Sample Selection for Fair and Robust Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Fairness and robustness are critical elements of Trustworthy AI that need to be addressed together. Fairness is about learning an unbiased model while robustness is about learning from corrupted data, and it is known that addressing only one of them may have an adverse affect on the other. In this work, we propose a sample selection-based algorithm for fair and robust training. To this end, we formulate a combinatorial optimization problem for the unbiased selection of samples in the presence of data corruption. Observing that solving this optimization problem is strongly NP-hard, we propose a greedy algorithm that is efficient and effective in practice. Experiments show that our method obtains fairness and robustness that are better than or comparable to the state-of-the-art technique, both on synthetic and benchmark real datasets. Moreover, unlike other fair and robust training baselines, our algorithm can be used by only modifying the sampling step in batch selection without changing the training algorithm or leveraging additional clean data.

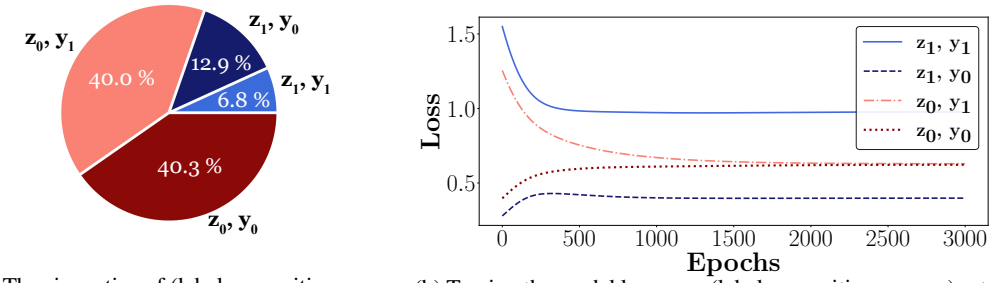
1 Introduction

Trustworthy AI is becoming essential for modern machine learning. While a traditional focus of machine learning is to train the most accurate model possible, companies that actually deploy AI including Microsoft [2021], Google [2020], and IBM [2020] are now pledging to make their AI systems fair, robust, interpretable, and transparent as well. Among the key objectives, we focus on fairness and robustness because they are closely related and need to be addressed together. Fairness aims to learn an unbiased model while robustness seeks to learn a resilient model even on noisy data, and both issues primarily root from the same training data. A recent work [Roh et al., 2020] shows that improving fairness while ignoring noisy data may lead to a worse accuracy-fairness tradeoff. Likewise, focusing on robustness only may have an adverse affect on fairness as we explain below.

There are several approaches for supporting fairness and robustness together. One is an in-processing approach where the model architecture itself is modified to optimize the two objectives. In particular, the state-of-the-art approach called FR-Train [Roh et al., 2020] uses adversarial learning to train a classifier and two discriminators for fairness and robustness. However, such an approach requires the application developer to use a specific model architecture and is thus restrictive. Another approach is to preprocess the data [Kamiran and Calders, 2011] and remove any bias and noise before model training, but this requires modifying the data itself. In addition, we are not aware of a general data cleaning method that is designed to explicitly improve model fairness and robustness together.

Instead, we propose an adaptive sample selection approach for the purpose of improving fairness and robustness. This approach can easily be deployed as a batch selection method, which does not require modifying the model or data. Our techniques build on top of two recent lines of sample selection research: clean selection for robust training [Rousseeuw, 1984, Song et al., 2019, Shen and Sanghavi, 2019] and batch selection for fairness [Roh et al., 2021]. Clean selection is a standard approach for discarding samples with high-loss values. More recently, batch selection for fairness has been

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.



(a) The size ratios of (label y, sensitive group z) sets within the COMPAS dataset.

(b) Tracing the model losses on (label y, sensitive group z) sets when training logistic regression on the COMPAS dataset.

Figure 1: The ProPublica COMPAS dataset [Angwin et al., 2016] has bias where the set $(\mathrm{y} = 1, \mathrm{z} = 1)$ has the smallest size in Figure 1a. However, when training a logistic regression model on this dataset, the same set has the highest loss at every epoch as shown in Figure 1b. Blindly applying clean selection at any point will thus discard $(\mathrm{y} = 1, \mathrm{z} = 1)$ samples the most, which results is even worse bias and possibly worse fairness.

proposed where the idea is to adaptively adjust sampling ratios among sensitive groups (e.g., black or white populations) so that the model training is not discriminative.

We note that state-of-the-art clean selection techniques like Iterative Trimmed Loss Minimization (ITLM) [Shen and Sanghavi, 2019] are not designed to address fairness and may actually worsen the data bias when used alone. For example, Figure 1 shows how the ProPublica COMPAS dataset [Angwin et al., 2016] can be divided into four sets where the label y is either 0 or 1, and the sensitive attribute z is either 0 or 1. The sizes of the sets are biased where $(\mathrm{y} = 1, \mathrm{z} = 1)$ is the smallest. However, the model loss on $(\mathrm{y} = 1, \mathrm{z} = 1)$ is the highest among the sets as shown in Figure 1b. Hence, blindly applying clean selection will discard $(\mathrm{y} = 1, \mathrm{z} = 1)$ samples the most, resulting in a worse bias that may negatively affect fairness as we demonstrate in Sec. 2.

We thus formulate a combinatorial optimization problem for fair and robust sample selection that achieves unbiased sampling in the presence of data corruption. To avoid discrimination on a certain group, we adaptively adjust the maximum number of samples per (y, z) set using the recently proposed FairBatch system [Roh et al., 2021], which solves a bilevel optimization problem of unfairness mitigation and standard empirical risk minimization through batch selection. We show that our optimization problem for sample selection can be viewed as a multidimensional knapsack problem and is thus strongly NP-hard [Garey and Johnson, 1979]. We then propose an efficient greedy algorithm that is effective in practice. Our method supports prominent group fairness measures: equalized odds [Hardt et al., 2016] and demographic parity [Feldman et al., 2015].

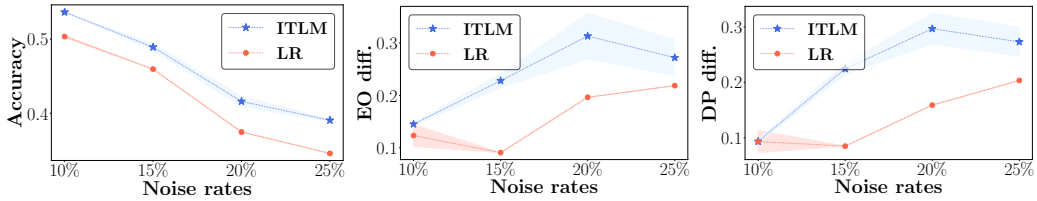
Experiments on synthetic and benchmark real datasets (COMPAS [Angwin et al., 2016] and Adult-Census [Kohavi, 1996]) show that our method obtains accuracy and fairness results that are better than baselines using fairness algorithms [Zafar et al., 2017a,b, Roh et al., 2021] and better than or on par with FR-Train [Roh et al., 2020].

Notation Let θ be the model weights, $\mathrm{x} \in \mathbb{X}$ be the input feature to the classifier, $\mathrm{y} \in \mathbb{Y}$ be the true class, and $\hat{\mathrm{y}} \in \mathbb{Y}$ be the predicted class where $\hat{\mathrm{y}}$ is a function of (x, θ). Let $\mathrm{z} \in \mathbb{Z}$ be a sensitive attribute, e.g., race or gender. Let $\mathrm{d} = (\mathrm{x}, \mathrm{y})$ be a training sample that contains a feature and label. Let S as the selected subset from the entire dataset D, where the cardinality of D is n. In the model training, we use a loss function $\ell_\theta(\cdot)$ that returns a loss value within the range $[0, 1]$ where a smaller value indicates a more accurate prediction.

2 Unfairness in Clean Selection

We demonstrate how clean selection that only focuses on robust training may lead to unfair models. Throughout this paper, we use ITLM [Shen and Sanghavi, 2019] as a representative clean selection method for its simplicity, although other methods can be used as well. We first introduce ITLM and demonstrate how ITLM can actually worsen fairness.

ITLM Cleaning samples based on their loss values is a traditional robust training technique [Rousseeuw, 1984] that is widely used due to its efficiency. Recently, ITLM was proposed



(a) Accuracy. (b) Equalized odds disparity. (c) Demographic parity disparity.

Figure 2: Performances of logistic regression with and without ITLM (ITLM and LR, respectively) on the ProPublica COMPAS dataset for different noise rates injected with label flipping [Paudice et al., 2018]. Although ITLM improves accuracy (higher is better), the fairness worsens according to the equalized odds and demographic parity disparity measures (lower is better).

as a scalable approach for sample selection where lower loss samples are considered cleaner. The following optimization problem is solved with theoretical guarantees of convergence:

$$\min_{S:|S|=\lfloor \tau n \rfloor} \sum_{s_i \in S} \ell_\theta(s_i)$$

where S is the a selected set, s_i is a sample in S, and τ is the ratio of clean samples in the entire data.

Potential Unfairness We show how applying ITLM may lead to unfairness through a simple experiment. We use the benchmark ProPublica COMPAS dataset [Angwin et al., 2016] where the label y indicates recidivism, and the sensitive attribute z is set to `gender`. We inject noise into the training set using label flipping techniques [Paudice et al., 2018]. For simplicity, we assume that ITLM knows the clean ratio τ of the data and selects $\lfloor \tau n \rfloor$ samples. We train a logistic regression model on the dataset.

We show the accuracy and fairness performances of logistic regression with and without ITLM for different noise ratios in Figure 2. As expected, ITLM increasingly improves the model accuracy for larger noise rates as shown in Figure 2a. At the same time, however, the fairness worsens according to the two measures we use in this paper (see definitions in Sec. 5) – equalized odds [Hardt et al., 2016] disparity and demographic parity [Feldman et al., 2015] disparity – as shown in Figures 2b and 2c, respectively. The results confirm our claims in Sec. 1.

3 Framework

We provide an optimization framework for training models on top of clean selection with fairness constraints. The entire optimization involves three joint optimizations, which we explain in a piecewise fashion: (1) how to perform clean selection with fairness constraints, (2) how to train a model on top of the selected samples, and (3) how to generate the fairness constraints. In Sec. 4 we present a concrete algorithm for solving all the optimizations.

3.1 Clean Selection with Fairness Constraints

The first optimization is to select clean samples such that the loss is minimized while fairness constraints are satisfied. For now, we assume that the fairness constraints are given where the selected samples of a certain set ($\mathrm{y} = y, \mathrm{z} = z$) cannot exceed a portion $\lambda_{(y,z)}$ of the selected samples of the class $\mathrm{y} = y$. Notice that $\sum_z \lambda_{(y,z)} = 1, \forall y \in \mathbb{Y}$. We explain how the λ values are determined in Sec. 3.3. We thus solve the following optimization for sample selection:

$$\min \sum_{i=1}^{n} \ell_\theta(d_i)\, p_i \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{n} p_i \leq \tau n, \;\; \sum_{i=1}^{n} \mathbb{I}_{D_{(y,z)}}(d_i)\, p_i \leq \lambda_{(y,z)} |S_y|, \tag{2}$$
$$y \in \mathbb{Y}, z \in \mathbb{Z}, \; p_i \in \{0,1\}, \; i = 1, ..., n$$

where p_i indicates whether the data sample d_i is selected or not, $D_{(y,z)}$ is a subset for the (y, z) class, $\mathbb{I}_{D_{(y,z)}}(d_i)$ is an indicator function that returns 1 if $d_i \in D_{(y,z)}$ and 0 otherwise, and S_y is the selected samples for $\mathrm{y} = y$. In Eq. 2, the first constraint comes from ITLM while the second constraint is for the fairness.

An important detail is that we use inequalities in all of our constraints, unlike ITLM. The reason is that we may not be able to find a feasible solution of selected samples if we only use equality constraints. For example, if $\lambda_{(1,0)} = \lambda_{(1,1)} = 0.5$, but we have zero $(\mathrm{y} = 1, \mathrm{z} = 0)$ samples, then the fairness equality constraint for $(\mathrm{y} = 1, \mathrm{z} = 0)$ can never be satisfied.

3.2 Model Training

The next optimization is to train a model on top of the selected samples from Sec. 3.1. Recall that the sample selection optimization only uses inequality constraints, which means that for some set $(\mathrm{y} = y, \mathrm{z} = z)$, its size $|S_{(y,z)}|$ may be smaller than $\lambda_{(y,z)}|S_y|$. However, we would like to train the model as if there were $\lambda_{(y,z)}|S_y|$ samples for the intended fairness performance. We thus reweight each set $(\mathrm{y} = y, \mathrm{z} = z)$ by $\frac{\lambda_{(y,z)}|S_y|}{|S_{(y,z)}|}$ and perform weighted empirical risk minimization (ERM) for model training. We then solve the bilevel optimization of model training and clean selection in Sec. 3.1 as follows:

$$\min_{\theta} \sum_{s_i \in S_{\boldsymbol{\lambda}}} \sum_{(y,z)\in(\mathbb{Y},\mathbb{Z})} \mathbb{I}_{S_{(y,z)}}(s_i) \; \frac{\lambda_{(y,z)}|S_y|}{|S_{(y,z)}|} \ell_\theta(s_i) \tag{3}$$

$$S_{\boldsymbol{\lambda}} = \underset{S=\{d_i|p_i=1\}:\text{Eq. 2}}{\arg\min} \text{Eq. 1} \tag{4}$$

where $\boldsymbol{\lambda}$ is the set of $\lambda_{(y,z)}$ for all $z \in \mathbb{Z},\ y \in \mathbb{Y}$.

3.3 Fairness Constraint Generation

To determine the $\lambda_{y,z}$ values, we adaptively adjust sampling ratios of sensitive groups using the techniques in FairBatch [Roh et al., 2021]. For each batch selection, we compute the fairness of the intermediate model and decide which sensitive groups should be sampled more (or less) in the next batch. The sampling rates become the $\lambda_{(y,z)}$ values used in the fairness constraints of Eq. 2.

FairBatch supports all the group fairness measures used in this paper. As an illustration, we state FairBatch's optimization when using equalized odds disparity (see the supplementary for the optimization using demographic parity disparity):

$$\min_{\boldsymbol{\lambda}} \max\{|L_{(y,z')} - L_{(y,z)}|\},\ z \neq z',\ z, z' \in \mathbb{Z},\ y \in \mathbb{Y} \tag{5}$$

where $L_{(y,z)} = 1/|S_{\boldsymbol{\lambda}(y,z)}| \sum_{S_{\boldsymbol{\lambda}(y,z)}} \ell_\theta(s_i)$, and $S_{\boldsymbol{\lambda}(y,z)}$ is a subset of $S_{\boldsymbol{\lambda}}$ from Eq. 4 for the $(\mathrm{y} = y, \mathrm{z} = z)$ set. FairBatch adjusts $\boldsymbol{\lambda}$ in each epoch via a signed gradient-based algorithm to the direction of reducing unfairness. For example, if the specific sensitive group z is less accurate on the $\mathrm{y} = y$ samples in the epoch t (i.e., z is discriminated in terms of equalized odds), FairBatch selects more data from the $(\mathrm{y} = y, \mathrm{z} = z)$ set to increase the model's accuracy on that set:

$$\lambda_{(y,z)}^{(t+1)} = \lambda_{(y,z)}^{(t)} - \alpha \cdot \mathrm{sign}(|L_{(y,z')} - L_{(y,z)}|),\ z \neq z' \tag{6}$$

where α is the step size of λ updates. To integrate FairBatch with our framework, we adaptively adjust $\boldsymbol{\lambda}$ as above for an intermediate model trained on the selected set $S_{\boldsymbol{\lambda}}$ from Eq. 4.

4 Algorithm

We present our algorithm for solving the optimization problem in Sec. 3. We first explain how the clean selection with fairness constraints problem in Sec. 3.1 can be converted to a multi-dimensional knapsack problem, which has known solutions. We then describe the full algorithm that includes the three functionalities: clean selection, model training, and fairness constraint generation.

Algorithm 1: Greedy-Based Clean and Fair Sample Selection

Input: loss $\ell_\theta(\boldsymbol{d})$, fair ratio lambda $\boldsymbol{\lambda}$, clean ratio τ

$profit$ = 1 - $\ell_\theta(\boldsymbol{d})$
$sortIdx$ = $\texttt{Sort}(profit)$
$S \leftarrow$ [ ]

for *idx in sortIdx* **do**
  if d_{idx} does not violate any constraint in Eq. 8 w.r.t. $\boldsymbol{\lambda}$ and τ **then**
    Append d_{idx} to S

Output : S

Algorithm 2: Overall Fair and Robust Model Training

Input: train data (x_{train}, y_{train}), clean ratio τ, loss function $\ell_\theta(\cdot)$

$\boldsymbol{d} \leftarrow (x_{train}, y_{train})$
$\theta \leftarrow$ initial model parameters
$\boldsymbol{\lambda} = \{\lambda_{y,z} | y \in \mathbb{Y}, z \in \mathbb{Z}\} \leftarrow$ random sampling ratios
for *each epoch* **do**
  $S = \texttt{Algorithm1}(\ell_\theta(\boldsymbol{d}), \boldsymbol{\lambda}, \tau_y)$
  Draw minibatches from S w.r.t. $\lambda_{y,z}|S_y|/|S_{(y,z)}|$
  for *each minibatch* **do**
    Update model parameters θ according to the minibatch
  Update $\forall \lambda_{y,z} \in \boldsymbol{\lambda}$ using the update rule as in Eq. 6

Output : model parameters θ

The clean selection problem can be converted to an equivalent multi-dimensional knapsack problem by (1) maximizing the sum of $(1 - \ell_\theta(\cdot))$'s instead of minimizing the sum of $\ell_\theta(\cdot)$'s and (2) rearranging the fairness constraints so that the right-hand side expressions become constants (instead of containing the variable S_y) as follows:

$$\max \sum_{i=1}^{n} (1 - \ell_\theta(d_i))\, p_i \tag{7}$$

$$\text{s.t.} \sum_{i=1}^{n} p_i \leq \tau n, \ \sum_{i=1}^{n} w_i p_i \leq \tau n, \text{where } w_i = \begin{cases} 1 & \text{if } d_i \notin D_y \\ 1 - \lambda_{(y,z)} & \text{if } d_i \in D_y \text{ and } d_i \notin D_z \\ 2 - \lambda_{(y,z)} & \text{if } d_i \in D_{y,z} \end{cases}, \tag{8}$$

$$y \in \mathbb{Y}, z \in \mathbb{Z}, p_i \in \{0, 1\}, i = 1, ..., n\ .$$

The multi-dimensional knapsack problem is known to be strongly NP-hard [Garey and Johnson, 1979]. Although exact algorithms have been proposed, they have exponential time complexities [Kellerer et al., 2004] and are thus impractical. There is a polynomial-time approximation scheme (PTAS) [Caprara et al., 2000], but the actual computation time increases exponentially for more accurate solutions [Kellerer et al., 2004]. Since our method runs within a single model training, we cannot tolerate such runtimes and thus opt for a greedy algorithm that is efficient and known to return reasonable solutions in practice [Akçay et al., 2007]. Algorithm 1 takes a greedy approach by sorting all the samples in descending order by their $(1 - \ell_\theta(\cdot))$ values and then sequentially selecting the samples that do not violate any of the fairness constraints. The computational complexity is thus $\mathcal{O}(n \log n)$ where n is the total number of samples.

We now present our overall model training process for weighted ERM in Algorithm 2. For each epoch, we first select a clean and fair sample set S using Algorithm 1. We then draw minibatches from S according to the (y $= y$, z $= z$)-wise weights described in Sec. 3.2 and update the model parameters θ via stochastic gradient descent. A minibatch selection with uniform sampling can be considered as an unbiased estimator of the ERM, so sampling each (y $= y$, z $= z$) set in proportion to its weight results in an unbiased estimator of the weighted ERM. Finally, at the end of the epoch, we update the $\boldsymbol{\lambda}$ values using the update rules as in Eq. 6.

5 Experiments

We now demonstrate the performances of the proposed algorithm. We use PyTorch and set logistic regression as the classifier for all experiments. The experimental results are drawn on separate clean test sets. All experiments are repeated with 5 different random seeds. We utilize Intel Xeon Silver 4210R CPUs and NVIDIA Quadro RTX 8000 GPUs. More detailed settings are in the supplementary.

Fairness Measures We focus on two representative group fairness measures: (i) equalized odds (EO) [Hardt et al., 2016], which aims for the same accuracy between sensitive groups conditioned on the true labels and (ii) demographic parity (DP) [Feldman et al., 2015], which is satisfied when the

sensitive groups have the same positive prediction ratio. We evaluate the fairness disparities over sensitive groups as follows: *EO disparity* = $\max_{z\in\mathbb{Z},y\in\mathbb{Y}} |\Pr(\hat{\mathrm{y}}=1|\mathrm{z}=z,\mathrm{y}=y)-\Pr(\hat{\mathrm{y}}=1|\mathrm{y}=y)|$ and *DP disparity* = $\max_{z\in\mathbb{Z}} |\Pr(\hat{\mathrm{y}}=1|\mathrm{z}=z)-\Pr(\hat{\mathrm{y}}=1)|$. Note that the classifier is perfectly fair when the fairness disparity is equal to zero.

Datasets We use total three datasets: one synthetic dataset and two real benchmark datasets.

We generate a synthetic dataset using a similar method to Zafar et al. [2017a]. The synthetic dataset has 3,000 samples and consists of two non-sensitive features $(\mathrm{x}_1,\mathrm{x}_2)$, one sensitive feature z, and one label class y. Each sample $(\mathrm{x}_1,\mathrm{x}_2,\mathrm{y})$ is drawn from the following Gaussian distributions: $(\mathrm{x}_1,\mathrm{x}_2)|\mathrm{y}=1\sim\mathcal{N}([1;1],[5,1;1,5])$ and $(\mathrm{x}_1,\mathrm{x}_2)|\mathrm{y}=0\sim\mathcal{N}([-1;-1],[10,1;1,3])$. We add the sensitive feature z to have a biased distribution: $\Pr(\mathrm{z}=1)=7\Pr((\mathrm{x}_1',\mathrm{x}_2')|\mathrm{y}=1)/[7\Pr((\mathrm{x}_1',\mathrm{x}_2')|\mathrm{y}=1)+\Pr((\mathrm{x}_1',\mathrm{x}_2')|\mathrm{y}=0)]$ where $(\mathrm{x}_1',\mathrm{x}_2')=(\mathrm{x}_1\cos(\pi/5)-\mathrm{x}_2\sin(\pi/5),\mathrm{x}_1\sin(\pi/5)+\mathrm{x}_2\cos(\pi/5))$. We visualize the synthetic dataset in the supplementary.

We utilize two real datasets, ProPublica COMPAS [Angwin et al., 2016] and AdultCensus [Kohavi, 1996], and use the same pre-processing in IBM Fairness 360 [Bellamy et al., 2019]. COMPAS and AdultCensus consist of 5,278 and 43,131 samples, respectively. The labels in COMPAS predict whether each criminal recidivates, and the labels in AdultCensus indicate whether each customer's annual income exceeds 50k dollars. We use `gender` as a sensitive attribute.

Noise Injection We use two types of label noise: (1) a general label noise [Paudice et al., 2018] that flips labels so as to maximize the accuracy degradation (General Label Flipping), and (2) a group-targeted label noise [Roh et al., 2020] that has the same attack goal (i.e., maximizing accuracy degradation) as the first method, but only flips labels of the specific sensitive group (Group-Targeted Label Flipping).

Baselines We compare our proposed algorithm with four types of baselines: (1) *vanilla* training by logistic regression; (2) *robust only* training by ITLM [Shen and Sanghavi, 2019]; (3) *fair only* training by FairBatch [Roh et al., 2021]; and (4) *fair and robust* training by a two-step integration of ITLM and FairBatch (ITLM→FB), a two-step integration of ITLM and the fairness constraints methods that adding penalty term in the loss function [Zafar et al., 2017a,b] (ITLM→Penalty), and FR-Train [Roh et al., 2020] that is a state-of-the-art fair and robust algorithm.

We describe more details of the fair and robust training baselines. ITLM→FB is a two-step integration of robustness and fairness, which first runs ITLM on the entire dataset and then trains FairBatch only with the final selected dataset from ITLM. ITLM→Penalty is a two-step integration with the fairness constraints method [Zafar et al., 2017a,b] that adds a fairness penalty term to the loss function for reducing the covariance between sensitive attributes and the predicted labels. We thus first runs ITLM on the entire dataset and then trains fairness penalty algorithm only with the final selected dataset from ITLM. We also considered another type of baseline that first considers fairness in the sample selection stage and then applies the fair algorithm to these data; however, we leave these baselines in the supplementary due to similar performances with the two-step baselines mentioned earlier. Finally, we utilize FR-Train [Roh et al., 2020] that has been recently proposed for fair and robust training, which is based on adversarial training between one classifier and fair and robust discriminators. To achieve robustness, FR-Train requires a small (say 10%) yet clean validation set.

Hyperparameters We choose the step size for updating λ (i.e., α in Eq. 6) within a candidate range [0.xxx, 0.xxx] using cross-validation. We assume the clean ratio τ is known. If the clean ratio is unknown, one can use cross-validation [Shen and Sanghavi, 2019, Liu and Tao, 2015].

5.1 Accuarcy and Fairness

Table 1 shows accuracy and fairness (w.r.t. equalized odds and demographic parity) performances of baselines and our algorithm on the synthetic data. We utilize two different types of label noise as described in the experimental setting. The results show that our algorithm achieves high accuracy and fairness compared to the baselines. Compared to the logistic regression and robust-only algorithm (ITLM), which do not consider fairness, our algorithm has much better fairness (i.e., lower disparity). For the fair-only algorithm (FB), fairness performance improves, but the accuracy severely decreases. This result shows a vulnerability to the noises of fair-only algorithms as known in the previous

Table 1: Performances on the **synthetic** test set w.r.t. **equalized odds (EO)** and **demographic parity (DP)**. We compare our algorithm with four types of baselines: (1) vanilla training: LR; (2) robust training: ITLM [Shen and Sanghavi, 2019]; (3) fair training: FB [Roh et al., 2021]; and (4) fair and robust training: ITLM→FB, ITLM→Penalty [Zafar et al., 2017b], and FR-Train [Roh et al., 2020]. Note that FR-Train utilizes a clean validation set with size as 10% of the training set. We inject label noises for 10% of the training data. Experiments are repeated 5 times.

	General Label Flipping				Group-Targeted Label Flipping			
Method	Acc.	EO Disp.	Acc.	DP Disp.	Acc.	EO Disp.	Acc.	DP Disp.
LR	.665±.003	.557±.015	.665±.003	.400±.010	.600±.002	.405±.008	.600±.002	.300±.006
ITLM	.716±.001	.424±.003	.716±.001	.380±.002	.719±.001	.315±.017	.719±.001	.287±.009
FB	.509±.017	.051±.018	.683±.002	.063±.019	.544±.005	.072±.010	.526±.003	.082±.002
ITLM→FB	.718±.003	.199±.020	.725±.002	.089±.032	.707±.001	.108±.030	.704±.003	.067±.027
ITLM→Penalty	.651±.051	.172±.046	.674±.012	.068±.014	.706±.001	.080±.004	.688±.004	.044±.004
FR-Train	.722±.005	.078±.010	.711±.003	.057±.027	.715±.003	.046±.011	.706±.011	.079±.030
Ours	.725±.001	.028±.002	.720±.001	.006±.001	.725±.001	.034±.002	.720±.001	.039±.007

Table 2: Performances on the **COMPAS** test set w.r.t. **equalized odds (EO)** and **demographic parity (DP)**. Other experimental settings are identical to Table 1.

	General Label Flipping				Group-Targeted Label Flipping			
Method	Acc.	EO Disp.	Acc.	DP Disp.	Acc.	EO Disp.	Acc.	DP Disp.
LR	.503±.002	.123±.021	.503±.002	.093±.020	.513±.000	.668±.000	.513±.000	.648±.000
ITLM	.536±.000	.145±.003	.536±.000	.094±.003	.539±.002	.573±.019	.539±.002	.547±.015
FB	.509±.001	.083±.013	.503±.002	.046±.010	.525±.004	.093±.003	.507±.004	.059±.010
ITLM→FB	.520±.001	.064±.013	.523±.003	.039±.009	.538±.008	.315±.074	.520±.013	.130±.105
ITLM→Penalty	.523±.003	.074±.027	.525±.001	.059±.008	.517±.014	.436±.028	.514±.007	.094±.022
FR-Train	.620±.009	.074±.015	.607±.016	.050±.013	.611±.029	.081±.020	.597±.039	.045±.014
Ours	.521±.000	.044±.000	.544±.000	.025±.000	.545±.009	.084±.012	.521±.000	.024±.000

study [Roh et al., 2020]. As the fair and robust baselines, we compare the two-step approaches and the state-of-the-art fair and robust algorithm called FR-Train. The two-step approaches, which combine ITLM and the state-of-the-art fairness techniques (ITLM→FB, ITLM→Penalty), show insufficient fairness or lower accuracy than our algorithm. These results imply that noise and bias are challenging to independently remove as these factors are not separable in the training data. FR-Train is the latest fair and robust training technique and requires an additional clean validation set to achieve robustness. Since FR-Train is based on adversarial learning, which is known to be possibly unstable, the non-converged training results have not been included as reported in the experimental setting of the FR-Train paper [Roh et al., 2020]. Although our algorithm does not use the additional validation set, it performs on par or better than FR-Train.

Tables 2 and 3 show the results on the COMPAS and AdultCensus datasets. Most of the comparisons with the baselines are similar to those in the synthetic data. In the COMPAS dataset, FR-Train is more accurate than our method, which implies that using a clean validation set is very effective on this dataset. Nonetheless, our method generally shows better fairness compared to FR-Train.

For a more detailed comparison with FR-Train, we now assume that the additional validation set cannot be completely clean, which is a general scenario in real-world applications. Under this scenario, the obtainable validation set is *noisy* data or *relatively clean* data by applying ITLM to noisy data. We compare FR-Train with clean, noisy, and relatively clean validation sets and our algorithm on the COMPAS dataset in Table 4. When the clean validation set is not available, the accuracy of FR-Train is degraded, and our technique shows similar or better accuracy with higher fairness compared to FR-Train.

5.2 Varying the Noise Ratio

We now compare the performances of the algorithms when varying the noise ratio in the data. Figure 3 shows the accuracy and fairness performances of logistic regression (LR), ITLM, FR-Train, and our algorithm when changing the amount of the general label flipping in the synthetic data. Experimental

Table 3: Performances on the **AdultCensus** test set w.r.t. **equalized odds (EO)** and **demographic parity (DP)**. Other experimental settings are identical to Table 1.

	General Label Flipping				Group-Targeted Label Flipping			
Method	Acc.	EO Disp.	Acc.	DP Disp.	Acc.	EO Disp.	Acc.	DP Disp.
LR	.746±.003	.070±.002	.746±.003	.073±.001	.748±.006	.095±.002	.748±.006	.127±.006
ITLM	.785±.018	.087±.031	.785±.018	.092±.016	.785±.011	.144±.023	.785±.011	.105±.006
FB	.748±.002	.022±.002	.758±.004	.046±.007	.739±.014	.086±.037	.693±.002	.015±.005
ITLM→FB	.772±.024	.047±.008	.776±.023	.073±.010	.773±.013	.047±.006	.769±.014	.053±.005
ITLM→Penalty	.776±.023	.082±.015	.774±.024	.054±.018	.755±.026	.161±.018	.757±.003	.013±.003
FR-Train	.765±.009	.059±.021	.771±.012	.093±.006	.722±.027	.069±.016	.773±.012	.049±.010
Ours	.771±.015	.029±.005	.782±.015	.049±.012	.761±.006	.047±.018	.760±.007	.034±.016

Table 4: Detailed comparison with FR-Train on the COMPAS test set with the general label flipping (10%). FR-Train is trained with three different types of validation set: (i) clean data, (ii) noisy data, and (iii) relatively clean data by applying ITLM to noisy data. For rows 1–3, the size of validation set is 10% of the training set.

Method	Acc.	EO Disp.	Acc.	DP Disp.
FR-Train with clean val. set	.620±.009	.074±.015	.607±.016	.050±.013
FR-Train with noisy val. set	.502±.034	.073±.021	.513±.035	.041±.024
FR-Train with noisy+ITLM val. set	.531±.033	.087±.018	.530±.024	.059±.018
Ours	.521±.000	.044±.000	.544±.000	.025±.000

results are consistent with those in Tables 1, 2, and 3. ITLM is more accurate than LR, but fairness is similarly bad or worse. For our algorithm and FR-Train, accuracy performances are higher than LR, and fairness performances are better than both LR and ITLM.

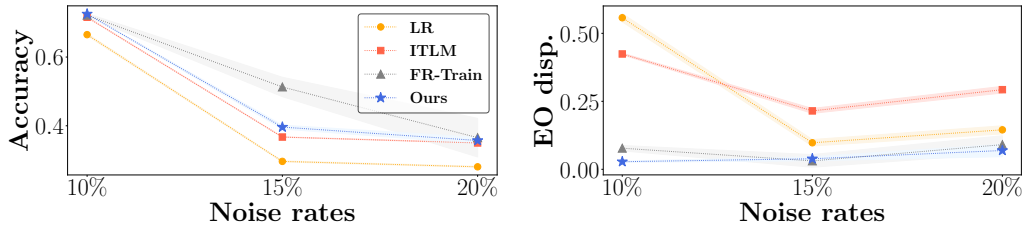
5.3 Ablation Study

In Table 5, we conduct an ablation study to investigate the effect of each step in our proposed algorithm. There are three ablation scenarios: (i) when removing the fairness constraints in the multi-dimensional knapsack problem that consider fairness in the data selection process, (ii) when removing the weighted ERM that ensures a fair distribution in the model training process, and (iii) when removing both functionalities (i.e., same as ITLM). In all ablation scenarios, the model training shows a more unfair result than our original algorithm. With these results, we can confirm that both the fairness constraints in the knapsack problem and the weighted ERM play significant roles in improving fairness. We thus take the synergetic effect of using the two functionalities together.

6 Related Work

Fair Training Model fairness consists of a variety of definitions that link social issues to machine learning research [Narayanan, 2018]. Among the definitions, we focus on group fairness definitions: equalized odds [Hardt et al., 2016] and demographic parity [Feldman et al., 2015]. To achieve the fairness measures, a variety of fair algorithms have been proposed, which can be categorized as follows: (i) pre-processing techniques that minimize bias in data [Kamiran and Calders, 2011, Zemel et al., 2013, Feldman et al., 2015, du Pin Calmon et al., 2017, Choi et al., 2020, Jiang and Nachum, 2020], (ii) in-processing techniques that revise the training process to prevent that model learns bias [Kamishima et al., 2012, Zafar et al., 2017a,b, Agarwal et al., 2018, Zhang et al., 2018, Cotter et al., 2019, Roh et al., 2020], and (iii) post-processing techniques that modify the outputs of the trained model to ensure fairness [Kamiran et al., 2012, Hardt et al., 2016, Pleiss et al., 2017, Chzhen et al., 2019]. Unlike other fairness approaches that assume clean training data, we propose a fair and robust training methodology that works even on noisy data.

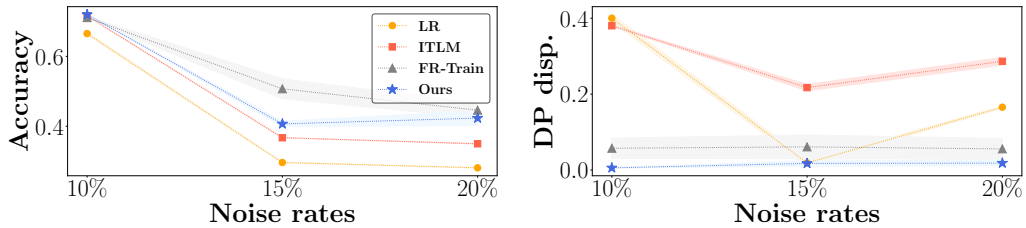
In addition, there is a recent line of work that adaptively selects training data to secure fairness [Roh et al., 2021]. The adaptive selecting approach only requires a single-line change of code in the batch selection, while many fair algorithms need significant non-trivial changes in the machine learning pipeline. Our algorithm has a similar spirit of achieving fairness by adjusting data size between groups, but ours further improves robustness by restricting the maximum number of selected samples.



(a) Accuracy.

(b) Equalized odds disparity.

Figure 3: Performances of logistic regression (LR), ITLM, FR-Train, and our algorithm (Ours) on the synthetic data for different noise rates injected with the general label flipping [Paudice et al., 2018]. Our algorithm and FR-Train show higher accuracy than LR and better fairness that LR and ITLM.



(a) Accuracy.

(b) Demographic parity disparity.

Figure 4: Performances of logistic regression (LR), ITLM, FR-Train, and our algorithm (Ours) on the synthetic data for different noise rates injected with the general label flipping [Paudice et al., 2018]. Our algorithm and FR-Train show higher accuracy than LR and better fairness that LR and ITLM.

Although not our current interests, we would like to introduce two important fairness definitions: individual fairness and causality-based fairness. Individual fairness [Dwork et al., 2012] aims to provide similar model predictions to the close examples. Causality-based fairness [Kilbertus et al., 2017, Kusner et al., 2017, Zhang and Bareinboim, 2018, Nabi and Shpitser, 2018, Khademi et al., 2019] focuses on understanding the causal relationship between attributes to address the limitations of the non-causal approaches.

Robust Training Robust training is a traditional research field of machine learning, which focuses on preventing performance degradations even when noisy or adversarial data is provided. Training data can contain various types of noisy data during data collection [Xiao et al., 2015]. Label noise is a vital research category of robust training, so numerous studies have been proposed to mitigate the impacts of label noises in the training data [Song et al., 2020], such as (i) robust architecture techniques that modify the model networks to be resilient to noises [Chen and Gupta, 2015, Jindal et al., 2016, Bekker and Goldberger, 2016, Han et al., 2018a], (ii) loss adjustment techniques that correct the loss value of each data point [Patrini et al., 2017, Chang et al., 2017, Ma et al., 2018, Arazo et al., 2019], and (iii) sample selection techniques that collect only clean samples for model training [Han et al., 2018b, Jiang et al., 2018, Chen et al., 2019, Shen and Sanghavi, 2019]. Unlike the traditional robust approaches, which generally ignore the bias issues in the training data, our algorithm achieves robustness while improving fairness in the holistic framework.

Fair & Robust Training There is a growing discussion that fairness and robustness should be considered together in model training [Lee et al., 2021]. A recent study reveals that the accuracy-fairness tradeoff can worsen when the fair algorithms ignore the noises in the training data [Roh et al., 2020]. To address this issue, they propose an adversarial learning-based algorithm to achieve fairness and robustness using a clean validation set. Compared to the existing fair and robust algorithm, our sample selection method does not require modifying the inner parts of model training and utilizing an additional clean validation set.

Another recent study shows that feature selection for robustness can lead to unfair model training [Khani and Liang, 2021]. This observation is related to our study in investigating the unfairness of robust-only training, but we focus on sample selection rather than feature selection.

Table 5: Ablation study on the synthetic and COMPAS test sets with the general label flipping (10%). We compare three ablation scenarios: (i) when removing the fairness constraints in the multi-dimensional knapsack problem that consider fairness in the data selection process, (ii) when removing the weighted ERM that ensures a fair distribution in the model training process, and (iii) when removing both functionalities (i.e., same as ITLM). As a result, the original algorithm with all functionalities shows the highest fairness than other ablation scenarios.

	Synthetic				COMPAS			
Method	Acc.	EO Disp.	Acc.	DP Disp.	Acc.	EO Disp.	Acc.	DP Disp.
Without knapsack const.	.xxx±.xxx	.xxx±.xxx	.723±.001	.036±.002	.xxx±.xxx	.xxx±.xxx	.520±.002	.073±.002
Without weighted ERM	.xxx±.xxx	.xxx±.xxx	.727±.000	.251±.009	.xxx±.xxx	.xxx±.xxx	.529±.007	.058±.020
Without both	.716±.001	.424±.003	.716±.009	.380±.002	.536±.000	.145±.003	.536±.000	.094±.003
Ours	.725±.001	.028±.002	.720±.001	.006±.001	.521±.000	.044±.000	.544±.000	.025±.000

Although we currently do not focus, there is another perspective that enables fair training even when the sensitive group information is noisy or missing [Hashimoto et al., 2018, Lahoti et al., 2020, Wang et al., 2020]. While these studies aim to achieve fairness with the unreliable sensitive group information, we focus on fair and robust training with the label noises. In addition, several fairness-targeted attack techniques are recently proposed [Chang et al., 2020, Solans et al., 2020]. We believe an interesting future work is to consider the above scenarios.

7 Conclusion

Societal Impact and Limitations

References

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In *ICML*, pages 60–69, 2018.

Yalçın Akçay, Haijun Li, and Susan H Xu. Greedy algorithm for the general multidimensional knapsack problem. *Annals of Operations Research*, 150(1):17–29, 2007.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. And its biased against blacks. ProPublica, 2016.

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *ICML*, pages 312–321, 2019.

Alan Joseph Bekker and Jacob Goldberger. Training deep neural-networks based on unreliable labels. In *IEEE ICASSP*, pages 2682–2686, 2016.

Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John T. Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J. Res. Dev.*, 63(4/5):4:1–4:15, 2019.

Alberto Caprara, Hans Kellerer, Ulrich Pferschy, and David Pisinger. Approximation algorithms for knapsack problems with cardinality constraints. *EJOR*, 123(2):333–345, 2000.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, 2017.

Hongyan Chang, Ta Duy Nguyen, Sasi Kumar Murakonda, Ehsan Kazemi, and Reza Shokri. On adversarial bias and the robustness of fair machine learning. *ArXiv*, 2020.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *ICML*, pages 1062–1070, 2019.

Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *ICCV*, pages 1431–1439, 2015.

Kristy Choi, Aditya Grover, Trisha Singh, Rui Shu, and Stefano Ermon. Fair generative modeling via weak supervision. In *ICML*, 2020.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Leveraging labeled and unlabeled data for consistent fair binary classification. In *NeurIPS*, pages 12760–12770. 2019.

Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In *ALT*, pages 300–332, 2019.

Flávio du Pin Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized pre-processing for discrimination prevention. In *NeurIPS*, pages 3995–4004, 2017.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In *ITCS*, pages 214–226, 2012.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *SIGKDD*, pages 259–268, 2015.

Michael Randolph Garey and David Stifler Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. W. H. Freeman, 1979.

Google. Responsible ai practices. `https://ai.google/responsibilities/responsible-ai-practices`, 2020.

Bo Han, Jiangchao Yao, Niu Gang, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, pages 5839–5849, 2018a.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *NeurIPS*, pages 3315–3323, 2016.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *ICML*, pages 1929–1938, 2018.

IBM. Trusting ai. `https://www.research.ibm.com/artificial-intelligence/trusted-ai/`, 2020.

Heinrich Jiang and Ofir Nachum. Identifying and correcting label bias in machine learning. In *AISTATS*, pages 702–712, 2020.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pages 2304–2313, 2018.

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *IEEE ICDM*, pages 967–972, 2016.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, 2011.

Faisal Kamiran, Asim Karim, and Xiangliang Zhang. Decision theory for discrimination-aware classification. In *IEEE ICDM*, pages 924–929, 2012.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *ECML PKDD*, pages 35–50, 2012.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer Nature Book Archives Millennium. 2004.

Aria Khademi, Sanghack Lee, David Foley, and Vasant Honavar. Fairness in algorithmic decision making: An excursion through the lens of causality. In *WWW*, pages 2907–2914, 2019.

Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. In *ACM FAccT*, pages 196–205, 2021.

Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. In *NeurIPS*, pages 656–666, 2017.

Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *SIGKDD*, pages 202–207, 1996.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *NeurIPS*, pages 4066–4076. 2017.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. In *NeurIPS*, pages 728–740, 2020.

Jae-Gil Lee, Yuji Roh, Hwanjun Song, and Steven Euijong Whang. Machine learning robustness, fairness, and their convergence. In *SIGKDD*, 2021.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE TPAMI*, 38(3):447–461, 2015.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, pages 3355–3364, 2018.

Microsoft. Responsible ai principles from microsoft. `https://www.microsoft.com/en-us/ai/responsible-ai`, 2021.

Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. In *AAAI*, pages 1931–1940, 2018.

Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *ACM FAccT*, volume 1170, 2018.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pages 1944–1952, 2017.

Andrea Paudice, Luis Muñoz-González, and Emil C. Lupu. Label sanitization against label flipping poisoning attacks. In *ECML PKDD*, pages 5–15, 2018.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon M. Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. In *NeurIPS*, pages 5684–5693, 2017.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. FR-Train: A mutual information-based approach to fair and robust training. In *ICML*, pages 8147–8157, 2020.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. FairBatch: Batch selection for model fairness. In *ICLR*, 2021.

Peter J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984.

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *ICML*, pages 5739–5748, 2019.

David Solans, Battista Biggio, and Carlos Castillo. Poisoning attacks on algorithmic fairness. In *ECML PKDD*, pages 162–177, 2020.

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, pages 5907–5915, 2019.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *ArXiv*, 2020.

Serena Wang, Wenshuo Guo, Harikrishna Narasimhan, Andrew Cotter, Maya Gupta, and Michael Jordan. Robust optimization for fairness with noisy protected groups. In *NeurIPS*, pages 5190–5203, 2020.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, pages 2691–2699, 2015.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *AISTATS*, pages 962–970, 2017a.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *WWW*, pages 1171–1180, 2017b.

Richard S. Zemel, Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, pages 325–333, 2013.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *AIES*, pages 335–340, 2018.

Junzhe Zhang and Elias Bareinboim. Fairness in decision-making - the causal explanation formula. In *AAAI*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract and Section 1 (Introduction).
   (b) Did you describe the limitations of your work? **[TODO]**
   (c) Did you discuss any potential negative societal impacts of your work? **[TODO]**
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? **[TODO]**
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? **[TODO]**
   (b) Did you include complete proofs of all theoretical results? **[TODO]**
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[TODO]**
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[TODO]**
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5 (Experiments).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5 (Experiments).
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? **[TODO]**
   (b) Did you mention the license of the assets? **[TODO]**
   (c) Did you include any new assets either in the supplemental material or as a URL? **[TODO]**
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[TODO]**
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[TODO]**

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

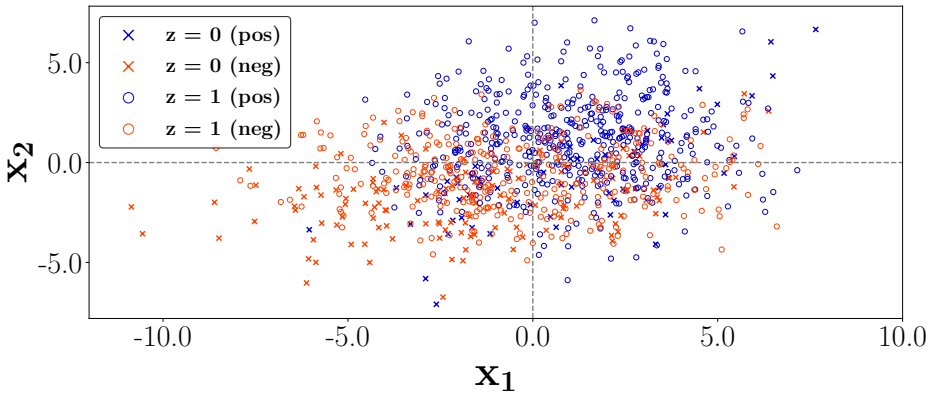


Figure 5: Synthetic data.

A Appendix - Framework and Algorithm

A.1 An Update Rule for Demographic Parity

A.2 Rearranging the Fairness Constraints in the Multidimensional Knapsack Problem

$$\sum_{i=1}^{n} \mathbb{I}_{D_{(y,z)}}(d_i)\, p_i \leq \lfloor \lambda_{(y,z)} |S_y| \rfloor = \lfloor \lambda_{(y,z)} \sum_{i=1}^{n} \mathbb{I}_{D_y}(d_i)\, p_i \rfloor$$

$$\sum_{i=1}^{n} \mathbb{I}_{D_{(y,z)}}(d_i)\, p_i - \lambda_{(y,z)} \sum_{i=1}^{n} \mathbb{I}_{D_y}(d_i)\, p_i \leq 0$$

$$\sum_{i=1}^{n} v_i p_i \leq 0, \text{where } v_i = \begin{cases} 0 & \text{if } d_i \notin D_y \\ -\lambda_{(y,z)} & \text{if } d_i \in D_y \text{ and } d_i \notin D_z \\ 1 - \lambda_{(y,z)} & \text{if } d_i \in D_{y,z} \end{cases}$$

$$\sum_{i=1}^{n} w_i p_i \leq \lfloor \tau n \rfloor, \text{where } w_i = \begin{cases} 1 & \text{if } d_i \notin D_y \\ 1 - \lambda_{(y,z)} & \text{if } d_i \in D_y \text{ and } d_i \notin D_z \\ 2 - \lambda_{(y,z)} & \text{if } d_i \in D_{y,z} \end{cases}.$$

B Appendix - Experiments

B.1 Other Experimental Settings

B.2 Synthetic Data

B.3 Other Fair and Robust Baselines

CS+Penalty is inspired by the fairness constraints method [Zafar et al., 2017a,b] that adds a fairness penalty term to the loss function for reducing the covariance between sensitive attributes and the predicted labels. We thus improve ITLM by adding the covariance term to the optimization: $\min_{S:|S|=\lfloor \tau N \rfloor} \sum_{s_i \in S} [l_\theta(s_i) + \mu\, |Cov(z_i, \hat{y}_i)|]$ where z_i and $\hat{y}_i$ are the sensitive group and the predicted label of the sample s_i, respectively.

Table 6: Performances on the **synthetic** test set w.r.t. **equalized odds (EO)**. We compare our algorithm with four types of baselines: (1) vanilla training: LR; (2) robust training: ITLM [Shen and Sanghavi, 2019]; (3) fair training: FB [Roh et al., 2021]; and (4) fair and robust training: ITLM→FB, ITLM→Penalty [Zafar et al., 2017b], and FR-Train [Roh et al., 2020]. Experiments are repeated 5 times.

	Full (10%)		Majority (10%)		Minority (10%)	
Method	Acc.	EO Disp.	Acc.	EO Disp.	Acc.	EO Disp.
LR	.665±.003	.557±.015	.600±.002	.405±.008	.621±.001	.250±.002
ITLM	.716±.009	.424±.003	.719±.001	.315±.017	.660±.002	.318±.006
FB	.509±.017	.051±.018	.544±.005	.072±.010	.576±.004	.060±.010
ITLM→FB	.718±.003	.199±.020	.707±.001	.108±.030	.669±.004	.202±.045
ITLM→Penalty	.651±.051	.172±.046	.706±.001	.080±.004	.656±.003	.209±.017
FR-Train	.722±.005	.078±.010	.715±.003	.046±.011	.713±.004	.026±.013
Ours	.725±.001	.028±.002	.725±.001	.034±.002	.691±.010	.048±.003

Table 7: Performances on the **synthetic** test set w.r.t. **demographic parity (DP)**. We compare our algorithm with four types of baselines: (1) vanilla training: LR; (2) robust training: ITLM [Shen and Sanghavi, 2019]; (3) fair training: FB [Roh et al., 2021]; and (4) fair and robust training: ITLM→FB, ITLM→Penalty [Zafar et al., 2017b], and FR-Train [Roh et al., 2020]. Experiments are repeated 5 times.

	Full (10%)		Majority (10%)		Minority (10%)	
Method	Acc.	DP Disp.	Acc.	DP Disp.	Acc.	DP Disp.
LR	.665±.003	.400±.010	.600±.002	.300±.006	.621±.001	.169±.003
ITLM	.716±.009	.380±.002	.719±.001	.287±.009	.660±.002	.195±.006
FB	.683±.002	.063±.019	.526±.003	.082±.002	.572±.002	.078±.022
ITLM→FB	.725±.002	.089±.032	.704±.003	.067±.027	.585±.002	.093±.004
ITLM→Penalty	.674±.012	.068±.014	.688±.004	.044±.004	.578±.008	.056±.012
FR-Train	.711±.003	.057±.027	.706±.011	.079±.030	.700±.011	.061±.037
Ours	.720±.001	.006±.001	.720±.001	.039±.007	.670±.016	.021±.008

Table 8: Performances on the **COMPAS** test set w.r.t. **equalized odds (EO)**. Other experimental settings are identical to Table 1.

	Full (10%)		Majority (10%)		Minority (10%)	
Method	Acc.	EO Disp.	Acc.	EO Disp.	Acc.	EO Disp.
LR	.503±.002	.123±.021	.513±.000	.668±.000	.631±.000	.580±.002
ITLM	.536±.000	.145±.003	.539±.002	.573±.019	.623±.000	.521±.000
FB	.509±.001	.083±.013	.525±.004	.093±.003	.550±.003	.060±.011
ITLM→FB	.520±.001	.064±.013	.538±.008	.315±.074	.578±.007	.299±.036
ITLM→Penalty	.523±.003	.074±.027	.517±.014	.436±.028	.548±.030	.294±.090
FR-Train	.620±.009	.074±.015	.611±.029	.081±.020	.639±.012	.118±.047
Ours	.521±.000	.044±.000	.545±.009	.084±.012	.541±.003	.057±.002

Table 9: Performances on the **COMPAS** test set w.r.t. **demographic parity (DP)**. We compare our algorithm with four types of baselines: (1) vanilla training: LR; (2) robust training: ITLM [Shen and Sanghavi, 2019]; (3) fair training: FB [Roh et al., 2021]; and (4) fair and robust training: ITLM→FB, ITLM→Penalty [Zafar et al., 2017b], and FR-Train [Roh et al., 2020]. Experiments are repeated 5 times.

	Full (10%)		Majority (10%)		Minority (10%)	
Method	Acc.	DP Disp.	Acc.	DP Disp.	Acc.	DP Disp.
LR	.503±.002	.093±.020	.513±.000	.648±.000	.631±.000	.453±.002
ITLM	.536±.000	.094±.003	.539±.002	.547±.015	.623±.000	.399±.000
FB	.503±.002	.046±.010	.507±.004	.059±.010	.441±.001	.076±.017
ITLM→FB	.523±.003	.039±.009	.520±.013	.130±.105	.574±.000	.209±.000
ITLM→Penalty	.525±.001	.059±.008	.514±.007	.094±.022	.533±.003	.065±.008
FR-Train	.607±.016	.050±.013	.597±.039	.045±.014	.621±.025	.079±.038
Ours	.544±.000	.025±.000	.521±.000	.024±.000	.511±.014	.076±.033

Table 10: Performances on the **AdultCensus** test set w.r.t. **equalized odds (EO)**. We compare our algorithm with four types of baselines: (1) vanilla training: LR; (2) robust training: ITLM [Shen and Sanghavi, 2019]; (3) fair training: FB [Roh et al., 2021]; and (4) fair and robust training: ITLM→FB, ITLM→Penalty [Zafar et al., 2017b], and FR-Train [Roh et al., 2020]. Experiments are repeated 5 times.

	Full (10%)		Majority (10%)		Minority (10%)	
Method	Acc.	EO Disp.	Acc.	EO Disp.	Acc.	EO Disp.
LR	.746±.003	.070±.002	.748±.006	.095±.002	.760±.002	.096±.002
ITLM	.785±.018	.087±.031	.785±.011	.144±.023	.810±.012	.039±.012
FB	.748±.002	.022±.002	.739±.014	.086±.037	.750±.004	.110±.011
ITLM→FB	.772±.024	.047±.008	.773±.013	.047±.006	.785±.014	.050±.016
ITLM→Penalty	.776±.023	.082±.015	.755±.026	.161±.018	.768±.008	.069±.008
FR-Train	.765±.009	.059±.021	.722±.027	.069±.016	.764±.007	.057±.009
Ours	.771±.015	.029±.005	.761±.006	.047±.018	.762±.014	.054±.015

Table 11: Performances on the **AdultCensus** test set w.r.t. **demographic parity (DP)**. We compare our algorithm with four types of baselines: (1) vanilla training: LR; (2) robust training: ITLM [Shen and Sanghavi, 2019]; (3) fair training: FB [Roh et al., 2021]; and (4) fair and robust training: ITLM→FB, ITLM→Penalty [Zafar et al., 2017b], and FR-Train [Roh et al., 2020]. Experiments are repeated 5 times.

	Full (10%)		Majority (10%)		Minority (10%)	
Method	Acc.	DP Disp.	Acc.	DP Disp.	Acc.	DP Disp.
LR	.746±.003	.073±.001	.748±.006	.127±.006	.760±.002	.036±.003
ITLM	.785±.018	.092±.016	.785±.011	.105±.006	.810±.012	.072±.015
FB	.758±.004	.046±.007	.693±.002	.015±.005	.747±.007	.093±.016
ITLM→FB	.776±.023	.073±.010	.769±.014	.053±.005	.786±.013	.057±.017
ITLM→Penalty	.774±.024	.054±.018	.757±.003	.013±.003	.798±.012	.031±.014
FR-Train	.771±.012	.093±.006	.773±.012	.049±.010	.781±.008	.045±.006
Ours	.782±.015	.049±.012	.743±.005	.012±.002	.792±.005	.027±.003