Counterfactually Fair Conformal Prediction

Ozgur Guldogan
UC Santa Barbara

Neeraj Sarna
Munich RE

Yuanyuan Li
Munich RE

Michael Berger
Munich RE

Abstract

While counterfactual fairness of point predictors is well studied, its extension to prediction *sets*—central to fair decision-making under uncertainty—remains underexplored. On the other hand, conformal prediction (CP) provides efficient, distribution-free, finite-sample valid prediction sets, yet does not ensure counterfactual fairness. We close this gap by developing *Counterfactually Fair Conformal Prediction* (CF-CP) that produces counterfactually fair prediction sets. Through symmetrization of conformity scores across protected-attribute interventions, we prove that CF-CP results in counterfactually fair prediction sets while maintaining the marginal coverage property. Furthermore, we empirically demonstrate that on both synthetic and real datasets, across regression and classification tasks, CF-CP achieves the desired counterfactual fairness and meets the target coverage rate with minimal increase in prediction set size. CF-CP offers a simple, training-free route to counterfactually fair uncertainty quantification.

1 INTRODUCTION

As machine learning models are being increasingly used in high-stakes applications like healthcare, criminal justice, test score estimation, etc.; it is crucial to quantify and mitigate the associated bias (Barocas and Selbst, 2016; Podesta et al., 2014; Corbett-Davies et al., 2023). This problem has been extensively explored in the past for both regression and classification problems (Berk et al., 2018; Chouldechova, 2017; Agarwal et al., 2019). The general idea is to ensure that subpopulations are treated equally (group fairness) or focus on individual fairness such as counterfactual fairness (Kusner et al., 2017). Although these fairness mitigation techniques are effective, most of them focus on point predictions and fail to capture the inherent uncertainty in ML models. Moreover, recent work (Cresswell et al., 2024) has shown that conformal prediction sets can improve decision quality in high-stakes tasks, underscoring the practical importance of uncertainty-aware prediction sets beyond point predictors.

An appealing methodology that promotes model fairness under uncertainty is the idea of *fair prediction sets* (Romano et al., 2020a; Liu et al., 2022). These methods provide marginal coverage guarantees while enforcing a group-level fairness criterion. Typically, they adapt classical group-fairness metrics for point predictors (e.g., demographic parity) to the set-valued setting via inclusion-based definitions (Vadlamani et al., 2025). To the best of our knowledge, all these past works on fair prediction sets have focused on group-level fairness. Furthermore, recent human-subject studies (Cresswell et al., 2025) suggest that enforcing certain group-fairness criteria for prediction sets, such as equalized coverage, may inadvertently increase disparate impact, highlighting the need for alternative fairness notions for conformal prediction beyond group-level criteria. Thus, although group-level fairness is a powerful tool, certain applications demand individual-level guarantees. Consider healthcare, for instance, where the intention could be to ensure a patient's recommended treatment is not systematically influenced by protected attributes; such concerns have been widely discussed in the context of fair machine learning for laboratory medicine (Azimi and Zaydman, 2023). A method can satisfy group-level parity yet still produce biased recommendations for particular individuals, which could be undesirable for such an application. Furthermore, as noted by Kusner et al. (2017), the notion of counterfactual fairness better captures the causality of bias in ML algorithms.

The above motivates prediction sets with individual-

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

level fairness. We extend counterfactual fairness notion to set-valued predictors: for a given individual, the prediction set should remain unchanged under counterfactual interventions on the protected attribute. For example, consider constructing a prediction set for a particular student's SAT score; a counterfactually fair prediction set would be identical for that student whether we intervene to change their protected attribute (e.g., gender) or not.

We focus on conformal prediction (CP) and develop a framework called *Counterfactually Fair Conformal Prediction* (CF-CP). Concretely, we compute conformity scores under each intervention on the protected attribute and then aggregate these per-intervention scores using a symmetric function to obtain a counterfactually fair score. We then use these scores with the standard split CP procedure. We show that under an invertible structural causal model (SCM) and the exchangeability assumptions, CF-CP guarantees both marginal coverage and set-level counterfactual fairness. We emphasize that the latter holds even when the underlying point predictor is not counterfactually fair. Empirically, we evaluate CF-CP on synthetic and real-world datasets, showing that it significantly reduces unfairness in the prediction sets compared to standard split CP, with only a modest increase in average set size.

We note that prediction sets generated by CP are not inherently fair. They can systematically differ between protected groups or change when we hypothetically alter an individual's sensitive attributes (Romano et al., 2020a; Vadlamani et al., 2025). CF-CP alters the scoring function of standard CP such that counterfactual fairness and coverage is ensured.

Our key contributions are summarized as follows:

- We extend the notion of counterfactual fairness to set-valued predictors and propose a post-training conformal prediction procedure (CF-CP) that symmetrizes conformity scores across protected attribute interventions.
- We prove that under an invertible SCM and exchangeability assumptions, CF-CP guarantees both marginal coverage and set-level counterfactual fairness.
- We empirically demonstrate that CF-CP effectively reduces counterfactual unfairness in prediction sets on synthetic and real-world datasets while maintaining desired coverage.

1.1 Related Work

Fairness in conformal prediction has primarily focused on group fairness notions (Romano et al., 2020a). Vadlamani et al. (2025) propose a method that adapts common group fairness notions (e.g., demographic parity, equalized odds) to conformal prediction without requiring group membership during test time. One line of work tailors CP-based regression to group-fairness goals, specifically Demographic Parity and Equal Opportunity (Liu et al., 2022; Wang et al., 2023). A separate line advances CP methods that seek equalized coverage across groups (Lu et al., 2022). Our work differs in that we focus on individual-level counterfactual fairness for set-valued predictors. Moreover, recent human-subject studies by Cresswell et al. (2025) show that enforcing equalized coverage across groups can inadvertently increase disparate impact in decision-making, further underscoring the need for alternative fairness notions for prediction sets. There is also a growing body of work that combines causality with uncertainty quantification, though not in the context of fairness. Lei and Candès (2021); Schröder et al. (2024); Chen et al. (2024) study conformal prediction for potential outcomes in causal inference. However, these methods do not consider fairness. We next summarize counterfactual fairness methods for point predictors.

Counterfactual fairness (CF), introduced by Kusner et al. (2017), is an individual-level fairness notion that requires a predictor's output to remain invariant under counterfactual changes to protected attributes. There are several strategies to design CF predictors. Kusner et al. (2017) propose to learn a predictor only on the features that are not descendants of the protected attribute A in the causal graph. Zuo et al. (2023) propose a method that achieves CF by mixing representations of factual and counterfactual instances before feeding to a predictor. Zhou et al. (2024b) propose a method that combines both factual and counterfactual predictions weighted by the probability of the corresponding protected attribute value. Di Stefano et al. (2020); Garg et al. (2019) study regularizers that penalize differences in predictions between factual and counterfactual instances. These methods focus on point predictors, while we extend CF to set-valued predictors and design a conformal prediction procedure that enforces this property.

Paper organization. The rest of the paper is organized as follows. In Section 2, we review background on conformal prediction. Section 3 introduces counterfactual fairness, first for point predictors and then extended to prediction sets. In Section 4, we present our CF-CP method, discuss its computational properties, and establish theoretical guarantees. Section 5 reports

empirical results on synthetic and real datasets. We conclude in Section 6.

2 BACKGROUND: CONFORMAL PREDICTION

Notation and Setup. We consider observed features $X \in \mathcal{X}$, a protected attribute $A \in \mathcal{A}$, and a label $Y \in \mathcal{Y}$. Throughout the paper, we assume $\mathcal{A}$ is finite (e.g., binary or categorical). We write a (possibly randomized) predictor as $\hat{f} : \mathcal{X} \times \mathcal{A} \to \mathcal{S}$, where $\mathcal{S} = \mathbb{R}$ for regression or $\mathcal{S} = \Delta^{K-1}$ for K-class classification. A *prediction set* procedure maps (X, A) to a subset $\mathcal{C}(X, A) \subseteq \mathcal{Y}$.

For causal notation, we adopt structural causal models (SCMs) (Pearl, 2009). An SCM is a tuple (U, V, F), where U are exogenous variables, $V = \{X, A, Y\}$ are endogenous variables, and F are structural equations that determine each variable in V as a function of its parents and some exogenous noise. Counterfactuals consider the following question: *"What would the value of observed variables be, had we intervened to set A to a different value?"* We denote by $X_{A \leftarrow a}$ the counterfactual version of X under the intervention $do(A = a)$, and similarly for $Y_{A \leftarrow a}$. For more details on SCMs and counterfactuals, see Pearl (2009).

2.1 Split Conformal Prediction

Here we briefly describe conformal prediction (CP); see Angelopoulos et al. (2023) for a recent survey. Conformal prediction (CP) constructs prediction sets with a pre-specified probability $1-\alpha$ of containing the true label, without distributional assumptions beyond exchangeability (Vovk et al., 2005; Shafer and Vovk, 2008). For simplicity, we focus on the popular *split* (or inductive) CP variant (Papadopoulos et al., 2002). The protected attribute A is not needed for standard CP, but we include it in the notation for clarity and to make explicit the dependence used in later sections. Suppose we are given a pre-trained predictor $\hat{f}$ (e.g., a neural network) and a calibration dataset of size n, $\{(X_i, A_i, Y_i)\}_{i=1}^n$, independent of $\hat{f}$. The goal is to construct a prediction set $\mathcal{C}_\alpha(X, A)$ that covers the true label Y with probability of at least $1 - \alpha$ over the randomness of (X, A, Y) and the calibration dataset. To this end, we define a *conformity score* function $s : \mathcal{X} \times \mathcal{A} \times \mathcal{Y} \to \mathbb{R}$ that measures how well a candidate label y conforms with the features (x, a) under the predictor $\hat{f}$. Without loss of generality, we use a negative-oriented score, where smaller values indicate better conformity. The score function inherently depends on the predictor $\hat{f}$ but for ease of notation we omit this dependence. Then, one computes conformity scores on the calibration set:

$$S_i = s\left(X_i, A_i, Y_i\right), \qquad i \in \mathcal{I}_{\text{cal}}, \tag{1}$$

where $\mathcal{I}_{\text{cal}}$ denotes the index set of the calibration data. Let $\hat{q}_{1-\alpha}$ be the $(1-\alpha)(1+1/|\mathcal{I}_{\text{cal}}|)$-empirical quantile of $\{S_i\}_{i \in \mathcal{I}_{\text{cal}}}$. For a test point (X_{n+1}, A_{n+1}), the split conformal prediction set is computed as follows:

$$\mathcal{C}_\alpha(X_{n+1}, A_{n+1}) = \{y \in \mathcal{Y} : s(X_{n+1}, A_{n+1}, y) \leq \hat{q}_{1-\alpha}\} \tag{2}$$

If (X_i, A_i, Y_i) and $(X_{n+1}, A_{n+1}, Y_{n+1})$ are exchangeable, then $\mathcal{C}_\alpha(X_{n+1}, A_{n+1})$ satisfies the *marginal coverage* guarantee:

$$\Pr\left\{Y_{n+1} \in \mathcal{C}_\alpha(X_{n+1}, A_{n+1})\right\} \geq 1 - \alpha. \tag{3}$$

If the score values are almost surely distinct, the probability above is upper bounded by $1 - \alpha + 1/(n+1)$. A detailed proof of this result can be found in Lei et al. (2018). The coverage guarantee is marginal, and the randomness is over both the calibration and test points.

3 COUNTERFACTUAL FAIRNESS

In this section, we formalize counterfactual fairness for point predictors and extend this definition to prediction set procedures. We also briefly recall past works that ensure counterfactual fairness for point predictors.

3.1 Point predictors

Building on the causal framework of Pearl (2009), Kusner et al. (2017) introduced the notion of counterfactual fairness (CF) which requires that a predictor's output remains invariant under counterfactual interventions on protected attributes. More formally:

Definition 3.1 (Counterfactual Fairness—point predictor)**.** A predictor $\hat{f} : \mathcal{X} \times \mathcal{A} \to \mathcal{S}$ is *counterfactually fair* (CF) if, for all $a, a' \in \mathcal{A}$ and $s \in \mathcal{S}$,

$$\Pr\left\{\hat{f}(X_{A \leftarrow a}, a) = s \mid A = a, U = u\right\} = \Pr\left\{\hat{f}(X_{A \leftarrow a'}, a') = s \mid A = a, U = u\right\} \tag{4}$$

That is, after intervening on A while holding the latent U fixed, the distribution of predictions should not change. Equivalently, $\hat{f}(X_{A \leftarrow a}, a) \mid (A = a, U = u)$ and $\hat{f}(X_{A \leftarrow a'}, a') \mid (A = a, U = u)$ have the same conditional distribution.

Several strategies exist to design CF predictors $\hat{f}$ satisfying (4). Kusner et al. (2017) propose to learn a

predictor on only on the features that are not descendants of A in the causal graph, then no causal path exists from A to $\hat{f}(X, A)$, ensuring CF. This approach may be impractical if all features are descendants of A. Counterfactually Fair Representations (CFR) (Zuo et al., 2023) satisfies fairness by learning a latent representation Z for factual and counterfactual instances, then combines them via a permutation-invariant operation (e.g., mean) before feeding to a predictor. Zhou et al. (2024b) propose a method that satisfies CF by combining both factual and counterfactual predictions weighted by the probability of the corresponding protected attribute value, called Plug-in Counterfactual Fairness (PCF).

3.2 Prediction sets

We extend counterfactual fairness notion from *point predictors* to *prediction sets* by requiring that the set produced for an individual remain invariant under counterfactual interventions on the protected attribute. We call this property *set-level counterfactual fairness*.

Definition 3.2 (Counterfactual fairness—prediction sets)**.** A set-valued procedure $\mathcal{C}_\alpha$ is *counterfactually fair* if, for all $a, a' \in \mathcal{A}$ and $y \in \mathcal{Y}$,

$$\Pr\{y \in \mathcal{C}_\alpha(X_{A\leftarrow a}, a) \mid A = a, U = u\} = \Pr\{y \in \mathcal{C}_\alpha(X_{A\leftarrow a'}, a') \mid A = a, U = u\}. \quad (5)$$

That is, for any individual with latent attributes $U = u$ and protected attribute $A = a$, the probability of including any label y in the prediction set is invariant to counterfactual changes in A.

4 COUNTERFACTUALLY FAIR CONFORMAL PREDICTION

We aim to design a split-CP procedure that preserves marginal coverage while enforcing set-level counterfactual invariance. Precisely, given a pre-trained (possibly counterfactually unfair) predictor $\hat{f}$, a calibration set $\mathcal{I}_{\text{cal}}$, and a target level $1-\alpha$, construct a conformal prediction procedure $\mathcal{C}_\alpha$ that (i) preserves the standard marginal coverage guarantee and (ii) satisfies set-level counterfactual fairness (Def. 3.2).

In this section, we propose an algorithm that satisfies these two properties. Then we compare CF-CP to baselines and discuss the differences. Finally, we demonstrate that this procedure results in counterfactually fair prediction sets along with marginal coverage guarantees.

Algorithm 1 Split CF-CP via score symmetrization

Require: Base predictor $\hat{f}$, score $s(\cdot)$, calibration set $\{(X_i, A_i, Y_i)\}_{i\in\mathcal{I}_{\text{cal}}}$, target $1-\alpha$, aggregator Agg

1: Define

$$s_{\text{cf}}(x, a, y) \leftarrow \text{Agg}\{s(x_{A\leftarrow a'}, a', y) : a' \in \mathcal{A}\}$$

2: **For each** $i \in \mathcal{I}_{\text{cal}}$: compute $S_i^{\text{cf}} \leftarrow s_{\text{cf}}(X_i, A_i, Y_i)$

3: Set $\hat{q}_{1-\alpha} \leftarrow \text{Quantile}_{\lceil(|\mathcal{I}_{\text{cal}}|+1)(1-\alpha)\rceil/|\mathcal{I}_{\text{cal}}|}\left(\{S_i^{\text{cf}}\}\right)$

4: **At test time** for (x, a), form

$$\mathcal{C}_\alpha^{\text{CF}}(x, a) \leftarrow \{y : s_{\text{cf}}(x, a, y) \le \hat{q}_{1-\alpha}\}$$

Ensure: Prediction set $\mathcal{C}_\alpha^{\text{CF}}(x, a)$

4.1 Score Symmetrization and CF-CP Sets

Let $s : \mathcal{X}\times\mathcal{A}\times\mathcal{Y} \to \mathbb{R}$ be any conformity score used by split CP with a base predictor $\hat{f}$. We define the *counterfactual score symmetrization* by aggregating scores over protected-attribute interventions:

$$s_{\text{cf}}(x, a, y) = \text{Agg}\{s(x_{A\leftarrow a'}, a', y) : a' \in \mathcal{A}\} \quad (6)$$

where Agg is a symmetric operator that is invariant to permutations of its inputs and $x_{A\leftarrow a'}$ denotes the counterfactual features obtained by setting $A{:=}a'$. Examples of the operator Agg include mean, max and min operations—we elaborate more on this below.

We then run split CP *with* s_{cf}: on calibration points $\{(X_i, A_i, Y_i)\}_{i\in\mathcal{I}_{\text{cal}}}$ compute $S_i^{\text{cf}} = s_{\text{cf}}(X_i, A_i, Y_i)$, take the usual quantile $\hat{q}_{1-\alpha}$ with finite sample correction. At test time, for a new point (x, a), the CF-CP set is

$$\mathcal{C}_\alpha^{\text{CF}}(x, a) = \{y \in \mathcal{Y} : s_{\text{cf}}(x, a, y) \le \hat{q}_{1-\alpha}\}. \quad (7)$$

This procedure is summarized in Algorithm 1.

Aggregator choices. The only requirement on Agg is to be permutation-invariant–i.e., the output of the aggregator should not change with the order of the inputs. In this paper, we consider three simple choices: (i) `mean`, (ii) `max`, and (iii) `min`. During the calibration threshold learning (step 3 of Algorithm 1), `max` leads to larger thresholds $\hat{q}_{1-\alpha}$, because it takes the maximum score across interventions for each sample in calibration dataset. However, at test time (step 4 of Algorithm 1), the `max` aggregator leads to intersection of the per-intervention sets, which balances out the effect of a larger threshold. Conversely, `min` leads to smaller thresholds at the calibration phase, but the test-time sets are unions of per-intervention sets, which can lead to larger prediction sets. The `mean` aggregator is a compromise between these cases in both calibration and test phases. While we focus on these simple aggregators, more complex choices are possible as long as they are permutation-invariant.

Computational overhead. Compared to standard split-CP, CF-CP introduces two sources of additional overhead: (i) generating counterfactual interventions, and (ii) computing conformity scores for each intervention $a \in \mathcal{A}$. Overall, CF-CP incurs an additional $\mathcal{O}(|\mathcal{A}|)$ factor compared to split-CP. Thus, when the protected attribute has small cardinality, the computational cost of CF-CP remains close to that of split-CP.

4.2 Baselines and Comparisons

We compare CF-CP to two types of baselines that either retrain a fair point predictor or apply post-hoc set manipulations, and we note their limitations.

Retraining-based approaches. One option is to first obtain a point predictor that is counterfactually fair—e.g., by excluding descendants of A in the SCM or via counterfactually fair representations (CFR) / plug-in counterfactual fairness (PCF)—and then apply standard split-CP on its score. This can yield set-level invariance by construction, but it typically trades predictive accuracy for fairness and thus inflates set sizes to maintain coverage; it also requires retraining, which may be infeasible. In addition, PCF variant relies on access to the probability distribution of the protected attribute, $\Pr(A)$, for post-training adjustments.

Post-hoc union over interventions. A second baseline constructs separate CP sets under each protected-attribute intervention and unions them:

$$\mathcal{C}_\alpha^{\cup}(x) := \bigcup_{a' \in \mathcal{A}} \mathcal{C}_\alpha(x_{A \leftarrow a'}, a') \tag{8}$$

This is again counterfactually fair by construction, but the union systematically enlarges sets and pushes effective coverage above the nominal $1-\alpha$.

How CF-CP differs. CF-CP calibrates a *single, symmetrized* conformity score that aggregates per-intervention scores at the instance level. In contrast to the post-hoc union, CF-CP avoids the union-induced overcoverage while still ensuring set-level counterfactual invariance. Compared to retraining-based approaches (CFR/PCF or descendants-exclusion), CF-CP is training-free, does not require modifying features or imposing invariance penalties, and does not need access to $\Pr(A)$; all aggregation occurs on scores, not on model parameters or population distributions.

4.3 Theoretical Guarantees

In this section we show that CF-CP satisfies the set-level counterfactual fairness of Definition 3.2 while preserving the marginal coverage guarantee of split CP. First, we state the main assumption needed for our theoretical analysis.

Assumption 4.1. The structural mapping between X and U is invertible for given A. Also, U and A are independent.

The independence assumption is standard in the counterfactual fairness literature. Even though the invertibility assumption may seem strong, it is commonly used in the counterfactual estimation and fairness literature (Nasr-Esfahany et al., 2023; Zhou et al., 2024a,b). We also note that after relaxing the assumption, one can still achieve approximate counterfactual fairness, as we show empirically in Section 5.

Assumption 4.1 asserts that there exists a mapping $g : \mathcal{U} \times \mathcal{A} \to \mathcal{X}$ such that $X = g(U, A)$ and, for each $a \in \mathcal{A}$, the mapping $g_a(u) := g(u, a)$ is invertible with inverse $g_a^{-1} : \mathcal{X} \to \mathcal{U}$. For any target $a' \in \mathcal{A}$, the corresponding counterfactual features are therefore

$$x_{A \leftarrow a'} = g_{a'}(U) = g_{a'}\left(g_a^{-1}(x)\right). \tag{9}$$

Also, the symmetrized score can be written as:

$$s_{\mathrm{cf}}(x, a, y) = \mathrm{Agg}\left\{ s\left(g_{a'}(g_a^{-1}(x)), a', y\right) : a' \in \mathcal{A} \right\} \tag{10}$$

This invertibility makes precise that the multiset $\{x_{A \leftarrow a'}\}_{a'}$ depends only on the shared latent U, not on the factual a. This results in invariance of the symmetrized score. We formalize this observation in the following lemma.

Lemma 4.1 (Invariance of the symmetrized score)**.** *Let Assumption 4.1 hold. Fix any $y \in \mathcal{Y}$. Then for all $a, a' \in \mathcal{A}$,*

$$s_{\mathrm{cf}}\left(X_{A \leftarrow a}, a, y\right) = s_{\mathrm{cf}}\left(X_{A \leftarrow a'}, a', y\right). \tag{11}$$

The proof is given in the Appendix A.1. The key idea is that, under Assumption 4.1, the latent U can be uniquely recovered from the factual features X and protected attribute $A = a$ via the inverse function g_a^{-1}. This allows us to express all counterfactual features $X_{A \leftarrow a'}$ as functions of $X_{A \leftarrow a}$ and $A = a$, and vice versa. The symmetry of s_{cf} then follows from the permutation-invariance of Agg.

This score-level invariance leads to the desired set-level counterfactual fairness. Since the score value is the same across counterfactuals, the prediction sets formed by thresholding the score at $\hat{q}_{1-\alpha}$ must also coincide.

Corollary 4.1 (Set-level counterfactual invariance)**.** *Under the conditions of Lemma 4.1, for all $a, a' \in \mathcal{A}$,*

$$\mathcal{C}_\alpha^{\mathrm{CF}}\left(X_{A \leftarrow a}, a\right) = \mathcal{C}_\alpha^{\mathrm{CF}}\left(X_{A \leftarrow a'}, a'\right). \tag{12}$$

The proof is immediate from Lemma 4.1 and the definition of CF-CP sets in (7).

The previous lemma and corollary establish that CF-CP enforces *set-level counterfactual invariance*. We now argue that replacing s with s_{cf} does *not* disturb the exchangeability required by split CP, so the standard marginal coverage guarantee continues to hold.

Theorem 4.1 (Exchangeability preservation)**.** *Assume that Assumption 4.1 holds and that the calibration points* $\{(X_i, A_i, Y_i)\}_{i\in\mathcal{I}_{\text{cal}}}$ *and the test point* $(X_{n+1}, A_{n+1}, Y_{n+1})$ *are exchangeable. Suppose the score function* s *and the aggregator* Agg *are measurable, and* Agg *is permutation-invariant. Define the counterfactual symmetrization*

$$s_{\text{cf}}(x, a, y) = \text{Agg}\left\{s\left(x_{A\leftarrow a'}, a', y\right) : a' \in \mathcal{A}\right\}.$$

Then, the scores

$$\{S_i^{\text{cf}} = s_{\text{cf}}(X_i, A_i, Y_i) : i \in \mathcal{I}_{\text{cal}} \cup \{n+1\}\} \tag{13}$$

are exchangeable.

The proof is given in the Appendix A.2. The main idea is that s_{cf} is a coordinate-wise measurable transformation of the exchangeable tuples (X_i, A_i, Y_i), hence the resulting scores remain exchangeable.

Theorem 4.1 shows that the exchangeability structure required by split CP is preserved when using the symmetrized score s_{cf}. This leads to the following coverage guarantee for CF-CP.

Corollary 4.2 (Marginal coverage for CF-CP)**.** *Under the conditions of Theorem 4.1, the split conformal set constructed with* s_{cf} *satisfies the standard marginal coverage guarantee:*

$$\Pr\left\{Y_{n+1} \in \mathcal{C}_\alpha^{\text{CF}}(X_{n+1}, A_{n+1})\right\} \geq 1 - \alpha. \tag{14}$$

The proof directly follows the standard split CP coverage proof (Vovk et al., 2005; Lei et al., 2018) applied to the exchangeable scores S_i^{cf}.

Combining Corollaries 4.1 and 4.2, we conclude that CF-CP achieves both marginal coverage and set-level counterfactual invariance, as desired.

5 EXPERIMENTS

In this section, we empirically evaluate CF-CP on synthetic and real datasets, spanning regression and classification tasks and show that it achieves the desired counterfactual fairness while meeting the target coverage rate with minimal increase in the size of the prediction sets is available in the following repository.[1]

[1] `https://github.com/guldoganozgur/cf_cp`

Methods. We compare the following methods: (i) **Split CP**: standard split conformal prediction using the original conformity score with the base predictor; (ii) **Post-hoc Union**: computes CP sets under each intervention and takes their union, specifically, $\mathcal{C}_\alpha^{\cup}(x) := \bigcup_{a'\in\mathcal{A}} \mathcal{C}_\alpha(x_{A\leftarrow a'}, a')$; (iii) **Counterfactual Fairness with** U (Kusner et al., 2017): train a counterfactually fair predictor by excluding descendants of A in the SCM, to simplify this we train base model only on U similar to the approach in Zhou et al. (2024b) followed by split CP; (iv) **Counterfactual Fair Representations (CFR)** (Zuo et al., 2023): trains a predictor on the augmented features as the mean of factual and counterfactual features, specifically, $\tilde{X} = \frac{1}{|\mathcal{A}|}\sum_{a'\in\mathcal{A}} X_{A\leftarrow a'}$ and then applies split CP; (v) **Plug-in Counterfactual Fairness (PCF)** (Zhou et al., 2024b): uses a weighted combination of the predictions for each intervention, specifically, $\hat{f}(X, A) = \sum_{a'\in\mathcal{A}} \Pr(A = a')\hat{f}(X_{A\leftarrow a'}, a')$ where $\hat{f}$ is the base predictor and then follows with split CP; and lastly our method (vi) **CF-CP**: our proposed method that uses the counterfactual symmetrized score in split CP, and we consider three aggregation functions: `mean`, `max`, and `min`. Following common practice, in the classification tasks if the prediction set is empty, we include the label with the highest predicted probability. The base predictor $\hat{f}$ is a linear model for both regression and classification tasks. More details on hyperparameters and training are in the Appendix B.1.

Score functions. For regression tasks, we use the absolute residual score $s(x, a, y) = |\hat{f}(x, a) - y|$. So, the less the absolute error, the better the conformity. For classification tasks, we use Least Ambiguous Classifier (LAC) (Sadinle et al., 2019) score $s(x, a, y) = 1 - \hat{f}_y(x, a)$, where $\hat{f}_y(x, a)$ is the predicted probability of class y. The more probable the class, the better the conformity. We also provide additional results with different base score functions in Appendix C.

Evaluation metrics. For each dataset and method, we report the following metrics—↑ and ↓ indicate that a larger and smaller value is desirable, respectively: (i) **MSE (↓)/Accuracy (↑)**: the mean squared error for regression and accuracy for classification on the test set; (ii) **Total Effect(TE) (↓)**: measures the counterfactual fairness of the base predictor $\hat{f}$, for each test point, we compute the change in the model's output when A is counterfactually flipped, and then average over the test set; (iii) **Coverage**: the empirical coverage of the prediction sets on the test set, $\frac{1}{|\mathcal{I}_{\text{test}}|}\sum_{i\in\mathcal{I}_{\text{test}}} \mathbb{I}[Y_i \in \mathcal{C}_\alpha(X_i, A_i)]$; (iv) **Avg. Set Size (↓)**: the average size of the prediction sets on the test set, for regression it is the average length of the inter-

Table 1: Synthetic results at $\alpha = 0.1$ (mean ± std over 10 runs). Left block: regression; right block: classification. (↓) and (↑) indicates that a smaller and larger value is desirable, respectively. Highlighting follows the convention described in the evaluation metrics paragraph.

	Regression					Classification				
Method	MSE (↓)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)
SplitCP	0.377 ± 0.012	1.219 ± 0.014	0.901 ± 0.008	2.032 ± 0.038	0.713 ± 0.009	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.025 ± 0.209	0.642 ± 0.043
Post-hoc Union	0.377 ± 0.012	1.219 ± 0.014	0.944 ± 0.005	3.251 ± 0.043	0.000 ± 0.000	0.730 ± 0.011	0.543 ± 0.037	0.936 ± 0.010	3.123 ± 0.255	0.000 ± 0.000
CFU	1.112 ± 0.017	0.000 ± 0.000	0.901 ± 0.008	3.447 ± 0.088	0.000 ± 0.000	0.589 ± 0.017	0.000 ± 0.000	0.907 ± 0.009	2.700 ± 0.125	0.000 ± 0.000
CFR	0.832 ± 0.014	0.000 ± 0.000	0.899 ± 0.010	2.964 ± 0.055	0.000 ± 0.000	0.589 ± 0.017	0.000 ± 0.000	0.908 ± 0.009	2.700 ± 0.125	0.000 ± 0.000
PCF	0.826 ± 0.013	0.000 ± 0.000	0.898 ± 0.008	2.951 ± 0.046	0.000 ± 0.000	0.571 ± 0.017	0.000 ± 0.000	0.905 ± 0.012	**2.526 ± 0.166**	0.000 ± 0.000
CF-CP-mean	**0.377 ± 0.012**	1.219 ± 0.014	0.899 ± 0.010	2.971 ± 0.050	0.000 ± 0.000	**0.730 ± 0.011**	0.543 ± 0.037	0.904 ± 0.014	2.542 ± 0.185	0.000 ± 0.000
CF-CP-max	**0.377 ± 0.012**	1.219 ± 0.014	0.900 ± 0.009	3.275 ± 0.053	0.000 ± 0.000	**0.730 ± 0.011**	0.543 ± 0.037	0.935 ± 0.006	3.696 ± 0.168	0.038 ± 0.008
CF-CP-min	**0.377 ± 0.012**	1.219 ± 0.014	0.900 ± 0.010	**2.897 ± 0.058**	0.000 ± 0.000	**0.730 ± 0.011**	0.543 ± 0.037	0.906 ± 0.013	2.581 ± 0.180	0.000 ± 0.000

vals, and for classification it is the average cardinality of the sets, specifically, $\frac{1}{|\mathcal{I}_{\text{test}}|}\sum_{i\in\mathcal{I}_{\text{test}}}|\mathcal{C}_\alpha(X_i, A_i)|$; and (v) **Counterfactual Set Disparity (CSD) (↓)**: quantifies how much the set changes across protected-attribute interventions for each test point. The details of CSD are as follows: For binary A, we use the Jaccard distance between the factual and counterfactual sets:

$$\text{CSD}(x) = 1 - \frac{|\mathcal{C}_\alpha(x_{A\leftarrow a}, a) \cap \mathcal{C}_\alpha(x_{A\leftarrow a'}, a')|}{|\mathcal{C}_\alpha(x_{A\leftarrow a}, a) \cup \mathcal{C}_\alpha(x_{A\leftarrow a'}, a')|}. \quad (15)$$

For multi-valued A, it can be extended by averaging pairwise Jaccard distances across interventions relative to the factual value. A lower CSD indicates more similar sets across counterfactuals, with 0 indicating identical sets and 1 indicating non-overlapping sets. We report the average counterfactual set disparity on the test set, specifically, $\frac{1}{|\mathcal{I}_{\text{test}}|}\sum_{i\in\mathcal{I}_{\text{test}}}\text{CSD}(X_i)$. The metrics for each method are computed over 10 random splits of the data into training, calibration, and test sets, and we report the mean and standard deviation in the tables and figures. For all tables in this paper, we use a common highlighting convention. For the tables that use oracle counterfactuals, bold (underlined) values mark the best (second-best) MSE/Accuracy and Avg. Set Size among the methods that achieve the target coverage and perfect counterfactual fairness for prediction sets (CSD=0). For the tables with noisy counterfactuals, bold (underlined) values mark the best (second-best) CSD, since in this setting we are primarily interested in how fairness behaves under imperfect counterfactuals.

Datasets. We evaluate on synthetic and real datasets spanning regression and multiclass classification. Throughout our experiments, we assume access to an underlying structural causal model that generates the data; for the synthetic datasets, the SCM is specified by construction, and for the real datasets, the SCMs are obtained by fitting structural equations to the observed data; full details on data generation, preprocessing, SCM specification and fitting, and counterfactual construction are in Appendix B.2.

Synthetic Regression. A nonlinear SCM with binary protected attribute A generates (X, Y); the task is to predict Y from (X, A) adopted from Zuo et al. (2023). Counterfactuals are formed by flipping A in the X-equation while holding exogenous noise fixed.

Synthetic Classification. A multi-class (K=10) SCM extends prior binary settings; the task is to predict Y from (X, A). It is inspired by Zhou et al. (2024b) but extended to multi-class targets. Counterfactuals replace A only in the structural equation for X (latent noise held fixed).

Law School. Predict first-year average (FYA) from LSAT, UGPA, race, and gender (Wightman, 1998); we take race as A (binary: white vs non-white). Counterfactuals are obtained by fitting a LiNGAM-based SCM to the observed data and intervening on A.

Bias in Bios. Predict profession (28 classes) from biography text; A is gender (binary) (De-Arteaga et al., 2019). Counterfactuals for this dataset are constructed by the method proposed in Lemberger and Saillenfest (2024), which involves first obtaining text embeddings via a pre-trained BERT model (Devlin et al., 2019), then decomposing the embedding into components aligned with and independent of A; counterfactuals flip the A component while keeping the independent component fixed.

Except for Bios dataset, we use $\alpha = 0.1$; for Bios we use $\alpha = 0.025$ since $\alpha = 0.1$ leads to near-singletons due to high accuracy of the base predictor.

5.1 Oracle Counterfactuals

We first present results when the counterfactuals are generated from the true SCM (oracle counterfactuals)– later, we consider the case where noise is added to the exogenous variables. Table 1 shows the results on synthetic regression and classification datasets. Table 2 shows the results on real datasets, Law School for regression and Bios for classification. First, we observe that vanilla split CP achieves the target coverage but has high counterfactual set disparity (CSD)

Table 2: Real data results. Left block: regression on Law School dataset with $\alpha = 0.1$; right block: classification on Bios dataset with $\alpha = 0.025$ (mean ± std over 10 runs). Highlighting follows the convention described in the evaluation metrics paragraph.

	Law School Regression					Bios Classification				
Method	MSE (↓)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)
SplitCP	0.758 ± 0.005	0.723 ± 0.019	0.898 ± 0.009	2.847 ± 0.070	0.405 ± 0.010	0.801 ± 0.001	0.094 ± 0.003	0.975 ± 0.003	3.311 ± 0.199	0.199 ± 0.005
Post-hoc Union	0.758 ± 0.005	0.723 ± 0.019	0.944 ± 0.007	3.570 ± 0.078	0.000 ± 0.000	0.801 ± 0.001	0.094 ± 0.002	0.978 ± 0.002	3.884 ± 0.227	0.000 ± 0.000
CFU	0.829 ± 0.007	0.000 ± 0.000	0.898 ± 0.011	2.968 ± 0.078	0.000 ± 0.000	-	-	-	-	-
CFR	0.827 ± 0.007	0.000 ± 0.000	0.897 ± 0.012	2.967 ± 0.084	0.000 ± 0.000	0.799 ± 0.000	0.000 ± 0.000	0.974 ± 0.003	**3.379 ± 0.183**	0.000 ± 0.000
PCF	0.828 ± 0.007	0.000 ± 0.000	0.897 ± 0.011	**2.958 ± 0.072**	0.000 ± 0.000	0.795 ± 0.001	0.000 ± 0.000	0.974 ± 0.003	3.471 ± 0.208	0.000 ± 0.000
CF-CP-mean	**0.758 ± 0.005**	0.723 ± 0.019	0.900 ± 0.013	3.106 ± 0.100	0.000 ± 0.000	**0.801 ± 0.001**	0.093 ± 0.003	0.974 ± 0.003	3.486 ± 0.191	0.000 ± 0.000
CF-CP-max	**0.758 ± 0.005**	0.723 ± 0.019	0.900 ± 0.013	3.106 ± 0.100	0.000 ± 0.000	**0.801 ± 0.001**	0.094 ± 0.003	0.975 ± 0.003	3.700 ± 0.208	0.000 ± 0.000
CF-CP-min	**0.758 ± 0.005**	0.723 ± 0.019	0.900 ± 0.013	3.106 ± 0.100	0.000 ± 0.000	**0.801 ± 0.001**	0.093 ± 0.003	0.974 ± 0.003	3.508 ± 0.199	0.000 ± 0.000

Table 3: Results with noisy estimations of counterfactuals (mean ± std over 10 runs). Highlighting follows the convention described in the evaluation metrics paragraph.

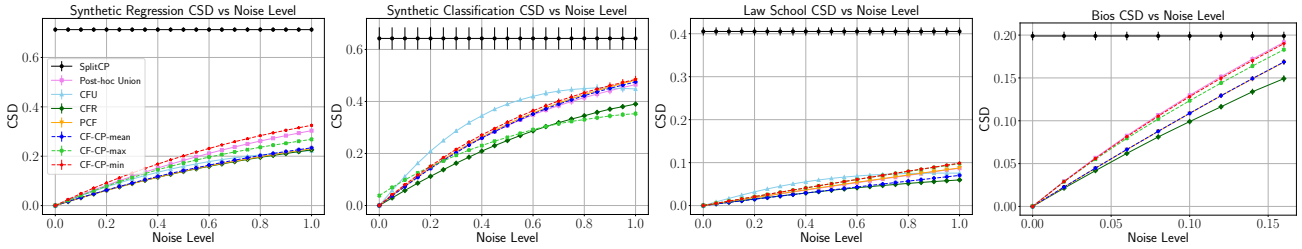
Method	Synth. Regression		Synth. Classification		Law School Regression		Bios Classification	
	Avg. Set Size (↓)	CSD (↓)	Avg. Set Size (↓)	CSD (↓)	Avg. Set Size (↓)	CSD (↓)	Avg. Set Size (↓)	CSD (↓)
SplitCP	2.032 ± 0.038	0.713 ± 0.009	2.025 ± 0.209	0.642 ± 0.043	2.847 ± 0.070	0.405 ± 0.010	3.311 ± 0.199	0.199 ± 0.005
Post-hoc Union	3.259 ± 0.044	0.152 ± 0.001	3.115 ± 0.254	0.260 ± 0.006	3.571 ± 0.078	0.036 ± 0.001	3.991 ± 0.231	0.130 ± 0.002
CFU	3.551 ± 0.088	0.135 ± 0.004	3.462 ± 0.239	0.345 ± 0.008	2.998 ± 0.063	0.055 ± 0.002	-	-
CFR	3.013 ± 0.078	**0.115 ± 0.003**	2.913 ± 0.209	**0.209 ± 0.004**	2.984 ± 0.064	**0.029 ± 0.001**	3.450 ± 0.226	**0.099 ± 0.003**
PCF	2.983 ± 0.061	**0.115 ± 0.003**	2.600 ± 0.210	0.263 ± 0.006	2.961 ± 0.071	0.037 ± 0.001	3.560 ± 0.197	0.109 ± 0.002
CF-CP-mean	3.016 ± 0.054	0.118 ± 0.002	2.609 ± 0.203	0.259 ± 0.006	3.108 ± 0.108	**0.029 ± 0.001**	3.581 ± 0.204	0.109 ± 0.002
CF-CP-max	3.374 ± 0.057	0.145 ± 0.003	4.271 ± 0.185	0.230 ± 0.004	3.114 ± 0.112	0.041 ± 0.001	3.813 ± 0.210	0.123 ± 0.002
CF-CP-min	2.935 ± 0.050	0.168 ± 0.003	2.657 ± 0.205	0.271 ± 0.005	3.107 ± 0.093	0.041 ± 0.001	3.599 ± 0.165	0.128 ± 0.003

ranging from 0.19 to 0.71. As expected, Post-hoc Union achieves counterfactual invariance for prediction sets (CSD=0), but at the cost of a significant increase in the size of the prediction sets and over-coverage due to inflation of the prediction sets. Methods that train counterfactually fair predictors (CFU, CFR, PCF) achieve counterfactual fairness for both point predictions and prediction sets (CSD=0) but at the cost of lower accuracy/MSE and prediction sets that are moderately larger than split CP. Our proposed CF-CP method achieves counterfactual fairness for prediction sets (CSD=0) while maintaining the accuracy/MSE of the base predictor, and it produces prediction sets that are similar in size to counterfactually fair baselines and moderately larger than split CP. Only `max` aggregation leads to a nonzero CSD in synthetic classification, because of non-empty prediction set rule. Also, CF-CP achieves comparable or better average set size compared to other counterfactual fairness methods (CFU, CFR, PCF) without requiring retraining or access to the probability distribution of A. Among the three aggregation functions, the results are quite similar, with `mean` achieving the smallest average set size in most cases.

5.2 Noisy Counterfactuals

In this section, we examine the robustness of CF-CP to the quality of the available counterfactuals, since its performance necessarily depends on how accurate these counterfactuals are. Specifically, we study how fairness and performance change as we introduce noise into the counterfactuals. For the datasets where we have access to the exogenous variables U (synthetic datasets and Law School), we add Gaussian noise to the true exogenous variables before generating counterfactuals. For the Bios dataset, we add Gaussian noise to the counterfactual embeddings. The results are shown in Table 3. Here we only report average set size and CSD due to space constraints and the other metrics and noise variances can be found in Appendix C. The noisy counterfactuals lead to a slight increase in CSD for all methods, but CF-CP still achieves a significant reduction in CSD compared to split CP, with only a modest increase in the size of the prediction sets. CF-CP achieves comparable average set size and CSD trade-off compared to other counterfactual fairness methods (CFU, CFR, PCF) even with noisy counterfactuals.

To further illustrate the robustness of CF-CP to noisy counterfactuals, we plot CSD against different noise levels in Figure 1. We vary the noise level from 0 (oracle counterfactuals) to 1 for synthetic datasets and Law School, and from 0 to 0.16 for Bios, as higher noise levels lead to poor performance. We observe that as the noise level increases, CSD also increases for all methods as expected. However, CF-CP methods show similar robustness to noise as other counterfactual fair models (CFU, CFR, PCF) and post-hoc union, and they consistently achieve lower CSD compared to split CP across all noise levels. This shows that CF-CP



(a) Synthetic Regression (b) Synthetic Classification (c) Law School Regression (d) Bios Classification

Figure 1: Counterfactual Set Disparity (CSD) vs noise level in counterfactual generation. It shows that CF-CP methods show similar robustness to noise as other counterfactual fair models.

is effective in reducing counterfactual unfairness even when counterfactuals are noisy.

We emphasize two key takeaways: (i) CF-CP does not require retraining the model or access to the probability distribution of A, making it a more practical choice in real-world scenarios where counterfactuals may be noisy or imperfect; and (ii) as stated earlier, CF-CP does not reduce the accuracy of the baseline predictor. Compared to CFU, CFR and PCF, the baseline predictor for CF-CP is more accurate–see Appendix C.

6 CONCLUSION

In this paper, we extended the notion of counterfactual fairness to set-valued predictors and then proposed a novel post-training conformal prediction method that enforces set-level counterfactual fairness by symmetrizing the conformity scores across protected attribute interventions. Under an invertible SCM, we showed that our method guarantees both marginal coverage and counterfactual invariance for prediction sets. Empirically, across synthetic and real-world datasets, including Law School and Bios, CF-CP substantially reduces counterfactual set disparity compared to standard split CP, with only a modest increase in average set size, and without requiring retraining or access to the probability distribution of the protected attribute. Our results suggest that enforcing counterfactual fairness at the set level is both feasible and effective. Two promising directions for future work are: (i) extending this fairness notion to path-dependent counterfactual fairness (Chiappa, 2019), where only certain causal pathways from the protected attribute to the outcome are considered unfair; (ii) generalizing CF-CP to dynamic settings where covariates and protected attributes evolve over time.

References

Agarwal, A., Dudík, M., and Wu, Z. S. (2019). Fair regression: Quantitative definitions and reduction-based algorithms. In *International conference on machine learning*, pages 120–129. PMLR.

Angelopoulos, A. N., Bates, S., et al. (2023). Conformal prediction: A gentle introduction. *Foundations and trends® in machine learning*, 16(4):494–591.

Angelopoulos, A. N., Bates, S., Jordan, M., and Malik, J. (2021). Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*.

Azimi, V. and Zaydman, M. A. (2023). Optimizing equity: Working towards fair machine learning algorithms in laboratory medicine. *The journal of applied laboratory medicine*, 8(1):113–128.

Barocas, S. and Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, 104(3):671–732.

Berk, R., Heidari, H., Jabbari, S., Kearns, M., and Roth, A. (2018). Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 1:42.

Chen, Z., Guo, R., Ton, J.-F., and Liu, Y. (2024). Conformal counterfactual inference under hidden confounding. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 397–408.

Chiappa, S. (2019). Path-specific counterfactual fairness. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7801–7808.

Chouldechova, A. (2017). Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163.

Corbett-Davies, S., Gaebler, J. D., Nilforoshan, H., Shroff, R., and Goel, S. (2023). The measure and mismeasure of fairness. *Journal of Machine Learning Research*, 24(312):1–117.

Cresswell, J. C., Kumar, B., Sui, Y., and Belbahri, M. (2025). Conformal prediction sets can cause disparate impact. In *The Thirteenth International Conference on Learning Representations*.

Cresswell, J. C., Sui, Y., Kumar, B., and Vouitsis, N. (2024). Conformal prediction sets improve human decision making. In *Forty-first International Conference on Machine Learning*.

De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., and Kalai, A. T. (2019). Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Di Stefano, P. G., Hickey, J. M., and Vasileiou, V. (2020). Counterfactual fairness: removing direct effects through regularization. *arXiv preprint arXiv:2002.10774*.

Garg, S., Perot, V., Limtiaco, N., Taly, A., Chi, E. H., and Beutel, A. (2019). Counterfactual fairness in text classification through robustness. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 219–226.

Kuchibhotla, A. K. (2020). Exchangeability, conformal prediction, and rank tests. *arXiv preprint arXiv:2005.06095*.

Kusner, M. J., Loftus, J., Russell, C., and Silva, R. (2017). Counterfactual fairness. *Advances in neural information processing systems*, 30.

Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R. J., and Wasserman, L. (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111.

Lei, L. and Candès, E. J. (2021). Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(5):911–938.

Lemberger, P. and Saillenfest, A. (2024). Explaining text classifiers with counterfactual representations. *arXiv preprint arXiv:2402.00711*.

Liu, M., Ding, L., Yu, D., Liu, W., Kong, L., and Jiang, B. (2022). Conformalized fairness via quantile regression. *Advances in Neural Information Processing Systems*, 35:11561–11572.

Lu, C., Lemay, A., Chang, K., Höbel, K., and Kalpathy-Cramer, J. (2022). Fair conformal predictors for applications in medical imaging. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 12008–12016.

Nasr-Esfahany, A., Alizadeh, M., and Shah, D. (2023). Counterfactual identifiability of bijective causal models. In *International conference on machine learning*, pages 25733–25754. PMLR.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. (2002). Inductive confidence machines for regression. In *European conference on machine learning*, pages 345–356. Springer.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Podesta, J., Pritzker, P., Moniz, E. J., Holdren, J., and Zients, J. (2014). Big data: Seizing opportunities and preserving values. Technical report, Executive Office of the President of the United States.

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M., and Goldberg, Y. (2020). Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256.

Romano, Y., Barber, R. F., Sabatti, C., and Candès, E. (2020a). With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 2(2).

Romano, Y., Sesia, M., and Candes, E. (2020b). Classification with valid and adaptive coverage. *Advances in neural information processing systems*, 33:3581–3591.

Sadinle, M., Lei, J., and Wasserman, L. (2019). Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 114(525):223–234.

Schröder, M., Frauen, D., Schweisthal, J., Heß, K., Melnychuk, V., and Feuerriegel, S. (2024). Conformal prediction for causal effects of continuous treatments. *arXiv preprint arXiv:2407.03094*.

Shafer, G. and Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3).

Vadlamani, A. T., Srinivasan, A., Maneriker, P., Payani, A., and srinivasan parthasarathy (2025). A generic framework for conformal fairness. In *The Thirteenth International Conference on Learning Representations*.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

Wang, F., Cheng, L., Guo, R., Liu, K., and Yu, P. S. (2023). Equal opportunity of coverage in fair regression. *Advances in Neural Information Processing Systems*, 36:7743–7755.

Wightman, L. F. (1998). Lsac national longitudinal bar passage study. lsac research report series.

Zheng, Y., Huang, B., Chen, W., Ramsey, J., Gong, M., Cai, R., Shimizu, S., Spirtes, P., and Zhang, K. (2024). Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8.

Zhou, Z., Bai, R., Kulinski, S., Kocaoglu, M., and Inouye, D. I. (2024a). Towards characterizing domain counterfactuals for invertible latent causal models. In *The Twelfth International Conference on Learning Representations*.

Zhou, Z., Liu, T., Bai, R., Gao, J., Kocaoglu, M., and Inouye, D. I. (2024b). Counterfactual fairness by combining factual and counterfactual predictions. *Advances in Neural Information Processing Systems*, 37:47876–47907.

Zuo, Z., Khalili, M., and Zhang, X. (2023). Counterfactually fair representation. *Advances in neural information processing systems*, 36:12124–12140.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes**.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes**.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. **Yes**.
   (b) Complete proofs of all theoretical results. **Yes**.
   (c) Clear explanations of any assumptions. **Yes**.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. **Yes**.
   (b) The license information of the assets, if applicable. **Not Applicable**.
   (c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable**.
   (d) Information about consent from data providers/curators. **Not Applicable**.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. **Not Applicable**.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**.

Supplementary Materials

A PROOFS

A.1 Proof of Lemma 4.1

Here we restate Lemma 4.1 for convenience.

Lemma A.1 (Invariance of the symmetrized score)**.** *Let Assumption 4.1 hold. Fix any $y \in \mathcal{Y}$. Then for all $a, a' \in \mathcal{A}$,*

$$s_{\text{cf}}\left(X_{A\leftarrow a}, a, y\right) = s_{\text{cf}}\left(X_{A\leftarrow a'}, a', y\right). \tag{16}$$

Proof. Fix $y \in \mathcal{Y}$ and $a, a' \in \mathcal{A}$. Under Assumption 4.1, for the factual pair $(X_{A\leftarrow a}, a)$ there exists a unique latent

$$U = g_a^{-1}\left(X_{A\leftarrow a}\right), \tag{17}$$

where g_a^{-1} is the inverse of the forward map g_a in the SCM that generates X from U under intervention $A \leftarrow a$. Likewise for $(X_{A\leftarrow a'}, a')$ we have the same $U = g_{a'}^{-1}(X_{A\leftarrow a'})$ because both arise from the same structural model applied to the same exogenous noise.

For any target intervention $a'' \in \mathcal{A}$, the corresponding counterfactual features are

$$X_{A\leftarrow a''} = g_{a''}(U), \tag{18}$$

independently of whether we start from $(X_{A\leftarrow a}, a)$ or $(X_{A\leftarrow a'}, a')$. Hence the multiset of per-intervention scores

$$\left\{s\left(g_{a''}(U), a'', y\right) : a'' \in \mathcal{A}\right\} \tag{19}$$

is identical in both cases (up to permutation). Since Agg in (6) is permutation-invariant, applying it to these equal multisets yields the same value:

$$s_{\text{cf}}\left(X_{A\leftarrow a}, a, y\right) = s_{\text{cf}}\left(X_{A\leftarrow a'}, a', y\right). \tag{20}$$

This proves (16). □

A.2 Proof of Theorem 4.1

Here we restate Theorem 4.1 for convenience.

Theorem A.1 (Exchangeability preservation)**.** *Assume that Assumption 4.1 holds and that the calibration points $\{(X_i, A_i, Y_i)\}_{i\in\mathcal{I}_{\text{cal}}}$ and the test point $(X_{n+1}, A_{n+1}, Y_{n+1})$ are exchangeable. Suppose the score function s and the aggregator* Agg *are measurable, and* Agg *is permutation-invariant. Define the counterfactual symmetrization*

$$s_{\text{cf}}(x, a, y) = \text{Agg}\left\{s\left(x_{A\leftarrow a'}, a', y\right) : a' \in \mathcal{A}\right\}.$$

Then, the scores

$$\left\{S_i^{\text{cf}} = s_{\text{cf}}(X_i, A_i, Y_i) : i \in \mathcal{I}_{\text{cal}} \cup \{n+1\}\right\} \tag{21}$$

are exchangeable.

Proof. Let π be any permutation of $\mathcal{I}_{\text{cal}} \cup \{n+1\}$. By assumption, $\{(X_{\pi(i)}, A_{\pi(i)}, Y_{\pi(i)})\}_i$ is distributionally identical to $\{(X_i, A_i, Y_i)\}_i$.

Define the coordinate-wise transformation

$$T(x, a, y) := \text{Agg}\left\{s\left(g_{a'}\left(g_a^{-1}(x)\right), a', y\right) : a' \in \mathcal{A}\right\}. \tag{22}$$

Under Assumption 4.1, g_a^{-1} and $g_{a'}$ are deterministic and measurable for all a, a', and s and Agg are measurable. Hence, T is measurable, and it is applied identically to each coordinate.

Set

$$S_i^{\text{cf}} := T(X_i, A_i, Y_i). \tag{23}$$

Exchangeability is preserved under identical measurable coordinate-wise transformations by Proposition 4 of Kuchibhotla (2020), it follows that

$$\{S_i^{\text{cf}}\}_i \stackrel{d}{=} \{s_{\text{cf}}(X_{\pi(i)}, A_{\pi(i)}, Y_{\pi(i)})\}_i. \tag{24}$$

Therefore, $\{S_i^{\text{cf}}\}_{i \in \mathcal{I}_{\text{cal}} \cup \{n+1\}}$ are exchangeable, as claimed. □

B EXPERIMENT DETAILS

B.1 Implementation Details

Here we provide additional implementation details. All GPU related experiments were run on GTX 1080 Ti. Remaining experiments were run on Apple M1 CPU.

Base predictor. For all datasets, the base predictor $\hat{f}$ is a linear model for both regression and classification tasks. For synthetic and Law School datasets, we use `scikit-learn`'s `LinearRegression` and `LogisticRegression` implementations (Pedregosa et al., 2011). The default hyperparameters are used except for the maximum number of iterations which is set to 1000 for `LogisticRegression`. For Bias in Bios dataset, we use `PyTorch` (Paszke et al., 2019) to implement the logistic regression model and utilize GPU to speed up optimization. We use LBFGS optimizer with a learning rate of 1 and a maximum of 100 iterations.

Conformal prediction sets. When constructing conformal prediction sets, we include at least one label in the prediction set for classification tasks if the score of the most likely label is above the threshold. This is to avoid empty prediction sets which can happen when the model is very uncertain about the prediction. It is a common practice in conformal prediction to ensure non-empty prediction sets.

B.2 Datasets

Here we introduce the datasets used in our experiments.

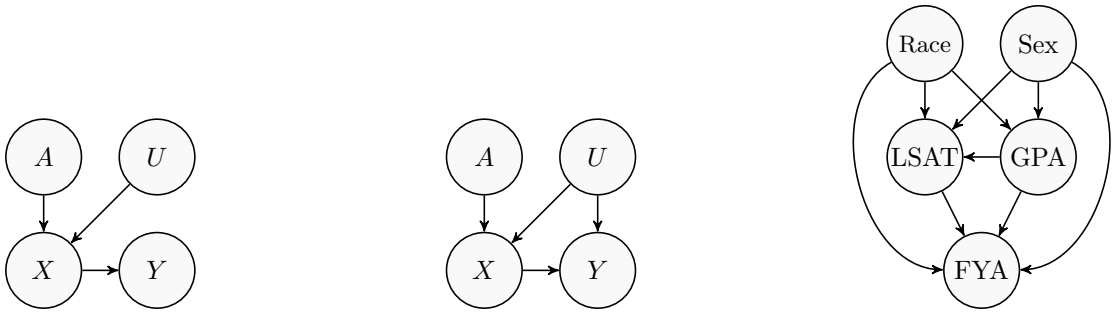
Synthetic Regression. We follow the data generation process in Zuo et al. (2023) to generate a synthetic regression dataset with a binary protected attribute A. The data is generated according to the following structural equations:

$$\begin{aligned}
&U_i \sim \mathcal{N}(0,1) \;\; i \in 1,2, \quad A \sim \text{Bernoulli}(0.4),\\
&X \leftarrow \sin(U_1) + \cos(AU_2) + A + 0.1,\\
&Y \leftarrow 0.2X^2 + 1.2X + 0.2 + \epsilon_Y, \quad \epsilon_Y \sim \mathcal{N}(0, 0.6).
\end{aligned}$$

The causal graph is shown in Figure 2a. Counterfactuals change only A in the X-equation (holding U_1, U_2, ϵ_Y fixed). For noisy experiments, we add i.i.d. Gaussian noise to U when constructing counterfactuals, i.e., we use $\tilde{U} = U + \epsilon_U$ where $\epsilon_U \sim \mathcal{N}(0, \sigma_U^2 I)$ and $\sigma_U = 0.4$ for the results in Table 3. We use $\alpha = 0.1$ and the split of the data is as follows: 5000 samples for training, 1000 samples for calibration, and 5000 samples for testing.

Synthetic Classification. We extend the binary classification setting in Zhou et al. (2024b) to a multi-class classification setting with $K = 10$ classes. The SCM is

$$\begin{aligned}
&U \sim \mathcal{N}(\mathbf{0}, I_d), \qquad A \sim \text{Bernoulli}(0.5),\\
&X \leftarrow (A - 0.5)w_A + UD_U,\\
&Y \sim \text{Categorical}\left(\text{softmax}(X^{\odot 3}W_X + UW_U + E)\right), \qquad E \sim \mathcal{N}(\mathbf{0}, \sigma^2 I_K).
\end{aligned}$$



(a) Synthetic Regression and Bias (b) Synthetic Classification (c) Law School

Figure 2: Causal Graphs of the Datasets Used in Experiments

The causal graph is shown in Figure 2b. Here $d = 10$ is the feature dimension and $K = 10$ the number of classes; $w_A \in \mathbb{R}^{1\times d}$ encodes the linear effect of A on X; $D_U \in \mathbb{R}^{d\times d}$ mixes the latent U into X (assumed full rank); $W_X, W_U \in \mathbb{R}^{d\times K}$ map the (nonlinear) features and latent into class logits; E is i.i.d. Gaussian *logit* noise with variance $\sigma = 0.2$ and $X^{\odot 3}$ denotes elementwise cubing of X. All the matrices are generated as identity matrix plus small uniform noise between $[0, 0.2]$ added to each entry, w_A has first $r = 3$ entries sampled from Uniform($[2, 2.2]$) and the rest are zeros. Counterfactuals change only A in the X-equation (holding U and E fixed). Similarly, for noisy experiments, we add i.i.d. Gaussian noise to U when constructing counterfactuals, i.e., we use $\tilde{U} = U + \epsilon_U$ where $\epsilon_U \sim \mathcal{N}(0, \sigma_U^2 I)$ and $\sigma_U = 0.4$ for the results in Table 3. We use $\alpha = 0.1$ and the split of the data is as follows: 5000 samples for training, 1000 samples for calibration, and 5000 samples for testing.

Law School. We use the Law School dataset from Wightman (1998), which has been used in prior work on counterfactual fairness (Kusner et al., 2017; Zuo et al., 2023). The task is to predict the first-year average grade (FYA) of law students based on their LSAT score, undergraduate GPA, race and gender. We consider race as the protected attribute A (binary: white vs non-white). We preprocess the LSAT and GPA features by standardizing them to have zero mean and unit variance. To construct counterfactuals, we fit a LiNGAM model to the data using the `causal-learn` package (Zheng et al., 2024). The fitted causal graph is shown in Figure 2c. We assume that race and gender are root nodes in the causal graph and there are no children of FYA node. For noisy experiments, we add i.i.d. Gaussian noise to U when constructing counterfactuals, and we assume U only effects LSAT and GPA, i.e., we use $\tilde{U} = U + \epsilon_U$ where $\epsilon_U \sim \mathcal{N}(0, \sigma_U^2 I)$ and $\sigma_U = 0.4$ for the results in Table 3. We use $\alpha = 0.1$ and the total number of samples is 21791, with a split of 1000 for calibration, 10000 for testing, and the rest for training.

Bias in Bios. We use the Bios dataset from De-Arteaga et al. (2019) which contains biographies of individuals in various professions. The task is to predict the profession of an individual based on their biography text, it has gender labels which we consider as the protected attribute A (binary: male vs female), there are 28 professions (classes) in total. We get the embeddings of the factual and counterfactual biographies via the method proposed by (Lemberger and Saillenfest, 2024), using their code is publicly available.[2] They assume a causal graph similar to Figure 2a where X is the text embedding and Y is the profession, and U is some latent variable that is not observed. We use the version of the dataset introduced in Ravfogel et al. (2020). First we obtain text embeddings using a pre-trained BERT model (Devlin et al., 2019), then we split this embedding into two orthogonal components, one that is aligned with the protected attribute A and one that is independent of A. To create counterfactual embeddings, we flip the component aligned with A while keeping the independent component fixed. In this dataset, we do not have access to U, so we cannot apply CFU method. For noisy experiments, we add i.i.d. Gaussian noise to the embedding when constructing counterfactuals, i.e., we use $\tilde{X}_{A\leftarrow a'} = X_{A\leftarrow a'} + \epsilon_X$ where $\epsilon_X \sim \mathcal{N}(0, \sigma_X^2 I)$ and $\sigma_X = 0.1$ for the results in Table 3. We use $\alpha = 0.025$ since the task is fairly easy, if we use $\alpha = 0.1$ all methods return singletons as the prediction sets most of the time. The total number of samples is 393423. We use 5000 samples for calibration, half of the dataset for testing, and the rest for training.

[2] `https://github.com/ToineSayan/counterfactual-representations-for-explanation`

Table 4: Synthetic results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs) with noisy counterfactuals. Left block: regression; right block: classification. Highlighting follows the convention described in the evaluation metrics paragraph.

	Regression					Classification				
Method	MSE (↓)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)
SplitCP	0.377 ± 0.012	1.219 ± 0.014	0.901 ± 0.008	2.032 ± 0.038	0.713 ± 0.009	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.025 ± 0.209	0.642 ± 0.043
Post-hoc Union	0.377 ± 0.012	1.219 ± 0.014	0.944 ± 0.006	3.259 ± 0.044	0.152 ± 0.001	0.730 ± 0.011	0.543 ± 0.037	0.935 ± 0.009	3.115 ± 0.254	0.260 ± 0.006
CFU	1.170 ± 0.020	0.267 ± 0.007	0.902 ± 0.008	3.551 ± 0.088	0.135 ± 0.004	0.512 ± 0.011	0.235 ± 0.004	0.908 ± 0.014	3.462 ± 0.239	0.345 ± 0.008
CFR	0.852 ± 0.014	0.191 ± 0.002	0.900 ± 0.012	3.013 ± 0.078	**0.115 ± 0.003**	0.563 ± 0.012	0.140 ± 0.002	0.907 ± 0.014	2.913 ± 0.209	**0.209 ± 0.004**
PCF	0.847 ± 0.014	0.189 ± 0.002	0.897 ± 0.010	2.983 ± 0.061	**0.115 ± 0.003**	0.550 ± 0.015	0.165 ± 0.004	0.908 ± 0.012	2.600 ± 0.210	0.263 ± 0.006
CF-CP-mean	0.377 ± 0.012	1.219 ± 0.014	0.900 ± 0.008	3.016 ± 0.054	0.118 ± 0.002	0.730 ± 0.011	0.543 ± 0.037	0.908 ± 0.013	2.609 ± 0.203	0.259 ± 0.006
CF-CP-max	0.377 ± 0.012	1.219 ± 0.014	0.895 ± 0.009	3.374 ± 0.057	0.145 ± 0.003	0.730 ± 0.011	0.543 ± 0.037	0.945 ± 0.004	4.271 ± 0.185	0.230 ± 0.004
CF-CP-min	0.377 ± 0.012	1.219 ± 0.014	0.905 ± 0.008	2.935 ± 0.050	0.168 ± 0.003	0.730 ± 0.011	0.543 ± 0.037	0.910 ± 0.012	2.657 ± 0.205	0.271 ± 0.005

Table 5: Real data results with noisy counterfactuals. Left block: regression on Law School dataset with $\alpha = 0.1$; right block: classification on Bios dataset with $\alpha = 0.025$ (mean $\pm$ std over 10 runs). Highlighting follows the convention described in the evaluation metrics paragraph.

	Law School Regression					Bios Classification				
Method	MSE (↓)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)
SplitCP	0.758 ± 0.005	0.723 ± 0.019	0.898 ± 0.009	2.847 ± 0.070	0.405 ± 0.010	0.801 ± 0.001	0.094 ± 0.003	0.975 ± 0.003	3.311 ± 0.199	0.199 ± 0.005
Post-hoc Union	0.758 ± 0.005	0.723 ± 0.019	0.943 ± 0.007	3.571 ± 0.078	0.036 ± 0.001	0.801 ± 0.001	0.094 ± 0.002	0.979 ± 0.002	3.991 ± 0.231	0.130 ± 0.002
CFU	0.835 ± 0.007	0.087 ± 0.003	0.900 ± 0.009	2.998 ± 0.063	0.055 ± 0.002	-	-	-	-	-
CFR	0.829 ± 0.007	0.045 ± 0.001	0.899 ± 0.009	2.984 ± 0.064	**0.029 ± 0.001**	0.796 ± 0.001	0.045 ± 0.001	0.974 ± 0.003	3.450 ± 0.226	**0.099 ± 0.003**
PCF	0.829 ± 0.007	0.056 ± 0.001	0.897 ± 0.010	2.961 ± 0.071	0.037 ± 0.001	0.792 ± 0.001	0.049 ± 0.001	0.975 ± 0.002	3.560 ± 0.197	0.109 ± 0.002
CF-CP-mean	0.758 ± 0.005	0.723 ± 0.019	0.899 ± 0.014	3.108 ± 0.108	**0.029 ± 0.001**	0.801 ± 0.001	0.093 ± 0.003	0.975 ± 0.003	3.581 ± 0.204	0.109 ± 0.002
CF-CP-max	0.758 ± 0.005	0.723 ± 0.019	0.900 ± 0.014	3.114 ± 0.112	0.041 ± 0.001	0.801 ± 0.000	0.094 ± 0.003	0.975 ± 0.003	3.813 ± 0.210	0.123 ± 0.002
CF-CP-min	0.758 ± 0.005	0.723 ± 0.019	0.899 ± 0.012	3.107 ± 0.093	0.041 ± 0.001	0.801 ± 0.001	0.093 ± 0.003	0.975 ± 0.002	3.599 ± 0.165	0.128 ± 0.003

C ADDITIONAL RESULTS

In this section, we provide additional experimental results that could not fit in the main text due to space constraints.

C.1 Results on Noisy Counterfactuals

Here we present additional results on the experiments with noisy counterfactuals. In Table 3, we report the results on all datasets with noisy counterfactuals. We only report average set size and CSD due to space constraints, the other metrics are reported here in Table 4 and Table 5. It can be seen that due to noisy counterfactuals, MSE or accuracy of counterfactually fair methods (CFU, CFR, PCF) are slightly worse than oracle counterfactuals. This also leads to a slight increase in average set size for these methods. However, counterfactually fair methods and CF-CP still achieve a significant reduction in CSD compared to split CP, with only a modest increase in the size of the prediction sets. The noise level for synthetic and Law School datasets is $\sigma_U = 0.4$, and for Bias in Bios dataset it is $\sigma_X = 0.1$.

C.2 Results on Synthetic Classification with Different Base Score Functions

Here we provide additional results on synthetic classification with different base score functions. In addition to LAC (used in the main text), we also consider the following two base score functions. The first one is Adaptive Prediction Sets (APS) score (Romano et al., 2020b):

$$s_{\text{APS}}(x, a, y) = \sum_{y' \in \mathcal{Y}, \hat{f}_{y'}(x,a) > \hat{f}_y(x,a)} \hat{f}_{y'}(x, a). \tag{25}$$

We use APS score without additional randomness for simplicity. The second one is the regularized APS (RAPS) score, which adds a regularization term to the APS score to penalize large prediction sets (Angelopoulos et al., 2021):

$$s_{\text{RAPS}}(x, a, y) = s_{\text{APS}}(x, a, y) + \lambda(\text{rank}(\hat{f}_y(x, a)) - k_{\text{reg}})_+, \tag{26}$$

where $\lambda > 0$ is the regularization parameter, k_{reg} is a threshold parameter, and $\text{rank}(\hat{f}(x, a, y))$ is the rank of the true label y in the sorted list of predicted probabilities. We set $\lambda = 0.5$ and $k_{\text{reg}} = 2$ in our experiments.

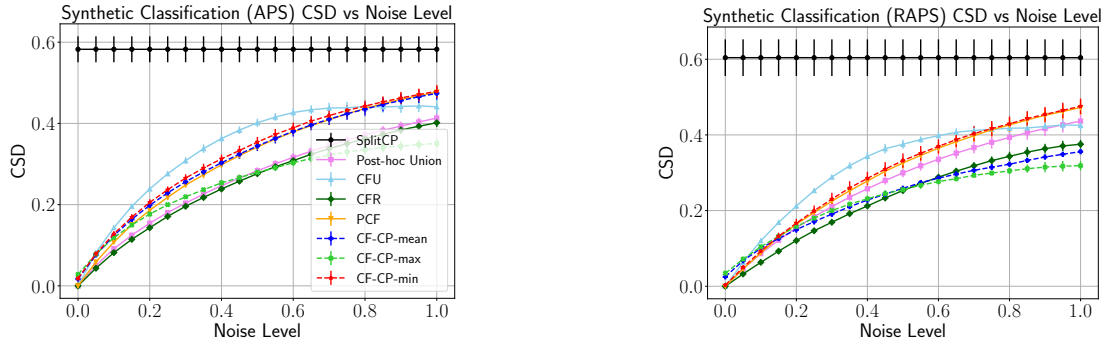
Table 6: Synthetic classification results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs) with APS and RAPS base scores with noiseless counterfactuals. Highlighting follows the convention described in the evaluation metrics paragraph. Since we do not use random sets for APS and RAPS, we relax the condition of achieving target coverage.

	APS					RAPS				
Method	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)
SplitCP	0.730 ± 0.011	0.543 ± 0.037	0.962 ± 0.006	3.517 ± 0.169	0.583 ± 0.032	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.393 ± 0.280	0.604 ± 0.048
Post-hoc Union	**0.730 ± 0.011**	0.543 ± 0.037	0.979 ± 0.004	5.021 ± 0.238	0.000 ± 0.000	**0.730 ± 0.011**	0.543 ± 0.037	0.940 ± 0.010	3.559 ± 0.316	0.000 ± 0.000
CFU	0.589 ± 0.017	0.000 ± 0.000	0.932 ± 0.006	3.695 ± 0.126	0.000 ± 0.000	0.589 ± 0.017	0.000 ± 0.000	0.907 ± 0.014	2.973 ± 0.183	0.000 ± 0.000
CFR	0.589 ± 0.017	0.000 ± 0.000	0.931 ± 0.006	3.685 ± 0.134	0.000 ± 0.000	0.589 ± 0.017	0.000 ± 0.000	0.907 ± 0.014	2.973 ± 0.182	0.000 ± 0.000
PCF	0.571 ± 0.017	0.000 ± 0.000	0.928 ± 0.008	**3.487 ± 0.145**	0.000 ± 0.000	0.571 ± 0.017	0.000 ± 0.000	0.906 ± 0.014	**2.859 ± 0.241**	0.000 ± 0.000
CF-CP-mean	**0.730 ± 0.011**	0.543 ± 0.037	0.941 ± 0.004	3.540 ± 0.119	0.017 ± 0.005	**0.730 ± 0.011**	0.543 ± 0.037	0.927 ± 0.009	3.298 ± 0.160	0.025 ± 0.006
CF-CP-max	**0.730 ± 0.011**	0.543 ± 0.037	0.947 ± 0.005	4.130 ± 0.127	0.028 ± 0.005	**0.730 ± 0.011**	0.543 ± 0.037	0.942 ± 0.008	3.986 ± 0.147	0.034 ± 0.007
CF-CP-min	**0.730 ± 0.011**	0.543 ± 0.037	0.944 ± 0.004	3.690 ± 0.139	0.018 ± 0.004	**0.730 ± 0.011**	0.543 ± 0.037	0.912 ± 0.009	2.948 ± 0.126	0.003 ± 0.004

Table 7: Synthetic classification results at $\alpha = 0.1$ (mean $\pm$ std over 10 runs) with APS and RAPS base scores with noisy counterfactuals, where the noise level is $\sigma_U = 0.4$. Highlighting follows the convention described in the evaluation metrics paragraph.

	APS					RAPS				
Method	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)	Accuracy (↑)	TE (↓)	Coverage	Avg. Set Size (↓)	CSD (↓)
SplitCP	0.730 ± 0.011	0.543 ± 0.037	0.962 ± 0.006	3.517 ± 0.169	0.583 ± 0.032	0.730 ± 0.011	0.543 ± 0.037	0.905 ± 0.011	2.393 ± 0.280	0.604 ± 0.048
Post-hoc Union	0.730 ± 0.011	0.543 ± 0.037	0.979 ± 0.004	4.945 ± 0.232	0.247 ± 0.008	0.730 ± 0.011	0.543 ± 0.037	0.938 ± 0.011	3.580 ± 0.322	0.258 ± 0.009
CFU	0.512 ± 0.011	0.235 ± 0.004	0.916 ± 0.012	4.007 ± 0.207	0.363 ± 0.010	0.512 ± 0.011	0.235 ± 0.004	0.904 ± 0.011	3.635 ± 0.239	0.343 ± 0.012
CFR	0.563 ± 0.012	0.140 ± 0.002	0.928 ± 0.008	3.790 ± 0.153	**0.238 ± 0.005**	0.563 ± 0.012	0.140 ± 0.002	0.909 ± 0.012	3.219 ± 0.173	**0.212 ± 0.004**
PCF	0.550 ± 0.015	0.165 ± 0.004	0.923 ± 0.010	3.484 ± 0.154	0.298 ± 0.008	0.550 ± 0.015	0.165 ± 0.004	0.907 ± 0.014	2.913 ± 0.244	0.278 ± 0.012
CF-CP-mean	0.730 ± 0.011	0.543 ± 0.037	0.945 ± 0.005	3.634 ± 0.118	0.303 ± 0.011	0.730 ± 0.011	0.543 ± 0.037	0.932 ± 0.007	3.511 ± 0.112	0.228 ± 0.006
CF-CP-max	0.730 ± 0.011	0.543 ± 0.037	0.952 ± 0.005	4.661 ± 0.203	0.254 ± 0.007	0.730 ± 0.011	0.543 ± 0.037	0.948 ± 0.006	4.410 ± 0.174	0.231 ± 0.008
CF-CP-min	0.730 ± 0.011	0.543 ± 0.037	0.947 ± 0.005	3.751 ± 0.126	0.313 ± 0.011	0.730 ± 0.011	0.543 ± 0.037	0.911 ± 0.010	2.991 ± 0.151	0.285 ± 0.017

The results are shown in Table 6 for oracle counterfactuals and Table 7 for noisy counterfactuals. The results are consistent with those using LAC score in the main text. APS score tends to produce larger prediction sets compared to LAC score, while RAPS score produces smaller prediction sets thanks to the regularization. Since we use deterministic version of APS, the average coverage is slightly larger than the target coverage value for all the methods. Here again counterfactually fair methods (CFU, CFR, PCF) and CF-CP methods achieve a significant reduction in CSD compared to split CP, with only a modest increase in the size of the prediction sets. In the noiseless case, CF-CP methods has CSD slightly larger than 0, which is due to the constraint that the prediction sets must include at least one label. For completeness, we also report how CSD changes with different noise levels in Figure 3 for APS and RAPS scores. The results are consistent with those using LAC score in the main text. When the noise level is higher, all methods have higher CSD, but counterfactually fair methods and CF-CP methods still achieve a significant reduction in CSD compared to split CP.



(a) Synthetic Classification with APS

(b) Synthetic Classification with RAPS

Figure 3: Counterfactual Set Disparity (CSD) vs noise level in counterfactual generation for APS(left) and RAPS (right) base scores on synthetic classification dataset (mean over 10 runs).