Multigroup Robustness

Lunjia Hu* [1] **Charlotte Peale*** [1] **Judy Hanwen Shen*** [1]

Abstract

To address the shortcomings of real-world datasets, robust learning algorithms have been designed to overcome arbitrary and indiscriminate data corruption. However, practical processes of gathering data may lead to patterns of data corruption that are localized to specific partitions of the training dataset. Motivated by critical applications where the learned model is deployed to make predictions about people from a rich collection of overlapping subpopulations, we initiate the study of *multigroup robust* algorithms whose robustness guarantees for each subpopulation only degrade with the amount of data corruption *inside* that subpopulation. When the data corruption is not distributed uniformly over subpopulations, our algorithms provide more meaningful robustness guarantees than standard guarantees that are oblivious to how the data corruption and the affected subpopulations are related. Our techniques establish a new connection between multigroup fairness and robustness.

1. Introduction

The gap between standard distributional assumptions and practical dataset limitations has been well-studied in machine learning literature – from unintended distribution shift to adversarially crafted data manipulations. Corresponding notions of *robustness* have been developed to reflect the goal of learning well in the presence of adversarially corrupted data, as well as attacks demonstrating that small amounts of corrupted data can seriously impact performance.

While attacks that target specific subgroups have been proposed (Jagielski et al., 2021), they give the adversary the power to change any point in the dataset to achieve its goal–even the ability to modify seemingly unrelated points outside the subgroup of interest. While this is reasonable as a

---

*Equal contribution [1]Stanford University, USA. Correspondence to: Charlotte Peale <cpeale@stanford.edu>.

strong worst-case adversarial assumption, data corruption issues are often far more localized in reality. In surveys, response bias may compromise answers from certain subpopulations (Meng, 2018; Bradley et al., 2021). As another example, when amassing internet data for training large models, certain sources can be less trustworthy or more toxic than others (Dodge et al., 2021). Since limitations to datasets may be isolated to certain groups, it is important to develop a fine-grained notion of robustness that ensures groups do not suffer undue harm due to corruption in unrelated data.

A fine-grained robustness notion We consider a new data-aware notion of robustness that we term *multigroup robustness*. At a high-level, a multigroup-robust learning algorithm guarantees that the effects of dataset corruption on every subpopulation-of-interest are bounded by the amount of corruption to data within that subpopulation (Figure 1).

More formally, we consider a binary-label learning problem where data from a domain X labeled with $y \in \{0, 1\}$ is inputted into a deterministic learning algorithm $\mathcal{A} : (X \times \{0, 1\})^* \to [0, 1]^X$ that outputs a predictor $p \in [0, 1]^X$. Given a set of subpopulations $\mathcal{C} \subseteq 2^X$, we say that $\mathcal{A}$ is multigroup robust with respect to $\mathcal{C}$ if, given a dataset $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ where the x_is are each drawn i.i.d. from some unknown distribution $\mathcal{D}_X$, given any other potentially corrupted dataset $S' = \{(x'_1, y'_1), ..., (x'_m, y'_m)\}$, we can guarantee that the difference in the mean prediction of *every* group $C \in \mathcal{C}$ is bounded by the amount of change to C between S and S', i.e.,

$$|\mathbb{E}_{\mathcal{D}_X}[(\mathcal{A}(S)(x) - \mathcal{A}(S')(x))\, \mathbf{1}[x \in C]]| \leq \mathsf{dist}_C(S, S')$$

where $\mathsf{dist}_C(S, S')$ is a measure of the changes to the set of points belonging to C between S and S', which we will formalize in Section 4.

Intuitively, this definition asks that when the points from a group are unchanged or change very little in S' (i.e., translating to a small $\mathsf{dist}_C(S, S')$), then the average prediction outputted by $\mathcal{A}$ on that group should also not change by much, thus preserving the group's accuracy-in-expectation. It is easy to construct algorithms that satisfy multigroup robustness, but are not very useful as learning algorithms; for example, the algorithm that always outputs the same predictor for any dataset. Thus, we will concentrate on the
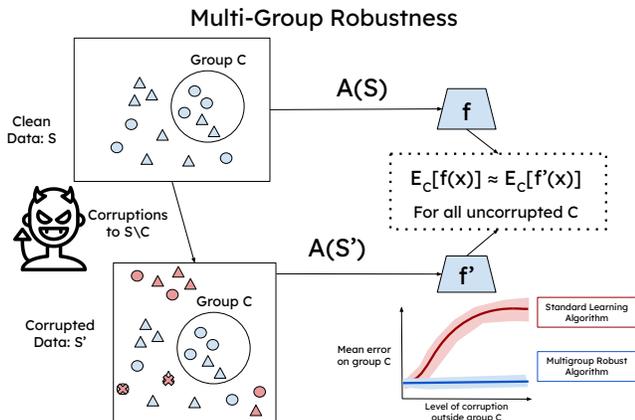
Figure 1: Intuitive illustration of multigroup robustness: for every group C, if points within the group are not modified, a multigroup robust algorithm produces a predictor that achieves marginal mean consistency with the clean data predictor (See Definition 4.1).

particular question of whether there exist efficient learning algorithms that provide multigroup robustness as well as an agnostic learning guarantee, i.e., that the predictor outputted by $\mathcal{A}$ performs at least as well as the best predictor from some benchmark set $\mathcal{P} \subseteq [0,1]^X$.

Standard algorithms are not multigroup robust Classic agnostic learning algorithms such as empirical risk minimization over the benchmark class can fail to satisfy multigroup robustness even for extremely simple families of sub-populations. Consider as an example the benchmark class containing the all ones or all zeros predictors, $\mathcal{P} = \{p_0, p_1\}$, and a uniform distribution over X where half of the points ($x \in X_0$) are labeled with 0s, and half of the points ($x \in X_1$) are labelled with ones. Given a sample from this half-and-half distribution, ERM will output the predictor matching the majority label in the dataset. However, because most datasets drawn will be close to half ones and half zeros, this means that only a tiny number of corrupted labels can drastically change the output from p_0 to p_1 or vice versa, impacting the predictions of a huge number of points despite not changing the overall accuracy by much.

In Section 7, we empirically demonstrate that several standard models for classification fail to preserve multigroup robustness under simple label-flipping and data addition attacks on the Adult Income Dataset.

Connections to multiaccuracy In light of these successful attacks, new ideas are necessary to develop algorithms that can match the performance guarantees of standard learning approaches while being multigroup robust.

When we only care about a small number of disjoint groups, a naive solution that provides multigroup robustness could

be to simply train a separate model on each group's data. However, this approach becomes inefficient as we consider a growing number of possibly overlapping groups. Moreover, the individually trained models may lose out on the predictive power that could be gained from using the dataset as a whole.

Instead, we turn to the notion of *multiaccuracy*, a learning objective originating in the algorithmic fairness literature that asks for a predictor to satisfy accuracy-in-expectation simultaneously on many groups (Hébert-Johnson et al., 2018; Kim et al., 2019). In Section 5 we outline how multiaccuracy's rigorous group-level guarantees can connect to our goal of multigroup robustness. However, because standard multiaccuracy algorithms assume access to i.i.d. data, we encounter challenges to using them in our setting where due to adversarial corruption, we cannot assume data is i.i.d.. We demonstrate how to bypass these obstacles by making appropriate modifications to standard multiaccuracy algorithms, and present a set of sufficient conditions for algorithms that provably achieve multigroup robustness. In Section 5.3, we supplement our results with a lower bound showing that multiaccuracy is a necessary property of any non-trivial algorithm that satisfies multigroup robustness.

Achieving multigroup robustness With these sufficient conditions in hand, in Section 6 we present an efficient post-processing approach that can be used to augment any existing learning algorithm to add both multigroup robustness and multiaccuracy guarantees, while preserving the performance guarantees of the original learning algorithm.

In Section 7, we supplement our theoretical results with experiments on real-world census datasets demonstrating that our post-processing approach can be added to existing learning algorithms to provide multigroup robustness protections without a drop in accuracy.

1.1. Our Contributions

To summarize, we list our main contributions below:

- Define a new notion of data-aware robustness, *multigroup robustness*, that ensures subgroups do not suffer undue harm due to corruptions in unrelated data (Section 4).

- Demonstrate general sufficient conditions for an algorithm to be multigroup robust by drawing connections to multiaccuracy (Section 5), and show that multiaccuracy is necessary for non-trivial multigroup robust algorithms (Section 5.3).

- Present an efficient post-processing algorithm that can provide an arbitrary learning algorithm with multi-

group robustness guarantees while preserving performance (Section 6).

- Empirically validate that standard learning algorithms are vulnerable to simple attacks on multigroup robustness, and that our postprocessing method successfully protects against these attacks while preserving accuracy (Section 7).

2. Related Work

Robust Machine Learning   Our work considers two notions of adversarial power similar to notions present in the literature on robust machine learning literature: distribution shift and datapoint corruption. Robustness to distribution shift considers worst-case guarantees when training data is drawn i.i.d. from a training distribution that does not match the true distribution, but is guaranteed to be in some uncertainty set around the true distribution, often a ball bounded by some notion of distance (see e.g. Ben-Tal et al. (2009)). Our notion of distance in the distribution shift setting is most similar to that of Ben-David et al. (2010) in that it also separates out the amount of covariate shift and label shift between distributions. We do so at a group level rather than at a population level, and consider label shift in terms of change in the mean rather than L1 distance, which means that their notion of distance is an upper bound to ours. Additionally, rather than bounding the size of the uncertainty set, we present adaptive guarantees that scale with the amount of distribution shift.

Moving beyond distribution shift, our strongest adversarial model is one of direct data corruption, in which an adversary can directly change points in a provided input dataset. This setting is similar to that of Kearns & Li (1988) and Bshouty et al. (2002), however the main difference is that most existing works limit the adversary's power by either allowing only a certain fraction of points to be corrupted, or allowing each point to be independently corrupted with some probability. We instead do not limit the adversary's power, but allow our bounds to adapt to the level of corruption and give better guarantees when the amount of corruption in a group is small. We choose to do this because we care about delivering heterogenous guarantees to a complex set of possibly overlapping groups in a population based on their level of corruption. Classic notions of constrained power such as allowing the adversary to change at most a certain fraction of datapoints are insufficient to distinguish between the case where corruptions are concentrated within a particular group, or more uniformly distributed throughout a population.

Algorithmic Stability   Our work also has connections to the area of *algorithmic stability*, in that it provides guarantees about the outputs of algorithms under small perturbations of the inputs (Bousquet & Elisseeff, 2000; Bassily et al., 2016). We choose to present our work as a robustness notion because it focuses on the goal of preserving group mean accuracy in the presence of corruption, and considers significant amounts of data corruption rather than the leave-one-out analysis usually employed in algorithmic stability works which considers only very small data perturbations (see e.g. (Bousquet & Elisseeff, 2000)).

Fairness and Data Poisoning/Robustness   Prior works have demonstrated that the fairness properties of machine learning models can be degraded by modifying small subsets of the training data (Solans et al., 2020; Van et al., 2022; Chai & Wang, 2023). Jagielski et al. (2021) show that subpopulations can be directly targeted in data poisoning attacks. Algorithms for finding fair predictors that are robust to data poisoning to any part of the dataset have also been proposed (Jin & Lai, 2023; Konstantinov & Lampert, 2022; Celis et al., 2021). These papers consider the task of learning a binary classifier in the presence of corrupted data, and they would like the classifier to be robust in that it satisfies certain fairness metrics such as equalized odds or demographic parity, whereas we measure robustness with respect to subgroup accuracy-in-expectation. These works are similar in that they consider robustness with respect to a set of groups, but differ in that they do not consider the location of corruption and how it relates to the affected groups. Moreover, they consider group fairness metrics that only consider disjoint groups whereas our multigroup approach can handle complex sets of overlapping groups. "Subgroup robustness" in recent literature refers to the performance of the worst (Martinez et al., 2021; Gardner et al., 2022). In distribution shift literature, "subgroup robustness" is used interchangeably with "worst group robustness" (Sagawa et al., 2019). In contrast, our definitions focus on how modifications to some groups in the training data can impact other unrelated groups.

Multiaccuracy and Multicalibration   Multiaccuracy and multicalibration are multigroup fairness notions that require a predictor to provide meaningful statistical guarantees (e.g., accuracy in expectation, calibration) on a large family of possibly overlapping subgroups of a population (Kleinberg et al., 2017; Hébert-Johnson et al., 2018; Kearns et al., 2018; Kim et al., 2019). Kim et al. (2022) show that multicalibration can ensure a predictor's robustness against distribution shift, achieving *universal adaptability*. They focus on covariate shift while assuming the conditional distribution of y given x remains the same and assume that the covariate shift can be represented by a propensity score function from a given family. Our work considers general forms of data corruption, both in covariate x and label y, and the corrupted data need not be i.i.d. from any distribution.

Robust Statistics Algorithms for estimating a variety of statistical quantities on corrupted data have been studied extensively in the literature, where the quantities to estimate include mean (Diakonikolas et al., 2019; Lai et al., 2016), covariance matrix (Diakonikolas et al., 2019; 2018), principal components (Candès et al., 2011), and beyond. In the typical setup of robust statistics, the corrupted data is formed by modifying or hiding a significant fraction of an otherwise i.i.d. dataset. The goal is to ensure that the estimation error is small relative to the fraction of corrupted data.

3. Preliminaries and Notation

A *deterministic learning algorithm* is defined as any algorithm $\mathcal{A} : (X \times \{0,1\})^* \to \mathcal{P}$ that takes a sample of data points $(x_1, y_1), ..., (x_n, y_n) \in X \times \{0,1\}$ for any $n \in \mathbb{Z}^+$ as input and outputs a predictor $p \in \mathcal{P}$ where X is a finite domain of points.

Given two datasets $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ and $S' = \{(x'_1, y'_1), ..., (x'_m, y'_m)\}$, we use the notation $S\Delta_X S'$ to denote a set containing the *symmetric difference* in terms of the multisets $\{x_1, ..., x_n\}$ and $\{x'_1, ..., x'_m\}$.

More formally, because these datasets could have duplicates of certain x-values, we cannot use a standard set definition of symmetric difference. This can be thought of as reinterpreting each $\{x_1, ..., x_n\}$ as a map μ from X to $\mathbb{Z}_{\geq 0}$ where $\mu(x)$ is equal to the number of times x appears in the sample, $\mu(x) = |\{i \in \{1, \ldots, n\} : x_i = x\}|$. Having similarly defined such a μ' for $\{x'_1, ..., x'_m\}$, the multiset symmetric difference of these two collections is defined as the map μ_Δ such that for all $x \in X$, $\mu_\Delta(x) = |\mu(x) - \mu'(x)|$.

Given two distributions $\mathcal{D}, \mathcal{D}'$ over a discrete domain X and a subpopulation $C \subseteq X$, we denote the statistical distance between $\mathcal{D}$ and $\mathcal{D}'$ restricted to C as

$$\Delta_C(\mathcal{D}, \mathcal{D}') :=$$
$$\sum_{x \in C} \left| \Pr_{X \sim \mathcal{D}}[X = x] - \Pr_{X \sim \mathcal{D}'}[X = x] \right|.$$

4. Multigroup Robustness

In this section, we define our notion of *multigroup robustness*. We consider a setting where a learner is provided with a sample of binary-labelled data points $(x_1, y_1), ..., (x_n, y_n) \in \mathcal{X} \times \{0,1\}$. We assume that the points could have been adversarially corrupted.

We consider two different levels of adversarial power. The strongest adversary we consider can make label-change adjustments to this dataset by replacing a point (x, y) with a new (x, y'), as well as add or delete new points to the dataset. Our main definition is designed to protect against

these strong data-dependent adversaries, and we state it formally below:

Definition 4.1 (Binary-label Multigroup Robustness). Let $\mathcal{C}$ be a subpopulation class consisting of subsets $C \subseteq X$. For any $n \in \mathbb{Z}^+, \varepsilon > 0, \delta \in [0,1]$, we say that a deterministic learning algorithm $\mathcal{A} : (X \times \{0,1\})^* \to \mathcal{P}$ is $(\mathcal{C}, n, \varepsilon, \delta)$-*multigroup robust* if for every distribution $\mathcal{D}_X$ over X, the following holds with probability at least $1 - \delta$ over $X_n = (x_1, ..., x_n)$ drawn i.i.d. from $\mathcal{D}_X$: for any $(y_1, ..., y_n) \in \{0,1\}^n$ and $(x'_1, y'_1), ..., (x'_m, y'_m) \in (X \times \{0,1\})^m$, let S and S' denote the two samples $\{(x_i, y_i)\}_{i=1}^n$ and $\{(x'_i, y'_i)\}_{i=1}^m$, respectively.

Defining $p := A(S)$ and $p' := A(S')$, we have

$$\left| \mathbb{E}_{x \sim \mathcal{D}_X}[(p(x) - p'(x))\mathbf{1}(x \in C)] \right|$$
$$\leq \frac{1}{n} \left| \sum_{i=1}^n y_i \mathbf{1}[x_i \in C] - \sum_{j=1}^m y'_j \mathbf{1}[x_j \in C] \right| \quad (1)$$
$$+ \frac{1}{n} |(S\Delta_X S') \cap C| + \varepsilon$$

for every $C \in \mathcal{C}$.

We highlight that our definition makes *no distributional assumptions* about the ground-truth y-values in the original dataset S, meaning that multigroup robustness implies a strong distribution-free robustness property that holds even when the original y-values were not i.i.d..

Here, the abstract $\text{dist}_C(S, S')$ used in the introduction is formalized to capture label flipping performed by the adversary (the first term) as well as addition and deletion of points (the second symmetric difference term). This means that this definition can also give a definition of robustness in settings where the adversary can only flip labels.

We also consider a weaker adversary that can only change the *distribution* the data is drawn from, and the particular training set is still drawn i.i.d. from that distribution:

Definition 4.2 (Binary-label Multigroup Robustness to Distribution Shift). Let $\mathcal{C}$ be a subpopulation class consisting of subsets $C \subseteq X$. For any $n \in \mathbb{Z}^+$, $\varepsilon > 0$, $\delta \in [0,1]$, we say that a deterministic learning algorithm $\mathcal{A} : (X \times \{0,1\})^* \to \mathcal{P}$ is $(\mathcal{C}, n, \varepsilon, \delta)$-*multigroup robust to distribution shift* if for any two distributions $\mathcal{D}, \mathcal{D}'$ over $X \times \{0,1\}$, the following holds with probability at least $1 - \delta$ over $S = ((x_1, y_1), ..., (x_n, y_n))$, $S' = ((x'_1, y'_1), ..., (x'_n, y'_n))$ drawn i.i.d. from $\mathcal{D}$ and $\mathcal{D}'$:

Defining $p := A(S)$ and $p' := A(S')$, we have

$$\left| \mathbb{E}_{x \sim \mathcal{D}_X}[(p(x) - p'(x))\mathbf{1}(x \in C)] \right|$$
$$\leq \left| \mathbb{E}_{(x,y) \sim \mathcal{D}}[y \mathbf{1}[x \in C]] - \mathbb{E}_{(x',y') \sim \mathcal{D}'}[y' \mathbf{1}[x' \in C]] \right|$$
$$+ \Delta_C(\mathcal{D}_X, \mathcal{D}'_X) + \varepsilon$$

$$(2)$$

for every $C \in \mathcal{C}$.

Note that here, the $\text{dist}_C(S, S')$ bound has been replaced with a measure of the distance between the corrupted and uncorrupted distribution, however the first term still captures the amount of label shift while the second term speaks to the covariate shift in the corrupted distribution.

5. Multigroup Robustness from Multiaccuracy

In this section, we give principled conditions for algorithms that provably achieve multigroup robustness. Our proofs are largely inspired by previous algorithms for *multiaccuracy*. We identify several key challenges in using those algorithms in our setting with corrupted data, and address these challenges by making appropriate modifications.

We first introduce the definition of multiaccuracy (Hébert-Johnson et al., 2018), a notion of algorithmic fairness that requires a predictor be accurate in expectation for every group in some collection of subgroups $C \subseteq 2^X$. Formally, we define multiaccuracy as follows:

Definition 5.1 (Multiaccuracy (Hébert-Johnson et al., 2018)). Let $\mathcal{C}$ be a subpopulation class consisting of subsets $C \subseteq X$ and $\mathcal{D}$ some target distribution over $X \times \{0, 1\}$. For $\varepsilon > 0$, we say a predictor $p : X \to [0, 1]$ is $(\mathcal{C}, \varepsilon)$-multiaccurate (MA) on $\mathcal{D}$ if for every $C \in \mathcal{C}$:

$$\left| \mathbb{E}_{(x,y) \sim D}\left[(y - p(x))\mathbf{1}[x \in C]\right] \right| \leq \varepsilon. \tag{3}$$

Throughout the paper, it will be useful to talk about algorithms that provide multiaccuracy guarantees. We formally define such an algorithm as follows:

Definition 5.2 (Multiaccurate Learning Algorithm). Let $\mathcal{C} \subseteq 2^X$ be a subpopulation class. Given $n \in \mathbb{Z}^+, \varepsilon > 0$, $\delta \in [0, 1]$, we say that a deterministic learning algorithm $\mathcal{A} : (X \times \{0, 1\})^* \to [0, 1]^X$ is $(\mathcal{C}, n, \varepsilon, \delta)$-multiaccurate if for any distribution $\mathcal{D}$ over $X \times \{0, 1\}$, given n i.i.d. data points $S = (x_1, y_1), ..., (x_n, y_n)$ from $\mathcal{D}$, the predictor $\mathcal{A}(S)$ satisfies $(\mathcal{C}, \varepsilon)$-MA on $\mathcal{D}$ with probability at least $1 - \delta$.

5.1. Standard Multiaccuracy Gives Multigroup Robustness under Distribution Shift

We begin by considering a setting where an adversary can only corrupt the data *distribution* rather than the sampled datapoints themselves, and show that in this setting, learners that output multiaccurate predictors with respect to a collection $\mathcal{C}$ are also multigroup robust to distribution shift with respect to $\mathcal{C}$.

For ease of notation, for a given distribution $\mathcal{D}$, predictor p, and subpopulation $C \subseteq X$ we denote $\text{MA-err}_{\mathcal{D}}(p, C) := \mathbb{E}_{(x,y) \sim \mathcal{D}}[(y - p(x))\mathbf{1}[x \in C]]$.

Lemma 5.3 (Robustness from MA). *Given two distributions $\mathcal{D}, \mathcal{D}'$ over $X \times \{0, 1\}$ and a collection of subsets $\mathcal{C} \subseteq 2^X$, let p and p' be $(\mathcal{C}, \varepsilon)$-MA predictors with respect to $\mathcal{D}$ and $\mathcal{D}'$, respectively. Then it holds that for all $C \in \mathcal{C}$,*

$$\begin{aligned}
&|\mathbb{E}_{\mathcal{D}}[(p(x) - p'(x))\mathbf{1}[x \in C]]| \\
&\leq |\text{MA-err}_{\mathcal{D}}(p', C) - \text{MA-err}_{\mathcal{D}'}(p', C)| + 2\varepsilon.
\end{aligned} \tag{4}$$

Intuitively, the left-hand-side of equation 4 already bears resemblance to a multigroup robustness statement with respect to p and p'. By reinterpreting the upper bound, we can show that multiaccurate learners provide multigroup robustness to distribution shift:

Lemma 5.4. *Let $\mathcal{C} \subseteq 2^X$ be a collection of subpopulations and let $\mathcal{A} : (X \times \{0, 1\})^n \to [0, 1]^X$ be a deterministic learning algorithm satisfying $(\mathcal{C}, n, \epsilon, \delta)$-MA. Then, $\mathcal{A}$ is also $(\mathcal{C}, n, 2\varepsilon, 2\delta)$-multigroup robust to distribution shift.*

5.2. Leveraging Uniform Convergence for Stronger Robustness

In total, we have shown so far that learning algorithms for multiaccuracy such as those of (Hébert-Johnson et al., 2018; Kim et al., 2019) *additionally* give out-of-the-box multigroup robustness guarantees against weak adversaries that can corrupt the data distribution. This result relies on the key assumption that while the corrupted distribution may be arbitrarily warped compared to the original distribution, we can still assume that the training data is drawn i.i.d. from the corrupted distribution. This is a key property that allows us to apply multiaccuracy algorithms that assume access to an i.i.d. datasource.

Moving beyond distributional shifts, we are also interested in stronger adversaries that can directly corrupt a data sample either through label change or addition/deletion of points. In the presence of these stronger adversaries, we can no longer assume access to an "clean" i.i.d. datasource. In the absence of i.i.d. data, we can still work with the empirical distribution: the uniform distribution over the data points. However, because multigroup robustness is a statement about the predictor similarity over the *true marginal distribution D_X*, it's not clear whether relying on the corrupted empirical distribution alone can give us these distributional guarantees necessary for robustness.

Despite this obstacle, our main result shows that by leveraging an additional uniform-convergence assumption, we can in fact achieve multigroup robustness against strong dataset-dependent adversaries for any algorithm that guarantees multiaccuracy on the empirical distribution. Our analysis for handling adversarially corrupted data differs from typical analyses where the uniform convergence assumption is applied to a learning algorithm with i.i.d. input data (see Remark 5.9).

We begin by showing in Lemma 5.6 that guaranteeing empirical multiaccuracy already yields a guarantee of predictor-closeness when measured over the empirical distribution of the uncorrupted dataset. We formally define an empirically robust learning algorithm as follows:

Definition 5.5 (Empirically Multiaccurate Learning Algorithm). Let $\mathcal{C}$ be a subpopulation class consisting of subsets $C \subseteq X$. Given $\varepsilon > 0$, we say that a deterministic learning algorithm $\mathcal{A} : (X \times \{0,1\})^* \to [0,1]^X$ is $(\mathcal{C}, \varepsilon)$-empirically-multiaccurate if when given as input data points $S = (x_1, y_1), ..., (x_n, y_n)$ from $X \times \{0,1\}$ for any $n \in \mathbb{Z}^+$, the predictor $p := \mathcal{A}(S)$ satisfies

$$\left| \frac{1}{n} \sum_{i=1}^{n} (y_i - p(x_i)) \mathbf{1}[x_i \in C] \right| \leq \varepsilon$$

for all $C \in \mathcal{C}$.

Lemma 5.6 (Pointwise Robustness from Empirical MA). Given two datasets $S = \{(x_i, y_i)\}_{i=1}^{n}$, $S' = \{(x'_j, y'_j)\}_{j=1}^{m}$ from $X \times \{0,1\}$ and a collection of subsets $\mathcal{C}$, let p and p' be $(\mathcal{C}, \varepsilon)$-MA and $(\mathcal{C}, \varepsilon')$-MA predictors with respect to $\mathsf{Uni}(S)$ and $\mathsf{Uni}(S')$, respectively. Then it holds that for all $C \in \mathcal{C}$,

$$\left| \mathbb{E}_{(x,y) \sim \mathsf{Uni}(S)}[(p'(x) - p(x))\mathbf{1}[x \in C]] \right|$$
$$\leq \frac{1}{n} \left| \sum_{(x,y) \in S} y\mathbf{1}[x \in C] - \sum_{(x',y') \in S'} y'\mathbf{1}[x' \in C] \right| \quad (5)$$
$$+ \frac{1}{n} |(S \Delta_X S') \cap C| + \varepsilon + \frac{m}{n}\varepsilon'.$$

While the right side of the statement of Lemma 5 matches the definition of multigroup robustness, the left side is still an expectation over the empirical distribution $\mathsf{Uni}(S)$ rather than the uncorrupted target distribution $\mathcal{D}_X$. This means that to show that multigroup robustness holds, it suffices to show that the empirical quantity $\left| \mathbb{E}_{(x,y) \sim \mathsf{Uni}(S)}[(p'(x) - p(x))\mathbf{1}[x \in C]] \right|$ is close to its population limit, $|\mathbb{E}_{x \sim D_x}[(p'(x) - p(x))\mathbf{1}[x \in C]]|$, for all $C \in \mathcal{C}$.

Our main result uses this reasoning to show that a learning algorithm that outputs a predictor multiaccurate with respect to the empirical distribution satisfies multigroup robustness whenever we can guarantee *uniform convergence* over all possible outputted predictors and subpopulations (See Definition 5.7 for a formal definition). We additionally show that under this assumption we can guarantee the outputted predictor is multiaccurate with respect to the target distribution when the learning algorithm is given uncorrupted i.i.d. data (Theorem 5.8).

Definition 5.7. Let $\mathcal{C} \subseteq 2^X$ be a class of subpopulations, and let $\mathcal{P} \subseteq [0,1]^X$ be a family of predictors. We say that $\mathcal{P}$ satisfies $(\mathcal{C}, n, \varepsilon, \delta)$-*uniform convergence* if for any

distribution $\mathcal{D}$ over $X \times \{0,1\}$, we are guaranteed that with probability at least $1 - \delta$ over the randomness of datapoints $(x_1, y_1), ..., (x_n, y_n)$ drawn i.i.d. from $\mathcal{D}$, we are guaranteed that the following inequalities hold for all $p \in \mathcal{P}$ and $C \in \mathcal{C}$:

$$\left| \frac{1}{n} \sum_{i=1}^{n} p(x_i)\mathbf{1}(x_i \in C) - \mathbb{E}_{\mathcal{D}}[p(x)\mathbf{1}(x \in C)] \right| \leq \varepsilon, \quad (6)$$

$$\left| \frac{1}{n} \sum_{i=1}^{n} y_i\mathbf{1}(x_i \in C) - \mathbb{E}_{\mathcal{D}}[y\mathbf{1}(x \in C)] \right| \leq \varepsilon. \quad (7)$$

Theorem 5.8. *Let* $\mathcal{A} : (X \times \{0,1\})^* \to \mathcal{P}$ *be a deterministic learning algorithm that outputs predictors from the family* $\mathcal{P} \subseteq [0,1]^X$. *For a family of subpopulations* $\mathcal{C} \subseteq 2^X$, *suppose that* $\mathcal{A}$ *satisfies empirical* $(\mathcal{C}, \varepsilon_1)$-*multiaccuracy and additionally suppose that* $\mathcal{P}$ *satisfies* $(\mathcal{C}, n, \varepsilon_2, \delta_2)$-*uniform convergence. Then,* $\mathcal{A}$ *is* $(\mathcal{C}, n, (1 + \frac{m}{n})\varepsilon_1 + 2\varepsilon_2, \delta_2)$-*multigroup robust as well as* $(\mathcal{C}, n, \varepsilon_1 + 2\varepsilon_2, \delta_2)$-*multiaccurate.*

Remark 5.9 (Strong use of uniform convergence). Uniform convergence is a standard technique for establishing the generalization of a learning algorithm and is typically applied to algorithms given uncorrupted, i.i.d. training data. The important difference in our approach is that we apply uniform convergence to algorithms whose input data is adversarially corrupted and cannot be treated as i.i.d. from any distribution. In our setting, i.i.d. data is first given to an *unrestricted* adversary to produce corrupted data, which is then given to the learning algorithm whose output model is *restricted* to a class. Since we do not restrict the behavior of the adversary, our analysis makes a stronger use of uniform convergence than typical analyses.

5.3. Lower Bounds

We now explore whether multiaccuracy is a necessary condition of a multigroup robust algorithm under a weak non-triviality assumption (see details in Appendix C).

Theorem 5.10 (Lower Bound). *Let* $\mathcal{C}$ *be a class of subpopulations,* $\mathcal{P} \subseteq [0,1]^X$ *a family of predictors containing the all ones predictor* $p(x) = 1$ *for all* $x \in X$, *and* $\mathcal{A} : (X \times \{0,1\}^*) \to \mathcal{P}$ *a deterministic learning algorithm. If* $\mathcal{A}$ *is* $(\mathcal{C}, n, \varepsilon_1, \delta_1)$-*multigroup robust and* $(n, \varepsilon_2, \delta_2)$-*accurate-in-expectation, and* $\mathcal{P}$ *satisfies* $(\mathcal{C}, n, \epsilon_3, \delta_3)$-*uniform convergence, then* $\mathcal{A}$ *is a* $(\mathcal{C}, n, \epsilon_1 + \epsilon_2 + 2\epsilon_3, 2\delta_1 + 2\delta_2 + \delta_3)$-*multiaccurate learning algorithm.*

6. Implementing Multigroup Robustness

So far, we have demonstrated sufficient conditions for a learning algorithm to satisfy multigroup robustness (empirical multiaccuracy and uniform convergence). We now show that multigroup robustness can be achieved efficiently in parallel with standard accuracy objectives. In particular,

we present a post-processing procedure that can convert any black-box learning algorithm into a multigroup robust learning algorithm that minimizes ℓ_2 error.

6.1. Post-processing Approach

We consider a setting where we are given access to an arbitrary deterministic learning algorithm $\mathcal{A} : (X \times \{0, 1\})^* \to [0, 1]^X$. We will demonstrate how to post-process this algorithm to produce a new algorithm $\mathsf{PP}_{\mathcal{A}} : (X \times \{0, 1\})^* \to [0, 1]^X$ (Algorithm 1) that provides comparable performance to that of $\mathcal{A}$ in terms of ℓ_2 error while also satisfying multigroup robustness. We present our post-processing approach in Algorithm 1.

Algorithm 1 Multiaccuracy Boost on Empirical Distribution

Parameters: $n \in \mathbb{Z}_{\geq 0}, \varepsilon \in \mathbb{R}_{\geq 0}, \mathcal{C} \subseteq \{0, 1\}^X$
Input: data points $(x_1, y_1), ..., (x_n, y_n) \in X \times \{0, 1\}$, learning algorithm $\mathcal{A} : \{0, 1\}^n \to [0, 1]^X$
Output: predictor $p : X \to [0, 1]$
Step 1: Training
initialize $p \leftarrow \mathcal{A}((x_1, y_1), ..., (x_n, y_n))$
Step 2: Post-processing
while $\exists C \in \mathcal{C}$ s.t. $|\frac{1}{n} \sum_{i=1}^n (p(x_i) - y_i)\mathbf{1}(x_i \in C)| > \varepsilon$
do
$\quad v_C := \mathsf{sgn}\left(\sum_{i=1}^n (p(x_i) - y_i)\mathbf{1}(x_i \in C)\right)$
$\quad$**for all** $x \in C$ **do**
$\quad\quad p(x) \leftarrow p(x) - v_C \epsilon$
$\quad\quad p(x) \leftarrow \max\{0, \min\{p(x), 1\}\}$
$\quad$**end for**
end while
Return: p

Algorithm 1 uses an iterative auditing approach to bring the predictor closer to being multiaccurate with each iteration of the While loop in Step 2. This is similar to standard algorithms for multiaccuracy presented in the literature (Hébert-Johnson et al., 2018; Kim et al., 2019), but is differentiated in that it audits using *the entire dataset* at each iteration, rather than sampling fresh data for each update step. This alteration is necessary in order to guarantee we achieve empirical multiaccuracy.

It follows immediately from the definition of Algorithm 1 that it outputs an empirically multiaccurate predictor (See Lemma B.3). We can also show that the algorithm is guaranteed to terminate in a bounded number of steps, and thus the class of predictors it can output is also bounded, giving us a uniform convergence result that we state formally and prove in Lemma B.6.

Lemmas B.3 and B.6 give us the two sufficient conditions for multigroup robustness described in Section 4 (empirical multiaccuracy and uniform convergence, respectively). Thus, we can show that $\mathsf{PP}_{\mathcal{A}}$ satisfies multigroup-robustness

and multiaccuracy:

Theorem 6.1. *Let $\mathcal{A} : (X \times \{0, 1\})^* \to \mathcal{P}$ be a deterministic learning algorithm that is guaranteed to output from a finite set of predictors $\mathcal{P} \subseteq [0, 1]^X$. Let $\mathsf{PP}_{\mathcal{A}}$ be the algorithm defined by Algorithm 1 on input $\mathcal{A}$ with input parameter $\epsilon > 0$. Then, for any $\delta \in [0, 1]$, $\mathsf{PP}_{\mathcal{A}}$ satisfies $(\mathcal{C}, n, (3 + \frac{m}{n})\epsilon, \delta)$-multigroup robustness and $(\mathcal{C}, n, 3\epsilon, \delta)$-multiaccuracy for any $n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^{1/\epsilon^2 + 1}/\delta)}{2\epsilon^2}$.*

Proof. The theorem follows immediately by combining Theorem 5.8 with Lemmas B.3 and B.6. $\quad\square$

6.2. Loss Minimization Guarantee

Lastly, we show that the post-processed predictor is not much worse than the initial predictor outputted by the original learning algorithm. Intuitively, this result follows from Lemma B.4, which tells us that each iteration of the post-processing step decreases the predictor's empirical ℓ_2-loss by at least ϵ^2. With this fact in hand, we can appeal to uniform convergence to show that the loss decrease also generalizes to the entire distribution. The following corollary is a general statement that holds for any run of the algorithm and follows from a stronger version of this result included in the appendix (Theorem B.7).

Corollary 6.2 (Corollary to Theorem B.7)**.** *Let $\mathcal{A} : (X \times \{0, 1\}^*) \to \mathcal{P}$ be a deterministic learning algorithm that is guaranteed to output from a finite set of predictors $\mathcal{P} \subseteq [0, 1]^X$. Let $\mathsf{PP}_{\mathcal{A}}$ be the algorithm defined by Algorithm 1 on input $\mathcal{A}$ with parameter $\epsilon > 0$. Given a sample $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ drawn i.i.d. from some distribution $\mathcal{D}$ over $X \times \{0, 1\}$, let $p := \mathcal{A}(S)$ be the predictor output by $\mathcal{A}$ on S, and let $p_{\mathsf{PP}} := \mathsf{PP}_{\mathcal{A}}$ be the predictor output after post-processing. Then, for any $\delta \in [0, 1]$ and $n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^{1/\epsilon^2})}{2\epsilon^2}$, we are guaranteed that with probability at least $1 - \delta$*

$$\mathbb{E}_{\mathcal{D}}[(y - p_{\mathsf{PP}}(x))^2] \leq \mathbb{E}_{\mathcal{D}}[(y - p(x))^2] + 2\epsilon.$$

6.3. Additional Extensions and Remarks

A data-efficient alternative algorithm. Previous algorithms for multicalibration require fresh data in each "boosting" iteration or apply adaptive data analysis techniques to ensure generalization (Hébert-Johnson et al., 2018; Kim et al., 2019). In contrast, Theorem 5.8 shows that the iterations can instead simply reuse the same data as in Algorithm 1, as long as the final model belongs to a class with bounded complexity. This property is often guaranteed for multicalibration algorithms (see e.g., Lemma B.6).[1] At a

---

[1] While our results are stated for multiaccuracy, they can be applied to multicalibration with appropriate modifications.

Adult Income Dataset: Effect of Label Shift in White Male Group on Other Subgroups
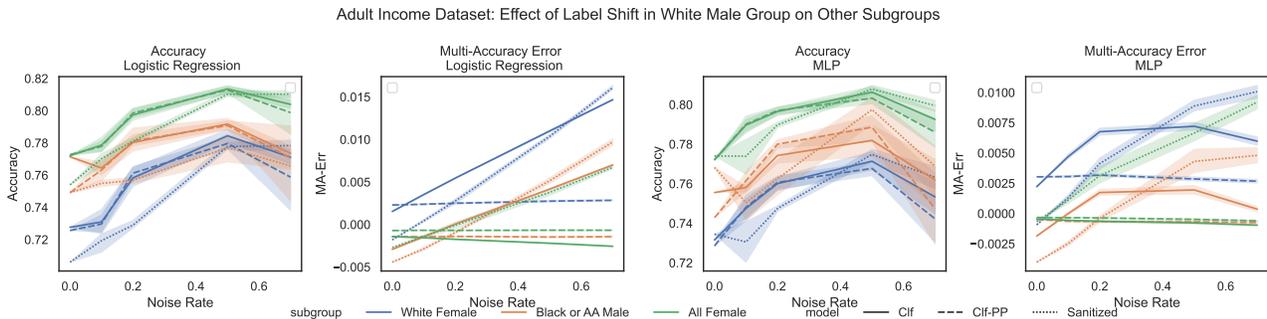
Figure 2: The effect of label change (0 to 1) in White male group on other subpopulations. For MA-err (closer to 0 is better), the base models (CLF) are susceptible to noise other groups, Algorithm 1 produces multigroup robust predictors (CLF-PP).

Adult Income Dataset: Data Addition of White Male Group Targeting White Female Group
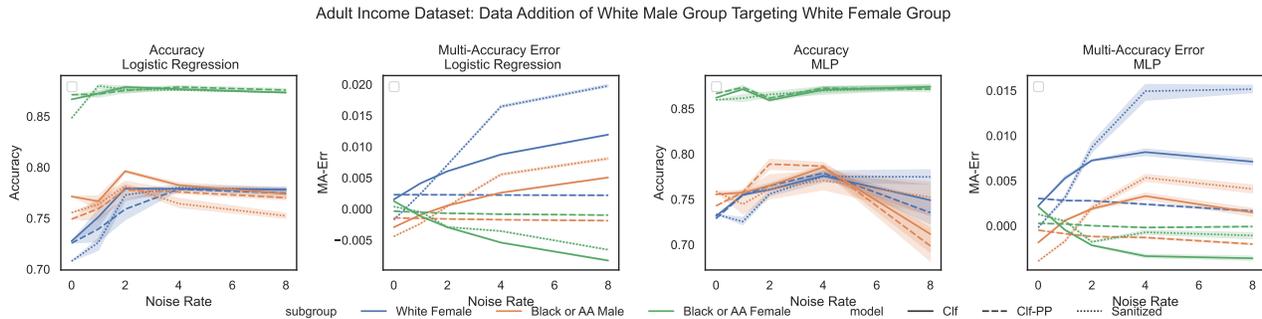
Figure 3: The effect of targeting the White female subgroup when only data addition from the White male subgroup is allowed. Multigroup robust predictors (CLF-PP) maintain a consistently low MA-err and high accuracy as more corrupted data is injected.

high level, previous multicalibration algorithms are analogous to the original boosting algorithm (Schapire, 1990), whereas ours are analogous to AdaBoost algorithm (Freund & Schapire, 1997). Using this alternative approach is crucial for our robustness guarantees.

Omniprediction. Recent work has explored the concept of an *omnipredictor*–a predictor whose predictions can be post-processed to select near-optimal actions for a large variety of loss functions, rather than needing to train the predictor to optimize for a specific loss function (Gopalan et al., 2022; 2023; Hu et al., 2023; Garg et al., 2024). It is known that multicalibration (a strengthening of multiaccuracy) is a sufficient condition for omniprediction. We note that our algorithm can be extended to the setting of multicalibration with appropriate modifications, which would provide a multigroup robust omniprediction algorithm that could provide accuracy guarantees beyond ℓ_2 loss. However, we note that while such an algorithm would be multigroup robust with respect to the outputted predictor, it is not clear if it would be multigroup robust with respect to the post-processed optimal actions. We leave this as a direction for future work.

Postprocessing on Fresh Data. While we envision using our post-processing step as part of an end-to-end learn-

ing pipeline and thus use the original data during post-processing, in certain settings, the original learning algorithm's training data may be unavailable and the post-processing step might need to use fresh data. In this case, we would continue to preserve accuracy guarantees, but could only guarantee multigroup robustness against strong adversaries with respect to the dataset used for post-processing. In the weaker distribution shift setting, fresh data for post-processing is enough to ensure overall multigroup robustness assuming that the distribution of the post-processing set is the same as that of the training data.

7. Experiments

Models and Datasets Due to the multigroup focus of our work, we examine several standard fairness datasets including Folktables-Income, Employment, Public Coverage (Ding et al., 2021), Bank (Moro & Cortez, 2012), and Law School (Sander, 2004) [2]. For the Folktables-Income task, we seek to predict whether the income of individuals was above $50k and we examine subgroups defined by race and sex. As an excerpt of our full results, we compare ro-

---

[2]See Section D.1 for full experiments for all datasets, different models classes, detailed data, and algorithm descriptions. Code to replicate experiments can be found at: `https://github.com/heyyjudes/multigroup-robust`

bustness to different attacks in three settings for two models: (1) CLF: Base classifier model: either logistic regression (LR) or a two-layer neural network (MLP), and (2) CLF-PP: Base classifier with post-processing using Algorithm 1, and (3) SANITATION: Label sanitation for input data points for each base classifier (Feng et al., 2014). We additionally investigate a robust linear regression baseline (Paudice et al., 2019) and find that the Multiacuracy performance was an order of magnitude worse than our models[3].

We report subgroup C accuracy and multi-accuracy error for each model at different noise rates:

- $\mathsf{Acc}_{\mathcal{D}}(p, C)$: $\frac{1}{|C|} \sum_{(x_i, y_i) \in C} \mathbf{1}[y_i = \mathbf{1}[p(x_i) > \gamma]]$

- $\mathsf{MA\text{-}err}_{\mathcal{D}}(p, C)$: $\frac{1}{n} \sum_{i=1}^{n} (p(x_i) - y_i)\mathbf{1}(x_i \in C)$

We seek to measure these quantities since the multi-accuracy error is the supremum of this $\mathsf{MA\text{-}err}_{\mathcal{D}}(p, C)$ across all identifiable subgroups $C \in \mathcal{C}$ and $\mathsf{Acc}_{\mathcal{D}}(p, C)$ gives an average classification metric where the γ threshold is optimized on the entire held-out validation set.

Label-Change We first consider robustness to label change when the set of training examples is unchanged. The noise rate represents the proportion of data points in the modified subgroup that has been shifted from 0 to 1[4]. Figure 2 shows the effect of randomly shifting labels in the White male group on three other groups that are unchanged: White female, Black or African American Male and Female subgroups. For Logistic Regression, the base classifier CLF, results in increased bias as the noise rate increases while the CLF-PP predictor retains a low multi-accuracy error while maintaining similar accuracy to the original model. A similar phenomenon is observed in the neural network (MLP) where CLF-PP remains multigroup robust. Label sanitization is more robust than the base models at small noise levels but is worse at larger noise levels; this is expected since the sanitation procedure uses neighboring labels.

Addition/Deletion We also demonstrate the robustness of Algorithm 1 through to additional data points designed to attack a specific target group. We employ a similar strategy as prior work designing subpopulation attacks (Jagielski et al., 2021) with the additional constraint that only data points outside of the target group can be used to create the poisoning dataset. To find data points to add that would affect the target group, we first cluster data points in a held-out set (not used for training or testing) using K-means. For each cluster where the target subgroup appears, we shift the labels of the of points we are allowed to modify

and add them to the training data. The amount of noise in this attack is scaled by how many times the identified data points are replicated before being added to the poisoned dataset. In Figure 3, we see that even at low levels of noise (i.e., poisoned points are replicated once or twice), the Logistic Regression and Neural Network (MLP) classifiers exhibit worsened multi-accuracy error while their post-processed counterparts (CLF-PP) exhibit consistently low MA-err without lower accuracy. Similar, to the label change setting, label sanitation helps in the small noise rate regime.

8. Discussion and Future Work

Motivated by practical scenarios where subgroups in datasets may be corrupted, we present *multigroup robustness* and provide an algorithm that gives meaningful robustness guarantees. Moreover, we empirically show that while standard models allow unrelated groups to suffer under data poisoning attacks, our algorithm applied to post-processing these predictors using the same poisoned data achieves multigroup robustness. While our analysis was limited to the binary setting, notions of multigroup robustness in the multi-class setting are an exciting direction for future work.

Acknowledgements

Impact Statement

Our work is motivated by the ethical impacts of subgroups modifying (adversarially or through sample bias) their data in order to manipulate outcomes for other groups. Standard learning algorithms allow for one group to have undue effects on another group and our work is a first step to addressing this shortcoming. Our work considers rich overlapping subgroups which is important in considering the many intersectional identities in today's society. Our use of sex and race variables and terminology is directly consistent with the US Census Bureau's standards.

References

Bassily, R., Nissim, K., Smith, A., Steinke, T., Stemmer, U., and Ullman, J. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pp. 1046–1059, 2016.

---

[3]We discuss detailed results in Section D.1

[4]Full algorithms can be found in Section D.2 for label change and D.3 for Addition/Deletion

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. *Robust optimization*, volume 28. Princeton university press, 2009.

Błasiok, J., Gopalan, P., Hu, L., Kalai, A. T., and Nakkiran, P. Loss minimization yields multicalibration for large neural networks. *arXiv preprint arXiv:2304.09424*, 2023.

Bousquet, O. and Elisseeff, A. Algorithmic stability and generalization performance. *Advances in neural information processing systems*, 13, 2000.

Bradley, V. C., Kuriwaki, S., Isakov, M., Sejdinovic, D., Meng, X.-L., and Flaxman, S. Unrepresentative big surveys significantly overestimated us vaccine uptake. *Nature*, 600(7890):695–700, 2021.

Bshouty, N. H., Eiron, N., and Kushilevitz, E. Pac learning with nasty noise. *Theoretical Computer Science*, 288(2): 255–275, 2002.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *J. ACM*, 58 (3), jun 2011. ISSN 0004-5411. doi: 10.1145/ 1970392.1970395. URL https://doi.org/10. 1145/1970392.1970395.

Celis, L. E., Mehrotra, A., and Vishnoi, N. Fair classification with adversarial perturbations. *Advances in Neural Information Processing Systems*, 34:8158–8171, 2021.

Chai, J. and Wang, X. To be robust and to be fair: Aligning fairness with robustness. *arXiv preprint arXiv:2304.00061*, 2023.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Robustly learning a gaussian: Getting optimal error, efficiently. In Czumaj, A. (ed.), *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pp. 2683–2702. SIAM, 2018. doi: 10.1137/1.9781611975031.171. URL https: //doi.org/10.1137/1.9781611975031.171.

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., and Stewart, A. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019. doi: 10.1137/ 17M1126680. URL https://doi.org/10.1137/ 17M1126680.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. Retiring adult: New datasets for fair machine learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Dodge, J., Sap, M., Marasović, A., Agnew, W., Ilharco, G., Groeneveld, D., Mitchell, M., and Gardner, M. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. *arXiv preprint arXiv:2104.08758*, 2021.

Feng, J., Xu, H., Mannor, S., and Yan, S. Robust logistic regression and classification. *Advances in neural information processing systems*, 27, 2014.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997. 1504. URL https://www.sciencedirect.com/ science/article/pii/S002200009791504X.

Gardner, J., Popovic, Z., and Schmidt, L. Subgroup robustness grows on trees: An empirical baseline investigation. *Advances in Neural Information Processing Systems*, 35: 9939–9954, 2022.

Garg, S., Jung, C., Reingold, O., and Roth, A. Oracle efficient online multicalibration and omniprediction. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2725–2792. SIAM, 2024.

Gopalan, P., Kalai, A. T., Reingold, O., Sharan, V., and Wieder, U. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

Gopalan, P., Hu, L., Kim, M. P., Reingold, O., and Wieder, U. Loss minimization through the lens of outcome indistinguishability. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

Hébert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pp. 1939–1948. PMLR, 2018.

Hu, L., Navon, I. R. L., Reingold, O., and Yang, C. Omnipredictors for constrained optimization. In *International Conference on Machine Learning*, pp. 13497–13527. PMLR, 2023.

Ilyas, A., Park, S. M., Engstrom, L., Leclerc, G., and Madry, A. Datamodels: Predicting predictions from training data. *arXiv preprint arXiv:2202.00622*, 2022.

Jagielski, M., Severi, G., Pousette Harger, N., and Oprea, A. Subpopulation data poisoning attacks. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3104–3122, 2021.

Jin, Y. and Lai, L. Fairness-aware regression robust to adversarial attacks. *IEEE Transactions on Signal Processing*, 2023.

Kearns, M. and Li, M. Learning in the presence of malicious errors. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pp. 267–280, 1988.

Kearns, M., Neel, S., Roth, A., and Wu, Z. S. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2564–2572. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kearns18a.html.

Kim, M. P., Ghorbani, A., and Zou, J. Multiaccuracy: Blackbox post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019.

Kim, M. P., Kern, C., Goldwasser, S., Kreuter, F., and Reingold, O. Universal adaptability: Target-independent inference that competes with propensity scoring. *Proceedings of the National Academy of Sciences*, 119(4):e2108097119, 2022. doi: 10.1073/pnas.2108097119. URL https://www.pnas.org/doi/abs/10.1073/pnas.2108097119.

Kleinberg, J., Mullainathan, S., and Raghavan, M. Inherent Trade-Offs in the Fair Determination of Risk Scores. In Papadimitriou, C. H. (ed.), *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 43:1–43:23, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/LIPIcs.ITCS.2017.43. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2017.43.

Konstantinov, N. and Lampert, C. H. Fairness-aware pac learning from corrupted data. *Journal of Machine Learning Research*, 23(160):1–60, 2022.

Lai, K. A., Rao, A. B., and Vempala, S. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 665–674, 2016. doi: 10.1109/FOCS.2016.76.

Martinez, N. L., Bertran, M. A., Papadaki, A., Rodrigues, M., and Sapiro, G. Blind pareto fairness and subgroup robustness. In *International Conference on Machine Learning*, pp. 7492–7501. PMLR, 2021.

Meng, X.-L. Statistical paradises and paradoxes in big data (i) law of large populations, big data paradox, and the 2016 us presidential election. *The Annals of Applied Statistics*, 12(2):685–726, 2018.

Moro, S., R. P. and Cortez, P. Bank Marketing. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5K306.

Paudice, A., Muñoz-González, L., and Lupu, E. C. Label sanitization against label flipping poisoning attacks. In *ECML PKDD 2018 Workshops: Nemesis 2018, UrbReas 2018, SoGood 2018, IWAISe 2018, and Green Data Mining 2018, Dublin, Ireland, September 10-14, 2018, Proceedings 18*, pp. 5–15. Springer, 2019.

Roh, Y., Lee, K., Whang, S., and Suh, C. Sample selection for fair and robust training. *Advances in Neural Information Processing Systems*, 34:815–827, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Sander, R. H. A systemic analysis of affirmative action in american law schools. *Stan. L. Rev.*, 57:367, 2004.

Sattigeri, P., Ghosh, S., Padhi, I., Dognin, P., and Varshney, K. R. Fair infinitesimal jackknife: Mitigating the influence of biased training data points without refitting. *Advances in Neural Information Processing Systems*, 35:35894–35906, 2022.

Schapire, R. E. The strength of weak learnability. *Machine Learning*, 5(2):197–227, Jun 1990. ISSN 1573-0565. doi: 10.1007/BF00116037. URL https://doi.org/10.1007/BF00116037.

Solans, D., Biggio, B., and Castillo, C. Poisoning attacks on algorithmic fairness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 162–177. Springer, 2020.

Van, M.-H., Du, W., Wu, X., and Lu, A. Poisoning attacks on fair machine learning. In *International Conference on Database Systems for Advanced Applications*, pp. 370–386. Springer, 2022.

A. Detailed Related Work

Fairness and influential Datapoints   The fairness outcomes of machine learning models can be attributed to training data (Roh et al., 2021). For example, demographic parity and equality of opportunity can depend on very few instances in the dataset according to influence scores. As a result, simply removing highly influential points can yield a dataset that results in models that are Pareto optimal for error and fairness metrics (Sattigeri et al., 2022). More generally beyond just fairness metrics, Ilyas et al. (2022) present a framework for prediction model prediction based on different subsets of the training data.

Fairness and Data Poisoning   Prior works have demonstrated that the fairness properties of machine learning models can be degraded by modifying small subsets of the training data (Solans et al., 2020; Van et al., 2022; Chai & Wang, 2023). Furthermore, Jagielski et al. (2021) shows empirically that subpopulations can be directly targeted in data poisoning attacks while minimizing the impact on non-target subpopulations. Algorithms for finding fair predictors that are robust to data poisoning to any part of the dataset have also been proposed for regression (Jin & Lai, 2023). The term "subgroup robustness" has also appeared in recent literature to mean the performance of the worst, often intersectional group (Martinez et al., 2021; Gardner et al., 2022). In the context of distribution shift, "subgroup robustness" is used interchangeably with "worst group robustness" (Sagawa et al., 2019). In contrast, our definitions focus on characterizing how modifications to training data, due to adversaries or sampling biases, impact different subgroups.

Multiaccuracy and Multicalibration   Multiaccuracy and multicalibration are multigroup fairness notions introduced by Hébert-Johnson et al. (2018) (see also Kearns et al., 2018; Kleinberg et al., 2017). These notions require a predictor to provide meaningful statistical guarantees (e.g., accuracy in expectation, calibration) on a large family of possibly overlapping subgroups of a population. Kim et al. (2019) apply post-processing to neural network models to achieve multiaccuracy. Recently, Błasiok et al. (2023) show that minimizing a proper loss over neural networks of a certain size yields multicalibration w.r.t. all subgroups identifiable by neural nets of a smaller size.

Kim et al. (2022) show that multicalibration can ensure a predictor's robustness against distribution shift, achieving *universal adaptability*. They focus on covariate shift, where the marginal distribution of x may change between training and testing, but the conditional distribution of y given x remains the same. Their results assume that the covariate shift can be represented by a propensity score function from a given class. Our work considers general forms of data corruption, both in covariate x and label y, and the corrupted data need not be i.i.d. from any distribution.

B. Missing Proofs

B.1. Section 5

Proof of Lemma 5.3. The proof follows by invoking the definition of multiaccuracy. Consider any $C \in \mathcal{C}$. Because p and p' are both ε-MA with respect to $\mathcal{D}$ and $\mathcal{D}'$, respectively, we are guaranteed that

$$|\text{MA-err}_{\mathcal{D}}(p, C) - \text{MA-err}_{\mathcal{D}'}(p', C)| \le 2\varepsilon \tag{8}$$

Expanding out the definition of MA-err gives us

$$\text{MA-err}_{\mathcal{D}}(p, C) - \text{MA-err}_{\mathcal{D}'}(p', C) \tag{9}$$
$$= \text{MA-err}_{\mathcal{D}}(p, C) - \text{MA-err}_{\mathcal{D}'}(p', C) \tag{10}$$
$$+ \mathbb{E}_{(x,y)\sim\mathcal{D}}[p'(x)\mathbf{1}[x \in C]] \tag{11}$$
$$- \mathbb{E}_{(x,y)\sim\mathcal{D}}[p'(x)\mathbf{1}[x \in C]] \tag{12}$$
$$= \mathbb{E}_{(x,y)\sim\mathcal{D}}[(p'(x) - p(x))\mathbf{1}[x \in C]] \tag{13}$$
$$+ \text{MA-err}_{\mathcal{D}}(p', C) - \text{MA-err}_{\mathcal{D}'}(p', C) \tag{14}$$

And thus we are guaranteed that the absolute value of the equation spanning lines 13 and 14 is at most 2ε. By moving the terms in line 14 to the right-hand-side, we get the lemma's statement:

$$|\mathbb{E}_{\mathcal{D}}[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{15}$$
$$\le |\text{MA-err}_{\mathcal{D}}(p', C) - \text{MA-err}_{\mathcal{D}'}(p', C)| + 2\varepsilon. \tag{16}$$

This completes the proof. □

Proof of Lemma 5.4. For an arbitrary predictor p' and $C \in \mathcal{C}$, we expand the quantity in the upper bound of Lemma 4:

$$\text{MA-err}_{\mathcal{D}}(p', C) - \text{MA-err}_{\mathcal{D}'}(p', C)$$
$$= \mathbb{E}_{\mathcal{D}}[(y - p'(x))\mathbf{1}[x \in C]] - \mathbb{E}_{\mathcal{D}'}[(y - p'(x))\mathbf{1}[x \in C]]$$
$$= \mathbb{E}_{\mathcal{D}}[y\mathbf{1}[x \in C]] - \mathbb{E}_{\mathcal{D}'}[y\mathbf{1}[x \in C]]$$
$$+ \mathbb{E}_{\mathcal{D}'_X}[p'(x)\mathbf{1}[x \in C]] - \mathbb{E}_{\mathcal{D}_X}[p'(x)\mathbf{1}[x \in C]]$$

Where we note that we can upper bound the quantity on the last line by

$$\mathbb{E}_{\mathcal{D}'_X}[p'(x)\mathbf{1}[x \in C]] - \mathbb{E}_{\mathcal{D}_X}[p'(x)\mathbf{1}[x \in C]]$$
$$= \sum_{x \in C} \left(\Pr_{X \sim \mathcal{D}'_X}[X = x] - \Pr_{X \sim \mathcal{D}_X}[X = x] \right) p'(x)$$
$$\leq \sum_{x \in C} | \Pr_{X \sim \mathcal{D}'_X}[X = x] - \Pr_{X \sim \mathcal{D}_X}[X = x]|$$
$$= \Delta_C(\mathcal{D}_X, \mathcal{D}'_X).$$

Therefore,

$$|\text{MA-err}_{\mathcal{D}}(p', C) - \text{MA-err}_{\mathcal{D}'}(p', C)|$$
$$\leq |\mathbb{E}_{\mathcal{D}}[y\mathbf{1}[x \in C]] - \mathbb{E}_{\mathcal{D}'}[y\mathbf{1}[x \in C]]| + \Delta_C(\mathcal{D}_X, \mathcal{D}'_X)$$

Thus, whenever p and p' are both ε-MA, we can apply Lemma 5.3 to conclude that

$$|\mathbb{E}_{\mathcal{D}}[(p(x) - p'(x))\mathbf{1}[x \in C]]|$$
$$\leq |\mathbb{E}_{\mathcal{D}}[y\mathbf{1}[x \in C]] - \mathbb{E}_{\mathcal{D}'}[y\mathbf{1}[x \in C]]| \qquad (17)$$
$$+ \Delta_C(\mathcal{D}_X, \mathcal{D}'_X) + 2\varepsilon.$$

It remains to show that this happens with probability at least $1 - 2\delta$ over the randomness of the samples from $\mathcal{D}$ and $\mathcal{D}'$. This follows from observing that the probability that $\mathcal{A}$ fails to output a multiaccurate predictor is at most δ, and so the probability $\mathcal{A}$ outputs a p or p' that is not multiaccurate is at most 2δ by a union bound. We conclude that equation 17 holds with probability at least $1 - 2\delta$ over the random samples from $\mathcal{D}$ and $\mathcal{D}'$, completing the proof. □

Proof of Lemma 5.6. For ease of notation, we denote $\mathbb{E}_{(x,y) \sim \text{Uni}(S)}$ and $\mathbb{E}_{(x,y) \sim \text{Uni}(S')}$ by $\mathbb{E}_S$ and $\mathbb{E}_{S'}$ respectively.

Consider any $C \in \mathcal{C}$. We rewrite the LHS of equation 5 as

$$|\mathbb{E}_S[(p'(x) - p(x))\mathbf{1}[x \in C]]|$$
$$= |\mathbb{E}_S[(y - p(x))\mathbf{1}[x \in C]]| + |\mathbb{E}_S[(p'(x) - y)\mathbf{1}[x \in C]]|$$
$$\leq |\mathbb{E}_S[(p'(x) - y)\mathbf{1}[x \in C]]| + \varepsilon$$

where the last line invokes the assumption that p is $(\mathcal{C}, \varepsilon)$-MA with respect to $\text{Uni}(S)$. Moreover, we have

$$|\mathbb{E}_S[(p'(x) - y)\mathbf{1}[x \in C]]| + \varepsilon$$
$$\leq \left| \mathbb{E}_S[(p'(x) - y)\mathbf{1}[x \in C]] - \frac{m}{n}\mathbb{E}_{S'}[(p'(x) - y)\mathbf{1}[x \in C]] \right|$$
$$+ \frac{m}{n}|\mathbb{E}_{S'}[(p'(x) - y)\mathbf{1}[x \in C]]| + \varepsilon$$
$$\leq \left| \mathbb{E}_S[p'(x)\mathbf{1}[x \in C]] - \frac{m}{n}\mathbb{E}_{S'}[p'(x)\mathbf{1}[x \in C]] \right|$$
$$+ \left| \mathbb{E}_S[y\mathbf{1}[x \in C]] - \frac{m}{n}\mathbb{E}_{S'}[y\mathbf{1}[x \in C]] \right| + \frac{m}{n}\varepsilon' + \varepsilon$$

Where this time we use the assumption that p' is $(\mathcal{C}, \varepsilon')$-MA with respect to $\mathsf{Uni}(S')$. Substituting in the definition of the uniform distributions over S and S', we have

$$\left| \mathbb{E}_S[p'(x)\mathbf{1}[x \in C]] - \frac{m}{n}\mathbb{E}_{S'}[p'(x)\mathbf{1}[x \in C]] \right| \tag{18}$$

$$+ \left| \mathbb{E}_S[y\mathbf{1}[x \in C]] - \frac{m}{n}\mathbb{E}_{S'}[y\mathbf{1}[x \in C]] \right| + \frac{m}{n}\varepsilon' + \varepsilon \tag{19}$$

$$= \frac{1}{n}\left| \sum_{i=1}^{n} p'(x_i)\mathbf{1}[x_i \in C]] - \sum_{j=1}^{m} p'(x'_j)\mathbf{1}[x'_j \in C]] \right| \tag{20}$$

$$+ \frac{1}{n}\left| \sum_{i=1}^{n} y_i\mathbf{1}[x_i \in C]] - \sum_{j=1}^{m} y'_j\mathbf{1}[x'_j \in C]] \right| + \frac{m}{n}\varepsilon' + \varepsilon \tag{21}$$

Thus, it suffices to show that 20 is upper-bounded by $\frac{1}{n}|(S\Delta_X S') \cap C|$. Indeed, we can rewrite in terms of maps μ_S and μ'_S for the multisets $\{x_1, ..., x_n\}$ and $\{x'_1, ..., x'_m\}$ to get

$$\frac{1}{n}\left| \sum_{i=1}^{n} p'(x_i)\mathbf{1}[x_i \in C]] - \sum_{j=1}^{m} p'(x'_j)\mathbf{1}[x'_j \in C]] \right|$$

$$= \frac{1}{n}\left| \sum_{x \in X} (\mu_S(x) - \mu_{S'}(x))p'(x)\mathbf{1}[x \in C] \right|$$

$$\leq \frac{1}{n}\left| \sum_{x \in X} |\mu_S(x) - \mu_{S'}(x)|\mathbf{1}[x \in C] \right|$$

$$= \frac{1}{n}|(S\Delta_X S') \cap C|.$$

This completes the proof. $\qquad\square$

Proof of Theorem 5.8. We restate the theorem as two separate lemmas (B.1 and B.2) and provide proofs for each below. The result immediately follows from the combination of these lemmas. $\qquad\square$

Lemma B.1. *Let $\mathcal{A} : (X \times \{0,1\})^* \to \mathcal{P}$ be a deterministic learning algorithm that outputs predictors from the family $\mathcal{P} \subseteq [0,1]^X$. For a family of subpopulations $\mathcal{C} \subseteq 2^X$, suppose that $\mathcal{A}$ satisfies empirical $(\mathcal{C}, \varepsilon_1)$-multiaccuracy and additionally suppose that $\mathcal{P}$ satisfies $(\mathcal{C}, n, \varepsilon_2, \delta_2)$-uniform convergence. Then, $\mathcal{A}$ is $(\mathcal{C}, n, (1 + \frac{m}{n})\varepsilon_1 + 2\varepsilon_2, \delta_2)$-multigroup robust.*

Lemma B.2. *Let $\mathcal{A} : (X \times \{0,1\})^* \to \mathcal{P}$ be a deterministic learning algorithm that outputs predictors from the family $\mathcal{P} \subseteq [0,1]^X$. For a family of subpopulations $\mathcal{C} \subseteq 2^X$, suppose that $\mathcal{A}$ satisfies empirical $(\mathcal{C}, \varepsilon_1)$-multiaccuracy and additionally suppose that $\mathcal{P}$ satisfies $(\mathcal{C}, n, \varepsilon_2, \delta_2)$-uniform convergence. Then for any distribution $\mathcal{D}$ over $X \times \{0,1\}$, $\mathcal{A}$ is a $(\mathcal{C}, n, \varepsilon_1 + 2\varepsilon_2, \delta_2)$-multiaccurate learning algorithm.*

Proof of Lemma B.1. Take any $C \in \mathcal{C}$ and distribution $\mathcal{D}_X$. Let $x_1, ..., x_n$ be points drawn i.i.d. from $\mathcal{D}_X$, and let $S = (x_1, y_1), ..., (x_n, y_n)$, $S' = (x'_1, y'_1), ..., (x'_m, y'_m)$ be datasets where the points other than $x_1, ..., x_n$ can be any value.

We consider the LHS of the multigroup robustness criterion:

$$|\mathbb{E}_{\mathcal{D}_X}[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{22}$$

$$\leq |\mathbb{E}_{\mathcal{D}_X}[(p(x) - p'(x))\mathbf{1}[x \in C]] - \mathbb{E}_S[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{23}$$

$$+ |\mathbb{E}_S[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{24}$$

$$\tag{25}$$

Due to our assumption of empirical multiaccuracy, we know that p and p' are $(\mathcal{C}, \varepsilon_1)$-MA predictors with respect to $\mathsf{Uni}(S)$ and $\mathsf{Uni}(S')$, respectively. Thus, we can apply Lemma 5.6 to upper bound 24 to get

$$|\mathbb{E}_{D_X}[(p(x) - p'(x))\mathbf{1}[x \in C]] - \mathbb{E}_S[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{26}$$
$$+ |\mathbb{E}_S[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{27}$$
$$\leq |\mathbb{E}_{D_X}[(p(x) - p'(x))\mathbf{1}[x \in C]] - \mathbb{E}_S[(p(x) - p'(x))\mathbf{1}[x \in C]]| \tag{28}$$
$$+ \frac{1}{n}\left|\sum_{i=1}^n y_i \mathbf{1}[x_i \in C] - \sum_{j=1}^m y'_j \mathbf{1}[x'_j \in C]\right| \tag{29}$$
$$+ \frac{1}{n}|(S \Delta_X S') \cap C| + \left(1 + \frac{m}{n}\right)\varepsilon_1. \tag{30}$$
$$\tag{31}$$

Finally, we use our uniform convergence assumption to upper-bound 28:

$$|\mathbb{E}_{D_X}[(p(x) - p'(x))\mathbf{1}[x \in C]] - \mathbb{E}_S[(p(x) - p'(x))\mathbf{1}[x \in C]]|$$
$$\leq |\mathbb{E}_{D_X}[p(x)\mathbf{1}[x \in C]] - \mathbb{E}_S[p(x)\mathbf{1}[x \in C]]|$$
$$+ |\mathbb{E}_{D_X}[p'(x)\mathbf{1}[x \in C]] - \mathbb{E}_S[p'(x)\mathbf{1}[x \in C]]|$$
$$\leq 2\varepsilon_2$$

where the final bound follows from uniform convergence and holds simultaneously for all $C \in \mathcal{C}$ with probability at least $1 - \delta_2$.

Substituting this into our original bound, we conclude that with probability at least $1 - \delta_2$, for all $C \in \mathcal{C}$, p and p' satisfy

$$|\mathbb{E}_{D_X}[(p(x) - p'(x))\mathbf{1}[x \in C]]|$$
$$\leq \frac{1}{n}\left|\sum_{i=1}^n y_i \mathbf{1}[x_i \in C] - \sum_{j=1}^m y'_j \mathbf{1}[x'_j \in C]\right|$$
$$+ \frac{1}{n}|(S \Delta_X S') \cap C| + \left(1 + \frac{m}{n}\right)\varepsilon_1 + 2\varepsilon_2.$$

Thus, we conclude that $\mathcal{A}$ is $(\mathcal{C}, n, \left(1 + \frac{m}{n}\right)\varepsilon_1 + 2\varepsilon_2, \delta_2)$-multigroup robust. $\qquad\square$

Proof of Lemma B.2. Consider any $\mathcal{D}$ and let $p = \mathcal{A}((x_1, y_1), ..., (x_n, y_n))$ where each (x_i, y_i) is drawn i.i.d. from $\mathcal{D}$ for sufficiently large $n \geq m(\mathcal{C}, \varepsilon_2, \delta_2)$.

Consider any $C \in \mathcal{C}$. We note that

$$|\mathbb{E}_D[(y - p(x))\mathbf{1}[x \in C]]|$$
$$= \left|\mathbb{E}_D[(y - p(x))\mathbf{1}[x \in C]] + \frac{1}{n}\sum_{i=1}^n (y_i - p(x_i))\mathbf{1}[x_i \in C] - \frac{1}{n}\sum_{i=1}^n (y_i - p(x_i))\mathbf{1}[x_i \in C]\right|$$
$$\leq \left|\mathbb{E}_D[(y - p(x))\mathbf{1}[x \in C]] - \frac{1}{n}\sum_{i=1}^n (y_i - p(x_i))\mathbf{1}[x_i \in C]\right| + \left|\frac{1}{n}\sum_{i=1}^n (y_i - p(x_i))\mathbf{1}[x_i \in C]\right|$$
$$\leq \left|\mathbb{E}_D[(y - p(x))\mathbf{1}[x \in C]] - \frac{1}{n}\sum_{i=1}^n (y_i - p(x_i))\mathbf{1}[x_i \in C]\right| + \varepsilon_1$$

Where the last step holds by our assumption that p is empirically multiaccurate. Continuing to simplify, we get

15

$$\left| \mathbb{E}_D[(y - p(x))\mathbf{1}[x \in C]] - \frac{1}{n}\sum_{i=1}^{n}(y_i - p(x_i))\mathbf{1}[x_i \in C] \right| + \varepsilon_1$$

$$\leq \left| \mathbb{E}_D[p(x)\mathbf{1}[x \in C]] - \frac{1}{n}\sum_{i=1}^{n}p(x_i)\mathbf{1}[x_i \in C] \right| + \left| \mathbb{E}_D[y\mathbf{1}[x \in C]] - \frac{1}{n}\sum_{i=1}^{n}y_i\mathbf{1}[x_i \in C] \right| + \varepsilon_1$$

By our uniform convergence guarantee, we are guaranteed that the above quantity is bounded by $2\varepsilon_2 + \varepsilon_1$ simultaneously for all $\mathcal{C}$ with probability at least $1 - \delta_2$, and thus we know that with probability at least $1 - \delta_2$, p satisfies

$$|\mathbb{E}_D[(y - p(x))\mathbf{1}[x \in C]]| \leq 2\varepsilon_2 + \varepsilon_1$$

for all $C \in \mathcal{C}$, and thus is $(\mathcal{C}, 2\varepsilon_2 + \varepsilon_1)$-MA with probability at least $1 - \delta_2$, completing the proof. $\qquad \square$

B.2. Section 6

Lemma B.3 (Empirical MA of Algorithm 1). *Algorithm 1 satisfies empirical $(\mathcal{C}, \varepsilon)$-multiaccuracy.*

Proof. Note that the stopping condition of Algorithm 1 implies that for all $C \in \mathcal{C}$,

$$\left| \frac{1}{n}\sum_{i=1}^{n}(y_i - p(x_i))\mathbf{1}[x_i \in C] \right| \leq \varepsilon.$$

Thus, p must be empirically $(\mathcal{C}, \varepsilon)$-MA. $\qquad \square$

While we have verified that the algorithm will satisfy empirical multiaccuracy upon termination, we have yet to verify whether it satisfies the uniform convergence property, which together with Lemma B.3 would imply multigroup robustness by our result in Theorem 5.8. Moreover, we need to show that the post-processing step does not decrease the predictor's ℓ_2 error by too much, and also that the algorithm is actually guaranteed to terminate in a small number of steps. All of these properties will follow from the following lemma, which shows that each update step can *only decrease* the predictor's ℓ_2 error on the empirical distribution.

Lemma B.4. *Given an arbitrary dataset $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, let $\nu_S : [0, 1]^X \to \mathbb{R}^{\geq 0}$ be a function capturing the empirical ℓ_2 error of a predictor on S:*

$$\nu_S(p) = \frac{1}{n}\sum_{i=1}^{n}(y_i - p(x_i))^2.$$

Given input data S, every iteration of the while loop in Algorithm 1 strictly decreases ν_S by at least ϵ^2, or terminates.

Proof of Lemma B.4. Let $p \in [0, 1]^X$ be an arbitrary predictor and consider one iteration of the while loop in Algorithm 1. If there exists no $C \in \mathcal{C}$ with $\left| \frac{1}{n}\sum_{i=1}^{n}(p(x_i) - y_i)\mathbf{1}[x_i \in C] \right| > \varepsilon$, then the algorithm terminates by definition.

In the other case, there exists some $C \in \mathcal{C}$ with $\left| \frac{1}{n}\sum_{i=1}^{n}(p(x_i) - y_i)\mathbf{1}[x_i \in C] \right| > \varepsilon$. Let $v_C = \mathsf{sgn}(\frac{1}{n}\sum_{i=1}^{n}(p(x_i) - y_i)\mathbf{1}[x_i \in C])$, and let p' be the updated predictor such that

$$p'(x) = p(x) - v_C \varepsilon \mathbf{1}[x \in C].$$

We consider the difference in potential functions:

$$\nu_S(p) - \nu_S(p') = \frac{1}{n}\sum_{i=1}^{n}(p(x_i) - y_i)^2 - (p'(x_i) - y_i)^2$$

we note that for all $x_i \notin C$, $p(x_i) = p'(x_i)$, so we can cancel out these terms to get

$$\frac{1}{n} \sum_{x_i, y_i \in S, x_i \in C} (p(x_i) - y_i)^2 - (p'(x_i) - y_i)^2$$

$$= \frac{1}{n} \sum_{x_i, y_i \in S, x_i \in C} (p(x_i) - y_i + p'(x_i) - y_i)(p(x_i) - y_i - p'(x_i) + y_i)$$

$$= \frac{1}{n} \sum_{x_i, y_i \in S, x_i \in C} (2(p(x_i) - y_i) - v_C \epsilon) v_C \epsilon$$

$$= 2\epsilon \left| \frac{1}{n} \sum_{i=1}^{n} (p(x_i) - y_i) \mathbf{1}[x_i \in C] \right| - \frac{|\{x_i, y_i \in S, x_i \in C\}| \epsilon^2}{n}$$

$$\geq 2\epsilon^2 - \frac{|\{x_i, y_i \in S, x_i \in C\}| \epsilon^2}{n}$$

$$\geq 2\epsilon^2 - \epsilon^2$$

$$= \epsilon^2$$

so, we have shown that $\nu_S(p) - \nu_S(p') \geq \epsilon^2$. Note that there is a final update to clip p' to be $[0, 1]$ values on all x. However, because every $y_i \in [0, 1]$, this can only reduce the ℓ_2 error (ν_S) of the updated predictor, and so we conclude that every non-terminating iteration of the algorithm reduces ν_S by at least ϵ^2, proving the claim. $\square$

As promised, a number of nice properties can now be derived using Lemma B.4. First, we can bound the number of iterations of the algorithm via a potential function argument (See the appendix B.2 for details):

Lemma B.5 (Stopping Time of Algorithm 1). *Algorithm 1 makes at most $1/\epsilon^2$ iterations of the while loop before terminating.*

Proof of Lemma B.5. Given input dataset $S = \{(x_1, y_1), ..., (x_n, y_n)\}$, we use the empirical ℓ_2 error,

$$\nu_S(p) = \frac{1}{n} \sum_{i=1}^{n} (y_i - p(x_i))^2$$

as a potential function. We note that by definition, for any p, we are guaranteed that $0 \leq \nu_S(p) \leq 1$.

Thus, applying the result of Lemma B.4, we can conclude that the number of iterations without termination can be at most $1/\epsilon^2$, otherwise ν_S would become negative, resulting in a contradiction. $\square$

Next, we show a uniform convergence result by bounding the set of predictors that can be outputted by Algorithm 1. Intuitively, we do this by using the result of Lemma B.5 to conclude that not too many updates are made to the predictor during post-processing, and thus the class of predictors output by the post-processing step is not too much more complex than that of the original learning algorithm.

Lemma B.6. *Let $\mathcal{A} : (X \times \{0, 1\})^* \to \mathcal{P}$ be a deterministic learning algorithm that is guaranteed to output from a finite set of predictors $\mathcal{P} \subseteq [0, 1]^X$. Let $\mathsf{PP}_\mathcal{A}$ be the algorithm defined by Algorithm 1 on input $\mathcal{A}$ and $\epsilon > 0$. Then, for any $\delta \in [0, 1]$ the family of predictors that can be output by $\mathsf{PP}_\mathcal{A}$ satisfies $(\mathcal{C}, n, \epsilon, \delta)$-uniform convergence for any $n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^{1/\epsilon^2 + 1}/\delta)}{2\epsilon^2}$.*

Proof of Lemma B.6. Let $\mathcal{P}_{\mathsf{PP}}$ be the family of predictors that can be output by $\mathsf{PP}_\mathcal{A}$. We proceed by bounding the size of $\mathcal{P}_{\mathsf{PP}}$.

Note that the predictor output by the initialization step of $\mathsf{PP}_\mathcal{A}$ must be from $\mathcal{P}$, and so there are $|\mathcal{P}|$ possible predictors that can be output at the initialization step of the algorithm.

From there, for any particular predictor output during initialization, note that at each iteration of the algorithm, there are at most $2|\mathcal{C}|$ possible different updates that can be made to the predictor.

Thus, applying the result of Lemma B.5 that says the number of iterations is bounded by $1/\epsilon^2$, we can conclude that the number of possible predictors that can be output by the algorithm is at most $(|\mathcal{P}|)\,(2|\mathcal{C}|)^{\frac{1}{\epsilon^2}}$.

By Chernoff bounds, for any particular p and C, the probability that

$$\left| \frac{1}{n} \sum_{i=1}^{n} p(x_i)\mathbf{1}[x_i \in C] - \mathbb{E}_{\mathcal{D}}[p(x)\mathbf{1}[x \in C]] \right| > \epsilon$$

is upper bounded by $2\exp(-2n\epsilon^2)$.

Including the ys noted in the definition of uniform convergence, this means that we need the above condition to hold for a total of

$$\left((|\mathcal{P}|)\,(2|\mathcal{C}|)^{\frac{1}{\epsilon^2}} + 1 \right) |\mathcal{C}|$$

pairs of p and C.

Thus, it suffices to take

$$n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^{1/\epsilon^2+1}/\delta)}{2\epsilon^2}$$

and thus $\mathcal{P}_{\mathsf{PP}}$ satisfies $(\mathcal{C}, n, \epsilon, \delta)$-uniform convergence for any $n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^{1/\epsilon^2+1}/\delta)}{2\epsilon^2}$. $\qquad\square$

Theorem B.7. *Let $\mathcal{A} : (X \times \{0,1\}^*) \to \mathcal{P}$ be a deterministic learning algorithm that is guaranteed to output from a finite set of predictors $\mathcal{P} \subseteq [0,1]^X$. Let $\mathsf{PP}_{\mathcal{A}}$ be the algorithm defined by Algorithm 1 on input $\mathcal{A}$ with parameter $\epsilon > 0$. Given a sample $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ drawn i.i.d. from some distribution $\mathcal{D}$ over $X \times \{0,1\}$, let $p := \mathcal{A}(S)$ be the predictor output by $\mathcal{A}$ on S, and let p_k be the predictor after $k \geq 0$ iterations of the post-processing update in $\mathsf{PP}_{\mathcal{A}}(S)$. Then, for any $\delta \in [0,1]$ and $n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^k)}{2\epsilon^2}$, we are guaranteed that with probability at least $1 - \delta$*

$$\mathbb{E}_{\mathcal{D}}[(y - p_k(x))^2] \leq \mathbb{E}_{\mathcal{D}}[(y - p(x))^2] + 2\epsilon - k\epsilon^2.$$

Proof of Theorem B.7. Note that by Lemma B.4, we are guaranteed that after k iterations, the empirical squared loss has decreased by at least $k\epsilon^2$, and thus

$$\frac{1}{n} \sum_{i=1}^{n} (y_i - p_k(x_i))^2 \leq \frac{1}{n} \sum_{i=1}^{n} (y_i - p(x_i))^2 - k\epsilon^2.$$

Note that by a Chernoff bound, for any fixed predictor p, we have that

$$\left| \mathbb{E}_{\mathcal{D}}[(y - p(x))^2] - \frac{1}{n} \sum_{i=1}^{n} (y_i - p(x_i))^2 \right| < \epsilon$$

with probability at most $2\exp(-2n\epsilon^2)$.

Using the same reasoning as in Lemma B.6, both p and p_k are guaranteed to belong to a subset of predictors of size at most $|\mathcal{P}|(2|\mathcal{C}|)^k$.

Thus, taking $n \geq \frac{\log(|\mathcal{P}|(2|\mathcal{C}|)^k)}{2\epsilon^2}$ guarantees that with probability at least $1 - \delta$, we have both

$$\left| \mathbb{E}_{\mathcal{D}}[(y - p(x))^2] - \frac{1}{n} \sum_{i=1}^{n} (y_i - p(x_i))^2 \right| < \epsilon$$

and

$$\left| \mathbb{E}_{\mathcal{D}}[(y - p_k(x))^2] - \frac{1}{n} \sum_{i=1}^{n} (y_i - p_k(x_i))^2 \right| < \epsilon$$

Thus, substituting these inequalities into our empirical expression gives us

$$\mathbb{E}_{\mathcal{D}}[(y - p_k(x))^2] \leq \mathbb{E}_{\mathcal{D}}[(y - p(x))^2] - k\epsilon^2 + 2\epsilon,$$

completing the proof. $\qquad\qquad\square$

C. Lower Bounds

Thus far, we've seen that certain multiaccurate learning algorithms can provide strong multigroup robustness guarantees while preserving performance guarantees when used as a post-processing step on an existing predictor.

In this section, we explore whether multiaccuracy is a necessary condition of a multigroup robust algorithm under a weak non-triviality assumption.

Clearly, a learning algorithm that always outputs the same predictor trivially satisfies multigroup robustness while not being multiaccurate, but is not a very useful learning algorithm. In order to circumvent these edge cases, we introduce a very weak non-triviality assumption, namely that an algorithm matches the overall mean of the true outcomes.

Definition C.1 (Accuracy in Expectation). We say a learning algorithm $\mathcal{A} : (X \times \{0, 1\})^* \to [0, 1]^X$ is (n, ε, δ)-accurate-in-expectation if given n i.i.d. datapoints $(x_1, y_1), ..., (x_n, y_n)$ from any distribution $\mathcal{D}$ over $X \times \{0, 1\}$, $\mathcal{A}$ outputs a predictor $p = \mathcal{A}((x_1, y_1), ..., (x_n, y_n))$ satisfying

$$|\mathbb{E}_D[y - p(x)]| \leq \varepsilon$$

with probability at least $1 - \delta$.

Note that this is a far weaker assumption than multiaccuracy, in that the predictor only needs to match the overall mean of the true outcomes, rather than on any particular groups. However, we will show that even this weak accuracy assumption paired with multigroup robustness implies multiaccuracy.

Theorem C.2 (Lower Bound, Theorem 5.10). *Let $\mathcal{C}$ be a class of subpopulations, $\mathcal{P} \subseteq [0, 1]^X$ a family of predictors containing the all ones predictor $p(x) = 1$ for all $x \in X$, and $\mathcal{A} : (X \times \{0, 1\}^*) \to \mathcal{P}$ a deterministic learning algorithm. If $\mathcal{A}$ is $(\mathcal{C}, n, \varepsilon_1, \delta_1)$-multigroup robust and $(n, \varepsilon_2, \delta_2)$-accurate-in-expectation, and $\mathcal{P}$ satisfies $(\mathcal{C}, n, \epsilon_3, \delta_3)$-uniform convergence, then $\mathcal{A}$ is a $(\mathcal{C}, n, \epsilon_1 + \epsilon_2 + 2\epsilon_3, 2\delta_1 + 2\delta_2 + \delta_3)$-multiaccurate learning algorithm.*

Proof of Theorem 5.10. Let $\mathcal{A}$ be the algorithm described in the theorem that is $(\mathcal{C}, n, \varepsilon_1, \delta_1)$-multigroup robust and $(n, \varepsilon_2, \delta_2)$-accurate-in-expectation.

Consider any $\mathcal{D}$ over $X \times \{0, 1\}$ with marginal $\mathcal{D}_X$ and let $S = \{(x_1, y_1), ..., (x_n, y_n)\}$ be drawn i.i.d. from $\mathcal{D}$.

Consider the two amended datasets $S_0 = \{(x_1, 0), ..., (x_n, 0)\}$ and $S_1 = \{(x_1, 1), ..., (x_n, 1)\}$. Let $p_0 = \mathcal{A}(S_0), p_1 = \mathcal{A}(S_1), p = \mathcal{A}(S)$.

Because S_0 and S_1 are identical to i.i.d. samples from distributions with the same marginal distribution but all zeros or all ones, we know that with probability at least $1 - 2\delta_2$, we have

$$\mathbb{E}_{D_X}[p_0(x)] \leq \varepsilon_2, \mathbb{E}_{D_X}[1 - p_1(x)] \leq \varepsilon_2.$$

When these two inequalities hold true, we also have for all $C \in \mathcal{C}$:

$$\mathbb{E}_{D_X}[p_0(x)\mathbf{1}[x \in C]] \leq \epsilon_2,$$

$$\mathbb{E}_{D_X}[p_1(x)\mathbf{1}[x \in C]] \geq \Pr_{D_X}[x \in C] - \epsilon_2.$$

Now, by multigroup robustness, we are guaranteed that with probability at least $1 - 2\delta_1$, we have the following two inequalities for all $C \in \mathcal{C}$:

$$|\mathbb{E}_{D_X}[(p_0(x) - p(x))\mathbf{1}[x \in C]]| \leq \frac{1}{n}\sum_{i=1}^{n} y_i \mathbf{1}[x_i \in C] + \epsilon_1$$

$$|\mathbb{E}_{D_X}[(p_1(x) - p(x))\mathbf{1}[x \in C]| \leq \frac{1}{n}\sum_{i=1}^{n}(1 - y_i)\mathbf{1}[x_i \in C] + \epsilon_1$$

Simplifying the first inequality gives us

$$\mathbb{E}_{D_X}[p(x)\mathbf{1}[x \in C]]$$

$$\leq \mathbb{E}_{D_X}[p_0(x)\mathbf{1}[x \in C]] + \frac{1}{n}\sum_{i=1}^{n}y_i\mathbf{1}[x_i \in C] + \epsilon_1$$

$$\leq \frac{1}{n}\sum_{i=1}^{n}y_i\mathbf{1}[x_i \in C] + \epsilon_1 + \epsilon_2$$

$$\leq \mathbb{E}_D[y\mathbf{1}[x \in C]] + \epsilon_1 + \epsilon_2 + \epsilon_3$$

Where the last step holds by uniform convergence for all $C \in \mathcal{C}$ with probability at least $1 - \delta_3$.

Similarly, the second inequality simplifies to

$$\mathbb{E}_{D_X}[p(x)\mathbf{1}[x \in C]]$$

$$\geq \mathbb{E}_{D_X}[p_1(x)\mathbf{1}[x \in C]] + \frac{1}{n}\sum_{i=1}^{n}y_i - \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}[x_i \in C] - \epsilon_1$$

$$\geq \Pr_{D_X}[x \in C] + \frac{1}{n}\sum_{i=1}^{n}y_i\mathbf{1}[x_i \in C] - \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}[x_i \in C] - \epsilon_1 - \epsilon_2$$

$$\geq \mathbb{E}_D[y\mathbf{1}[x \in C]] - \epsilon_1 - \epsilon_2 - 2\epsilon_3$$

where the last step follows by two applications of uniform convergence and our assumption that $\mathcal{P}$ contains the constant ones predictor, and also hold simultaneously for all $C \in \mathcal{C}$ with probability at least $1 - \delta_3$.

Thus, we conclude that with probability at least $1 - (2\delta_1 + 2\delta_2 + \delta_3)$, it holds that

$$|\mathbb{E}_D[(p(x) - y)\mathbf{1}[x \in C]| \leq \epsilon_1 + \epsilon_2 + 2\epsilon_3$$

for all $C \in \mathcal{C}$ and thus $\mathcal{A}$ is $(\mathcal{C}, n, \epsilon_1 + \epsilon_2 + 2\epsilon_3, 2\delta_1 + 2\delta_2 + \delta_3)$-multiaccurate. $\qquad\square$

D. Full Experiments

D.1. Models and Datasets

Due to the fairness focus of our work, we use Folktables (Ding et al., 2021), a modern version of the UCI Adult Dataset based on the yearly American Community Survey. In this section, we expand on the Income experiments in the main text to other tasks. We look at 3 binary tasks with different feature dimensions, subgroup ratios, and positive class ratios (Table 1). These datasets were taken from a mid-sized state where there is significant racial diversity: Louisiana (the 25th most populous state in the US). For each task, we consider subgroups defined by race and sex. We consider the three most common racial groups as coded by: *White Alone* (62%), *Black or African American Alone* (32.8%), and *Asian Alone* (2%). Although the Hispanic and Latino populations constitute 5.6% of the population in Louisiana, [5] the Census Bureau officially states that *"People who identify their origin as Hispanic, Latino, or Spanish may be of any race"*[6]. Thus we include the three aforementioned racial identifiable groups for our experiments.

For this task, we seek to predict whether the income of individuals was above \$50k and we examine subgroups defined by race and sex. We compare robustness to different attacks in three settings for 5 different models:

- MAEMP: Multiaccuracy Boost on Empirical Distribution (Algorithm 1)

- CLF: Base classifier model: We evaluate performance across 5 models of different model classes

---

[5]https://www.census.gov/quickfacts/fact/table/LA/RHI725222#RHI725222
[6]https://www.census.gov/quickfacts/fact/note/US/RHI625222

Dataset	Task	n	Features	Positive Class Ratio	White Ratio	Black or AA Ratio	Asian Ratio	Male Ratio
ACS Income	Income $>$ $50k$	20667	47	0.332	0.710	0.235	0.020	0.507
ACS Employment	Employed	43589	69	0.414	0.675	0.266	0.018	0.483
ACS Public Coverage	Covered by Public Health Insurance	16879	76	0.406	0.589	0.344	0.023	0.424
Bank Dataset	Open a Term Deposit	45,221	17	0.113	N/A	N/A	N/A	N/A
Law School	Bar Exam Passage	20800	11	0.890	0.841	0.058	0.038	0.062

Table 1: Task summary for Folktables, Law School and Bank Datasets. Since race information is not available for the bank dataset, we will use communication preference and age.

- - LR: Logistic Regression Classifier. Probabilities from this model are probability estimates of each class from the logistic function.
  - DT: Decision Tree classifier. Probabilities from this model are the fraction of samples of the same class in a leaf[7].
  - kNN: k-Nearest Neighbors algorithm with parameter search from 3, 5, and 7 nearest neighbors. Probabilities from this model represent the proportion of neighbors in the predicted class.
  - XGB: XGBoost algorithm using the default *binary-logistic* objective. Probabilities are class probabilities summed and normalized across trees within the boosting system.
  - MLP: Multi-Layer Perceptron neural network with 2 layers after performing hyperparameter search over the learning rate and l_2 regularization weight. Probabilities are the output of the logistic function for each class.

- CLF-PP: Base classifier with post-processing using Multiaccuracy Boost on Empirical Distribution.

- SANITIZATION: Data sanitization before training with each base classifier following prior work (Feng et al., 2014)

- ROBUST LOGISTIC REGRESSION: Logistic Regression Algorithm to minimize the effect of outliers proven to be effective against data poisoning attacks in general (Paudice et al., 2019).

To evaluate each model, we report subgroup accuracy and multi-accuracy error for each model at different noise rates. More formally, for subgroup C:

- Accuracy$_{\mathcal{D}}(p, C)$: $\frac{1}{|C|} \sum_{(x_i, y_i) \in C} \mathbf{1}[y_i = \mathbf{1}[p(x_i) > \gamma]]$

- MA-err$_{\mathcal{D}}(p, C)$: $\frac{1}{n} \sum_{i=1}^{n} (p(x_i) - y_i) \mathbf{1}(x_i \in C)$

We seek to measure these quantities since multi-accuracy error is the supremum of this MA-err$_{\mathcal{D}}(p, C)$ across all identifiable subgroups $C \in \mathcal{C}$ and Accuracy gives an average classification metric where the γ threshold is optimized on a held-out validation set. We measure all of these results on a test set while both training and post-process with Algorithm 1 are done on the training set.

D.2. Label-Change

---

[7]https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

---

Algorithm 2 Label Change Algorithm

Parameters: target: $t \in \{0, 1, *\}$, modify group $C \subseteq \{0, 1\}^X$, noise ratio $\sigma \in [0, 1]$.
Input: clean dataset $D = (x_1, y_1), ..., (x_n, y_n) \in X \times \{0, 1\}$
Output: corrupted dataset $D' = (x_1, y'_1), ..., (x_n, y'_n) \in X \times \{0, 1\}$
$D' = \{\}$
for all $(x_i, y_i) \in D$ **do**
  $z \sim Uniform([0, 1])$
  if $z < \sigma$ and $y_i == t$ **then**
    $y'_i = 1 - y_i$
  else
    $y'_i = y_i$
  end if
  $D' = D' \cup (x_i, y'_i)$
end for
Return: D'

Adult Income Dataset: Different Label Change of White Male Group on other groups
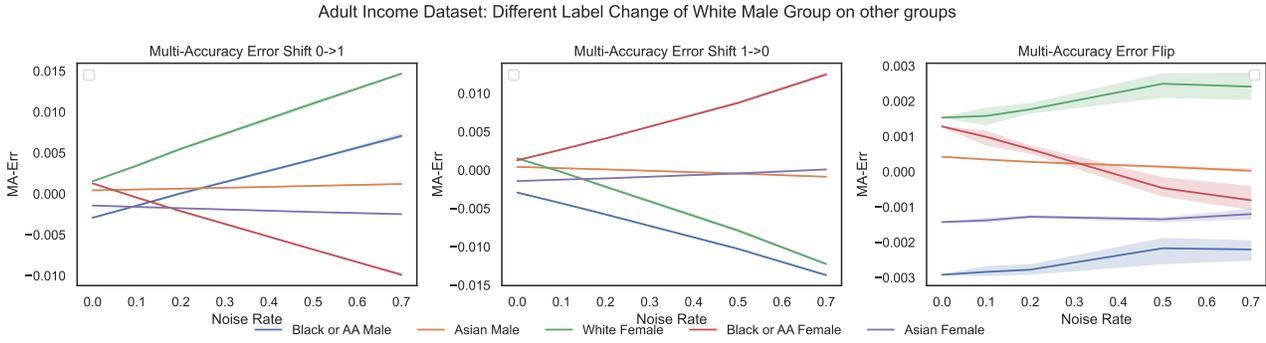
Figure 4: Effect of label change in the white male group on other groups with a Logistic Regression Classifier. As the amount of noise increases, the MA-err of the resulting predictor grows away from 0. The direction depends on whether the shift in label is from 0 to 1 or from 1 to 0.

Designing Label Shift   First, we consider robustness to label noise when the set of training examples is unchanged (See Section 4). We randomly (1) flip labels, (2) shift labels to 1, and (3) shift labels to 0 within a subgroup and observe the effects on other subgroups. Algorithm 2 describes the generation of all three types of label change. The noise rate in this corruption represents the proportion of data points in the modified subgroup that has been shifted.

We compare the different effects of label shifts first on the Logistic Regression model to understand the impact of label change. Figures 4, 5, and 6 show the effect of different types to label change when different groups are modified. We observed significant changes in MA-err on other groups particularly when the white male group and white female group are modified by shifting the labels from 0 to 1. We proceed with experiments in label change by modifying the white male group by shifting labels from 0 to 1 in different ratios. These preliminary results also show that due to the low ratio of the Asian population, MA-err will remain small for this population since the normalization term is the total test set size. Moving forward, we will use the 4 larger subpopulations white-male, white-female, black or AA-male, and black or AA female to measure the results of our proposed algorithm.

Robust Logistic Regression and Label Shift   Table 3 summarizes the overall accuracy at different levels of label shift. The robust LR algorithm actually achieves higher accuracy. However, in Table 2, the multiaccuracy of robust logistic regression is an order of magnitude larger than the baselines and our empirical multiaccuracy algorithm.

Robustness to label change through Multi-Accuracy Boosting and Post Processing   For the ACSIncome dataset, Figure 7 (Also Figure 1 in the main text) shows the effect of randomly shifting labels in the White Male group on three other groups: White Female, Black or African American Male, and Black or African American Female. For Logistic Regression, we see that the base classifier (dotted line), results in increased bias as the noise rate increases while the MAEMP predictor

Adult Income Dataset: Different Label Change of White Female Group on other groups
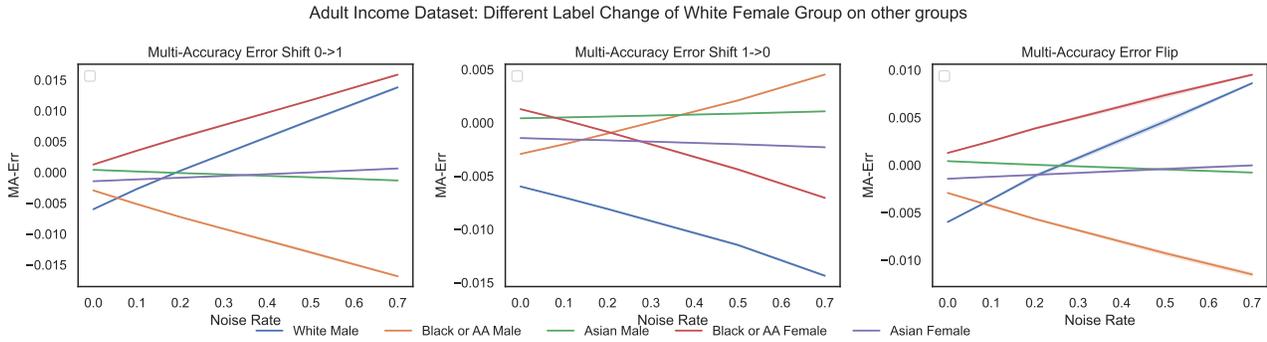
Figure 5: Effect of label change in the white female group on other groups with a Logistic Regression Classifier. In this group, flipping the labels has a large effect on the Black population.

Adult Income Dataset: Different Label Change of Black Male Group on other groups
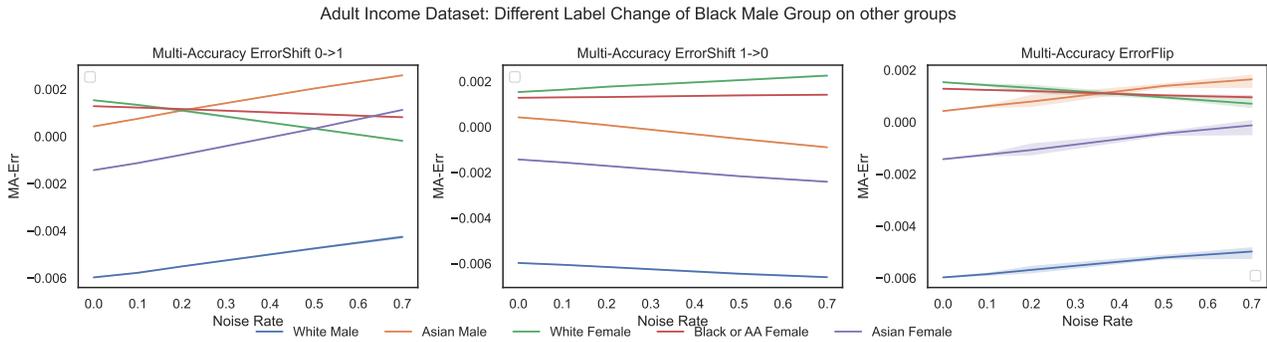
Figure 6: Effect of label change in the black male group on other groups with a Logistic Regression Classifier. Since this group is a smaller fraction of the population, the induced change in MA-err is much smaller.

Adult Income Dataset: Label Change of White Male Group
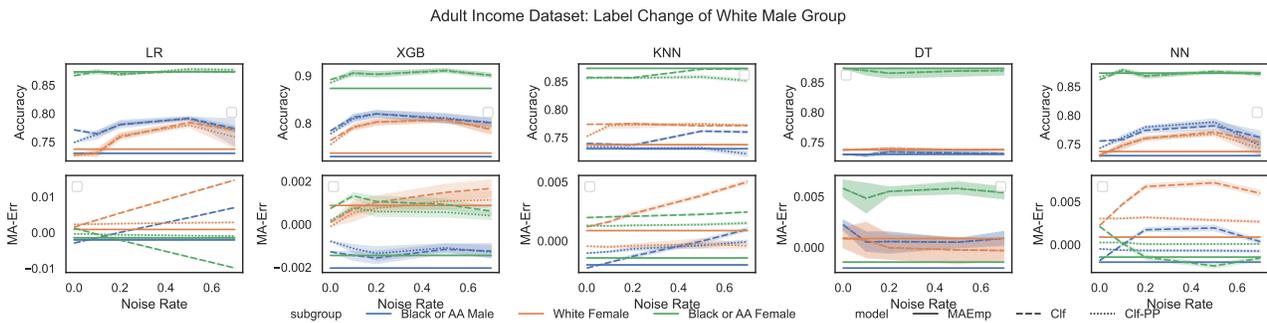
Figure 7: Comparison of different predictors on corrupted data with label change at various ratios on the ACSIncome Dataset. When labels are shifted from 0 to 1 in the white male group, baseline classifiers such as Logistic Regression, k-Nearest Neighbors, and Neural Network (MLP) also become more biased for other subgroups (i.e., increasing MA-err). However, both the uniformly initialized and post processed predictor using Algorithm 1 are robust to label change in other groups.

Adult Income Dataset: Label Change of White Female Group
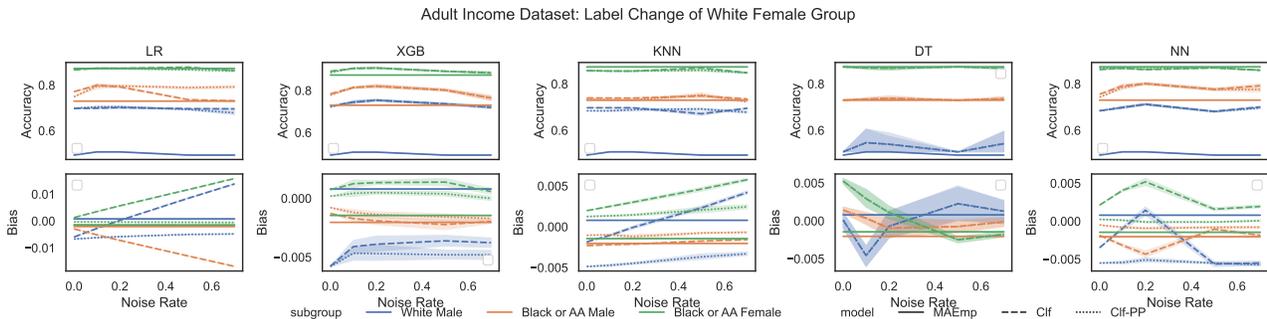
Figure 8: Comparison of different predictors on corrupted data with label change at various ratios on the ACSIncome Dataset: 0 to 1 label shift in the white female group. A similar result as modifying the white male group follows.

Table 2: ACS Income MultiAccuracy (Signed Group Mean Error) with increasing noise from label shift. Modified Group: White Men - Target/Effect Group: White Women (standard error i.e. mean over 10 runs)

Algorithm	0.0 ($\times 10^{-2}$)	0.1 ($\times 10^{-2}$)	0.2 ($\times 10^{-2}$)	0.5 ($\times 10^{-2}$)	0.7 ($\times 10^{-2}$)
LR Standard	0.15 (0.000)	0.35 (0.003)	0.55 (0.003)	1.10 (0.005)	1.14 (0.003)
LR with Sanitization	-0.17 (0.000)	0.04 (0.009)	0.29 (0.011)	1.04 (0.012)	1.62 (0.014)
Robust LR	8.06 (0.000)	7.94 (0.010)	7.93 (0.009)	7.91 (0.008)	7.89 (0.006)
LR-MA-Emp	0.22 (0.000)	0.23 (0.003)	0.25 (0.003)	0.27 (0.005)	0.28 (0.003)

retains a low and consistent level of multi-accuracy guarantee. Furthermore, post-processing the Linear Regression classifier with Empirical MABoost also results in similar accuracy to the original model and multi-accuracy regardless of the level of noise. A similar phenomenon is observed in the neural network (NN) and k-nearest neighbors models where MAEMP and CLF-PP are robust to label noise. Moreover, we observe that reducing MA-err comes at no cost to the overall accuracy: the accuracies of the post-processed predictors are comparable to the original predictors. In Figure 8, we observe a similar phenomenon but in the white male group as well as both black male and female groups.

For the ACS employment dataset, there is a significantly larger positive class ratio (Table 1) but similar subgroup ratios since this dataset comes from the same state. In Figure 9, we see a similar pattern to the income data set in terms of the multi-accuracy boosting algorithm producing predictors that are robust to changes in disjoint groups. Here we see that the accuracy of the post proceed predictor can be higher than before pre-processing (e.g., Logistic Regression - black female and black male subgroups). We see similar patterns in the predictor robustness when the white female group is modified in the ACSEmployment dataset.

For the ACS Coverage dataset, the predictors are trained to predict the probability that an individual is covered by public health insurance. Figure 11 and 12 illustrate the results for label shift from 0 to 1 in the white male and white female subgroups respectively. For shifts in both these subgroups, the impact on other groups is only significant for the logistic regression classifier. For logistic regression, we observe the base models suffering from more biased predictions (MA-err) on unmodified groups as the level of noise increases while the post-processed logistic regression predictor retains a consistently low MA-err for all unmodified subgroups.

Boosting Classifiers and Multi-Accuracy Error Across all datasets (e.g., Income, Coverage, and Employment), we observe that the boosting classifiers are relatively calibrated already. For Decision Trees, in particular, the base and post-processed predictors are the same. This is due to the fact that we did not limit the depth of the model, the tree branches on subgroups since subgroup labels are a part of the features. A similar phenomenon exists for XGBoost where there is some randomization that allows some post-processing improvements but the base classifier is already multi-group robust. This is not surprising since for both Decision Trees and XGBoost, the ability to branch exactly based on group membership would reduce the effect of noise and corruption of unrelated subgroups.

Table 3: ACS Income Accuracy - (Signed Group Mean Error) with increasing noise from label shift. Modified Group: White Men - Target/Effect Group: White Women (standard error i.e. mean over 10 runs)

Algorithm	0.0	0.1	0.2	0.5	0.7
LR Standard	0.73 (0.000)	0.73 (0.004)	0.76 (0.006)	0.78 (0.002)	0.77 (0.002)
LR with Sanitization	0.71 (0.000)	0.71 (0.004)	0.73 (0.003)	0.77 (0.001)	0.78 (0.001)
Robust LR	0.76 (0.000)	0.75 (0.002)	0.77 (0.003)	0.79 (0.000)	0.75 (0.001)
LR-MA-Emp	0.73 (0.000)	0.73 (0.003)	0.76 (0.002)	0.78 (0.002)	0.76 (0.010)
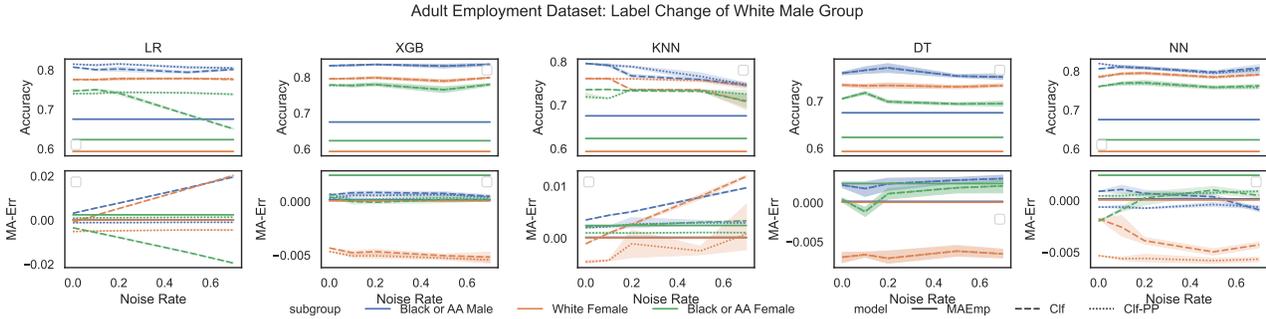
Figure 9: Comparison of different predictors on corrupted data in the white male group with label change at various ratios on the ACS Employment.
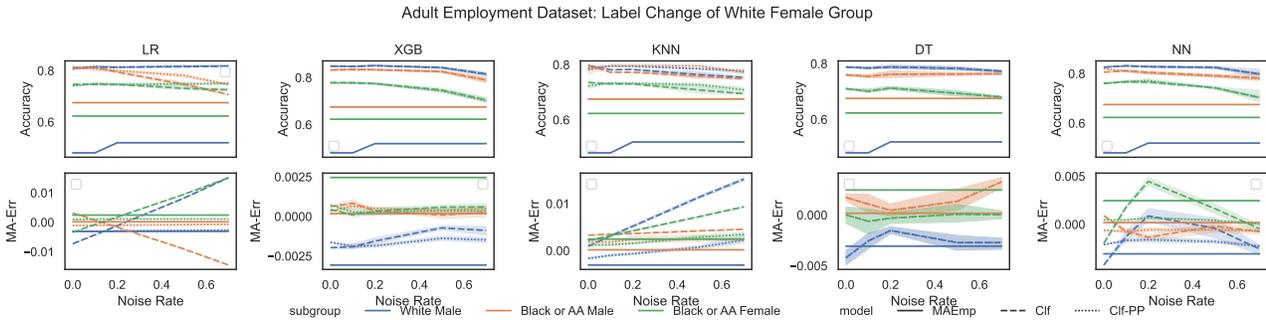
Figure 10: Comparison of different predictors on corrupted data in the white female group with label change at various ratios on the ACS Employment dataset.

D.3. Addition/Deletion

We also demonstrate the robustness of Algorithm 1 through corruptions to data in the form of additional data points. We employ a similar strategy as prior work designing subpopulation attacks (Jagielski et al., 2021) with the additional constraint that only data points outside of the target group can be added to create the poisoning dataset. To find data points to add that would affect the target group, we first cluster data points in a held-out set (not used for training or testing) using K-means. For each cluster where the target subgroup appears, we shift the labels of the group we are allowed to modify. We then add these shifted points from the held-out group to the training data. The amount of noise in this attack is scaled by how many times the identified data points are replicated before being added to the poisoned dataset. Algorithm 3 describes the algorithm we use to generate a corrupted dataset D' of size m. For these data poisoning attacks, we also target 0 labels in each cluster to shift. This attack is more precise than random shifts in subgroups since there is a specific targeted subgroup.

Robust Logistic Regression and Data Addition Table 5 summarizes the overall accuracy at different levels of label shift. Similar to label shift, the robust LR algorithm achieves higher accuracy than the other baselines. However, in Table 4, the multiaccuracy of robust logistic regression is an order of magnitude larger than the baselines and our empirical multiaccuracy algorithm. These patterns both in data addition and label shift illustrate that robust logistic regression might introduce

Adult Coverage Dataset: Label Change of White Female Group
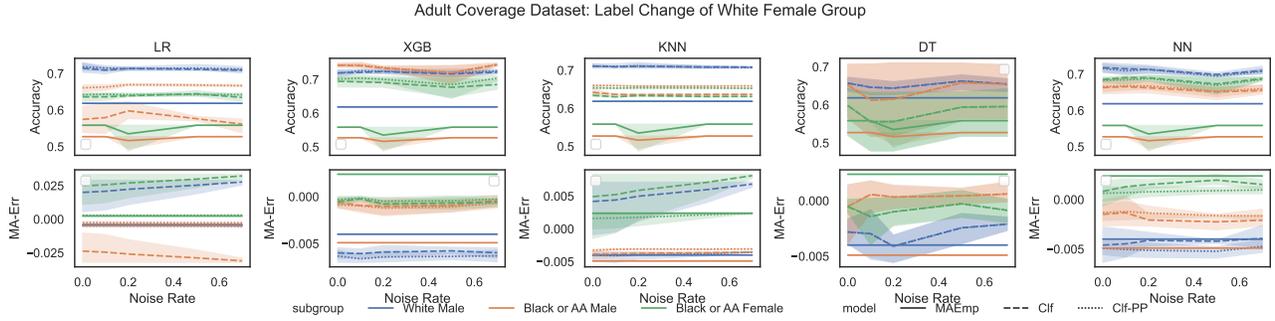
Figure 11: Comparison of different predictors on corrupted data in the white male group with label change at various ratios on the ACS Coverage dataset.

Adult Coverage Dataset: Label Change of White Female Group
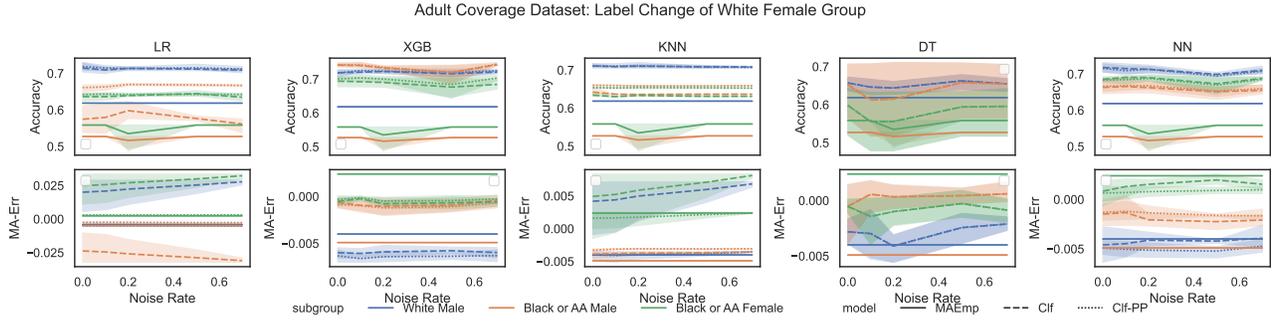
Figure 12: Comparison of different predictors on corrupted data in the white female group with label change at various ratios on the ACS Coverage dataset.

Algorithm 3 Data Addition Algorithm

Parameters: modify group and target group: $C_{mod}, C_{tgt} \subseteq \{0,1\}^X$, noise factor $\alpha \in [1,10]$, num clusters k, cluster threshold γ, target: $t \in \{0,1,*\}$
Input: clean dataset $D = (x_1, y_1), ..., (x_n, y_n) \in X \times \{0,1\}$, held out dataset D_{aux}
Output: corrupted dataset $D' = (x_1, y'_1), ..., (x_m, y'_m) \in X \times \{0,1\}$
$D' = \{\}$
centers = CLUSTER(D_{aux}, k)
for all $c \in$ centers **do**
    $D_c = \{(x,y)|\text{CLOSESTCENTER}((x,y)) = c, (x,y) \in D_{aux}\}$
    $D_{tgt} = \{(x,y)|(x,y) \in C_{tgt}, (x,y) \in D_c\}$
    $D_{mod} = \{(x,y)|(x,y) \in C_{mod}, (x,y) \in D_c\}$
    if $|D_{tgt}| \geq \gamma$ **then**
        for all $(x,y) \in D_{mod}$ **do**
            if $y == t$ **then**
                $y' = 1 - y$
                $D' = D' \cup (x, y')^\alpha$
            end if
        end for
    end if
end for
Return: $D' = D' \cup D$

---

undesirable group mean bias for unaffected groups even though overall performance is not affected.

Table 4: ACS Income MultiAccuracy with increasing noise from data addition. Modified Group: White Men - Target/Effect Group: White Women (standard error i.e. mean over 10 runs)

Algorithm	$\mathbf{0}$ $(\times 10^{-2})$	$\mathbf{1}$ $(\times 10^{-2})$	$\mathbf{2}$ $(\times 10^{-2})$	$\mathbf{4}$ $(\times 10^{-2})$	$\mathbf{8}$ $(\times 10^{-2})$
LR Standard	0.15 (0.000)	0.41 (0.002)	0.61 (0.003)	0.87 (0.004)	1.19 (0.005)
LR with Sanitization	-0.17 (0.000)	0.25 (0.003)	0.71 (0.003)	1.65 (0.003)	1.97 (0.003)
Robust LR	8.06 (0.000)	7.95 (0.010)	7.94 (0.009)	7.91 (0.008)	7.88 (0.007)
MA-Emp	0.22 (0.000)	0.22 (0.001)	0.22 (0.001)	0.22 (0.002)	0.22 (0.003)

Table 5: ACS Income Accuracy with increasing noise from data addition. Modified Group: White Men - Target/Effect Group: White Women (standard error i.e. mean over 10 runs)

Algorithm	$\mathbf{0}$	$\mathbf{1}$	$\mathbf{2}$	$\mathbf{4}$	$\mathbf{8}$
LR Standard	0.73 (0.000)	0.75 (0.005)	0.78 (0.001)	0.78 (0.001)	0.78 (0.001)
LR with Sanitization	0.71 (0.000)	0.73 (0.001)	0.76 (0.008)	0.78 (0.001)	0.78 (0.001)
Robust LR	0.76 (0.000)	0.75 (0.002)	0.77 (0.003)	0.79 (0.004)	0.75 (0.001)
MA-Emp	0.73 (0.000)	0.74 (0.006)	0.76 (0.005)	0.78 (0.001)	0.77 (0.001)

In Figure 13 (Figure 3 in the main text), we see that even at low levels of noise (i.e., poisoned points are replicated once or twice), the Logistic Regression, k-Nearest Neighbors, and Neural Network base classifiers (CLF) exhibit worsened multi-accuracy error. However, with post-processing (CLF-PP), the mult-accuracy error remains consistent around 0 regardless of the noise ratio. Moreover, the benefits of low multi-accuracy error come without cost to the accuracy. These effects are not just on the target group, white female, but also appear in other subgroups such as the black male and female subgroups. Figure 14 shows the effects when the target group is the black male group where a similar effect exists.

D.4. Law School Dataset

Since the positive class ratio is high for the law school dataset, we use label shift labels from 1 to 0 in the modified group. Figure 19 shows the impact of MA post-processing of different models on the law school dataset. Label shift and data addition severely impact the logistic regression model model but our post-processing manages to maintain the multi-accuracy for unaffected groups. It is interesting to note that for KNN, the post-processed model also achieves better accuracy with lower MA. Figure 20 shows a similar phenomenon for logistic regression. In addition, the neural network models area also significantly impacted by noise. Our empirical MA algorithms can maintain multiaccuracy with increasing noise.

D.5. Bank Dataset

We also run our experiments on the Bank Data. Since the positive class is only 11% of the dataset, we also flip the labels from 0 to 1 in our label shift attack. Figure 21 shows that shifting the labels in the group of cell phone users aged 18-29 affects the other groups of cell phone users and 18-29 telephone users. Figure 22 shows that for the data addition attacks, a similar pattern occurs. Our post-processed predictors consistently maintain multiaccuracy while not negatively affecting the overall accuracy.
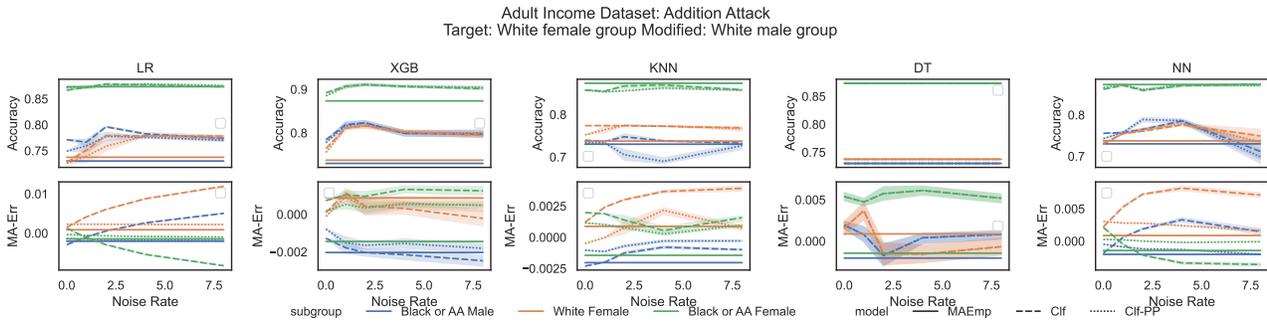
Figure 13: Comparison of different predictors on a corrupted dataset with injected data (Algorithm 3) at various levels on the ACSIncome Dataset. Baseline classifiers such as Logistic Regression, k-nearest Neighbors, and Neural Networks (MLP) become more biased (i.e., increasing MA-err) for the target subgroup (C_{tgt}: White female) even though only data from the white male group (C_{mod}) has been added. However, both the uniformly initialized and post-processed predictor using Algorithm 1 are robust to label change in other groups while preserving comparable accuracy.
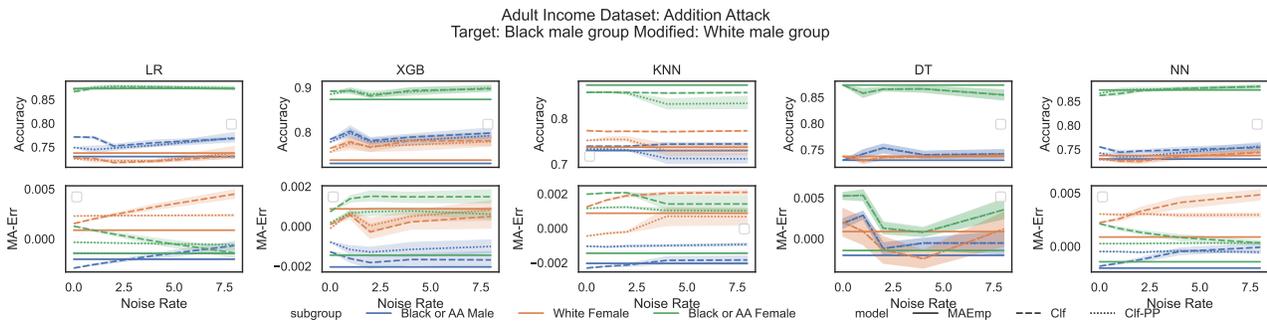
Figure 14: Comparison of different predictors on a corrupted dataset with injected data from the white male group (C_{mod}) with the black male group as the target group (C_{tgt}) at various levels on the ACSIncome Dataset.

Adult Employment Dataset: Addition Attack
Target: White female group Modified: White male group
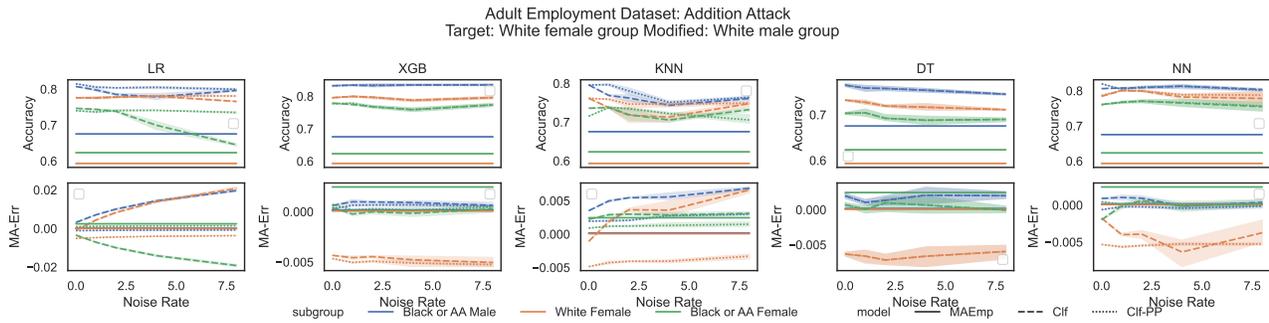
Figure 15: Comparison of different predictors on a corrupted dataset with injected data from the white male group (C_{mod}) with the white female group as the target group (C_{tgt}) at various levels on the ACSEmployment Dataset.

Adult Employment Dataset: Addition Attack
Target: Black male group Modified: White male group
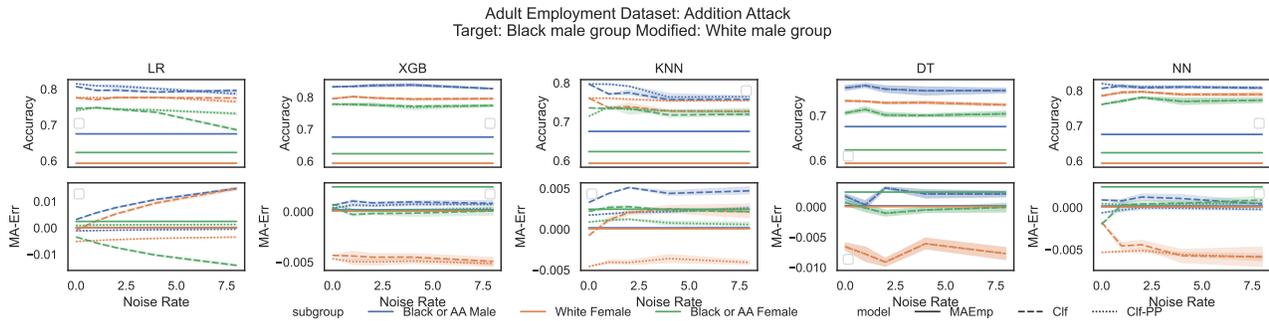
Figure 16: Comparison of different predictors on a corrupted dataset with injected data from the white male group (C_{mod}) with the black male group as the target group (C_{tgt}) at various levels on the ACSEmployment Dataset.
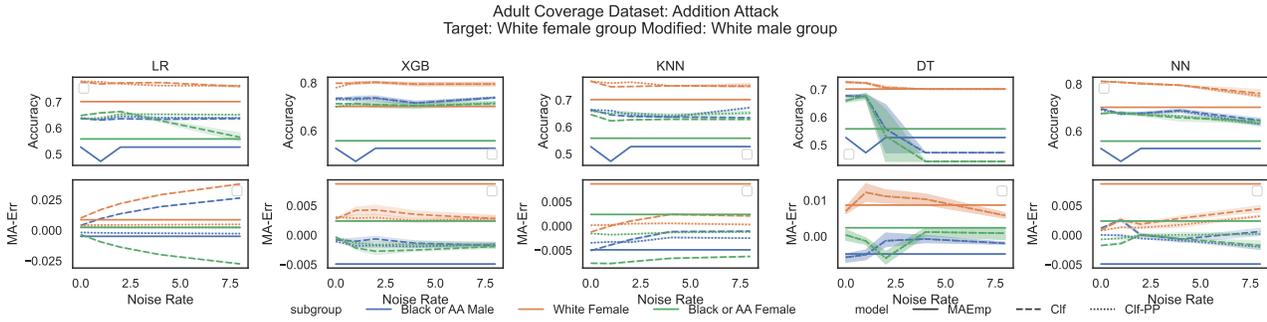
Figure 17: Comparison of different predictors on a corrupted dataset with injected data from the white male group (C_{mod}) with the white female group as the target group (C_{tgt}) at various levels on the ACS Coverage Dataset.
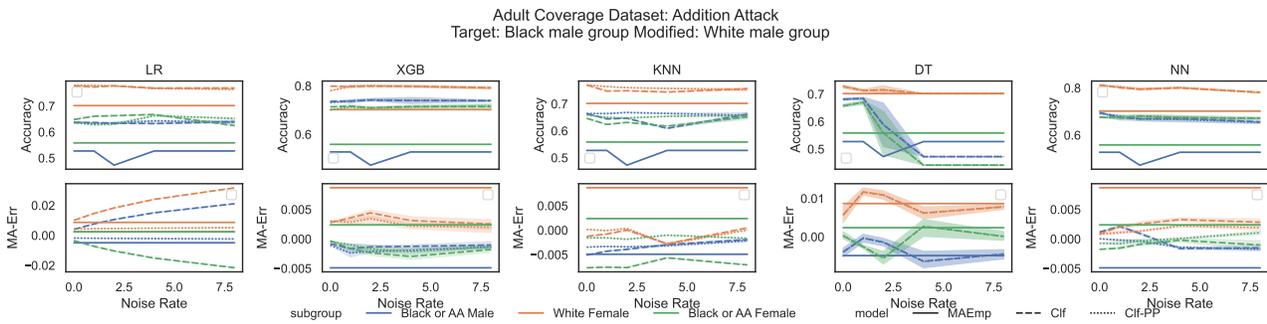
Figure 18: Comparison of different predictors on a corrupted dataset with injected data from the white male group (C_{mod}) with the black male group as the target group (C_{tgt}) at various levels on the ACS Coverage Dataset.
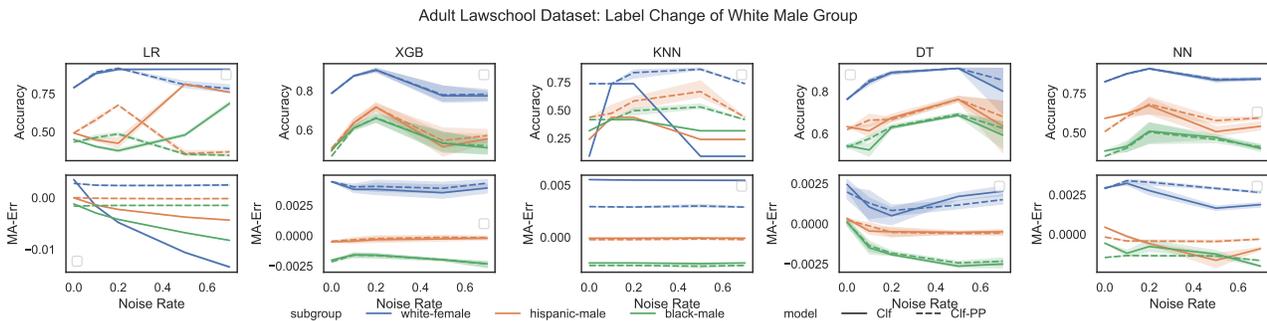
Figure 19: Comparison of different predictors on a corrupted dataset with shifted labels in the white male group (C_{mod}) at various levels of noise the law school dataset.
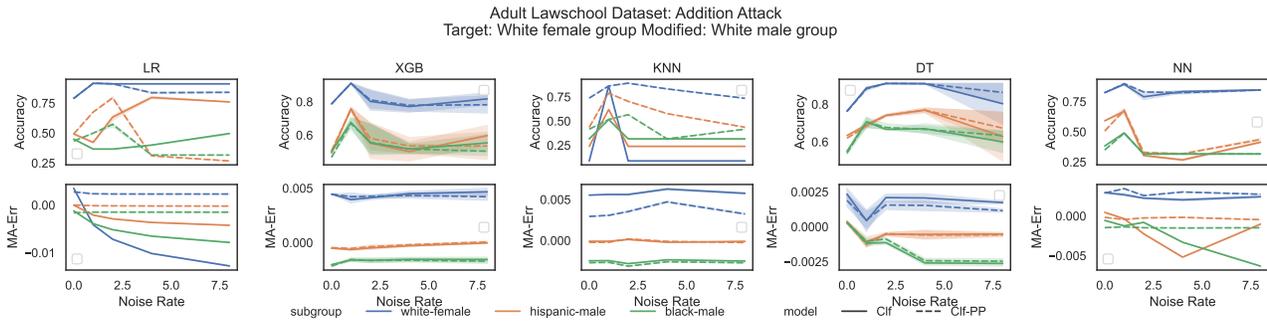
Figure 20: Comparison of different predictors on a corrupted dataset with injected data from the white male group (C_{mod}) with the white female group as the target group (C_{tgt}) at various levels of noise on the Law School Dataset.
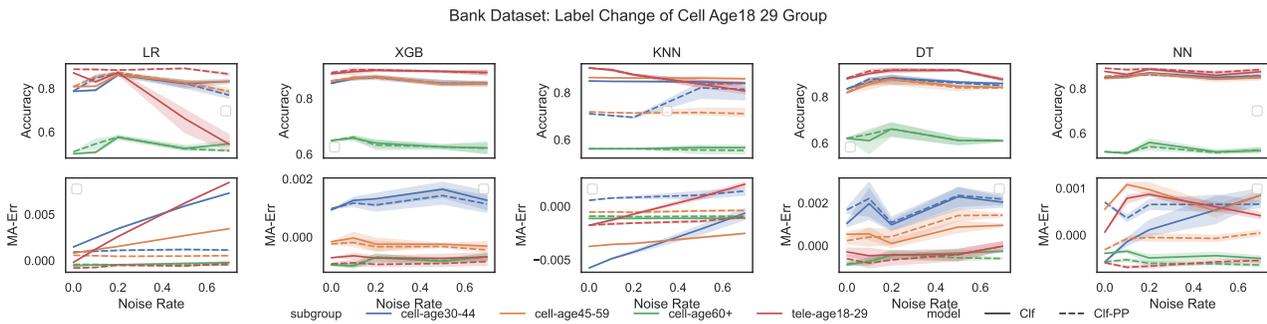
Figure 21: Comparison of different predictors on a corrupted dataset with shifted labels in the aged 18-29 cell phone users group (C_{mod}) at various levels of noise in the Bank Dataset.
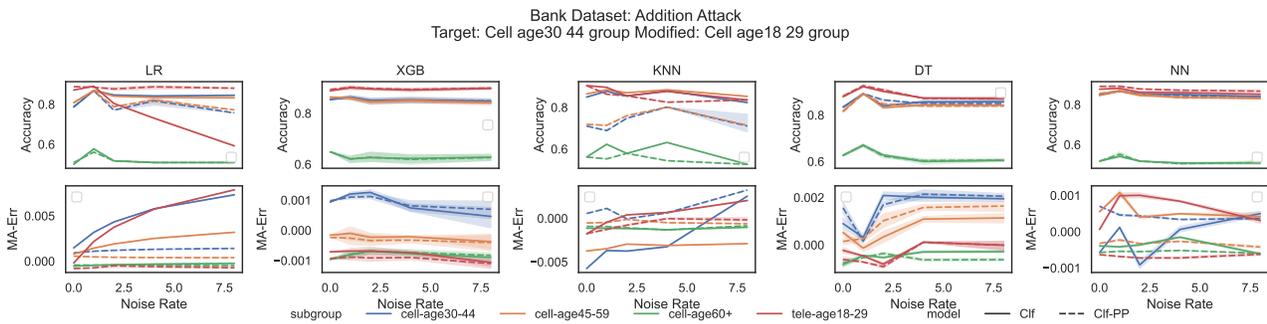
Figure 22: Comparison of different predictors on a corrupted dataset with injected data from the aged 18-29 cell phone users group (C_{mod}) with the aged 30-44 cell phone users group as the target group (C_{tgt}) at various levels of noise on the Bank Dataset.