CONFEX: UNCERTAINTY-AWARE COUNTERFACTUAL EXPLANATIONS WITH CONFORMAL GUARANTEES

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual explanations (CFXs) provide human-understandable justifications for model predictions, enabling actionable recourse and enhancing interpretability. To be reliable, CFXs must avoid regions of high predictive uncertainty, where explanations may be misleading or inapplicable. However, existing methods often neglect uncertainty or lack principled mechanisms for incorporating it with formal guarantees. We propose CONFEX, a novel method for generating uncertainty-aware counterfactual explanations using Conformal Prediction (CP) and Mixed-Integer Linear Programming (MILP). CONFEX explanations are designed to provide local coverage guarantees, addressing the issue that CFX generation violates exchangeability. To do so, we develop a novel localised CP procedure that enjoys an efficient MILP encoding by leveraging an offline tree-based partitioning of the input space. This way, CONFEX generates CFXs with rigorous guarantees on both predictive uncertainty and optimality. We evaluate CONFEX against state-of-the-art methods across diverse benchmarks and metrics, demonstrating that our uncertainty-aware approach yields robust and plausible explanations.

1 INTRODUCTION

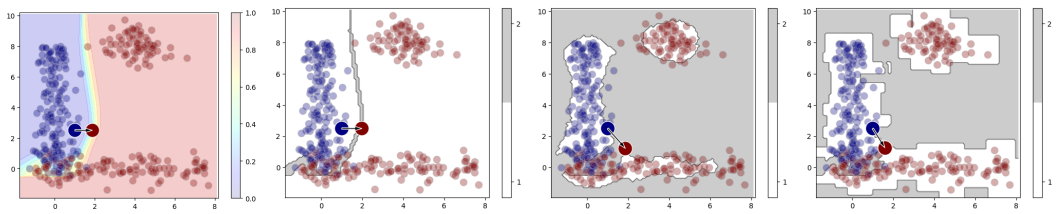
Machine learning models are deployed in high-stakes decision-making scenarios like loan approvals, medical diagnoses, and employment screening. In these contexts, algorithmic recourse—providing actionable feedback to individuals influenced by these decisions—is not just a technical concern but also an ethical and legal imperative. Although the legal status of "right to explanations" under the EU's General Data Protection Regulation (GDPR) remains contested (Wachter et al., 2017; Selbst & Barocas, 2018), there is growing consensus that individuals should be offered meaningful information about algorithmic decisions that impact them (Edwards & Veale, 2017; Binns et al., 2018).

Counterfactual explanations (*CFX*) were formally introduced by Wachter et al. (2017) as a method for algorithmic recourse. CFXs answer questions like: "What minimal changes to my input features would have altered the model's decision desirably?", and Wachter's formalisation focuses on finding counterfactual explanations that are minimally close to the original point (*factual instance*) or have sparse feature changes. These criteria of closeness and sparseness have been extended in later methods to other desiderata such as diversity, causality, actionability, and plausibility, to generate explanations that work better as a recourse path and are distinguished from adversarial examples.

However, most existing CFX methods fail to account for the inherent uncertainty in both data and model predictions. This is problematic because explanations that ignore uncertainty may lead to false confidence in suggested changes, potentially resulting in ineffective recourse actions when deployed in practice. Uncertainty quantification in CFX is thus crucial for generating reliable and actionable insights.

We introduce CONFEX, an uncertainty-aware CFX generator that builds on *Conformal Prediction* (*CP*) (Vovk et al., 2022; Angelopoulos et al., 2023). CP is a popular uncertainty quantification framework that offers distribution-free and finite-sample coverage guarantees. It works by using calibration data to construct prediction regions that contain the true (unknown) outcome with a user-specified probability. CP does not require assumptions on the data distribution and the underlying model, except that the calibration data and the test point must be exchangeable. The core idea of our CONFEX method is to constrain the search space for CFXs only to those points leading to a

singleton prediction region $\{y^+\}$, i.e., points that yield the desired outcome y^+ with a high degree of certainty, since non-singleton CP regions represent uncertain predictions.



(a) Minimum distance (b) CONFEX-Naive (c) CONFEX-LCP (d) CONFEX-Tree

Figure 1: Counterfactuals produced for the same factual instance (marked in blue) for a MLP classifier using approaches MILP-MinDist, CONFEX-Naive, CONFEX-LCP, CONFEX-Tree. CONFEX approaches use bandwidth as 35% of the median pairwise distance between calibration points, and alpha as 2%.

To illustrate our methods, Fig. 1a displays CFXs produced over a synthetic 2D dataset inspired from Poyiadzi et al. (2020). We can observe that counterfactuals produced by the minimal distance approach and by a naive application of CP to the CFX generation problem, called CONFEX-Naive (Section 3), fail to be plausible with respect to the data distribution.

These issues with naively applying CP to CFX generation stem from the fact that the generated (test-time) CFX may not be exchangeable with the calibration points, thereby affecting the validity of CP's guarantees. We solve this by imposing stricter coverage requirements for CP: we build prediction regions that approximately[1] attain *local* (aka *test-conditional*) guarantees, i.e., the target coverage probability is achieved for *any* test point. In contrast, normally, CP guarantees are marginal, i.e., the coverage probability is averaged over the joint calibration and test distribution.

Our CONFEX method relies on a *Mixed-Integer Linear Programming* (*MILP*) encoding of the optimisation problem, which not only guarantees optimality of solutions but also ensures satisfaction of the CP constraints. We present two methods for incorporating local coverage constraints. The first is *localised CP* (Guan, 2023), which frames conditional coverage as a covariate shift problem (Tibshirani et al., 2019). However, it requires encoding and solving calibration quantiles in MILP, which is computationally expensive and scales poorly with the dataset size. The second, more efficient, method is a KD-tree-based encoding of local calibration quantiles. For this method, we use regression trees, which can be efficiently encoded in MILP.

In summary, our main contributions are:

1. a mathematical formulation for distribution-free uncertainty-aware counterfactual explanations, the first to apply conformal prediction in a principled manner (i.e., by addressing the exchangeability problem via test-conditional coverage);
2. a novel localised CP procedure which, with an efficient MILP encoding, for generation of CFXs, which can be used more generally to incorporate (test-conditional) CP uncertainty constraints in any search problem;
3. an extensive experimental evaluation demonstrating that our CONFEX method outperforms competing generators by providing more plausible and stable explanations, as well as enjoying formal guarantees on uncertainty.

2 Background

Counterfactual Explanations Let $\hat{f} : \mathcal{X} \to \mathcal{Y}$ denote a trained classifier for which we seek to generate counterfactual explanations. Given an instance $x_0 \in \mathcal{X}$ such that $\hat{f}(x_0) \neq y^+$, the goal is to identify a counterfactual instance x' such that $\hat{f}(x') = y^+$. Wachter et al. (2017) frame this as an

[1]Exact conditional guarantees for CP are known to be impossible unless the inputs are discrete (Vovk, 2012; Barber et al., 2020).

optimisation problem and solve it via gradient descent.

$$x_{\text{cf}} \in \arg\min_{x'} \max_{\lambda} \left(\lambda \ \text{yloss}\left(\hat{f}\left(x'\right), y^{+}\right) + \text{dist}\left(x_0, x'\right)\right). \tag{1}$$

The loss function aims to find an explanation that changes the predicted class to the target class (first term), while also ensuring that the explanation is close to the input instance (second term). Closeness is often defined as an L_p norm, which can be weighted based on the observed data (e.g. the inverse median absolute deviation), or to reflect domain knowledge (Dandl et al., 2020). However, by optimising solely for closeness, this formulation often leads to counterfactual explanations that resemble adversarial examples and may not be actionable or robust.

Desirable properties of CFXs include *validity* (prediction flips to y^{+}), *proximity* (closeness to the factual instance), *sparsity* (few feature changes), *plausibility* (realistic and likely under the data distribution), *actionability* (only mutable features are altered), *causality* (identified counterfactual satisfies causal relationships) and *robustness* (stability under input perturbations); see (Verma et al., 2020; Karimi et al., 2021).

Uncertainty-aware CFX methods show promise for enhancing the robustness and plausibility of CFXs. In this line of work, Schut et al. (2021) propose minimising predictive entropy across an ensemble of models to consider the effect of uncertain regions. Bayesian approaches, such as CLUE (Antorán et al., 2020), leverage predictive uncertainty from Bayesian neural networks to generate epistemically informative counterfactuals.

Conformal Prediction and CFXs CP is a distribution-free inference framework that complements any predictive model with rigorous uncertainty quantification. CP outputs prediction sets guaranteed to contain the true (unknown) outcome with a user-specified probability $1-\alpha$ without relying on asymptotic or parametric assumptions (Vovk et al., 2022; Angelopoulos et al., 2023). To construct these sets, CP performs the following steps:

1. **Calibration**: use a held-out calibration dataset $\mathcal{D}_{\text{cal}} = \{(x_i, y_i)\}_{i=1}^{n}$ to find the critical value $q_{1-\alpha}$ (i.e., the $1-\alpha$ quantile) of a chosen test statistic called the *(non-conformity) score* $s(x, y)$, which is normally chosen to quantify the deviation between the model prediction $\hat{f}(x)$ and the ground truth y. This step is performed only once, offline. Formally,

$$q_{1-\alpha} = Q_{1-\alpha}\left(\sum_{i=1}^{n} \frac{1}{n+1}\delta_{s(x_i,y_i)} + \frac{1}{n+1}\delta_{+\infty}\right), \tag{2}$$

   where $Q_{1-\alpha}$ is the $1-\alpha$ quantile function and δ_v is the Dirac distribution centered at v.
2. **Inference**: for a test input x^*, construct a prediction region $C(x^*)$ by including all labels y whose score is below the critical value (i.e., such that $s(x^*, y) \leq q_{1-\alpha}$).

The CP procedure provides the following marginal guarantee for an unseen test point (x^*, y^*):

$$\mathop{\mathbb{P}}_{\mathcal{D}_{\text{cal}}, (x^*, y^*)} (y^* \in C_{1-\alpha}(x^*)) \geq 1 - \alpha. \tag{3}$$

The above holds in finite sample regimes (as opposed to asymptotic) under the mild condition of exchangeability (a weaker assumption than IID), i.e., the joint distribution of calibration and test points is invariant under permutations. By marginal guarantees, we mean that the coverage probability of equation 3 is achieved on average over the joint calibration and test distribution.

To our knowledge, there exist only two methods which apply conformal prediction to CFX generation: ECCCo (Altmeyer et al., 2024) and CPICF (Adams et al., 2025).

CPICF (Adams et al., 2025) assumes an alternative "individualised" setting, where an institution holds a private black-box classifier and aims to provide CFXs to individuals without disclosing the classifier. The knowledge of each individual is modelled by their own classifier, and the organisation produces a CFX to reduce uncertainty in the global classifier via CP. This is a fundamentally different setting to ours, furthermore CPICF's formulation does not retain any formal CP guarantees.

In the standard setting, ECCCo extends Wachter's formulation (equation 1) with two additional terms: one that optimises the energy of the identified counterfactual to enhance plausibility, and

one that minimises uncertainty through the smooth conformal set size loss of Stutz et al. (2022). However, ECCCo has the following drawbacks: 1) it incorporates conformal prediction, but in a way that does not address exchangeability issues, which we detail in Section 3.1; 2) the procedure does not guarantee CP regions will have the required size (e.g., singletons); 3) it relies on energy-based training to obtain plausible CFXs. As we will show, our approach instead induces plausible CFXs solely by using CP constraints, formulating these constraints to enforce local validity (thereby solving the exchangeability issues), and thanks to the MILP formulation, it ensures satisfaction of the set size constraints whilst being optimally close.

Mixed Integer Linear Programming (MILP) and CFXs MILP provides a framework for formulating and deriving CFXs as a constraint-solving problem. The problem is of finding a point x' which minimises the distance to the original instance x_0 whilst being classified as y^+.

$$x_{\text{cf}} \in \arg\min_{x'} \text{dist}(x_0, x') \quad \text{s.t. } \hat{f}(x') = y^+ \tag{4}$$

We refer to this method as MILP-MinDist, and it serves as a baseline for our CONFEX method.

For the model f to be encoded in MILP, f must be linearly representable; this is the case for e.g. linear classifiers and multilayer perceptrons with ReLU activations, as well as non-differentiable models such as decision trees. Neural network layers like sigmoid or softmax are not linearly representable, but can be omitted from the MILP encoding if used at the last layer since we can identify if $f(x_{\text{cf}}) = y^+$ based on the logits alone.

When presented to an MILP solver, this approach is guaranteed to yield a valid and optimal CFX, if such an explanation exists. Gradient-based methods, on the other hand, are incomplete, meaning that they may fail to find valid CFXs or may return suboptimal solutions.

We note that properties like causality and actionability can be incorporated in equation 4 through MILP constraints on the input variables; similarly, a set of diverse explanations (as opposed to an individual one) can be generated by repeatedly solving the problem and adding constraints or objective function terms to block or penalize explanations similar to those already identified (Kanamori et al., 2020). By adding such constraints, our method can accommodate these desiderata as well.

3 CFXs WITH CP CONSTRAINTS: A NAIVE ATTEMPT

We first present a naive approach to apply conformal prediction to minimise the uncertainty in the generated CFX, which we call CONFEX-Naive. This approach extends MILP-MinDist (see equation 4) by restricting the search space to points yielding the singleton CP region $\{y^+\}$, i.e., points attaining the target class and with a high degree of certainty:

$$x_{\text{cf}} \in \arg\min_{x'} \text{dist}(x_0, x') \quad \text{s.t. } C_{1-\alpha}(x') = \{y^+\} \tag{5}$$

Note that the above constraint is equivalent to the constraints $s(x', y^+) \leq q_{1-\alpha}$ and $\bigwedge_{y \neq y^+} s(x', y) > q_{1-\alpha}$. The quantile $q_{1-\alpha}$ is pre-computed on the held-out calibration set.

For multi-layer perceptrons, we use the following log-likelihood ratio as the score function

$$s(x, y) = \log\left(\frac{\max_{y' \neq y} p(x)_{y'}}{p(x)_y}\right), \tag{6}$$

where $p(x)_y$ is the softmax probability of y predicted by the model f for input x. When the correct class is predicted, the ratio is below 1 and we obtain a negative score. When the model is wrong, the ratio is positive and the score grows bigger as the model confidence on y decreases relative to that on the predicted class. Importantly, equation 6 can be equivalently expressed in a linear form as $s(x, y) = -l(x)_y + \max_{y' \neq y} l(x)_{y'}$, where $l(x)$ is the predicted vector of logits, making it efficiently representable in MILP.

Relation with CFX-base We note that our score function is well-formed, i.e., $s(x, y)$ is lowest when y is the label predicted by the model f (and, in particular, $s(x, y)$ increases as the softmax probability of y decreases). Thus, when a CP prediction region returns the singleton $\{y^+\}$, then

y^+ is the class with the lowest score, i.e., the class predicted by f. That is, for any $\alpha \in (0, 1)$, $C_{1-\alpha}(x) = \{y^+\} \rightarrow f(x) = y^+$. This implies that the feasible set of CONFEX is a subset of that of CFX-base, and so, CONFEX explanations can never attain smaller (better) distances than CFX-base. Importantly, since the above property holds for any α, it also holds for any choice of quantile $q_{1-\alpha}$. This property also applies to the localised CP methods described later, which define a different quantile value.

3.1 Need for Conditional Guarantees

A visual example of using CONFEX-Naive to generate a counterfactual explanation is shown in Figure 1 (plot b). We observe that that the effect of including the singleton set size constraint is to push the counterfactual explanation closer to the further past the decision boundary compared to MILP-MinDist (plot a), which is desirable since the identified CFX would resemble less an adversarial example. However, the counterfactual explanation the identified CFX is somewhat counterintuitive: it lies in an area without local datapoints, i.e., away from the data support (see plot d). Since the CP constraints enforce low-uncertainty predictions, we would expect to find the CFX in a region where datapoints unambiguously belong to the target class, and not in regions near the decision boundary, where multiple classes overlap, or with no or little data support.

The main issue is that CONFEX-Naive can return CFXs that are not exchangeable with the calibration points, violating CP's marginal guarantees. Hence, our prediction regions should be valid *for any* choice of test inputs (not just exchangeable ones), requiring the coverage requirements to be strengthened to enforce *conditional validity*, i.e., for *any* choice of $x = x'$, the following must hold:

$$\mathop{\mathbb{P}}_{\mathcal{D}_{\text{cal}},(x,y)} (y \in C_{1-\alpha}(x) \mid x = x') \geq 1 - \alpha. \tag{7}$$

However, unless the inputs are discrete, the above exact conditional guarantees are known to be impossible if we require distribution-free and finite-sample guarantees (Vovk, 2012; Barber et al., 2020). To solve this issue, among the several methods recently proposed for CP with approximate conditional validity (Jung et al., 2022; Hore & Barber, 2023; Ding et al., 2023; Gibbs et al., 2025; Cabezas et al., 2025), we focus on the *localised CP (LCP)* method of Guan (2023), described next.

4 The CONFEX Approach

Our method CONFEX uses Localised Conformal Prediction (LCP) to generate CFXs with more principled, local coverage guarantees. We introduce two variants: CONFEX-LCP, which encodes LCP constraints via MILP, and CONFEX-Tree, which also provides local guarantees via MILP but is more computationally efficient thanks to an offline tree-based representation of the local quantiles.

4.1 Localised Conformal Prediction (LCP) and CONFEX-LCP

Localised Conformal Prediction (LCP) (Guan, 2023) relaxes strict conditional coverage (see equation 7) by requiring coverage to hold only within a local neighbourhood around a test input x^*. To achieve this, LCP reweights the calibration points as if they were drawn under the localised distribution of x^*, thereby restoring exchangeability. The reweighted probabilities are computed by a *localiser kernel* $H : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$, which measures how "close" x' is to x, with $H(x, x) = 1$. In our method, we use the L_1-box kernel

$$H(x, x') = \mathbf{1}(\|x - x'\|_1 \leq h), \tag{8}$$

where h is the kernel bandwidth controlling the degree of localisation. For numerical and ordinal features, the L_1 distance is computed after normalisation; for categorical features, we require exact matches over all or some categorical features, else $H(x, x') = 0$. Other kernels (e.g., based on infinity norm or Gaussian smoothing) are also possible.

For a test input x^*, the local quantile is

$$q^{\text{LCP}}_{1-\alpha}(x^*) = Q_{1-\alpha}\left(\sum_{i=1}^{n} w_i \delta_{s(x_i,y_i)} + w^* \delta_{+\infty}\right), \tag{9}$$

where $w_i = \frac{H(x^*, x_i)}{W}$ for $i = 1, \ldots, n$ and $w^* = \frac{H(x^*, x^*)}{W} = \frac{1}{W}$, with $W = 1 + \sum_{i=1}^{n} H(x^*, x_i)$ being a normalizing factor.

This reweighting step and the resulting prediction region $C_{1-\alpha}^{LCP}(x^*) = \{y : s(x^*, y) \leq q_{1-\alpha}^{\text{LCP}}(x^*)\}$ ensure, for any test point x^*, the following approximate conditional guarantee:

$$\mathbb{P}_{\mathcal{D}_{\text{cal}} \sim P_{X,Y}^n, (x,y) \sim P_{X,Y}^*} (y \in C_{LCP,1-\alpha}(x)) \geq 1 - \alpha, \tag{10}$$

where $P_{X,Y}^n$ is the (product) distribution of the n calibration points, and $P_{X,Y}^* = P_{Y|X} \times P_X^*$ is the localised test distribution, with $P_X^* = P_X \circ H(x^*, X)$ being the distribution of X obtained by applying to P_X the kernel H centered at x^*.

CONFEX-LCP We extend CONFEX-Naive by replacing CP regions with LCP regions, yielding more principled and adaptive counterfactual generation. Formally,

$$x_{\text{cf}} \in \arg\min_{x'} \text{dist}(x_0, x') \quad \text{s.t. } C_{1-\alpha}^{LCP}(x') = \{y^+\}, \tag{11}$$

which enforces $s(x', y^+) \leq q_{1-\alpha}^{LCP}(x')$ and $s(x', y) > q_{1-\alpha}^{LCP}(x')$ for all $y \neq y^+$. Unlike CONFEX-Naive, which uses a single global quantile $\hat{q}$, here the quantile depends on the candidate x', requiring explicit encoding in the MILP formulation (see Algorithm 2 in the Appendix). This introduces additional variables and big-M constraints linear in the calibration set size. Fig. 1 (plot c) shows a CFX computed using CONFEX-LCP.

Properties. Thanks to the LCP method, CONFEX-LCP computes quantiles using only points local to the test input x, where locality is defined by the L1 kernel. This yields more adaptive and reliable uncertainty estimates than vanilla CP (and CONFEX-Naive), with larger prediction sets in sparse or ambiguous regions, whilst ensuring that counterfactual is grounded with the data, i.e., similar (local) individuals which are correctly predicted to be in the target class. We note that features in the kernel can be assigned different weights based on domain knowledge. The choice of the kernel bandwidth h is application-specific and it allows us to balance between local and marginal coverage.

4.2 CONFEX-TREE: FAST VARIANT OF CONFEX-LCP

Due to the increased cost of resolving quantiles using MILP, LCP is infeasible for practical use with large calibration sets. In this section, we introduce CONFEX-Tree, an efficient alternative formulation of Localised CP which retains formal guarantees. CONFEX-Tree leverages that decision trees are efficiently representable in MILP and uses precomputed local quantiles. While LCP operates at test-time by retaining only the calibration points within distance h of the point, CONFEX-Tree works offline to determine locality constraints: it splits the feature space recursively to obtain local neighbourhoods of calibration points having kernel width of at most h.

The construction procedure is inspired by kd-trees (Skrodzki, 2019) and detailed in Algorithm 1. Each leaf specifies a precomputed local quantile using only calibration points within that leaf. From these points, we also compute the midpoint of the smallest enclosing hyper-rectangle. The tree construction ensures that no two points in a leaf can have a bigger L_∞ distance than the kernel bandwidth h. Then, each new test point x' is assigned to a leaf of the tree and is associated with the corresponding quantile if x' is within L_∞ distance of $h/2$ from the midpoint, which means that it is within distance of h from any calibration point of that leaf. To handle categorical features, we stratify the dataset by each combination of (all or select) categorical values and generate a tree for each stratum (which is equivalent to first splitting on all categorical features).

The resulting tree is encoded in MILP and used to provide the quantile value for the test point, replacing the LCP regions from CONFEX-LCP . Formally, explanations are derived by solving

$$x_{\text{cf}} \in \arg\min_{x'} \text{dist}(x_0, x') \quad \text{s.t. } C_{1-\alpha}^{\text{Tree}}(x') = \{y^+\}, \tag{12}$$

where $C_{1-\alpha}^{\text{Tree}}$ is constructed using the local tree-based quantiles returned by Algorithm 1.

Properties of CONFEX-Tree. The tree constructed by the CONFEX-Tree defines a partitioning of the feature space into disjoint regions $\{\mathcal{X}_g\}_{g \in \mathcal{G}}$. Each g has an associated quantile value $q_{1-\alpha,g}$

Algorithm 1: CONFEX-Tree: Tree-based encoding of local quantiles

Input : Calibration set $\mathcal{D}_{\text{cal}}$, score function s, coverage level $1-\alpha$, bandwidth h
Output: Tree-based quantile encoding
Categorical Stratification:

1. Stratify the calibration dataset by each distinct combination of categorical feature values.
2. Generate a tree for each group using the Tree Construction procedure over the normalised numerical and ordinal values only.

Tree Construction:

1. If the maximum range along any feature dimension of all calibration points in the node is less than h, stop and create a leaf node. At each leaf, compute and store:
   - the $1-\alpha$ quantile of the scores $s(x, y)$ of the calibration points assigned to the leaf;
   - the midpoint of the calibration features in the leaf.
2. Otherwise, split the current node along the feature with the maximum spread, using the midpoint of that feature's values as the split point. Recurse on the left and right subsets to build subtrees.

Prediction for test point x^*:

1. Select the correct tree based on the test point's categorical values.
2. Traverse the tree using x' until reaching a leaf. Let c and q be its stored midpoint and quantile.
3. Reject point if assigned to the leaf but not local: if $\|x^* - c\|_\infty > h/2$, return ∞; o/w, return q.

computed using only calibration points in g. This results in the following finite-sample group-conditional coverage guarantee

$$\mathbb{P}\big(y \in C^{\text{Tree}}_{1-\alpha}(x^*) \mid x^* \in \mathcal{X}_g\big) \;\geq\; 1-\alpha \qquad \text{for all } g \in \mathcal{G}, \tag{13}$$

as per Vovk (2012). Note that our method overapproximates the group-conditional quantiles as it assigns a quantile of ∞ when x^* has L_∞ distance more than $h/2$ from the midpoint of g. For this reason, it still satisfies the above guarantee.

Moreover, by construction, the groups created by CONFEX-Tree are local regions of calibration points in the feature space. Hence, we obtain an approximate conditional guarantee, as the tree approximates the conditional quantile $Q_{1-\alpha}(s|x)$ with the granularity of the approximation being controlled by the bandwidth h.

Finally, CONFEX-Tree can be viewed as an instance of LCP using the following kernel

$$H(x, x') = \mathbf{1}(\|x - x'\|_\infty \leq h \wedge \exists g.x, x' \in \mathcal{X}_g), \tag{14}$$

i.e., both points need to belong to the same leaf and have L_∞ distance bounded by h. Using this kernel, the guarantees of equation 10 also apply to CONFEX-Tree.

5 Evaluation

In this section, we evaluate our method against competing CFX methods, assessing the cost (distance), plausibility and sensitivity of CFXs generated by CONFEX-Tree. We explore the impact of varying the kernel bandwidth and the user-specified coverage rate, and we verify the formal coverage guarantees of CONFEX methods. We find that over all benchmarks, CONFEX produces more stable and plausible CFXs, provided the kernel bandwidth is appropriately chosen.

Experimental setup For our experiments, two classes of models are considered: multi-layer perceptrons (MLPs) and random forests (RFs). We selected four tabular datasets commonly found in the CFX literature: AdultIncome (Becker & Kohavi, 1996), CaliforniaHousing (Pace & Barry, 1997), GiveMeSomeCredit and GermanCredit (Hofmann, 1994), using a training-calibration-test split of 60%-20%-20% for each.

To evaluate CONFEX, we compare our efficient tree-based approach CONFEX-Tree (CTree) against competing uncertainty-aware generators: ECCCo (Altmeyer et al., 2024), the only other CFX

	CaliforniaHousing			GermanCredit		
	Distance	Plausibility	Sens (10^{-1})	Distance	Plausibility	Sens (10^{-1})
Multi-Layer Perceptron						
MinDist	**0.02 ± 0.00**	0.38 ± 0.04	41.83 ± 8.98	1.69 ± 0.04	0.50 ± 0.06	0.09 ± 0.01
ECCCo	0.35 ± 0.01	-0.61 ± 0.03	0.24 ± 0.01	0.94 ± 0.01	0.21 ± 0.05	0.05 ± 0.02
Greedy	1.71 ± 0.18	-0.98 ± 0.02	**0.13 ± 0.00**	0.98 ± 0.05	-0.02 ± 0.04	0.09 ± 0.00
Wachter	0.08 ± 0.01	0.47 ± 0.07	1.57 ± 0.26	**0.40 ± 0.01**	0.77 ± 0.03	0.25 ± 0.00
CNaive	0.03 ± 0.01	0.36 ± 0.04	15.70 ± 2.36	1.83 ± 0.05	0.23 ± 0.15	0.08 ± 0.03
CTree	0.21 ± 0.01	**0.72 ± 0.02**	0.26 ± 0.09	2.58 ± 0.00	**1.00 ± 0.00**	**0.00 ± 0.00**
Random Forest						
MinDist	**0.01 ± 0.00**	0.45 ± 0.03	35.36 ± 15.39	1.69 ± 0.05	0.36 ± 0.02	0.09 ± 0.01
FT	0.11 ± 0.01	0.46 ± 0.02	0.53 ± 0.06	0.57 ± 0.05	0.86 ± 0.00	**0.09 ± 0.00**
FOCUS	**0.09 ± 0.00**	0.44 ± 0.06	5.58 ± 1.73	0.55 ± 0.13	0.88 ± 0.02	0.48 ± 0.02
CNaive	0.03 ± 0.01	0.47 ± 0.05	9.92 ± 2.08	1.71 ± 0.04	0.67 ± 0.07	0.09 ± 0.02
CTree	0.18 ± 0.01	**0.65 ± 0.03**	**0.42 ± 0.16**	2.58 ± 0.70	**1.00 ± 0.00**	0.28 ± 0.26

Table 1: Results for CaliforniaHousing and GermanCredit datasets. We set $\alpha = 0.1$, and report the best result in terms of plausibility for CONFEX-Tree, which is with bandwidth 0.05 for both datasets. Note that other methods seem to attain smaller distances than MinDist in some configurations; this is because these methods not always return valid counterfactuals. See the appendix for further discussion and full results.

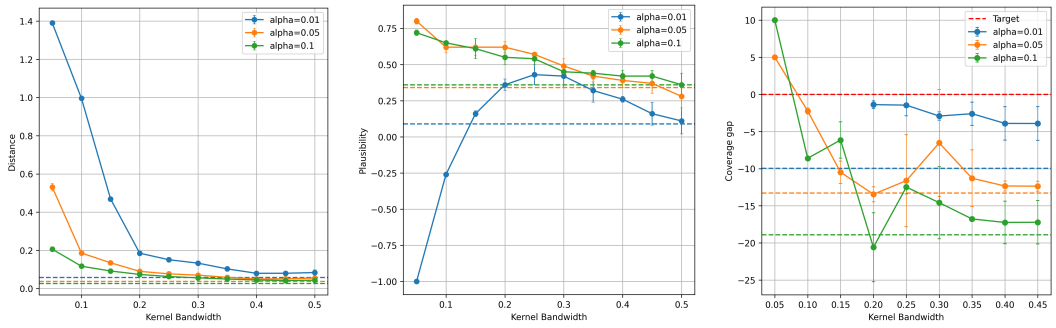
method which uses CP, and a modified version of Schut (Schut et al., 2021) (called 'Greedy' in our table) which uses a single MLP instead of an ensemble, as well as the Wachter et al. (2017) baseline. For tree-based models, we compare against the popular methods FeatureTweak (FT) (Tolomei et al., 2017), which searches for possible paths which can change the classification, and FOCUS (Lucic et al., 2021), which optimises for distance over a differentiable relaxation of the tree models. As baselines, we include MILP-MinDist (MinDist) and CONFEX-Naive (CNaive). As discussed previously, CONFEX-LCP is very expensive (3 minutes per instance, on average) due to its "direct" (and inefficient) quantile encoding. Hence, we did not conduct extensive experiments for it.

Metrics To evaluate the CFXs, we focus on two main dimensions: plausibility and sensitivity. *Plausibility* evaluates whether counterfactuals lie close to the data distribution, and is measured with the Local Outlier Factor (LOF) stratified per target class, with higher scores indicating more realistic examples. *Sensitivity* (Sens) captures robustness to small perturbations of the input instance x; counterfactuals with low sensitivity remain consistent under such perturbations.

For each model and generator, we compute metrics from 100 generated CFXs for factual points taken from the test set, plus an additional 100 for the sensitivity metric. This process is repeated twice per dataset, and the metrics obtained are then computed and averaged to ensure statistical reliability. We also record the distance, implausibility, stability, and validity of the method. Further details on the metrics and experimental setup can be found in the appendix.

Evaluation of conformal guarantees In the main setup, CFXs are generated for each test instance, but since their ground truth is unknown, coverage cannot be computed. We therefore run an additional simulated setup, identical to CONFEX in that it finds the *closest test point* whose CP region is a singleton comprising the target class. This way, true labels are known and we can compute the empirical coverage $\mathbb{E}(\mathbf{1}(y \in C_{1-\alpha}(x)))$ over this resampling of the test set. We measure the gap between the observed coverage and its gap from the target $1 - \alpha$. Note that this resampling considers only CFX-like points and hence breaks exchangeability. So, we expect CONFEX-Naive to miss the coverage target and the localised procedures to fare better.

Results discussion In Table 1, we observe that CONFEX-Tree consistently outperforms competing methods by producing more plausible, and in many cases less sensitive explanations. In some instances, this comes with the added benefit of shorter distances, as seen with the CaliforniaHousing dataset. This is in contrast to CONFEX-Naive which shows substantially lower plausibility and



(a) Distance (b) Plausibility (c) Coverage Gap

Figure 2: Effect of coverage rate and kernel bandwidth on metrics for CONFEX-Tree on the CaliforniaHousing dataset. CONFEX-Naive is represented by dashed horizontal lines.

higher sensitivity, validating the issues illustrated in Figure 1 and further motivating the use of localisation in CP. We observe similar trends for the AdultIncome and GiveMeSomeCredit datasets, but these are reported in the appendix for space reasons.

Fig. 2 illustrates the effect of varying the kernel bandwidth and coverage rate in the CONFEX-Tree method. Increasing the coverage rate $1 - \alpha$ leads to larger distances, since prediction sets become more conservative and singleton regions less frequent. Larger bandwidths yield shorter distances but at the cost of lower plausibility, as the notion of locality becomes weaker[2]. These observations are consistent with the fact that, as the kernel bandwidth grows, localised CP converges to standard marginal CP, as seen with CONFEX-Naive in the figures.

In the (simulated) CFX setting, the Coverage Gap results confirm that vanilla CP (used by CONFEX-Naive) fails to reach the target coverage, while localised CP with a suitably chosen kernel bandwidth succeeds. For small bandwidths (i.e., "strong" locality), all three choices of α attain or are close to the target coverage level, but the gap grows as the bandwidth increases and localisation diminishes. For $\alpha = 0.01$ and small bandwidths, no data is obtained since no test points produced a singleton prediction region (as required by our CONFEX constraints). These figures demonstrate that picking a correct bandwidth is crucial for obtaining good plausibility and coverage guarantees.

6 Conclusions

We introduced a novel MILP-based framework for generating uncertainty-aware counterfactual explanations with formal, distribution-free guarantees. By developing an efficient encoding of localised conformal prediction, we address the critical issue of exchangeability violation in the CFX search process. This allows us to enforce approximate test-conditional guarantees, ensuring the generation of provably reliable, plausible, and robust explanations.

Limitations Since our approach uses MILP to solve for CFXs, it will struggle scaling to very large models; gradient-based methods like Wachter and ECCCo are less prone to this problem, but they sacrifice guarantees on CFX validity. Moreover, CP requires a held-out calibration dataset, which may be problematic when data is scarce. Fortunately, CP guarantees hold regardless of the calibration set size (but small sets will lead to more conservative prediction regions). Finally, picking an appropriate kernel bandwidth is an additional task which requires domain knowledge or evaluation on a validation set.

References

James M Adams, Gesine Reinert, Lukasz Szpruch, Carsten Maple, and Andrew Elliott. Individualised counterfactual examples using conformal prediction intervals. *arXiv preprint arXiv:2505.22326*, 2025.

[2]For very small α (0.01) and small kernel bandwidths, we observe low plausibility: we conjecture this could be due to the CP method localising on outlier points.

Patrick Altmeyer, Mojtaba Farmanbar, Arie van Deursen, and Cynthia CS Liem. Faithful model explanations through energy-constrained conformal counterfactuals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 10829–10837, 2024.

Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. *arXiv preprint arXiv:2006.06848*, 2020.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. The limits of distribution-free conditional predictive inference, 2020. URL https://arxiv.org/abs/1903.04684.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996.

Reuben Binns, Max Van Kleek, Michael Veale, Ulrik Lyngs, Jun Zhao, and Nigel Shadbolt. 'it's reducing a human being to a percentage': Perceptions of justice in algorithmic decisions. In *CHI*, pp. 377. ACM, 2018.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104, 2000.

Luben MC Cabezas, Mateus P Otto, Rafael Izbicki, and Rafael B Stern. Regression trees for fast and adaptive prediction intervals. *Information Sciences*, 686:121369, 2025.

Susanne Dandl, Christoph Molnar, Martin Binder, and Bernd Bischl. Multi-objective counterfactual explanations. In *International conference on parallel problem solving from nature*, pp. 448–469. Springer, 2020.

Tiffany Ding, Anastasios Angelopoulos, Stephen Bates, Michael Jordan, and Ryan J Tibshirani. Class-conditional conformal prediction with many classes. *Advances in neural information processing systems*, 36:64555–64576, 2023.

Sanghamitra Dutta, Jason Long, Saumitra Mishra, Cecilia Tilli, and Daniele Magazzeni. Robust counterfactual explanations for tree-based ensembles. In *International conference on machine learning*, pp. 5742–5756. PMLR, 2022.

Lilian Edwards and Michael Veale. Slave to the algorithm? why a 'right to an explanation' is probably not the remedy you are looking for. volume 16, pp. 1–65, 2017.

Isaac Gibbs, John J Cherian, and Emmanuel J Candès. Conformal prediction with conditional guarantees. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, pp. qkaf008, 2025.

Leying Guan. Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50, 2023.

Gurobi, 2022. URL https://github.com/Gurobi/gurobi-machinelearning.

Hans Hofmann. Statlog (German Credit Data). UCI Machine Learning Repository, 1994.

Rohan Hore and Rina Foygel Barber. Conformal prediction with local weights: randomization enables local guarantees. *arXiv preprint arXiv:2310.07850*, 2023.

Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Batch multivalid conformal prediction. *arXiv preprint arXiv:2209.15145*, 2022.

Kentaro Kanamori, Takuya Takagi, Ken Kobayashi, and Hiroki Arimura. Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization. In *IJCAI*, pp. 2855–2862, 2020.

Amir-Hossein Karimi, Gilles Barthe, Bernhard Schölkopf, and Isabel Valera. A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects. In *Proceedings of the 2021 International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Ana Lucic, Harrie Oosterhuis, Hinda Haned, and Maarten de Rijke. Focus: Flexible optimizable counterfactual explanations for tree ensembles, 2021. URL https://arxiv.org/abs/1911.12199.

Kyosuke Morita, 2023. URL https://github.com/kyosek/CFXplorer.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3): 291–297, 1997.

Martin Pawelczyk, Sascha Bielawski, Johannes van den Heuvel, Tobias Richter, and Gjergji Kasneci. Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms, 2021.

Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. Face: Feasible and actionable counterfactual explanations. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 344–350, 2020.

Lisa Schut, Oscar Key, Rory McGrath, Luca Costabello, Bogdan Sacaleanu, Medb Corcoran, and Yarin Gal. Generating interpretable counterfactual explanations by implicit minimisation of epistemic and aleatoric uncertainties. In *AISTATS*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1756–1764. PMLR, 2021.

Andrew D. Selbst and Solon Barocas. The intuitive appeal of explainable machines. *Fordham Law Review*, 87: 1085, 2018.

Martin Skrodzki. The kd tree data structure and a proof for neighborhood computation in expected logarithmic time. *arXiv preprint arXiv:1903.04936*, 2019.

David Stutz, Krishnamurthy, Dvijotham, Ali Taylan Cemgil, and Arnaud Doucet. Learning optimal conformal classifiers, 2022. URL https://arxiv.org/abs/2110.09192.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Gabriele Tolomei, Fabrizio Silvestri, Andrew Haines, and Mounia 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*, KDD '17, pp. 465–474. ACM, August 2017. doi: 10.1145/3097983.3098039. URL http://dx.doi.org/10.1145/3097983.3098039.

Srikumar Verma, John P Dickerson, and Kevin Hines. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*, 2020.

Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pp. 475–490. PMLR, 2012.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer Nature, 2022.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

A Appendix

AI Use Declaration Generative AI tools were used sparingly for polishing writing.

A.1 Related Works

Our work integrates three research areas: counterfactual explanations (CFXs), uncertainty quantification in explanations, and the application of conformal prediction (CP) to optimization problems. Counterfactual explanations, introduced by Wachter et al. (2017), provide recourse by identifying minimal feature changes to alter a model's prediction. While initial work focused on validity and distance, the field has expanded to include desiderata like plausibility and actionability (Verma et al., 2020; Karimi et al., 2021). Methodologies have also diversified from gradient-based optimization to tree-specific algorithms (Tolomei et al., 2017; Lucic et al., 2021) and constraint-based methods using Mixed-Integer Linear Programming (MILP) (Kanamori et al., 2020). However, a critical limitation of many approaches is their failure to account for model uncertainty, which can result in misleading or brittle explanations (Schut et al., 2021). To address this, prior works have employed Bayesian methods (Antorán et al., 2020) or model ensembles (Schut et al., 2021). CONFEX contributes a novel, principled alternative by using Conformal Prediction. More relevant is ECCCo (Altmeyer et al., 2024), which uses a loss term based on the conformal set size (Stutz et al., 2022) but crucially does not address the violation of the exchangeability assumption inherent in the CFX search process.

A.2 MILP encoding of localised CP

The following algorithm Algorithm 2 computes the LCP quantile value in MILP. To do this, all calibration scores and calibration points must be accessible to the optimiser. Variables are constrained as distances from the test point to each calibration point, and another set of variables compute the corresponding weight according to the L1 kernel. These weights are used alongside calibration scores to identify the desired weighted quantile. This encoding is linear in the size of the calibration set.

Algorithm 2: Localised CP constraints in MILP

Input : Calibration dataset $\{(x_i, y_i)\}_{i=1}^n$, corresponding scores $\{s_i\}_{i=1}^n$, test input x^*, L1 localisation kernel with bandwidth h, level $\alpha \in (0, 1)$

Output: Local quantile $q_{1-\alpha}^{LCP}$

1 Sort $\{(x_i, y_i)\}_{i=1}^n$ in ascending order w.r.t. scores.

2 Add n real variables $d_1, \ldots, d_n$.

3 For $i = 1, \ldots, n$, add the L1 distance constraint $d_i = \|x_i - x^*\|_1$.

4 Add n binary variables $w_1, \ldots, w_n$ as the weights induced by the L1 kernel.

5 For $i = 1, \ldots, n$, add the constraint $w_i = \mathbf{1}(d_i \leq h)$, implemented for arbitrarily large $M > 0$ as

$$d_i \leq h + M(1 - w_i) \wedge d_i \geq h - Mw_i$$

6 Add n binary variables $in_1, \ldots, in_n$; each in_i keeps track if the score s_i is below the quantile.

7 Add integer variables W and $W_{1-\alpha}$ denoting, respectively, the sum of all weights and of those weights whose score is below the quantile.

8 Add constraints $W = \sum_{i=1}^n w_i$, $W_{1-\alpha} = \sum_{i=1}^n in_i \cdot w_i$ and $W_{1-\alpha} \geq \lceil (1-\alpha)W \rceil$. The latter expresses that the scores below the quantile have probability at least $1 - \alpha$.

9 Define $W'_{1-\alpha} = \sum_{i=1}^n (1 - in_i) \cdot w_i$ and add constraint $W'_{1-\alpha} \geq \lfloor \alpha W \rfloor$

10 Solve constraints and return s_k.

11 $q_{1-\alpha}^{LCP}$ will be the largest calibration score s_i for which $in_i = 1$. To identify it, add an integer variable $k \in \{1, \ldots, n\}$.

12 For $i = 1, \ldots, n$, add the constraint $in_i = \mathbf{1}(i \leq k)$ using a big-M encoding as done in line 5.

A.3 METRICS

In order to evaluate the quality of the generated counterfactual explanations, we adopt a set of quantitative metrics that measure different aspects of their usefulness and reliability. Specifically, we focus on three core dimensions: *plausibility*, *sensitivity*, and *stability*. In addition, we report auxiliary metrics such as the distance of counterfactuals to the original instance, the proportion of failures, and the validity rate of generated explanations. Together, these metrics provide a comprehensive view of both the fidelity and robustness of counterfactual explanations.

Plausibility. A counterfactual explanation should lie close to the underlying data distribution so that it represents a realistic and interpretable alternative. To assess this, we measure plausibility using the Local Outlier Factor (LOF) (Breunig et al., 2000), which quantifies how isolated a sample is with respect to its nearest neighbours. A LOF score of $+1$ indicates that the counterfactual is consistent with observed data, whereas -1 suggest that the counterfactual is implausible. We use the `scikit-learn` implementation of LOF with `novelty=True` and $n\_\text{neighbors} = 20$, stratified by the target class.

Sensitivity. Beyond plausibility, we also want to assess whether counterfactuals are *robust* to small changes in the input instance. Sensitivity measures how much a counterfactual explanation changes when the original instance x is perturbed within a small neighbourhood. Formally, given an input x and its counterfactual x_c, we uniformly sample a perturbed instance $x' \sim U_b(x)$ from the ℓ_2 centred around the factual, compute a new counterfactual x'_c. Sensitivity is then defined as the relative deviation between the two counterfactuals, normalised by the cost of the initial counterfactual:

$$\text{CFX Sensitivity} = \mathbb{E}_{x' \sim U_b(x)} \left[\frac{\|x'_c - x_c\|_2}{\|x_c - x\|_2} \right].$$

In practice, we sample 4 neighbours from 25 test points to inform our sensitivity metric. Intuitively, low sensitivity indicates that the explanation remains stable when the factual input undergoes small variations, thereby suggesting robustness and consistency.

In our experiments, we choose the budget b of the uniform sampling to correspond to a ball with 0.1% of the volume of the feature space.

$$V_{\text{ball}} = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)} r^d = b V_{\text{total}}$$

where d is the number of non-categorical features in the space. Solving for r,

$$r = \left(\frac{b V_{\text{total}}}{\pi^{d/2} / \Gamma\left(\frac{d}{2}+1\right)} \right)^{1/d}$$

This allows the same budget to be used across datasets with differing numbers of features. When sampling neighbours, we do not change categorical values and we fix ordinal values to their closest valid value.

Stability. Complementary to sensitivity, stability measures how consistent the counterfactual is under perturbations applied directly to the counterfactual itself. That is, we perturb x_c within a budgeted neighbourhood and evaluate the variance in the model predictions across these perturbed samples. Following an adaptation of (Dutta et al., 2022), stability is computed as:

$$\text{CFX Stability} = \frac{1}{K} \sum_{x' \in N_x} \hat{f}\left(x'\right)_{y^+} - \sqrt{\frac{1}{K} \sum_{x' \in N_x} \left(\hat{f}\left(x'\right)_{y^+} - \frac{1}{K} \sum_{x' \in N_x} \hat{f}\left(x'\right)_{y^+} \right)^2},$$

where N_x is a set of K points sampled as $x' \sim U_b(x_c)$.

where $\hat{f}\left(x'\right)_{y^+}$ refers to the predicted probability of the target class. The metric neighbours a large mean value for the predicted probability of sampled neighbours, whilst penalising variations in these

values by subtracting the standard deviation to ensure that that mean is not a combination of very high and very low values. Similarly to the Sensivity metric, $U_b(x_c)$ denotes sampling from the ℓ_2 ball centred around the counterfactual, computing the radius in the same way, taking the budget to represent 0.1% of the total feature volume.

Stability is high when the predictions across perturbed counterfactuals remain close to each other, which indicates that the explanation is not overly sensitive to minor fluctuations in its actualisation.

Auxiliary metrics. In addition to the three core dimensions, we report the following supplementary measures:

- *Distance:* the average L1 distance between the original instance and the counterfactual,
$$\text{Distance} = \mathbb{E}\big(\|x' - x\|_1\big),$$
which quantifies the minimality of the intervention required.
- *Validity:* the proportion of counterfactuals that successfully change the prediction to the desired class,
$$\text{Validity} = \mathbb{E}(1\{\hat{f}(x') = y^+\}).$$
For example, invalidity could be due to numerical artefacts in encoding the models in MILP, or failure for SGD procedures to converge to a flipped class. We report whenever a method a method produces less than 90% validity, and exclude invalid CFXs from the computation of other metrics.
- *Failure rate:* the proportion of runs where the generator fails to produce a counterfactual, for example due to infeasible constraints in optimisation-based methods such as MILP.
- *Implausibility:* The average distance from the counterfactual to the closest 10% of points of the target class, similar to Altmeyer et al. (2024).

A.4 Experimental Setup

All generators, except FOCUS (using the CFXplorer package Morita (2023)) and FeatureTweak (implementation taken from CARLA Pawelczyk et al. (2021)), were implemented as part of a Python library to generate CFXs to be released in the future. For solving MILP instances, we utilise the Gurobi solver, and utilise the Gurobi Machine Learning Gurobi (2022) library to formulate the trained classifiers as constraints.

We evaluate our method on four widely-used datasets for classification tasks.

California Housing. We use the California Housing dataset Pace & Barry (1997) from the StatLib repository through scikit-learn's `sklearn.datasets.fetch_california_housing` function[3]. The original regression problem was changed into a binary classification task by categorizing houses based on whether the median income exceeds \$20,000 (42% above, 58% below). The dataset contains 8 numeric features, which we scaled to the range $(0, 1)$ using MinMax scaling.

GiveMeSomeCredit. This dataset, obtained through Kaggle[4], contains credit scoring data with 8 numeric features that were scaled to $(0, 1)$ using MinMax scaling. Because this dataset is larger, we used a batch size of 256 for training.

German Credit. We use the German Credit dataset from the UCI Machine Learning Repository Hofmann (1994), with the cleaned version obtained through Kaggle[5]. The preprocessing included: (i) scaling numeric features (Age, Credit amount, Duration) to $(0, 1)$ using MinMax scaling, (ii) ordinal encoding of categorical features (job: 0-3, savings account: 0-4, checking account: binary), then normalised, and (iii) replacing missing values in savings and checking account fields with 'none'. The Purpose feature was excluded from our analysis.

Adult Income. This dataset Becker & Kohavi (1996), obtained through Kaggle[6], predicts whether an individual's income exceeds \$50,000. We processed the following features: numeric features

[3] `https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html`
[4] `https://www.kaggle.com/competitions/GiveMeSomeCredit`
[5] `https://www.kaggle.com/datasets/uciml/german-credit/data`
[6] `https://www.kaggle.com/datasets/wenruliu/adult-income-dataset`

(Age, Capital Gain, Capital Loss, Hours per week) scaled to $(0, 1)$, ordinal features (education), and categorical features (Workclass, Occupation, Race, Relationship, Gender, Marital status) using one-hot encoding. Similar to GiveMeSomeCredit, we used a batch size of 256 for this dataset.

Model Configuration. For all datasets, we used a multilayer perceptron (MLP) with 50 hidden units. The batch size was set to 64 for California Housing and German Credit, trained for 100 epochs, and 256 for GiveMeSomeCredit and Adult Income, trained for 50 epochs. For the random forest model, we also evaluated a Random Forest classifier with 5 estimators and number of leaves limited to 500 for the GiveMeSomeCredit and AdultIncome models.

B Further results

B.1 Further discussion of Table 1

For GermanCredit, whilst Wachter obtained the closest counterfactuals, had a validity rate of 84%, demonstrating how gradient-based methods may fail to correctly change prediction to the target class. ECCCo (79%) and FeatureTweak (52%) also suffered validity issues. On the other hand, MILP-MinDistalways found a valid counterfactual, including satisfying correct categorical and ordinal encoding unlike some of the competing tree generators, and this is reflected with an increased distance.

Note that in all figures, bandwidth is measured as multiples of the median pairwise distance between all points in the dataset.

The rest of this section displays results with respect to all metrics featured in Appendix A.3.

B.2 California Housing

B.2.1 Model evaluation results

Repeat	Accuracy (%)	Precision (%)	F1 Score (%)
repeat0, MLP	83.58	83.61	83.59
repeat1, MLP	82.95	83.59	82.95
repeat0, RF	78.05	80.60	77.83
repeat1, RF	78.10	80.60	77.90

Table 2: Model evaluation results, CaliforniaHousing.

B.2.2 CFX GENERATION RESULTS

	Distance	Plausibility	Implausibility	Sensitivity (10^{-1})	Stability
MinDist	0.02 ± 0.00	0.38 ± 0.04	0.21 ± 0.00	41.83 ± 8.98	0.06 ± 0.02
CNaive,0.01	0.06 ± 0.01	0.09 ± 0.05	0.22 ± 0.00	3.27 ± 0.19	0.07 ± 0.02
CNaive,0.05	0.04 ± 0.01	0.34 ± 0.00	0.21 ± 0.00	8.01 ± 1.58	0.06 ± 0.02
CNaive,0.1	0.03 ± 0.01	0.36 ± 0.04	0.21 ± 0.00	15.70 ± 2.36	0.06 ± 0.02
CTree0.01,0.05*	1.39 ± 0.01	-1.00 ± 0.00	0.39 ± 0.00	0.00 ± 0.00	1.00 ± 0.00
CTree0.01,0.1	1.00 ± 0.00	-0.26 ± 0.00	0.16 ± 0.01	0.02 ± 0.01	0.06 ± 0.03
CTree0.01,0.15	0.47 ± 0.01	0.16 ± 0.02	0.16 ± 0.01	0.08 ± 0.01	0.09 ± 0.01
CTree0.01,0.2	0.19 ± 0.01	0.36 ± 0.04	0.16 ± 0.00	0.58 ± 0.34	0.10 ± 0.02
CTree0.05,0.05	0.53 ± 0.02	0.80 ± 0.02	0.15 ± 0.00	0.09 ± 0.07	0.10 ± 0.02
CTree0.05,0.1	0.19 ± 0.01	0.62 ± 0.04	0.16 ± 0.00	0.44 ± 0.15	0.10 ± 0.03
CTree0.05,0.15	0.13 ± 0.01	0.62 ± 0.06	0.17 ± 0.00	1.09 ± 0.67	0.10 ± 0.03
CTree0.05,0.2	0.09 ± 0.01	0.62 ± 0.04	0.18 ± 0.00	2.54 ± 0.93	0.08 ± 0.02
CTree0.1,0.05	0.21 ± 0.01	0.72 ± 0.02	0.16 ± 0.00	0.26 ± 0.09	0.10 ± 0.03
CTree0.1,0.1	0.12 ± 0.01	0.65 ± 0.01	0.18 ± 0.00	1.40 ± 0.98	0.09 ± 0.02
CTree0.1,0.15	0.09 ± 0.01	0.61 ± 0.07	0.18 ± 0.00	1.94 ± 1.36	0.08 ± 0.02
CTree0.1,0.2	0.07 ± 0.01	0.55 ± 0.05	0.19 ± 0.00	3.78 ± 0.82	0.07 ± 0.02
CTree0.01,0.25	0.15 ± 0.01	0.43 ± 0.07	0.16 ± 0.00	0.85 ± 0.33	0.10 ± 0.02
CTree0.01,0.3	0.13 ± 0.01	0.42 ± 0.08	0.18 ± 0.00	1.17 ± 0.37	0.09 ± 0.02
CTree0.01,0.35	0.10 ± 0.01	0.32 ± 0.08	0.18 ± 0.00	1.87 ± 0.78	0.09 ± 0.03
CTree0.01,0.4	0.08 ± 0.01	0.26 ± 0.02	0.19 ± 0.00	2.54 ± 0.53	0.08 ± 0.03
CTree0.05,0.25	0.08 ± 0.01	0.57 ± 0.01	0.18 ± 0.00	3.90 ± 0.31	0.08 ± 0.02
CTree0.05,0.3	0.07 ± 0.01	0.49 ± 0.05	0.19 ± 0.00	4.76 ± 0.76	0.08 ± 0.02
CTree0.05,0.35	0.06 ± 0.01	0.42 ± 0.04	0.19 ± 0.00	6.04 ± 0.36	0.07 ± 0.02
CTree0.05,0.4	0.05 ± 0.01	0.39 ± 0.05	0.20 ± 0.00	7.58 ± 1.03	0.07 ± 0.02
CTree0.1,0.25	0.06 ± 0.01	0.54 ± 0.04	0.19 ± 0.00	6.10 ± 1.08	0.07 ± 0.02
CTree0.1,0.3	0.06 ± 0.01	0.45 ± 0.01	0.19 ± 0.00	6.70 ± 1.35	0.07 ± 0.02
CTree0.1,0.35	0.05 ± 0.01	0.44 ± 0.02	0.19 ± 0.00	9.81 ± 0.27	0.07 ± 0.02
CTree0.1,0.4	0.04 ± 0.01	0.42 ± 0.04	0.20 ± 0.00	12.14 ± 1.80	0.07 ± 0.02
CTree0.01,0.45	0.08 ± 0.01	0.16 ± 0.08	0.20 ± 0.00	2.41 ± 0.03	0.08 ± 0.03
CTree0.01,0.5	0.08 ± 0.01	0.11 ± 0.09	0.20 ± 0.00	2.14 ± 0.31	0.08 ± 0.03
CTree0.05,0.45	0.05 ± 0.01	0.37 ± 0.07	0.20 ± 0.00	7.78 ± 1.44	0.07 ± 0.02
CTree0.05,0.5	0.05 ± 0.01	0.28 ± 0.08	0.20 ± 0.00	7.16 ± 0.70	0.07 ± 0.02
CTree0.1,0.45	0.04 ± 0.01	0.42 ± 0.04	0.20 ± 0.00	14.51 ± 0.21	0.07 ± 0.02
CTree0.1,0.5	0.04 ± 0.01	0.36 ± 0.08	0.20 ± 0.00	13.87 ± 0.66	0.07 ± 0.02
Wachter	0.08 ± 0.01	0.47 ± 0.07	0.20 ± 0.00	1.57 ± 0.26	0.07 ± 0.02
Schut	1.71 ± 0.18	-0.98 ± 0.02	0.82 ± 0.06	0.13 ± 0.00	0.46 ± 0.03
ECCCo,0.01	0.37 ± 0.02	-0.67 ± 0.01	0.20 ± 0.00	0.22 ± 0.02	0.33 ± 0.05
ECCCo,0.05	0.36 ± 0.01	-0.62 ± 0.04	0.20 ± 0.00	0.24 ± 0.01	0.32 ± 0.06
ECCCo,0.1	0.35 ± 0.01	-0.61 ± 0.03	0.20 ± 0.00	0.24 ± 0.01	0.32 ± 0.06

Table 3: CFX generation results, CaliforniaHousing, MLP. CTree0.01,0.05 has 59% failues

	Distance	Plausibility	Implausibility	Sensitivity (10^{-1})	Stability
MinDist	0.01 ± 0.00	0.45 ± 0.03	0.20 ± 0.00	35.36 ± 15.39	0.23 ± 0.02
CNaive,0.01	0.03 ± 0.01	0.47 ± 0.05	0.20 ± 0.00	9.92 ± 2.08	0.24 ± 0.02
CNaive,0.05	0.03 ± 0.01	0.47 ± 0.05	0.20 ± 0.00	9.92 ± 2.08	0.24 ± 0.02
CNaive,0.1	0.03 ± 0.01	0.47 ± 0.05	0.20 ± 0.00	9.92 ± 2.08	0.24 ± 0.02
CTree0.01,0.05	-	-	-	-	-
CTree0.01,0.1	-	-	-	-	-
CTree0.01,0.15	0.60 ± 0.11	0.59 ± 0.29	0.15 ± 0.00	0.10 ± 0.03	0.26 ± 0.02
CTree0.01,0.2	0.15 ± 0.01	0.45 ± 0.05	0.16 ± 0.00	0.68 ± 0.19	0.24 ± 0.02
CTree0.05,0.05	0.45 ± 0.00	0.47 ± 0.03	0.16 ± 0.00	0.10 ± 0.06	0.26 ± 0.03
CTree0.05,0.1	0.16 ± 0.00	0.61 ± 0.09	0.17 ± 0.00	0.50 ± 0.12	0.23 ± 0.01
CTree0.05,0.15	0.11 ± 0.01	0.60 ± 0.06	0.17 ± 0.00	1.01 ± 0.46	0.23 ± 0.01
CTree0.05,0.2	0.07 ± 0.00	0.57 ± 0.09	0.18 ± 0.00	3.21 ± 2.37	0.23 ± 0.02
CTree0.1,0.05	0.18 ± 0.01	0.65 ± 0.03	0.17 ± 0.00	0.42 ± 0.16	0.24 ± 0.01
CTree0.1,0.1	0.10 ± 0.01	0.64 ± 0.02	0.18 ± 0.01	2.29 ± 1.65	0.23 ± 0.02
CTree0.1,0.15	0.08 ± 0.01	0.58 ± 0.02	0.18 ± 0.00	2.01 ± 1.26	0.23 ± 0.02
CTree0.1,0.2	0.06 ± 0.01	0.50 ± 0.06	0.19 ± 0.00	4.65 ± 3.14	0.23 ± 0.03
FOCUS	0.09 ± 0.00	0.44 ± 0.06	0.20 ± 0.00	5.58 ± 1.73	0.26 ± 0.02
FeatureTweak	0.11 ± 0.01	0.46 ± 0.02	0.19 ± 0.01	0.53 ± 0.06	0.26 ± 0.02

Table 4: CFX generation results, CaliforniaHousing, RF. Note that methods with no data had 100% failures. FeatureTweak had 55% validity.

B.2.3 Conditional coverage results

This table reports the performance of different counterfactual generators under four evaluation settings: marginal coverage, class-conditional coverage, random binning, and counterfactual similarity. In the paper we discussed the counterfactual simulation, however we also evaluate the marginal coverage over a test set, average class-conditional coverage, average coverage over a random paritioning of the test set into 3 bins. In each case, we report the (averaged) coverage gap and set size

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000
CTree,0.01,0.15	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000
CTree,0.01,0.1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000
CTree,0.01,0.25	1.968 ± 0.007	1.970 ± 0.002	1.970 ± 0.002	1.924 ± 0.066
CTree,0.01,0.2	1.976 ± 0.000	1.979 ± 0.002	1.979 ± 0.002	1.934 ± 0.074
CTree,0.01,0.35	1.894 ± 0.005	1.877 ± 0.004	1.877 ± 0.004	1.617 ± 0.126
CTree,0.01,0.3	1.955 ± 0.002	1.953 ± 0.002	1.953 ± 0.002	1.895 ± 0.031
CTree,0.01,0.45	1.811 ± 0.014	1.815 ± 0.017	1.815 ± 0.017	1.372 ± 0.060
CTree,0.01,0.4	1.825 ± 0.034	1.829 ± 0.017	1.829 ± 0.017	1.393 ± 0.087
CTree,0.05,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000
CTree,0.05,0.15	1.977 ± 0.012	1.982 ± 0.001	1.982 ± 0.001	1.981 ± 0.009
CTree,0.05,0.1	1.987 ± 0.005	1.989 ± 0.001	1.989 ± 0.001	1.984 ± 0.008
CTree,0.05,0.25	1.907 ± 0.002	1.915 ± 0.004	1.915 ± 0.004	1.837 ± 0.026
CTree,0.05,0.2	1.947 ± 0.024	1.949 ± 0.002	1.949 ± 0.002	1.852 ± 0.075
CTree,0.05,0.35	1.667 ± 0.041	1.657 ± 0.036	1.657 ± 0.036	1.394 ± 0.002
CTree,0.05,0.3	1.864 ± 0.000	1.853 ± 0.012	1.853 ± 0.012	1.730 ± 0.063
CTree,0.05,0.45	1.584 ± 0.050	1.569 ± 0.040	1.569 ± 0.040	1.271 ± 0.005
CTree,0.05,0.4	1.618 ± 0.053	1.590 ± 0.039	1.590 ± 0.039	1.296 ± 0.029
CTree,0.1,0.05	1.999 ± 0.002	1.999 ± 0.000	1.999 ± 0.000	2.000 ± 0.000
CTree,0.1,0.15	1.950 ± 0.002	1.944 ± 0.001	1.944 ± 0.001	1.876 ± 0.055
CTree,0.1,0.1	1.966 ± 0.003	1.972 ± 0.001	1.972 ± 0.001	1.972 ± 0.015
CTree,0.1,0.25	1.823 ± 0.003	1.804 ± 0.018	1.804 ± 0.018	1.595 ± 0.019
CTree,0.1,0.2	1.892 ± 0.036	1.894 ± 0.007	1.894 ± 0.007	1.818 ± 0.033
CTree,0.1,0.35	1.533 ± 0.012	1.511 ± 0.023	1.511 ± 0.023	1.324 ± 0.041
CTree,0.1,0.3	1.691 ± 0.033	1.682 ± 0.021	1.682 ± 0.021	1.435 ± 0.011
CTree,0.1,0.45	1.434 ± 0.055	1.433 ± 0.025	1.433 ± 0.025	1.271 ± 0.005
CTree,0.1,0.4	1.465 ± 0.029	1.466 ± 0.024	1.466 ± 0.024	1.296 ± 0.029
CNaive,0.01	1.606 ± 0.129	1.607 ± 0.133	1.607 ± 0.133	1.000 ± 0.000
CNaive,0.05	1.353 ± 0.156	1.338 ± 0.147	1.338 ± 0.147	1.000 ± 0.000
CNaive,0.1	1.155 ± 0.086	1.154 ± 0.087	1.154 ± 0.087	1.000 ± 0.000

Table 5: Average set size, CaliforniaHousing, MLP.

B.3 German Credit

B.3.1 Model evaluation results

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
CTree,0.01,0.15	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
CTree,0.01,0.1	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
CTree,0.01,0.25	1.000 ± 0.000	0.817 ± 0.067	0.806 ± 0.068	0.552 ± 0.257
CTree,0.01,0.2	0.996 ± 0.005	0.908 ± 0.034	0.903 ± 0.034	0.770 ± 0.188
CTree,0.01,0.35	1.000 ± 0.000	0.630 ± 0.124	0.612 ± 0.137	-1.204 ± 2.672
CTree,0.01,0.3	0.996 ± 0.005	0.745 ± 0.062	0.734 ± 0.069	0.540 ± 0.274
CTree,0.01,0.45	0.995 ± 0.000	0.549 ± 0.220	0.522 ± 0.017	-3.457 ± 3.083
CTree,0.01,0.4	0.993 ± 0.003	0.549 ± 0.188	0.515 ± 0.017	-3.433 ± 3.083
CTree,0.05,0.05	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000
CTree,0.05,0.15	0.994 ± 0.005	4.802 ± 0.050	4.794 ± 0.051	4.782 ± 0.206
CTree,0.05,0.1	1.000 ± 0.000	4.909 ± 0.000	4.903 ± 0.000	4.976 ± 0.000
CTree,0.05,0.25	0.996 ± 0.005	4.307 ± 0.057	4.285 ± 0.051	4.491 ± 0.240
CTree,0.05,0.2	1.000 ± 0.000	4.635 ± 0.021	4.625 ± 0.017	4.709 ± 0.171
CTree,0.05,0.35	0.972 ± 0.002	2.259 ± 0.794	1.633 ± 0.068	1.475 ± 0.805
CTree,0.05,0.3	0.995 ± 0.000	3.877 ± 0.197	3.825 ± 0.188	4.419 ± 0.206
CTree,0.05,0.45	0.944 ± 0.000	3.400 ± 0.902	0.920 ± 0.308	-0.766 ± 1.233
CTree,0.05,0.4	0.960 ± 0.005	3.377 ± 0.902	0.750 ± 0.125	-0.741 ± 1.233
CTree,0.1,0.05	1.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000
CTree,0.1,0.15	0.998 ± 0.003	9.719 ± 0.067	9.709 ± 0.068	9.782 ± 0.206
CTree,0.1,0.1	0.998 ± 0.000	9.807 ± 0.016	9.794 ± 0.017	9.939 ± 0.017
CTree,0.1,0.25	0.982 ± 0.005	8.176 ± 0.330	8.110 ± 0.308	6.342 ± 1.199
CTree,0.1,0.2	0.990 ± 0.010	9.022 ± 0.071	8.995 ± 0.051	9.516 ± 0.137
CTree,0.1,0.35	0.919 ± 0.009	5.252 ± 1.335	1.909 ± 0.103	4.525 ± 1.439
CTree,0.1,0.3	0.970 ± 0.005	7.182 ± 0.511	7.093 ± 0.479	6.221 ± 1.165
CTree,0.1,0.45	0.902 ± 0.005	5.709 ± 1.376	1.284 ± 0.061	4.234 ± 1.233
CTree,0.1,0.4	0.917 ± 0.007	5.700 ± 1.358	1.594 ± 0.247	4.259 ± 1.233
CNaive,0.01	0.983 ± 0.007	0.963 ± 0.678	0.854 ± 0.770	-9.962 ± 0.668
CNaive,0.05	0.938 ± 0.033	2.496 ± 0.075	2.422 ± 0.021	-5.962 ± 0.668
CNaive,0.1	0.890 ± 0.026	2.443 ± 0.079	2.434 ± 0.059	-0.962 ± 0.668

Table 6: Coverage gap, CaliforniaHousing, MLP.

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.15	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.2	1.992 ± 0.005	1.991 ± 0.001	1.991 ± 0.001	1.000 ± 0.000
CTree,0.05,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	1.000 ± 0.000
CTree,0.05,0.15	1.985 ± 0.010	1.987 ± 0.002	1.987 ± 0.002	1.000 ± 0.000
CTree,0.05,0.1	1.992 ± 0.002	1.992 ± 0.001	1.992 ± 0.001	1.000 ± 0.000
CTree,0.05,0.2	1.971 ± 0.007	1.970 ± 0.005	1.970 ± 0.005	1.000 ± 0.000
CTree,0.1,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	1.000 ± 0.000
CTree,0.1,0.15	1.967 ± 0.019	1.967 ± 0.004	1.967 ± 0.004	1.000 ± 0.000
CTree,0.1,0.1	1.979 ± 0.002	1.981 ± 0.001	1.981 ± 0.001	1.000 ± 0.000
CTree,0.1,0.2	1.941 ± 0.015	1.943 ± 0.004	1.943 ± 0.004	1.000 ± 0.000
CNaive,0.01	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CNaive,0.05	1.540 ± 0.005	1.513 ± 0.011	1.513 ± 0.011	1.000 ± 0.000
CNaive,0.1	1.521 ± 0.029	1.513 ± 0.011	1.513 ± 0.011	1.000 ± 0.000

Table 7: Average set size, CaliforniaHousing, RF

B.3.2 CFX GENERATION RESULTS

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.15	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.1	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.2	1.000 ± 0.000	0.989 ± 0.016	0.988 ± 0.017	0.988 ± 0.017
CTree,0.05,0.05	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000
CTree,0.05,0.15	0.999 ± 0.002	4.898 ± 0.016	4.891 ± 0.017	-11.921 ± 4.402
CTree,0.05,0.1	1.000 ± 0.000	4.927 ± 0.071	4.927 ± 0.069	2.832 ± 3.032
CTree,0.05,0.2	0.996 ± 0.005	4.758 ± 0.145	4.746 ± 0.154	-27.570 ± 22.080
CTree,0.1,0.05	1.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000
CTree,0.1,0.15	0.998 ± 0.003	9.612 ± 0.351	9.588 ± 0.377	-8.580 ± 13.635
CTree,0.1,0.1	0.996 ± 0.002	9.812 ± 0.073	9.806 ± 0.069	-7.236 ± 1.182
CTree,0.1,0.2	0.994 ± 0.002	9.348 ± 0.050	9.334 ± 0.051	-33.193 ± 0.617
CNaive,0.01	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CNaive,0.05	0.945 ± 0.009	4.255 ± 0.060	0.921 ± 0.159	-42.481 ± 0.000
CNaive,0.1	0.951 ± 0.003	4.759 ± 0.133	4.477 ± 0.137	-37.481 ± 0.000

Table 8: Coverage gap, CaliforniaHousing, RF

Repeat	Accuracy (%)	Precision (%)	F1 Score (%)
repeat0,RF	70.00	68.27	68.77
repeat1,RF	69.50	68.31	68.76
repeat0,MLP	72.00	72.00	72.00
repeat1,MLP	71.00	70.01	70.39

Table 9: Model evaluation results, GermanCredit.

B.3.3 CONDITIONAL COVERAGE RESULTS

	Distance	Plausibility	Implausibility	Sensitivity (10^{-1})	Stability
MinDistance	1.69 ± 0.04	0.50 ± 0.06	0.73 ± 0.01	0.09 ± 0.01	0.58 ± 0.01
CNaive,0.01	2.19 ± 0.10	-0.06 ± 0.24	0.83 ± 0.02	0.03 ± 0.00	0.96 ± 0.03
CNaive,0.05	2.02 ± 0.00	0.01 ± 0.21	0.79 ± 0.02	0.06 ± 0.01	0.84 ± 0.06
CNaive,0.1	1.83 ± 0.05	0.23 ± 0.15	0.75 ± 0.02	0.08 ± 0.03	0.70 ± 0.03
CTree0.01,0.05	-	-	-	-	-
CTree0.01,0.1	-	-	-	-	-
CTree0.01,0.15	-	-	-	-	-
CTree0.01,0.2	-	-	-	-	-
CTree0.05,0.05	-	-	-	-	-
CTree0.05,0.1	-	-	-	-	-
CTree0.05,0.15	-	-	-	-	-
CTree0.05,0.2	2.48 ± 0.05	1.00 ± 0.00	0.27 ± 0.02	0.01 ± 0.01	0.89 ± 0.00
CTree0.1,0.05	2.58 ± 0.00	1.00 ± 0.00	0.34 ± 0.00	0.00 ± 0.00	0.46 ± 0.02
CTree0.1,0.1	2.37 ± 0.09	1.00 ± 0.00	0.24 ± 0.01	0.01 ± 0.00	0.83 ± 0.03
CTree0.1,0.15	3.14 ± 0.00	1.00 ± 0.00	0.31 ± 0.00	0.01 ± 0.00	0.64 ± 0.00
CTree0.1,0.2	2.34 ± 0.65	1.00 ± 0.00	0.38 ± 0.01	0.01 ± 0.00	0.62 ± 0.00
Wachter	0.40 ± 0.01	0.77 ± 0.03	0.59 ± 0.00	0.25 ± 0.00	0.22 ± 0.01
Schut	0.98 ± 0.05	-0.02 ± 0.04	0.80 ± 0.02	0.09 ± 0.00	0.67 ± 0.03
ECCCo,0.01	1.01 ± 0.03	0.16 ± 0.04	0.78 ± 0.02	0.05 ± 0.02	0.73 ± 0.01
ECCCo,0.05	0.98 ± 0.03	0.14 ± 0.10	0.77 ± 0.02	0.05 ± 0.02	0.72 ± 0.02
ECCCo,0.1	0.94 ± 0.01	0.21 ± 0.05	0.75 ± 0.01	0.05 ± 0.02	0.72 ± 0.02

Table 10: CFX generation results, GermanCredit, MLP. Note that methods with no data had 100% failues, Wachter and all ECCCo methods had 84% validity and Schut 79%.

B.4 GIVEMESOMECREDIT

B.4.1 MODEL EVALUATION RESULTS

	Distance	Plausibility	Implausibility.1	Sensitivity (10^{-1})	Stability
MinDist	1.69 ± 0.05	0.34 ± 0.12	0.77 ± 0.02	0.09 ± 0.01	0.37 ± 0.00
CNaive,0.01	1.71 ± 0.04	0.55 ± 0.11	0.71 ± 0.01	0.09 ± 0.02	0.45 ± 0.01
CNaive,0.05	1.71 ± 0.04	0.55 ± 0.11	0.71 ± 0.01	0.09 ± 0.02	0.45 ± 0.01
CNaive,0.1	1.72 ± 0.04	0.58 ± 0.08	0.71 ± 0.00	0.09 ± 0.02	0.45 ± 0.01
CTree0.01,0.15	-	-	-	-	-
CTree0.01,0.2	-	-	-	-	-
CTree0.01,0.25	-	-	-	-	-
CTree0.01,0.5	-	-	-	-	-
CTree0.01,0.75	2.35 ± 0.01	0.43 ± 0.17	0.65 ± 0.01	0.07 ± 0.02	0.44 ± 0.01
CTree0.05,0.15	-	-	-	-	-
CTree0.05,0.2	2.17 ± 0.19	1.00 ± 0.00	0.34 ± 0.06	0.02 ± 0.00	0.47 ± 0.27
CTree0.05,0.25	2.58 ± 0.70	1.00 ± 0.00	0.43 ± 0.09	0.28 ± 0.26	0.30 ± 0.01
CTree0.05,0.5	1.99 ± 0.04	0.75 ± 0.03	0.55 ± 0.03	0.05 ± 0.00	0.39 ± 0.01
CTree0.05,0.75	1.91 ± 0.13	0.63 ± 0.05	0.69 ± 0.01	0.09 ± 0.00	0.50 ± 0.03
CTree0.05,1	1.88 ± 0.06	0.56 ± 0.06	0.73 ± 0.01	0.09 ± 0.02	0.47 ± 0.02
CTree0.1,0.05	2.25 ± 0.07	1.00 ± 0.00	0.35 ± 0.01	0.01 ± 0.01	0.33 ± 0.21
CTree0.1,0.1	2.07 ± 0.02	1.00 ± 0.00	0.34 ± 0.00	0.00 ± 0.00	0.21 ± 0.03
CTree0.1,0.15	1.64 ± 0.08	1.00 ± 0.00	0.37 ± 0.01	0.05 ± 0.02	0.21 ± 0.01
CTree0.1,0.2	1.78 ± 0.26	1.00 ± 0.00	0.38 ± 0.03	0.03 ± 0.02	0.21 ± 0.03
CTree0.01,1	2.35 ± 0.03	0.42 ± 0.14	0.65 ± 0.02	0.08 ± 0.02	0.44 ± 0.01
CTree0.1,0.25	1.35 ± 0.07	0.98 ± 0.02	0.49 ± 0.02	0.13 ± 0.07	0.33 ± 0.03
CTree0.1,0.5	1.47 ± 0.05	0.79 ± 0.05	0.62 ± 0.00	0.10 ± 0.01	0.35 ± 0.03
CTree0.1,0.75	1.75 ± 0.04	0.50 ± 0.08	0.71 ± 0.00	0.10 ± 0.01	0.43 ± 0.03
CTree0.1,1	1.76 ± 0.07	0.50 ± 0.12	0.72 ± 0.00	0.09 ± 0.02	0.43 ± 0.02
FOCUS	0.54 ± 0.13	0.83 ± 0.01	0.58 ± 0.02	0.48 ± 0.02	0.26 ± 0.01
FeatureTweak	0.52 ± 0.05	0.82 ± 0.02	0.57 ± 0.03	0.09 ± 0.00	0.18 ± 0.01

Table 11: CFX generation results, GermanCredit, RF. Note that methods with no data had 100% failures. FeatureTweak had 52% validity.

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.15	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.25	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.2	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.15	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.2	1.975 ± 0.035	1.990 ± 0.014	1.990 ± 0.014	-
CTree,0.1,0.05	2.000 ± 0.000	1.995 ± 0.000	1.995 ± 0.000	1.000 ± 0.000
CTree,0.1,0.15	2.000 ± 0.000	1.985 ± 0.000	1.985 ± 0.000	1.000 ± 0.000
CTree,0.1,0.1	2.000 ± 0.000	1.995 ± 0.000	1.995 ± 0.000	1.000 ± 0.000
CTree,0.1,0.2	1.950 ± 0.000	1.935 ± 0.014	1.935 ± 0.014	1.000 ± 0.000
CNaive,0.01	1.950 ± 0.071	1.913 ± 0.025	1.913 ± 0.025	1.000 ± 0.000
CNaive,0.05	1.600 ± 0.071	1.737 ± 0.095	1.737 ± 0.095	1.000 ± 0.000
CNaive,0.1	1.450 ± 0.071	1.407 ± 0.004	1.407 ± 0.004	1.000 ± 0.000

Table 12: Average set size, GermanCredit, MLP.

B.4.2 CFX GENERATION RESULTS

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.15	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.1	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.2	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.05,0.05	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.15	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.1	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.2	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.1,0.05	1.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000
CTree,0.1,0.15	1.000 ± 0.000	9.640 ± 0.000	9.500 ± 0.000	7.500 ± 0.000
CTree,0.1,0.1	1.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000
CTree,0.1,0.2	1.000 ± 0.000	9.640 ± 0.000	9.500 ± 0.000	7.500 ± 0.000
CNaive,0.01	1.000 ± 0.000	1.280 ± 1.159	1.900 ± 0.424	-56.000 ± 4.950
CNaive,0.05	0.925 ± 0.035	4.608 ± 1.088	3.250 ± 0.354	-46.500 ± 0.707
CNaive,0.1	0.925 ± 0.106	12.236 ± 0.509	5.500 ± 2.121	-33.750 ± 4.596

Table 13: Coverage gap, GermanCredit, MLP.

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.15	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.25	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.2	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.5	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,0.75	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.01,1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.15	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.1	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.25	2.000 ± 0.000	1.990 ± 0.014	1.990 ± 0.014	-
CTree,0.05,0.2	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CTree,0.05,0.5	1.950 ± 0.071	1.940 ± 0.028	1.940 ± 0.028	-
CTree,0.05,0.75	1.800 ± 0.141	1.865 ± 0.148	1.865 ± 0.148	-
CTree,0.05,1	1.850 ± 0.212	1.862 ± 0.152	1.862 ± 0.152	-
CTree,0.1,0.05	2.000 ± 0.000	1.998 ± 0.004	1.998 ± 0.004	-
CTree,0.1,0.15	2.000 ± 0.000	1.993 ± 0.011	1.993 ± 0.011	-
CTree,0.1,0.1	1.975 ± 0.035	1.998 ± 0.004	1.998 ± 0.004	-
CTree,0.1,0.25	1.900 ± 0.071	1.873 ± 0.018	1.873 ± 0.018	-
CTree,0.1,0.2	1.950 ± 0.071	1.945 ± 0.014	1.945 ± 0.014	-
CTree,0.1,0.5	1.850 ± 0.071	1.677 ± 0.011	1.677 ± 0.011	-
CTree,0.1,0.75	1.675 ± 0.106	1.710 ± 0.071	1.710 ± 0.071	-
CTree,0.1,1	1.725 ± 0.106	1.708 ± 0.067	1.708 ± 0.067	-
CNaive,0.01	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CNaive,0.05	2.000 ± 0.000	2.000 ± 0.000	2.000 ± 0.000	-
CNaive,0.1	1.700 ± 0.141	1.677 ± 0.025	1.677 ± 0.025	-

Table 14: Average set size, GermanCredit, RF

B.4.3 CONDITIONAL COVERAGE RESULTS

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.15	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.1	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.25	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.2	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.5	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,0.75	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.01,1	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CTree,0.05,0.05	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.15	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.1	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.25	1.000 ± 0.000	4.820 ± 0.254	4.750 ± 0.354	-
CTree,0.05,0.2	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CTree,0.05,0.5	0.975 ± 0.035	3.730 ± 0.638	3.750 ± 1.061	-
CTree,0.05,0.75	0.900 ± 0.071	5.009 ± 2.331	3.250 ± 1.061	-
CTree,0.05,1	0.975 ± 0.035	5.009 ± 2.331	3.750 ± 0.354	-
CTree,0.1,0.05	1.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000	-
CTree,0.1,0.15	1.000 ± 0.000	9.820 ± 0.254	9.750 ± 0.354	-
CTree,0.1,0.1	1.000 ± 0.000	10.000 ± 0.000	10.000 ± 0.000	-
CTree,0.1,0.25	1.000 ± 0.000	5.232 ± 0.438	7.250 ± 1.061	-
CTree,0.1,0.2	0.975 ± 0.035	9.051 ± 0.325	9.000 ± 0.000	-
CTree,0.1,0.5	1.000 ± 0.000	7.168 ± 0.580	4.750 ± 1.768	-
CTree,0.1,0.75	0.900 ± 0.071	8.887 ± 3.152	5.000 ± 0.707	-
CTree,0.1,1	0.925 ± 0.035	8.887 ± 3.152	5.000 ± 0.707	-
CNaive,0.01	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	-
CNaive,0.05	1.000 ± 0.000	5.000 ± 0.000	5.000 ± 0.000	-
CNaive,0.1	0.975 ± 0.035	9.526 ± 2.247	4.250 ± 0.354	-

Table 15: Coverage gap, GermanCredit, RF.

Repeat	Accuracy (%)	Precision (%)	F1 Score (%)
repeat0,MLP	93.54	91.82	91.81
repeat1,MLP	93.49	91.79	91.96
repeat0,RF	93.40	91.57	91.78
repeat1,RF	93.40	91.53	91.69

Table 16: Model evaluation results, GiveMeSomeCredit.

B.5 Adult Income

B.5.1 Model evaluation results

	Distance	Plausibility	Implausibility	Sensitivity (10^{-1})	Stability
MinDist	0.03 ± 0.00	0.94 ± 0.00	0.09 ± 0.00	1.16 ± 0.07	0.17 ± 0.03
CNaive,0.01	0.25 ± 0.00	-0.96 ± 0.02	0.21 ± 0.01	0.20 ± 0.03	0.12 ± 0.07
CNaive,0.05	0.04 ± 0.00	0.76 ± 0.04	0.09 ± 0.00	0.76 ± 0.02	0.17 ± 0.03
CNaive,0.1	0.09 ± 0.01	-0.03 ± 0.43	0.11 ± 0.00	0.46 ± 0.03	0.16 ± 0.03
CTree0.01,0.05	0.24 ± 0.01	0.75 ± 0.25	0.07 ± 0.00	0.02 ± 0.01	0.10 ± 0.02
CTree0.01,0.1	0.18 ± 0.00	0.94 ± 0.00	0.07 ± 0.00	0.10 ± 0.02	0.19 ± 0.02
CTree0.01,0.15	0.23 ± 0.01	0.95 ± 0.01	0.08 ± 0.01	0.12 ± 0.02	0.18 ± 0.03
CTree0.01,0.2	0.15 ± 0.01	0.89 ± 0.07	0.08 ± 0.00	0.20 ± 0.00	0.17 ± 0.04
CTree0.05,0.05	0.31 ± 0.05	0.96 ± 0.02	0.06 ± 0.00	0.06 ± 0.01	0.21 ± 0.01
CTree0.05,0.1	0.07 ± 0.01	0.96 ± 0.02	0.08 ± 0.00	0.54 ± 0.06	0.18 ± 0.03
CTree0.05,0.15	0.05 ± 0.00	0.92 ± 0.04	0.08 ± 0.00	0.61 ± 0.04	0.17 ± 0.03
CTree0.05,0.2	0.05 ± 0.00	0.92 ± 0.04	0.09 ± 0.00	0.62 ± 0.00	0.17 ± 0.03
CTree0.1,0.05	0.23 ± 0.06	0.98 ± 0.00	0.08 ± 0.01	0.13 ± 0.03	0.20 ± 0.01
CTree0.1,0.1	0.06 ± 0.01	0.95 ± 0.01	0.08 ± 0.00	0.66 ± 0.05	0.17 ± 0.03
CTree0.1,0.15	0.04 ± 0.00	0.95 ± 0.01	0.09 ± 0.00	0.78 ± 0.01	0.17 ± 0.03
CTree0.1,0.2	0.03 ± 0.00	0.95 ± 0.01	0.09 ± 0.00	0.76 ± 0.01	0.17 ± 0.03
Wachter	0.08 ± 0.00	0.92 ± 0.00	0.09 ± 0.00	0.98 ± 0.03	0.18 ± 0.03
Schut	0.57 ± 0.32	-0.32 ± 0.40	0.43 ± 0.20	0.56 ± 0.38	0.16 ± 0.15
ECCCo,0.01	0.62 ± 0.29	-0.99 ± 0.01	0.27 ± 0.14	0.23 ± 0.01	0.24 ± 0.03
ECCCo,0.05	0.58 ± 0.25	-0.98 ± 0.00	0.25 ± 0.12	0.24 ± 0.01	0.24 ± 0.03
ECCCo,0.1	0.58 ± 0.26	-0.98 ± 0.00	0.26 ± 0.13	0.24 ± 0.00	0.23 ± 0.02

Table 17: CFX generation results, GiveMeSomeCredit, MLP. Note Wachter had validity 28%, Schut 80%

	Distance	Plausibility	Implausibility	Sensitivity	Stability
MinDistanceCF	0.01 ± 0.00	0.95 ± 0.01	0.09 ± 0.00	100.94 ± 11.42	0.29 ± 0.01
CNaive,0.01	0.04 ± 0.00	0.86 ± 0.00	0.09 ± 0.00	1.23 ± 0.05	0.30 ± 0.01
CNaive,0.05	0.01 ± 0.00	0.95 ± 0.01	0.09 ± 0.00	42.08 ± 36.87	0.29 ± 0.01
CNaive,0.1	0.01 ± 0.00	0.96 ± 0.00	0.09 ± 0.00	40.40 ± 35.97	0.29 ± 0.01
CTree0.01,0.05	0.39 ± 0.01	0.49 ± 0.49	0.07 ± 0.01	0.01 ± 0.00	0.22 ± 0.04
CTree0.01,0.1	0.08 ± 0.01	0.95 ± 0.01	0.07 ± 0.00	0.38 ± 0.06	0.29 ± 0.01
CTree0.01,0.15	0.07 ± 0.00	0.96 ± 0.04	0.07 ± 0.00	0.47 ± 0.03	0.29 ± 0.01
CTree0.01,0.2	0.05 ± 0.00	0.96 ± 0.02	0.08 ± 0.00	0.71 ± 0.14	0.29 ± 0.01
CTree0.05,0.05	0.11 ± 0.01	0.96 ± 0.02	0.07 ± 0.00	0.35 ± 0.07	0.29 ± 0.02
CTree0.05,0.1	0.03 ± 0.00	0.97 ± 0.01	0.08 ± 0.00	1.29 ± 0.17	0.29 ± 0.01
CTree0.05,0.15	0.03 ± 0.00	0.97 ± 0.01	0.08 ± 0.00	1.77 ± 0.49	0.29 ± 0.01
CTree0.05,0.2	0.02 ± 0.00	0.97 ± 0.01	0.08 ± 0.00	1.91 ± 0.27	0.29 ± 0.01
CTree0.1,0.05	0.07 ± 0.00	0.99 ± 0.01	0.08 ± 0.00	0.56 ± 0.12	0.29 ± 0.01
CTree0.1,0.1	0.03 ± 0.00	0.98 ± 0.02	0.08 ± 0.00	1.37 ± 0.18	0.29 ± 0.01
CTree0.1,0.15	0.02 ± 0.00	0.97 ± 0.01	0.08 ± 0.00	4.83 ± 2.84	0.29 ± 0.01
CTree0.1,0.2	0.02 ± 0.00	0.98 ± 0.02	0.09 ± 0.00	5.01 ± 2.90	0.29 ± 0.01
FOCUS	0.05 ± 0.00	0.92 ± 0.02	0.09 ± 0.00	2.14 ± 0.84	0.30 ± 0.01
FeatureTweak	0.03 ± 0.00	0.96 ± 0.00	0.09 ± 0.00	1.37 ± 0.07	0.26 ± 0.01

Table 18: CFX generation results, GiveMeSomeCredit, RF. Note FeatureTweak had validity 46%. CFX0.01,0.05 had 49% failures (i.e. no singleton regions existed for one label at these alpha and bandwidth levels)

B.5.2 CFX GENERATION RESULTS

Generator	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.998 ± 0.001	1.998 ± 0.000	1.998 ± 0.000	1.000 ± 0.000
CTree,0.01,0.15	1.806 ± 0.010	1.800 ± 0.001	1.800 ± 0.001	1.000 ± 0.000
CTree,0.01,0.1	1.810 ± 0.002	1.813 ± 0.001	1.813 ± 0.001	1.000 ± 0.000
CTree,0.01,0.2	1.667 ± 0.010	1.667 ± 0.009	1.667 ± 0.009	1.000 ± 0.000
CTree,0.05,0.05	1.736 ± 0.001	1.727 ± 0.000	1.727 ± 0.000	1.000 ± 0.000
CTree,0.05,0.15	1.328 ± 0.005	1.322 ± 0.000	1.322 ± 0.000	1.000 ± 0.000
CTree,0.05,0.1	1.361 ± 0.002	1.349 ± 0.001	1.349 ± 0.001	1.000 ± 0.000
CTree,0.05,0.2	1.211 ± 0.009	1.208 ± 0.002	1.208 ± 0.002	1.000 ± 0.000
CTree,0.1,0.05	1.497 ± 0.005	1.492 ± 0.000	1.492 ± 0.000	1.000 ± 0.000
CTree,0.1,0.15	1.173 ± 0.004	1.170 ± 0.001	1.170 ± 0.001	1.000 ± 0.000
CTree,0.1,0.1	1.205 ± 0.003	1.203 ± 0.000	1.203 ± 0.000	1.000 ± 0.000
CTree,0.1,0.2	1.102 ± 0.002	1.103 ± 0.000	1.103 ± 0.000	1.000 ± 0.000
CNaive,0.01	1.465 ± 0.004	1.471 ± 0.001	1.471 ± 0.001	1.000 ± 0.000
CNaive,0.05	1.031 ± 0.004	1.031 ± 0.002	1.031 ± 0.002	1.000 ± 0.000
CNaive,0.1	0.939 ± 0.007	0.940 ± 0.001	0.940 ± 0.001	1.000 ± 0.000

Table 19: Average set size, GiveMeSomeCredit, MLP

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.000 ± 0.000	0.976 ± 0.000	0.997 ± 0.000	0.804 ± 0.000
CTree,0.01,0.15	0.997 ± 0.001	3.045 ± 0.052	0.585 ± 0.007	-9.372 ± 0.138
CTree,0.01,0.1	0.996 ± 0.000	2.825 ± 0.017	0.615 ± 0.002	-8.907 ± 0.242
CTree,0.01,0.2	0.993 ± 0.001	4.856 ± 0.259	0.338 ± 0.035	0.058 ± 0.031
CTree,0.05,0.05	0.993 ± 0.000	5.479 ± 0.035	4.253 ± 0.005	-14.472 ± 0.069
CTree,0.05,0.15	0.973 ± 0.001	19.022 ± 0.131	2.365 ± 0.002	-56.163 ± 10.446
CTree,0.05,0.1	0.978 ± 0.000	16.801 ± 0.038	2.702 ± 0.002	-31.957 ± 5.063
CTree,0.05,0.2	0.967 ± 0.000	23.661 ± 0.582	1.667 ± 0.057	-37.027 ± 1.018
CTree,0.1,0.05	0.986 ± 0.001	11.022 ± 0.006	8.478 ± 0.012	-48.498 ± 48.317
CTree,0.1,0.15	0.959 ± 0.008	27.718 ± 0.313	6.057 ± 0.005	-43.707 ± 4.978
CTree,0.1,0.1	0.968 ± 0.003	24.271 ± 0.205	6.628 ± 0.002	-50.795 ± 13.815
CTree,0.1,0.2	0.953 ± 0.002	31.741 ± 0.547	5.410 ± 0.019	-36.828 ± 0.738
CNaive,0.01	0.992 ± 0.000	6.274 ± 0.017	0.185 ± 0.012	-13.384 ± 0.899
CNaive,0.05	0.950 ± 0.002	35.012 ± 1.558	0.347 ± 0.075	-28.925 ± 5.145
CNaive,0.1	0.897 ± 0.009	47.846 ± 0.342	0.418 ± 0.026	-3.522 ± 10.552

Table 20: Coverage gap, GiveMeSomeCredit, MLP

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.998 ± 0.000	1.999 ± 0.000	1.999 ± 0.000	1.995 ± 0.000
CTree,0.01,0.15	1.773 ± 0.000	1.775 ± 0.005	1.775 ± 0.005	1.759 ± 0.010
CTree,0.01,0.1	1.788 ± 0.005	1.785 ± 0.006	1.785 ± 0.006	1.763 ± 0.010
CTree,0.01,0.2	1.633 ± 0.012	1.632 ± 0.002	1.632 ± 0.002	1.591 ± 0.005
CTree,0.05,0.05	1.720 ± 0.005	1.712 ± 0.002	1.712 ± 0.002	1.717 ± 0.002
CTree,0.05,0.15	1.290 ± 0.004	1.291 ± 0.003	1.291 ± 0.003	1.271 ± 0.004
CTree,0.05,0.1	1.326 ± 0.006	1.317 ± 0.004	1.317 ± 0.004	1.287 ± 0.004
CTree,0.05,0.2	1.181 ± 0.001	1.180 ± 0.001	1.180 ± 0.001	1.156 ± 0.002
CTree,0.1,0.05	1.459 ± 0.012	1.455 ± 0.002	1.455 ± 0.002	1.468 ± 0.022
CTree,0.1,0.15	1.109 ± 0.006	1.109 ± 0.003	1.109 ± 0.003	1.126 ± 0.001
CTree,0.1,0.1	1.147 ± 0.013	1.144 ± 0.002	1.144 ± 0.002	1.144 ± 0.002
CTree,0.1,0.2	1.036 ± 0.000	1.042 ± 0.001	1.042 ± 0.001	1.069 ± 0.001
CNaive,0.01	1.348 ± 0.005	1.349 ± 0.002	1.349 ± 0.002	1.000 ± 0.000
CNaive,0.05	1.036 ± 0.002	1.033 ± 0.002	1.033 ± 0.002	1.000 ± 0.000
CNaive,0.1	0.940 ± 0.007	0.942 ± 0.002	0.942 ± 0.002	1.000 ± 0.000

Table 21: Average set size, GiveMeSomeCredit, RF

	Marginal	Class Conditional	Random Binning	Counterfactual Sim
CTree,0.01,0.05	1.000 ± 0.000	0.988 ± 0.017	0.998 ± 0.002	1.000 ± 0.000
CTree,0.01,0.15	0.995 ± 0.001	2.838 ± 0.179	0.565 ± 0.012	-1.789 ± 0.069
CTree,0.01,0.1	0.995 ± 0.002	2.796 ± 0.108	0.585 ± 0.007	-1.740 ± 0.000
CTree,0.01,0.2	0.994 ± 0.001	4.403 ± 0.063	0.352 ± 0.021	-4.088 ± 0.761
CTree,0.05,0.05	0.986 ± 0.000	5.080 ± 0.116	3.558 ± 0.078	1.086 ± 0.069
CTree,0.05,0.15	0.959 ± 0.002	18.307 ± 0.004	1.105 ± 0.007	-5.445 ± 0.450
CTree,0.05,0.1	0.962 ± 0.001	16.091 ± 0.421	1.428 ± 0.045	-5.323 ± 0.346
CTree,0.05,0.2	0.955 ± 0.003	23.382 ± 0.025	0.532 ± 0.035	-8.283 ± 0.450
CTree,0.1,0.05	0.958 ± 0.009	9.461 ± 0.114	5.690 ± 0.108	2.637 ± 0.173
CTree,0.1,0.15	0.913 ± 0.008	25.681 ± 0.027	1.577 ± 0.085	-4.310 ± 0.519
CTree,0.1,0.1	0.925 ± 0.009	21.947 ± 0.073	2.225 ± 0.002	-4.139 ± 0.623
CTree,0.1,0.2	0.905 ± 0.002	30.943 ± 0.144	0.905 ± 0.082	-5.313 ± 0.415
CNaive,0.01	0.991 ± 0.002	7.387 ± 0.000	0.157 ± 0.042	-15.952 ± 0.104
CNaive,0.05	0.949 ± 0.002	35.148 ± 0.149	0.347 ± 0.057	-11.952 ± 0.104
CNaive,0.1	0.895 ± 0.007	46.000 ± 0.162	0.418 ± 0.021	-6.952 ± 0.104

Table 22: Coverage gap, GiveMeSomeCredit, RF.

Repeat	Accuracy (%)	Precision (%)	F1 Score (%)
repeat0,RF	85.73	85.20	85.14
repeat1,RF	85.32	84.76	84.72
repeat0,MLP	85.41	85.05	85.17
repeat1,MLP	85.04	84.70	84.83

Table 23: Model evaluation results, AdultIncome.

	Distance	Plausibility	Implausibility	Sensitivity (10^{-1})	Stability
MinDistance	1.23 ± 0.02	-0.17 ± 0.05	2.00 ± 0.00	0.06 ± 0.01	0.31 ± 0.00
CNaive,0.01	1.27 ± 0.07	-0.16 ± 0.02	2.00 ± 0.00	0.05 ± 0.00	0.41 ± 0.01
CNaive,0.05	1.24 ± 0.05	-0.18 ± 0.02	2.00 ± 0.00	0.05 ± 0.01	0.35 ± 0.00
CNaive,0.1	1.23 ± 0.02	-0.16 ± 0.04	2.00 ± 0.01	0.06 ± 0.01	0.33 ± 0.00
CTree0.01,0.05	1.86 ± 0.08	-0.16 ± 0.12	1.84 ± 0.00	0.04 ± 0.01	0.27 ± 0.00
CTree0.01,0.1	1.22 ± 0.00	-0.02 ± 0.04	1.87 ± 0.01	0.06 ± 0.00	0.32 ± 0.01
CTree0.01,0.15	1.05 ± 0.01	-0.08 ± 0.00	1.90 ± 0.00	0.05 ± 0.01	0.33 ± 0.02
CTree0.01,0.2	1.17 ± 0.02	-0.12 ± 0.04	1.95 ± 0.01	0.05 ± 0.01	0.34 ± 0.01
CTree0.05,0.05	1.19 ± 0.02	0.01 ± 0.01	1.89 ± 0.01	0.07 ± 0.00	0.27 ± 0.00
CTree0.05,0.1	1.07 ± 0.01	-0.05 ± 0.03	1.94 ± 0.01	0.06 ± 0.01	0.28 ± 0.00
CTree0.05,0.15	1.09 ± 0.03	-0.10 ± 0.02	1.96 ± 0.01	0.06 ± 0.00	0.30 ± 0.01
CTree0.05,0.2	1.12 ± 0.04	-0.20 ± 0.00	1.97 ± 0.01	0.05 ± 0.00	0.33 ± 0.01
CTree0.1,0.05	1.09 ± 0.04	-0.02 ± 0.06	1.94 ± 0.01	0.07 ± 0.00	0.26 ± 0.01
CTree0.1,0.1	1.09 ± 0.03	-0.11 ± 0.03	1.95 ± 0.01	0.06 ± 0.01	0.29 ± 0.01
CTree0.1,0.15	1.08 ± 0.03	-0.13 ± 0.03	1.96 ± 0.01	0.06 ± 0.01	0.30 ± 0.00
CTree0.1,0.2	1.06 ± 0.02	-0.16 ± 0.02	1.94 ± 0.01	0.06 ± 0.01	0.32 ± 0.00
Wachter	0.39 ± 0.05	0.36 ± 0.02	1.87 ± 0.01	0.14 ± 0.01	0.14 ± 0.02
Schut	0.85 ± 0.02	0.11 ± 0.07	2.07 ± 0.00	0.01 ± 0.00	0.84 ± 0.01
ECCCo,0.01	0.57 ± 0.01	0.13 ± 0.01	1.88 ± 0.01	0.05 ± 0.00	0.37 ± 0.01
ECCCo,0.05	0.57 ± 0.00	0.12 ± 0.02	1.88 ± 0.01	0.05 ± 0.00	0.37 ± 0.02
ECCCo,0.1	0.56 ± 0.01	0.12 ± 0.02	1.88 ± 0.01	0.05 ± 0.00	0.37 ± 0.02

Table 24: CFX generation results, AdultIncome, using categorical 'Workclass' to split.

	Distance	Plausibility	Implausibility	Sensitivity (10^{-1})	Stability
MinDist	0.93 ± 0.00	-0.02 ± 0.08	1.94 ± 0.00	0.14 ± 0.02	0.26 ± 0.01
CNaive,0.01	1.43 ± 0.20	-0.19 ± 0.11	1.91 ± 0.04	0.09 ± 0.02	0.31 ± 0.02
CNaive,0.05	1.03 ± 0.07	0.01 ± 0.01	1.95 ± 0.01	0.08 ± 0.01	0.26 ± 0.01
CNaive,0.1	0.97 ± 0.08	-0.08 ± 0.06	1.95 ± 0.02	0.13 ± 0.02	0.26 ± 0.01
CTree0.01,0.05	1.67 ± 0.17	-0.08 ± 0.06	1.84 ± 0.02	0.05 ± 0.01	0.19 ± 0.03
CTree0.01,0.1	1.39 ± 0.17	-0.01 ± 0.17	1.80 ± 0.01	0.11 ± 0.01	0.25 ± 0.02
CTree0.01,0.15	1.28 ± 0.25	-0.03 ± 0.19	1.83 ± 0.01	0.09 ± 0.01	0.26 ± 0.02
CTree0.01,0.2	1.41 ± 0.20	-0.16 ± 0.14	1.92 ± 0.02	0.11 ± 0.00	0.26 ± 0.02
CTree0.05,0.05	0.96 ± 0.04	0.13 ± 0.05	1.90 ± 0.00	0.11 ± 0.03	0.18 ± 0.00
CTree0.05,0.1	0.95 ± 0.10	0.04 ± 0.04	1.92 ± 0.01	0.08 ± 0.02	0.23 ± 0.01
CTree0.05,0.15	0.92 ± 0.08	0.08 ± 0.12	1.93 ± 0.01	0.14 ± 0.04	0.22 ± 0.01
CTree0.05,0.2	0.94 ± 0.01	-0.06 ± 0.02	1.96 ± 0.01	0.10 ± 0.01	0.24 ± 0.02
CTree0.1,0.05	0.75 ± 0.06	0.17 ± 0.07	1.90 ± 0.00	0.22 ± 0.08	0.18 ± 0.01
CTree0.1,0.1	0.85 ± 0.02	0.04 ± 0.06	1.93 ± 0.01	0.11 ± 0.00	0.22 ± 0.01
CTree0.1,0.15	0.89 ± 0.01	0.00 ± 0.06	1.94 ± 0.00	0.11 ± 0.01	0.23 ± 0.01
CTree0.1,0.2	0.93 ± 0.03	0.00 ± 0.08	1.91 ± 0.01	0.09 ± 0.01	0.25 ± 0.01
FOCUS	0.55 ± 0.12	0.36 ± 0.06	1.86 ± 0.00	0.21 ± 0.09	0.17 ± 0.00
FeatureTweak	0.34 ± 0.07	0.23 ± 0.09	1.87 ± 0.01	0.05 ± 0.01	0.14 ± 0.00

Table 25: CFX generation results, AdultIncome, RF. Note that FeatureTweak had a validity of 73%