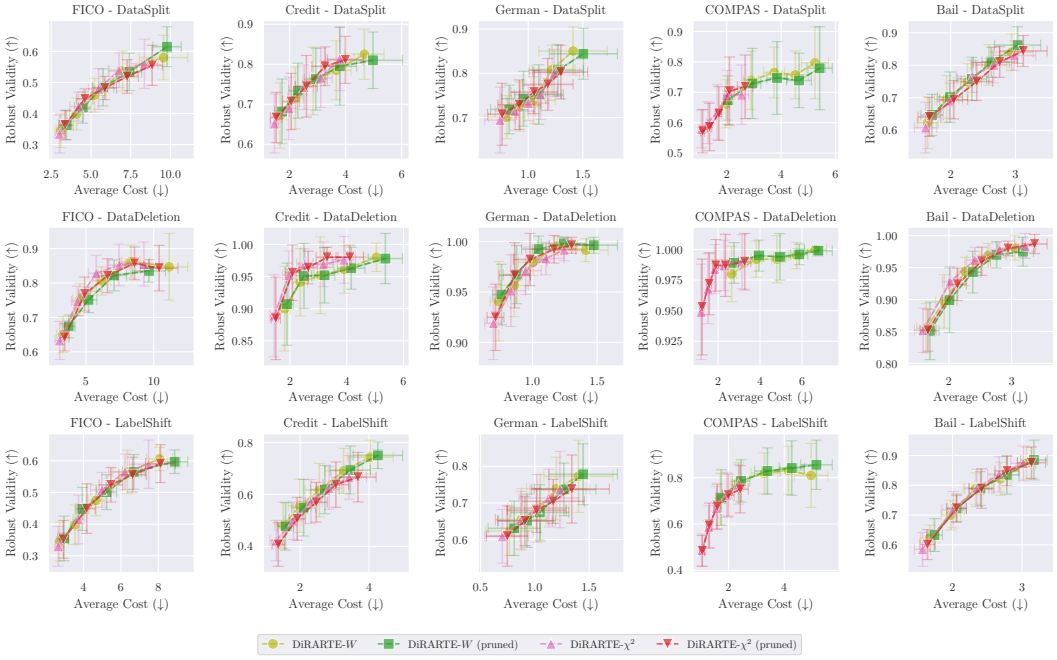


(a) XGBoost

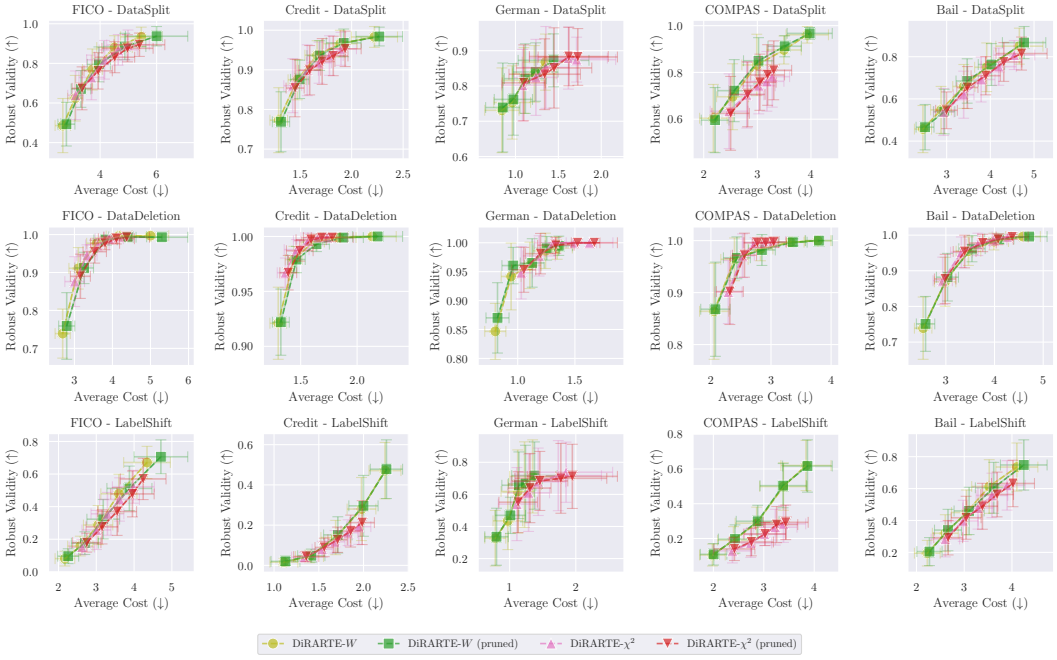


(b) Random Forest

Figure 7: Experimental results on weighted ℓ_1-norm for the ablation study of the merging level L with respect to the standard validity. The shaded area indicates the standard deviation across 10 different splits. We observed that while the validity of $L = 1$ was often degraded when the radius ε was large, the validity of $L = 2$ was stably high regardless of ε. These results indicate that expanding the search space by setting $L = 2$ enables us to obtain feasible solutions to our robust AR problem, even when the radius ε is large.

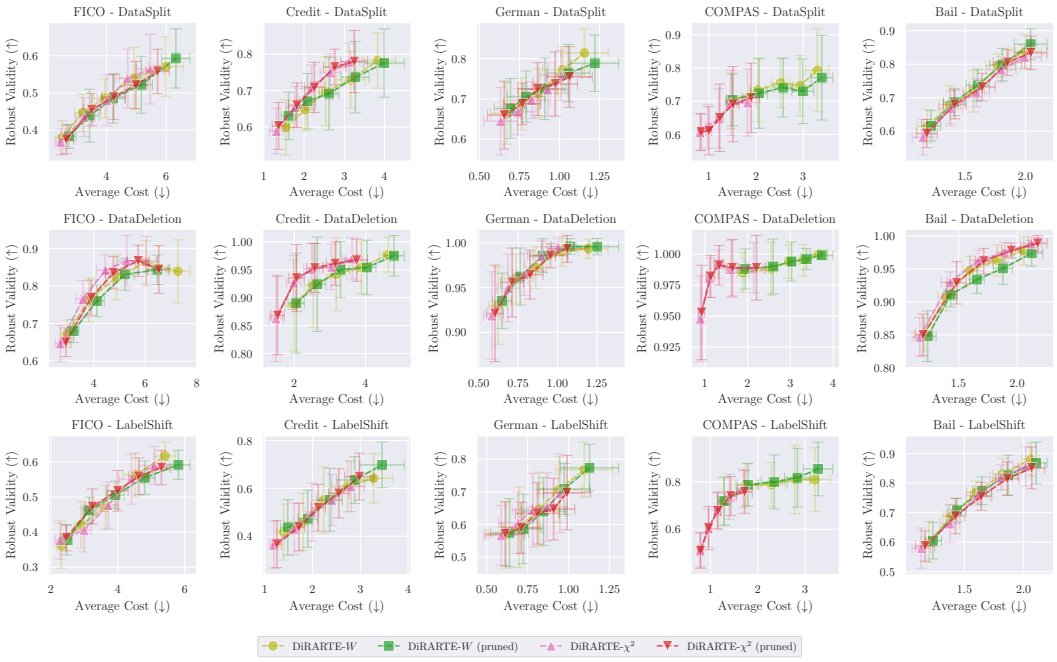


(a) XGBoost

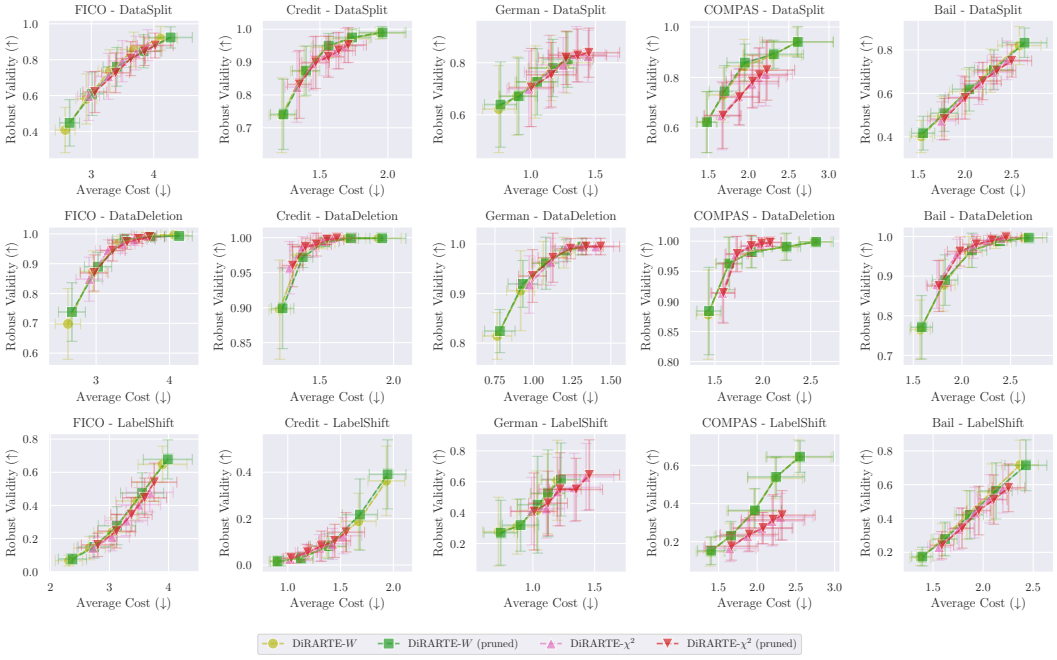


(b) Random Forest

Figure 8: Experimental results on TLPS for the ablation study of the pruning strategy with respect to the robust validity and the average cost. Error bars indicate the standard deviation across 10 different splits. We observed that the pruning strategy did not significantly affect the performance of the method, which indicates that the pruning strategy can be safely applied to our method without degrading the performance.



(a) XGBoost



(b) Random Forest

Figure 9: Experimental results on weighted ℓ_1-norm for the ablation study of the pruning strategy with respect to the robust validity and the average cost. Error bars indicate the standard deviation across 10 different splits. We observed that the pruning strategy did not significantly affect the performance of the method, which indicates that the pruning strategy can be safely applied to our method without degrading the performance.

Table 9: Experimental results for the ablation study of the pruning strategy with respect to the average runtime [s] (lower is better). Here, W and χ^2 stand for DiRARTE-W and DiRARTE-χ^2, respectively. We observed that the pruning strategy could reduce the runtime of the method, which indicates that the pruning strategy can be effectively applied to our method to reduce the runtime without degrading the performance.

(a) XGBoost (TLPS)

Method	Strategy	FICO	Credit	German	COMPAS	Bail
W	no pruning	0.078 ± 0.01	0.055 ± 0.01	0.019 ± 0.00	0.028 ± 0.00	0.057 ± 0.01
	50% pruning	0.044 ± 0.00	0.035 ± 0.00	0.011 ± 0.00	0.020 ± 0.00	0.036 ± 0.00
	relative change	-43.27%	-37.42%	-39.08%	-31.17%	-35.75%
χ^2	no pruning	0.102 ± 0.01	0.072 ± 0.01	0.024 ± 0.01	0.034 ± 0.00	0.071 ± 0.01
	50% pruning	0.059 ± 0.01	0.046 ± 0.01	0.015 ± 0.00	0.023 ± 0.00	0.046 ± 0.00
	relative change	-41.76%	-36.87%	-37.48%	-31.32%	-36.16%

(b) Random Forest (TLPS)

Method	Strategy	FICO	Credit	German	COMPAS	Bail
W	no pruning	0.191 ± 0.01	0.138 ± 0.01	0.075 ± 0.01	0.163 ± 0.01	0.127 ± 0.01
	50% pruning	0.099 ± 0.01	0.074 ± 0.00	0.043 ± 0.01	0.084 ± 0.01	0.068 ± 0.00
	relative change	-47.95%	-46.32%	-42.21%	-48.42%	-46.23%
χ^2	no pruning	0.210 ± 0.01	0.152 ± 0.01	0.083 ± 0.01	0.171 ± 0.01	0.138 ± 0.01
	50% pruning	0.110 ± 0.01	0.081 ± 0.00	0.048 ± 0.01	0.088 ± 0.01	0.074 ± 0.00
	relative change	-47.55%	-46.78%	-41.99%	-48.19%	-46.64%

(c) XGBoost (weighted ℓ_1-norm)

Method	Strategy	FICO	Credit	German	COMPAS	Bail
W	no pruning	0.068 ± 0.01	0.051 ± 0.01	0.018 ± 0.00	0.027 ± 0.00	0.052 ± 0.01
	50% pruning	0.038 ± 0.00	0.032 ± 0.00	0.011 ± 0.00	0.019 ± 0.00	0.033 ± 0.00
	relative change	-43.76%	-37.67%	-39.41%	-31.11%	-36.14%
χ^2	no pruning	0.091 ± 0.01	0.067 ± 0.01	0.023 ± 0.00	0.032 ± 0.00	0.067 ± 0.01
	50% pruning	0.053 ± 0.01	0.042 ± 0.00	0.015 ± 0.00	0.022 ± 0.00	0.042 ± 0.00
	relative change	-42.14%	-37.42%	-36.36%	-31.24%	-36.59%

(d) Random Forest (weighted ℓ_1-norm)

Method	Strategy	FICO	Credit	German	COMPAS	Bail
W	no pruning	0.167 ± 0.01	0.128 ± 0.01	0.072 ± 0.01	0.159 ± 0.01	0.118 ± 0.01
	50% pruning	0.088 ± 0.01	0.069 ± 0.00	0.042 ± 0.01	0.082 ± 0.01	0.063 ± 0.00
	relative change	-47.51%	-46.39%	-42.31%	-48.48%	-46.22%
χ^2	no pruning	0.186 ± 0.01	0.143 ± 0.01	0.079 ± 0.01	0.166 ± 0.01	0.130 ± 0.01
	50% pruning	0.098 ± 0.01	0.076 ± 0.00	0.048 ± 0.01	0.086 ± 0.01	0.069 ± 0.00
	relative change	-47.17%	-46.91%	-39.97%	-48.30%	-46.63%

Table 10: Average validity, cost, and runtime [s] of each standard AR method for tree ensembles (higher validity and lower cost and time are better). We observed that FT and FT++ could stably find low-cost valid actions with the fastest runtime among the methods, regardless of the dataset.

(a) XGBoost

	Method	FICO	Credit	German	COMPAS	Bail
Validity	FT	0.99 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	OAE	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	FOCUS (0.001)	1.0 ± 0.0	1.0 ± 0.0	0.98 ± 0.0	0.97 ± 0.0	1.0 ± 0.0
	FOCUS (0.01)	1.0 ± 0.0	1.0 ± 0.0	0.98 ± 0.0	0.97 ± 0.0	1.0 ± 0.0
	FOCUS (0.1)	0.95 ± 0.1	0.92 ± 0.0	0.75 ± 0.1	0.92 ± 0.0	1.0 ± 0.0
	FT++	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Cost	FT	1.55 ± 0.2	0.79 ± 0.0	0.46 ± 0.1	0.58 ± 0.1	0.71 ± 0.1
	OAE	18.03 ± 3.4	4.22 ± 0.4	0.24 ± 0.1	0.41 ± 0.1	0.36 ± 0.1
	FOCUS (0.001)	10.94 ± 1.1	8.35 ± 0.5	5.73 ± 0.5	10.34 ± 3.6	6.31 ± 0.4
	FOCUS (0.01)	8.86 ± 0.8	6.19 ± 0.6	3.36 ± 0.5	6.45 ± 2.4	5.04 ± 0.5
	FOCUS (0.1)	2.85 ± 0.3	1.86 ± 0.3	1.12 ± 0.2	1.26 ± 0.3	2.21 ± 0.3
	FT++	1.44 ± 0.2	0.79 ± 0.0	0.45 ± 0.1	0.57 ± 0.1	0.69 ± 0.1
Time [s]	FT	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0	0.01 ± 0.0
	OAE	179.69 ± 6.2	170.18 ± 11.4	22.29 ± 12.8	5.0 ± 0.2	48.8 ± 11.8
	FOCUS (0.001)	11.59 ± 0.5	12.25 ± 0.7	5.98 ± 0.2	7.65 ± 0.4	11.8 ± 0.5
	FOCUS (0.01)	11.54 ± 0.5	12.22 ± 0.7	5.98 ± 0.2	7.67 ± 0.4	11.8 ± 0.5
	FOCUS (0.1)	11.57 ± 0.5	12.19 ± 0.6	6.0 ± 0.2	7.67 ± 0.4	11.82 ± 0.5
	FT++	0.07 ± 0.0	0.06 ± 0.0	0.02 ± 0.0	0.03 ± 0.0	0.06 ± 0.0

(b) Random Forest

	Method	FICO	Credit	German	COMPAS	Bail
Validity	FT	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	OAE	0.44 ± 0.3	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
	FOCUS (0.001)	0.77 ± 0.1	1.0 ± 0.0	1.0 ± 0.0	0.94 ± 0.1	1.0 ± 0.0
	FOCUS (0.01)	0.72 ± 0.1	1.0 ± 0.0	1.0 ± 0.0	0.93 ± 0.1	1.0 ± 0.0
	FOCUS (0.1)	0.4 ± 0.1	0.37 ± 0.1	0.39 ± 0.1	0.62 ± 0.0	0.71 ± 0.1
	FT++	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0	1.0 ± 0.0
Cost	FT	2.32 ± 0.7	1.12 ± 0.1	0.66 ± 0.1	1.22 ± 0.2	1.32 ± 0.2
	OAE	16.2 ± 6.5	3.32 ± 0.7	0.58 ± 0.2	1.14 ± 0.2	1.35 ± 0.2
	FOCUS (0.001)	11.09 ± 1.2	11.88 ± 0.7	6.49 ± 0.5	4.29 ± 0.7	8.59 ± 0.5
	FOCUS (0.01)	5.19 ± 0.3	6.67 ± 0.3	3.45 ± 0.4	3.65 ± 0.3	5.85 ± 0.2
	FOCUS (0.1)	1.32 ± 0.3	1.49 ± 0.2	0.58 ± 0.1	1.14 ± 0.1	1.61 ± 0.2
	FT++	2.2 ± 0.6	1.11 ± 0.1	0.65 ± 0.1	1.21 ± 0.2	1.31 ± 0.2
Time [s]	FT	0.02 ± 0.0	0.02 ± 0.0	0.01 ± 0.0	0.02 ± 0.0	0.02 ± 0.0
	OAE	188.75 ± 6.3	187.94 ± 2.8	60.87 ± 12.7	16.12 ± 2.2	94.16 ± 10.6
	FOCUS (0.001)	18.92 ± 0.7	18.58 ± 0.3	12.23 ± 0.3	15.62 ± 0.2	17.16 ± 0.3
	FOCUS (0.01)	18.95 ± 0.8	18.64 ± 0.2	12.23 ± 0.3	15.62 ± 0.3	17.65 ± 0.8
	FOCUS (0.1)	19.03 ± 0.9	18.59 ± 0.1	12.22 ± 0.3	15.6 ± 0.3	17.8 ± 0.7
	FT++	0.18 ± 0.0	0.14 ± 0.0	0.08 ± 0.0	0.16 ± 0.0	0.12 ± 0.0

Table 11: Held-out coverage of the radius ε_α calibrated at each level α, averaged over the three scenarios (closer to α is better). (a) Marginal coverage, i.e., the fraction of the pairs of test instances and held-out models with $\Delta_{W_1}(q_\theta(\cdot \mid \boldsymbol{x}), q_{\theta'}(\cdot \mid \boldsymbol{x})) \leq \varepsilon_\alpha$. (b) Uniform coverage, i.e., the fraction of the held-out models whose maximum discrepancy over the test instances is at most ε_α. We observed that the coverage closely matched α in the marginal form and tracked α in the uniform form, though less tightly.

(a) Marginal coverage

Dataset	**Calibration level** α								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
FICO	0.101	0.202	0.303	0.403	0.502	0.601	0.702	0.801	0.900
Credit	0.100	0.200	0.300	0.399	0.499	0.599	0.699	0.799	0.900
German	0.103	0.205	0.310	0.408	0.507	0.607	0.699	0.799	0.899
COMPAS	0.102	0.202	0.303	0.402	0.498	0.598	0.700	0.800	0.900
Bail	0.099	0.199	0.299	0.401	0.501	0.601	0.700	0.800	0.900

(b) Uniform coverage

Dataset	**Calibration level** α								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
FICO	0.119	0.216	0.300	0.383	0.492	0.596	0.699	0.805	0.901
Credit	0.120	0.223	0.313	0.409	0.533	0.625	0.733	0.823	0.903
German	0.093	0.164	0.269	0.369	0.468	0.564	0.668	0.757	0.832
COMPAS	0.107	0.191	0.283	0.391	0.468	0.576	0.695	0.795	0.901
Bail	0.125	0.208	0.319	0.427	0.513	0.619	0.708	0.799	0.883

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: See the abstract and Section 1.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: See the last paragraph of Section 6.

   Guidelines:

   - The answer [N/A] 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: See Sections 2 to 4 and Section B.

Guidelines:

- The answer [N/A] 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: See Section 5 and Sections D to F.

   Guidelines:

   - The answer [N/A] 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: See Section 5 and Section F. We provide our code and data in our GitHub repository at `https://github.com/kelicht/dirarte`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

   Answer: [Yes]

   Justification: See Section 5 and Sections D and E.

   Guidelines:

   - The answer [N/A] 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: All averages in figures are reported with error bars. All averages in tables are reported with the standard deviation across multiple runs.

   Guidelines:

   - The answer [N/A] 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: See Sections 5 and F.

   Guidelines:

   - The answer [N/A] 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: We have read the ethics guidelines and confirmed that our paper conforms to them.

   Guidelines:

   - The answer [N/A] 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: See Section G.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] 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: See Section F.

    Guidelines:

    - The answer [N/A] 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: See `https://github.com/kelicht/dirarte`.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core method of this paper does not involve LLMs.

Guidelines:

- The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.