Mitigating Source Bias for Fairer Weak Supervision

Changho Shin, Sonia Cromp, Dyah Adila, Frederic Sala

Department of Computer Sciences
University of Wisconsin-Madison
{cshin23, cromp, adila, fsala}@wisc.edu

March 30, 2023

Abstract

Weak supervision overcomes the label bottleneck, enabling efficient development of training sets. Millions of models trained on such datasets have been deployed in the real world and interact with users on a daily basis. However, the techniques that make weak supervision attractive—such as integrating any source of signal to estimate unknown labels—also ensure that the pseudolabels it produces are highly biased. Surprisingly, given everyday use and the potential for increased bias, weak supervision has not been studied from the point of view of fairness. This work begins such a study. Our departure point is the observation that even when a fair model can be built from a dataset with access to ground-truth labels, the corresponding dataset labeled via weak supervision can be arbitrarily unfair. Fortunately, not all is lost: we propose and empirically validate a model for source unfairness in weak supervision, then introduce a simple counterfactual fairness-based technique that can mitigate these biases. Theoretically, we show that it is possible for our approach to simultaneously improve both accuracy and fairness metrics—in contrast to standard fairness approaches that suffer from tradeoffs. Empirically, we show that our technique improves accuracy on weak supervision baselines by as much as 32% while reducing demographic parity gap by 82.5%.

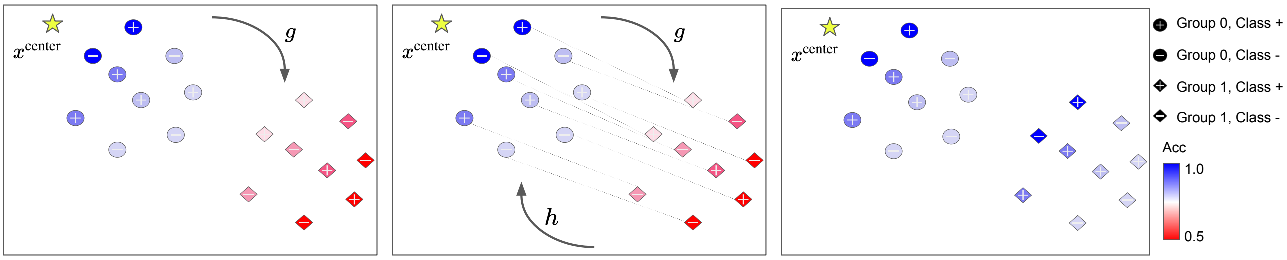
1 Introduction

Weak supervision (WS) is a powerful set of techniques aimed at overcoming the labeled data bottleneck [19, 42, 46]. Instead of manually annotating points, users assemble noisy label estimates obtained from multiple sources, model them by learning source accuracies, and combine them into a high-quality pseudolabel to be used for downstream training. All of this is done without any ground truth labels. Simple, flexible, yet powerful, weak supervision is now a standard component in machine learning workflows [2] in industry, academia, and beyond. Most excitingly, WS has been used to build models deployed to billions of devices.

Real-life deployment of models, however, raises crucial questions of fairness and bias. Such questions are tackled in the burgeoning field of fair machine learning [15, 25]. However, weak supervision *has not been studied from this point of view.* This is not a minor oversight. The properties that make weak supervision effective (i.e., omnivorously ingesting any source of signal for labels) are precisely those that make it likely to suffer from harmful biases. This motivates the need to understand and mitigate the potentially disparate outcomes that result from using weak supervision.

The starting point for this work is a simple result. Even when perfectly fair classifiers are possible when trained on ground-truth labels, weak supervision-based techniques can nevertheless produce arbitrarily unfair outcomes. Because of this, simply applying existing techniques for producing fair outcomes to the datasets produced via WS is insufficient—delivering highly suboptimal datasets. Instead, a new approach, specific to weak supervision, must be developed.

We introduce a simple technique for improving the fairness properties of weak supervision-based models. Intuitively, a major cause of bias in WS is that particular sources are targeted at certain groups, and so produce far more accurate label estimates for these groups—and far more noise for others. We counterfactually ask what outgroup points would most be like if they were part of the 'privileged' group (with respect to each source), enabling us to borrow from the more powerful signal in the sources applied to this group. Thus the problem is reduced to finding a transformation between groups that satisfies this counterfactual. Most excitingly, while in standard fairness



(a) Data & LF accuracy distribution (b) Reverse group transformation (c) Recovered labels and accuracy

Figure 1: Intuitive illustration for our setting and approach. (a): circles and diamonds are datapoints from group 0 and 1, respectively. Labeling function vote accuracy is colored-coded, with blue being perfect (1.0) and red random (0.5). Note that accuracy degrades as data points get farther from center x^{center} (star). (b) We can think of group 1 as having been moved far from the center via a transformation g, producing lower-quality estimates and violating fairness downstream. (c) Our technique uses counterfactual fairness to undo this transformation, obtaining higher quality estimates. and improved fairness.

approaches there is a typical tradeoff between fairness and accuracy, with our approach, both the fairness and performance of WS-based techniques can be (sometimes dramatically) improved.

Theoretically, in certain settings, we provide finite-sample rates for recovering the counterfactual transformation. Empirically, we propose several ways to craft an efficiently-computed transformation building on optimal transport and some simple variations. We validate our claims on a diverse set of experiments. These include standard real-world fairness datasets, where we observe that our method can improve both fairness and accuracy by as much as 82.5% and 32.5%, respectively, versus weak supervision baselines. Our method can also be combined with other fair ML methods developed for fully supervised settings, further improving fairness. Finally, our approach has implications for WS beyond bias: as a proof of concept, we applied our technique to synthetically crafted groups in a dataset from the WRENCH benchmark [54] and obtained mild improvements—suggesting that it could be used to generically improve weak supervision.

The contributions of this work include,

- The first study of fairness in weak supervision,
- A new empirically-validated model for weak supervision that captures labeling function bias,
- A simple counterfactual fairness-based correction to mitigate such bias, compatible with any existing weak supervision pipeline, as well as with downstream fairness techniques,
- Theoretical results showing that (1) even with a fair labeled dataset, a weakly-supervised counterpart can be arbitrarily biased and (2) a finite-sample recovery result for the correction algorithm we propose,
- Experiments validating our claims, including on weakly-supervised forms of popular fairness evaluation datasets, showing gains in fairness metrics—and often simultaneously improvements in accuracy.

2 Background and Related Work

We present some high-level background on weak supervision and fairness in machine learning. Afterward, we provide setup and present the problem statement.

2.1 Weak Supervision

Weak supervision frameworks build labeled training sets *with no access to ground truth labels*. Instead, they exploit multiple sources that provide noisy estimates of the label. These sources include heuristic rules, knowledge base lookups, pretrained models, and more [11, 22, 28, 38, 44]. Because these sources may have different—and unknown—accuracies and dependencies, their outputs must be modeled in order to produce a combination that can be used as a high-quality pseudolabel.

Concretely, there is a dataset $\{(x_1,y_1),...,(x_n,y_n)\}$, but the true label $y_i \in \{-1,+1\}$ is not observed. Instead we can access the outputs of m sources (also called labeling functions) $\lambda^1,\lambda^2,...,\lambda^m \in \{-1,+1\}$ outputting noisy estimates of the labels. These outputs are modeled via a generative model called the *label model*, $p_\theta(\lambda^1,...,\lambda^m,y)$. The goal is to estimate the parameters θ of this model, without accessing the latent y, and to produce a pseudolabel estimate $p_{\hat{\theta}}(y|\lambda^1,...,\lambda^m)$. For more background, see [55].

2.2 Machine Learning and Fairness

Fairness in machine learning is a large and active field that seeks to understand and mitigate biases. Because of the depth of this area, we can only briefly introduce high-level notions that will be useful in the weak supervision setting.

In order to measure fairness, it is necessary to introduce a metric. Many such metrics have been proposed; two popular choices are demographic parity [15] and equal opportunity [25]. Demographic parity is based on the notion that individuals of different groups should have equal treatment, i.e., if A is the group attribute, $P(\hat{Y}=1|A=1)=P(\hat{Y}=1|A=0)$. The equal opportunity principle requires that predictive error should be equal across groups, i.e., $P(\hat{Y}=1|Y=1,A=1)=P(\hat{Y}=1|Y=1,A=0)$.

A large number of works study, measure, and seek to improve fairness in different machine learning settings based on these metrics. Typically, the assumption is that the underlying dataset differs within groups in such a way that a trained model will violate, for example, the equal opportunity principle. In contrast, in this work, we focus on additional violations of fairness that are induced by weak supervision pipelines—which can create substantial unfairness even when the true dataset is perfectly fair. In the same spirit is [52], which considers fairness in positive-and-unlabeled (PU) settings, where true labels are available, but only for one class, while other points are unlabeled.

Counterfactual Fairness Most closely related to the notion of fairness we use in this work is *counterfactual fairness*. [33] introduced such a counterfactual fairness notion, which implies that changing the sensitive attribute A, while keeping other variables causally not dependent on A, should not affect the outcome. While this notion presumes the causal structure behind the ML task, it is related to our work in the sense that our proposed method tries to remove the causal effect by A with a particular transformations.

A more recent line of work has proposed bypassing the need for causal structures and directly tackling counterfactual fairness through optimal transport [4, 6, 20, 47, 48]. The idea is to detect or mitigate unfairness by mapping one group to another group via such techniques. In this paper, we build on these tools to help improve fairness while avoiding the accuracy-fairness tradeoff common to most settings.

3 Mitigating Labeling Function-Induced Unfairness

We are ready to explain our approach to mitigating unfairness in weak supervision sources. First, we provide a flexible model that captures such behavior, along with empirical evidence supporting it. Next, we propose a simple solution to correct unfair source behavior via optimal transport.

Modeling Group Bias in Weak Supervision Weak supervision models the accuracies and correlations in labeling functions. The standard model, used in [19, 43] and others is

$$P(\lambda^1,...,\lambda^m,y)=\frac{1}{Z}\exp(\theta_y y+\sum_{j=1}^{m}\theta_j\lambda^j y), \tag{1}$$

with $\theta_j \geq 0$. We leave out the correlations for simplicity; all of our discussion below holds when considering correlations as well. Here, Z is the normalizing partition function. The θ are *canonical parameters* for the model. θ_y sets the class balance. The θ_i's capture how accurate LF i is: if $\theta_i=0$, the LF is independent of y and so produces random guesses. If θ_i is relatively large, the LF is highly accurate.

A weakness of model (1) is that it *ignores the feature vector x*. It implies that LFs are uniformly accurate over the feature space—a highly unrealistic assumption. A more general model was presented in [7], where there is a model for each feature vector x, i.e.,

$$P_x(\lambda^1,\ldots,\lambda^m,y) = \frac{1}{Z}\exp(\theta_y y + \sum_{j=1}^{m} \theta_{j,x}\lambda^j(x)y). \tag{2}$$

However, as we see only one sample for each x, it is impossible to recover the parameters θ_x. Instead, the authors assume a notion of *smoothness*. This means that the $\theta_{j,x}$'s do not vary in small neighborhoods, so that the feature space can be partitioned and a single model learned per part. This model is more general, but still requires the smoothness assumption. It also does not encode any notion of bias. Instead, we propose a model that encodes both smoothness and bias.

Concretely, let us assume that the data is drawn from some distribution on $\mathcal{Z}\times\mathcal{Y}$, where $\mathcal{Z}$ is a latent space. We do not observe samples from $\mathcal{Z}$. Instead, there are l transformation functions $g_1,\ldots,g_l$, where $g_k:\mathcal{Z}\to\mathcal{X}$. For each point z_i, there is an assigned group k and we observe $x_i = g_k(z_i)$. Then, our model is the following:

$$P(\lambda^1(z),\ldots,\lambda^m(z),y) = \frac{1}{Z}\exp\left(\theta_y y + \sum_{j=1}^{m} \frac{\theta_j}{1+d(x^{\text{center}_j}, g_k(z))}\lambda^j(g_k(z))y\right). \tag{3}$$

We explain this model as follows. We can think of it as a particular version of (2). However, instead of arbitrary $\theta_{j,x}$ parameters for each x, we explicitly model these parameters as two components:

- A feature-independent accuracy parameter θ_j,
- A denominator term that modulates the accuracy based on the distance between feature vector x and some fixed center x^{center_j}.

The center represents, for each LF, the most accurate point. Here the LF accuracy is maximized at a level set by the parameter θ_j. However, as the feature vector $x = g_k(z)$ moves away from this center, the LF votes increasingly poorly. Note that this is an explicit form of smoothness.

Below, for simplicity, we assume there are two groups, indexed by 0,1, that $\mathcal{X} = \mathcal{Z}$, and that $g_0(z) = z$. In other words, the transformation for group 0 is the identity, while this may not be the case for group 1. We note that simple extensions of our approach can handle cases where none of these assumptions are met.

Labeling Function Bias The model (3) explains how and when labeling functions might be biased. Suppose that g_k takes points z far from x^{center_j}. Then, the denominator term in (3) grows—and so the penalty for $\lambda(x)$ to disagree with y is reduced, making the labeling function less accurate.

Indeed, this is common in practice. For example, consider a situation where a bank uses features that include credit scores for loan review. Suppose the group variable is the applicant's nationality. Immigrants typically have a shorter period to build credit; this is reflected in a transformed distribution $g_1(z)$. A labeling function using a credit score threshold may be accurate for non-immigrants, but may end up being highly inaccurate when applied to immigrants.

We validate this notion empirically as well. We used the Adult dataset [32], commonly used for fairness studies, with a set of custom-built labeling functions. In Figure 2, we track the accuracies of these LFs as a function of distance from an empirically-discovered center x^{center_j}. On the left is the high-accuracy group; as expected in our model, as we increase the distance, the accuracy decreases.

On the right-hand side, we see the lower-accuracy group, whose labeling functions are voting $x_i = g_1(z_i)$. This transformation has sent these points further away from the center (note the larger distances). As a result, the overall accuracies have also decreased. Note, for example, how LF 5, in purple, varies between 0.9 and 1.0 accuracy in one group and is much worse—between 0.6 and 0.7—in the other.

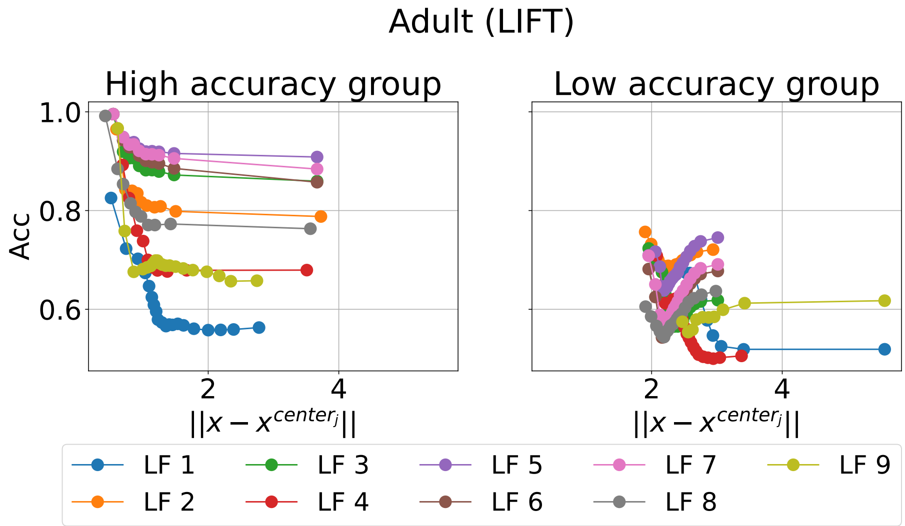


Figure 2: Average accuracy (y-axis) depending on the distance to the center point (x-axis). The center is obtained by evaluating accuracy of all data points based on their neighborhood.

Algorithm 1: SOURCE BIAS MITIGATION (SBM)

Parameters: Features X_0, X_1 and LF outputs $\Lambda_0 = [\lambda_0^1, \dots, \lambda_0^m]$, $\Lambda_1 = [\lambda_1^1, \dots, \lambda_1^m]$ for groups 0, 1, transport threshold ε
Returns: Modified weak labels $\Lambda = [\lambda^1, \dots, \lambda^m]$
Estimate accuracy of λ^j in each group, $\hat{a}_0^j, \hat{a}_1^j$ from Λ_0, Λ_1 with Algorithm 2
for $j \in \{1, 2, \dots, m\}$ **do**
  if $\hat{a}_1^j \geq \hat{a}_0^j + \varepsilon$ **then**
    Update λ_0^j by transporting X_0 to X_1 (Algorithm 3)
  else if $\hat{a}_0^j \geq \hat{a}_1^j + \varepsilon$ **then**
    Update λ_1^j by transporting X_1 to X_0 (Algorithm 3)
  end if
end for
return $\Lambda = [\lambda^1, \dots, \lambda^m]$

3.1 Correcting Unfair LFs

Given the model (3), how can we reduce the bias induced by the g_k functions? A simple idea is to *reverse* the effect of the g_k's. If we could invert these functions, violations of fairness would be mitigated, since the accuracies of labeling functions would be uniformized over the groups.

Concretely, suppose that g_k is invertible and that h_k is this inverse. If we knew h_k, then we could ask the labeling functions to vote on $h_k(x) = h_k(g_k(x)) = z$, rather than on $x = g_k(z)$, and we could do so for any group, yielding equal-accuracy estimates for all groups. The technical challenge is how to estimate the inverses of the g_k's, without any parametric form for these functions. To do so, we deploy optimal transport (OT) [40]. OT transports a probability distribution to another probability distribution by finding a minimal cost coupling. We use OT to recover the reverse map $h_k : \mathcal{X} \to \mathcal{Z}$ by

$$\hat{h}_k = \arg \inf_{T(\nu)=\omega} \left\{ \int_{x \in \mathcal{X}} c(x, T(x)) d\nu(x) \right\},$$

where c is a cost functon, ν is a probability measure in $\mathcal{X}$ and ω is a probability measure in $\mathcal{Z}$.

Our proposed approach, building on the use of OT, is called *source bias mitigation* (SBM). It seeks to reverse the group transformation g_k via OT. The core routine is described in Algorithm 1. The first step of the algorithm is to estimate the accuracies of each group so that we can identify which group is privileged, i.e., which of the transformations g_0, g_1 is the identity map. To do this, we use Algorithm 2 [19] by applying it to each group separately.

After identifying the high-accuracy group, we transport data points from the low-accuracy group to it. Since not every transported point perfectly matches an existing high-accuracy group point, we find a nearest neighbor and borrow its label. We do this only when there is a sufficient inter-group accuracy gap, since the error in transport might otherwise offset the benefit.

In practice, if the transformation is sufficiently weak, it is possible to skip optimal transport and simply use nearest neighbors. Doing this turned out to be effective in some experiments (Section 5.1). Finally, after running SBM, modified weak labels are used in a standard weak supervision pipeline, which is described in Appendix C.

4 Theoretical Results

We provide two types of theoretical results. First, we show that labeling function bias can be arbitrarily bad—resulting in substantial unfairness—regardless of whether the underlying dataset is fair. Next, we show that in certain settings, we can consistently recover the fair labeling function performance when using Algorithm 1, and provide a finite-sample error guarantee. Finally, we comment on extensions. All proofs are located in Appendix D.

Setting and Assumptions We assume that the distributions $P_0(x)$ and $P_1(x')$ are subgaussian with means μ_0 and μ_1 and positive-definite covariance matrices Σ_0 and Σ_1, respectively. Note that by assumption, $P_0(x) = P(z)$ and $P_1(x')$ is the pushforward of $P_0(x)$ under g_1. Let $\mathbf{r}(\Sigma)$ denote the effective rank of Σ [50].

We observe n_0 and n_1 i.i.d. samples from groups 0 and 1, respectively. We use Euclidean distance as the distance $d(x,y) = ||x-y||$ in model (3). For the unobserved ground truth labels, y_i is drawn from some distribution $P(y|z)$. Finally, the labeling functions voting on our points are drawn via the model (3).

4.1 Labeling Functions can be Arbitrarily Unfair

We show that, as a result of the transformation g_1, the predictions of labeling functions can be arbitrarily unfair even if the dataset is fair. The idea is simple: the average group 0 accuracy, $\mathbb{E}_{z\in\mathcal{Z}}[P(\lambda(I(z)) = y)]$, is independent of g_1, so it suffices to show that $\mathbb{E}_{x'\in g_1(\mathcal{Z})}[P(\lambda(x') = y)]$ can deteriorate when g_1 moves data points far from the center x^{center_0}. As such, we consider the change in $\mathbb{E}_{x'\in g_1(\mathcal{Z})}[P(\lambda(x') = y)]$ as the group 1 points are transformed increasingly far from x^{center_0} in expectation.

Theorem 4.1. *Let $g_1^{(k)}$ be an arbitrary sequence of functions such that $\lim_{k\to\infty} \mathbb{E}_{x'\in g_1^{(k)}(\mathcal{Z})}[||x' - x^{center_0}||] \to \infty$. Suppose our assumptions above our met; in particular, that the label y is independent of the observed features $x = I(z)$ or $x' = g_1^{(k)}(z), \forall k$, conditioned on the latent features z. Then,*

$$\lim_{k\to\infty} \mathbb{E}_{x'\in g_1^{(k)}(\mathcal{Z})}[P(\lambda(x') = y)] = \frac{1}{2},$$

which corresponds to random guessing.

It is easy to construct such a sequence of functions $g_1^{(k)}$, for instance by letting $g_1^{(k)}(z) = z + ku$, where u is a d-dimensional vector of ones.

When the distribution of group 1 points lies far from x^{center_0} while the distribution of group 0 points lies near to x^{center_0}, the accuracy parity of λ suffers. With adequately large expected $d(x^{\text{center}_0}, g_1^{(k)}(z))$, the performance of λ on group 1 points approaches random guessing.

Table 1: Summary of real datasets

Domain	Datasets	Y	A
Tabular	Adult	Income ($\geq$50000)	Sex
	Bank	Subscription	Age ($\geq$25)
NLP	CivilComments	Toxicity	Black
	HateXplain	Toxicity	Race
Vision	CelebA	Gender	Young
	UTKFace	Gender	Asian

4.2 Finite-Sample Bound for Mitigating Unfairness

Next, we provide a result bounding the difference in LF accuracy between group 0 points, $\mathbb{E}_{x\in\mathcal{Z}}[P(\lambda(x)=y)]$, and group 1 points transformed using our method, $\mathbb{E}_{x'\in\mathcal{X}}[P(\lambda(\hat{h}(x'))=y)]$. A tighter bound on this difference corresponds to better accuracy parity.

Theorem 4.2. *Set τ to be*

$$\tau=\max\left(\frac{\mathbf{r}(\Sigma_0)}{n_0},\frac{\mathbf{r}(\Sigma_1)}{n_1},\frac{t}{\min(n_0,n_1)},\frac{t^2}{\max(n_0,n_1)^2}\right),$$

and let C be a constant. Under the assumptions described above, when using Algorithm 1, for any $t>0$, we have that

$$|\mathbb{E}_{x\in\mathcal{Z}}[P(\lambda(x)=y)]-\mathbb{E}_{x'\in\mathcal{X}}[P(\lambda(\hat{h}(x'))=y)]|\leq 4\theta_0 C\sqrt{\tau\mathbf{r}(\Sigma_1)},$$

with probability $1-e^{-t}-\frac{1}{n_1}$.

Next we interpret Theorem 4.2. LF accuracy parity scales with $\max\left(1/\sqrt{n_1},1/\sqrt{n_2}\right)$. This does not present any additional difficulties compared to vanilla weak supervision—it is the same rate we need to learn LF accuracies. In other words, there is no sample complexity penalty for using our approach. Furthermore, LF accuracy parity scales inversely to $\max\left(\sqrt{\mathbf{r}(\Sigma_0)\mathbf{r}(\Sigma_1)},\mathbf{r}(\Sigma_1)\right)$. That is, when the distributions $P_0(x)$ or $P_1(x')$ have greater spread, it is more difficult to recover and so restore fair behavior.

Finally, we briefly comment on extensions. It is not hard to extend these results to setting with less strict assumptions. For example, we can take P to be a mixture of Gaussians. In this case, it is possible to combine algorithms for learning mixtures [8] with the approach we presented.

5 Experiments

The primary objective of our experiments is to validate that SBM improves fairness while often enhancing model performance as well. In real data experiments, we confirm that our methods work well with real-world fairness datasets (Section 5.1). In the synthetic experiments, we validate our theory claims in a fully controllable setting—showing that our method can achieve perfect fairness and performance recovery (Section 5.2). In addition, we show that our method is compatible with other fair ML techniques developed for fully supervised learning (Section 5.3). Finally, as a proof-of-concept, we present evidence that our method can actually improve weak supervision performance beyond fairness by creating and operating on artificial groups that have accuracy gaps (Section 5.4). Our code is available at https://github.com/SprocketLab/fair-ws.

5.1 Real data experiments

Claims Investigated In real data settings, we hypothesize that our methods can improve the performance and fairness of LFs, leading to better fairness and improved performance of the weak supervision end model.

Table 2: Tabular dataset result

Dataset	Methods	Acc (↑)	F1 (↑)	Δ_{DP} (↓)	Δ_{EO} (↓)
Adult	FS	0.824	0.564	0.216	0.331
	WS (Baseline)	0.717	0.587	0.475	0.325
	SBM (w/o OT)	0.720	**0.592**	0.439	0.273
	SBM (OT linear)	0.560	0.472	0.893	0.980
	SBM (OT sinkhorn)	**0.723**	0.590	**0.429**	**0.261**
Bank	FS	0.912	0.518	0.128	0.117
	WS (Baseline)	0.674	0.258	0.543	0.450
	SBM (w/o OT)	0.876	**0.550**	0.106	**0.064**
	SBM (OT linear)	**0.892**	0.304	**0.095**	0.124
	SBM (OT sinkhorn)	0.847	0.515	0.122	0.080

Table 3: Tabular dataset result (LIFT)

Dataset	Methods	Acc (↑)	F1 (↑)	Δ_{DP} (↓)	Δ_{EO} (↓)
Adult	FS	0.824	0.564	0.216	0.331
	WS (Baseline)	0.717	**0.584**	0.475	0.325
	SBM (w/o OT)	0.704	0.366	0.032	0.192
	SBM (OT linear)	0.700	0.520	0.015	**0.138**
	SBM (OT sinkhorn)	**0.782**	0.448	**0.000**	0.178
Bank	FS	0.912	0.518	0.128	0.117
	WS (Baseline)	0.674	0.258	0.543	0.450
	SBM (w/o OT)	0.698	0.255	**0.088**	0.137
	SBM (OT linear)	**0.892**	**0.305**	0.104	0.121
	SBM (OT sinkhorn)	0.698	0.080	0.109	**0.072**

Setup and Procedure We used 6 datasets in three different domains: tabular (Adult and Bank Marketing), NLP (CivilComments and HateXplain), and vision (CelebA and UTKFace). Their task and group variables are summarized in Table 1.

LFs are either heuristics or pretrained models. More details are included in Appendix E.3.

For the weak supervision pipeline, we followed a standard procedure. First, we generate weak labels from labeling functions in the training set. Secondly, we train the label model on weak labels. In this experiment, we used Snorkel [2] as the label model in weak supervision settings. Afterwards, we generate pseudolabels from the label model, train the end model on the these, and evaluate it on the test set. We used logistic regression as the end model.

The only difference between our method and the original weak supervision pipeline is a procedure to fix weak labels from each labeling function. As a sanity check, a fully supervised learning result (FS), which is the model performance trained on the true labels, is also provided. Crucially, however, *in weak supervision, we do not have such labels*, and therefore fully supervised learning is simply an upper bound to performance—and not a baseline.

We ran three variants of our method. *SBM (w/o OT)* is a 1-nearest neighbor mapping to another group without any transformation. *SBM (OT linear)* is a 1-nearest neighbor mapping with a linear map learned via optimal transport. *SBM (OT sinkhorn)* is a 1-nearest neighbor mapping with a Monge mapping learned via the Sinkhorn algorithm.

To see if our method can improve both fairness and performance, we measured the demographic parity gap (Δ_{DP}) and the equal opportunity gap (Δ_{EO}) as fairness metrics, and computed accuracy and F1 score as performance metrics as well. The definitions for these are restated below,

Table 4: NLP dataset result

Dataset	Methods	Acc (↑)	F1 (↑)	Δ_{DP} (↓)	Δ_{EO} (↓)
Civil	FS	0.893	0.251	0.083	0.091
	WS (Baseline)	0.854	**0.223**	0.560	0.546
	SBM (w/o OT)	0.879	0.068	0.048	0.047
	SBM (OT linear)	0.880	0.070	0.042	0.039
	SBM (OT sinkhorn)	**0.882**	0.047	**0.028**	**0.026**
Hate	FS	0.698	0.755	0.238	0.121
	WS (Baseline)	0.584	0.590	0.170	0.133
	SBM (w/o OT)	0.592	0.637	0.159	0.138
	SBM (OT linear)	**0.670**	0.606	0.120	0.101
	SBM (OT sinkhorn)	0.612	**0.687**	**0.072**	**0.037**

Table 5: Vision dataset result

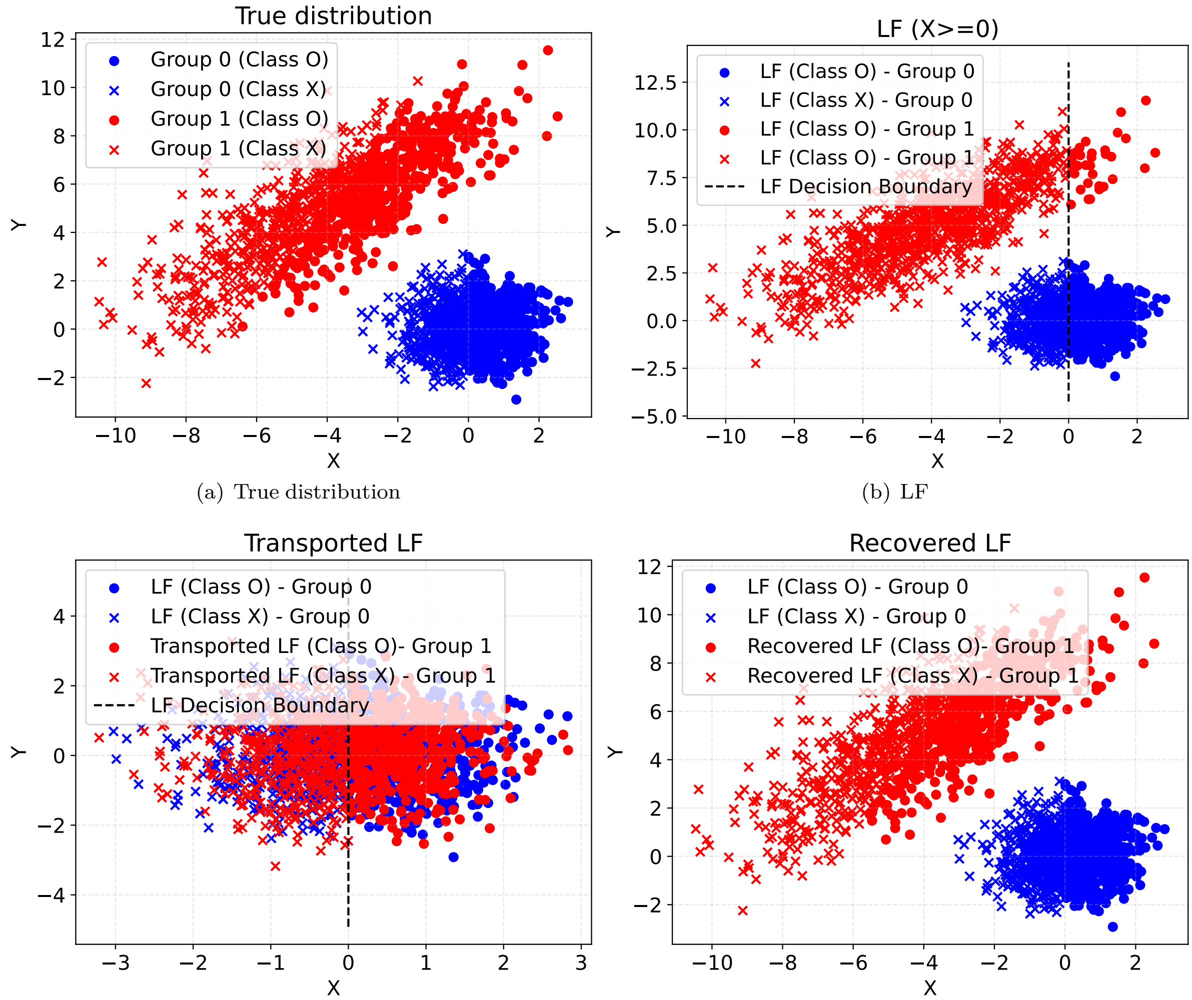
Dataset	Methods	Acc (↑)	F1 (↑)	Δ_{DP} (↓)	Δ_{EO} (↓)
CelebA	FS	0.897	0.913	0.307	0.125
	WS (Baseline)	0.866	0.879	0.308	0.193
	SBM (w/o OT)	0.870	0.883	0.309	0.192
	SBM (OT linear)	0.870	0.883	**0.306**	0.185
	SBM (OT sinkhorn)	**0.872**	**0.885**	**0.306**	**0.184**
UTKFace	FS	0.810	0.801	0.133	0.056
	WS (Baseline)	0.791	0.791	0.172	0.073
	SBM (w/o OT)	0.797	0.790	0.164	0.077
	SBM (OT linear)	0.800	0.793	0.135	0.043
	SBM (OT sinkhorn)	**0.804**	**0.798**	**0.130**	**0.041**

- Demographic parity gap
$$\Delta_{DP} := |P(\hat{Y}=1|A=1) - P(\hat{Y}=1|A=0)|.$$
- Equal opportunity gap
$$\Delta_{EO} := |P(\hat{Y}=1|Y=1,A=1) - P(\hat{Y}=1|Y=1,A=0)|.$$

Results The tabular dataset result is reported in Table 2. As expected, our method improves accuracy while reducing demographic parity gap and equal opportunity gap. However, we observed *SBM (OT linear)* critically fails at Adult dataset, contrary to what we anticipated. We suspected this originates in one-hot coded features, which might distort computing distances in the nearest neighbor search. To work around one-hot coded values in nearest neighbor search, we deployed LIFT [14], which encodes the input as natural language (e.g. "She/he is <race attribute>. She/he works for <working hour attribute> per week. ...") and embeds them with language models (LMs). We provide heuristic rules to convert feature columns into languages in Appendix E.2, and used BERT as the LM. The result is given in Table 3. While it sacrifices a small amount of accuracy, it substantially reduces the unfairness as expected.

The results for NLP datasets are provided in Table 4. In the CivilComments and HateXplain datasets, we observed our methods mitigate bias consistently, as we hoped. While our methods improve performance as well in the HateXplain dataset, enhancing other metrics in CivilComments results in drops in the F1 score. We believe that a highly unbalanced class setting ($P(Y=1) \approx 0.1$) is the cause of this result.

The results for vision datasets are given in Table 5. Though not as dramatic as other datasets since here the LFs are pretrained models, none of which are crucially biased, our methods can still improve accuracy and fairness. In



(a) True distribution

(b) LF

(c) Transported LF

(d) Recovered LF

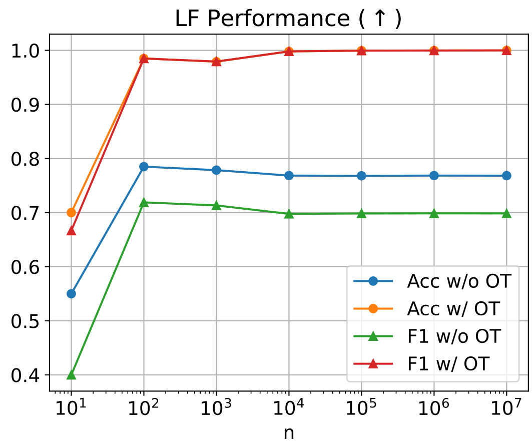
Figure 3: Synthetic dataset scenario. In (a), seemingly different data distributions from the two groups actually have perfect achievable fairness. However, the labeling function in (b) only works well in group 0, which leads to unfairness. Via OT (c), the input distribution can be matched and the LF applies to similar groups—one original, one recovered. As a result, LFs on group 1 works as well as on group 0 (d).

particular, our approach shows clear improvement over the baseline, which yields performance closer to the fully supervised learning setting while offering less bias.

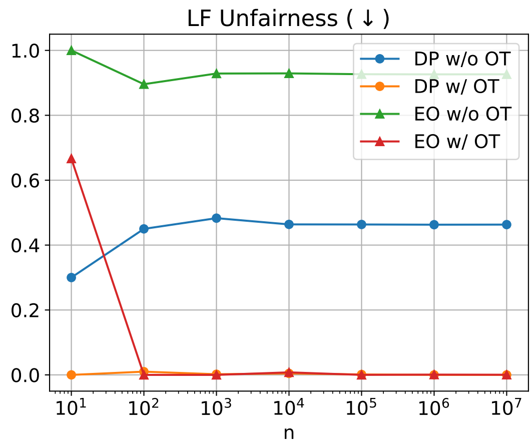
5.2 Synthetic experiment

Claim Investigated We hypothesized that our method can recover both fairness and accuracy (as a function of the number of samples available) by transporting the distribution of one group to another group when our theoretical assumptions are almost exactly satisfied. To show this, we generate unfair synthetic data and LFs and see if our method can remedy LF fairness and improve LF performance.

Setup and Procedure We generated a synthetic dataset that has perfect fairness as follows. First, n input features in $\mathbb{R}^2$ are sampled as $X_0 \sim \mathcal{N}(\mathbf{0}, I)$ for group 0, and labels Y_0 are set by $Y_0 = \mathbb{1}(X_0[0] \geq 0.5)$, i.e. 1 if the first dimension is positive or equal. Afterwards, n input features in $\mathbb{R}^2$ are sampled as $\tilde{X}_1 \sim \mathcal{N}(0, I)$ for group 1, and the labels are also



(a) LF Performance w/ OT



(b) LF Unfairness w/ OT

Figure 4: Synthetic experiment result. As OT recovers the transformation induces by the group attribute, the performance improves and the unfairness drops.

Table 6: Compatibility with other fair ML methods (HateXplain dataset)

	Acc (↑)	F1 (↑)	Δ_{DP} (↓)	Δ_{EO} (↓)
FS	0.698	0.755	0.238	0.121
WS (Baseline)	0.584	0.590	0.171	0.133
SBM (OT sinkhorn)	**0.612**	0.687	0.072	0.037
WS (Baseline) + Optimal Threshold (DP)	0.539	0.515	0.005	0.047
SBM (OT sinkhorn) + Optimal Threshold (DP)	0.607	**0.694**	**0.002**	**0.031**

set by $Y_1 = \mathbb{1}(\tilde{X}_1[0] \geq 0.5)$. Then, a linear transformation is applied to the input distribution: $X_1 = \Sigma \tilde{X}_1 + \mu$ where $\mu = \begin{bmatrix} -4 \\ 5 \end{bmatrix}$, $\Sigma = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, which is the distribution of group 1. Clearly, we can see that $X_1 = \Sigma X_0 + \mu \sim \mathcal{N}(\mu, \Sigma)$. Here we applied the same labeling function $\lambda(x) = \mathbb{1}(x[0] \geq 0)$, which is the same as the true label distribution in group 0. We apply our method (SBM OT linear) since our data model fits its basic assumption. Again, we evaluated the results by measuring accuracy, F1 score, Δ_{DP}, and Δ_{EO}. The setup and procedure are illustrated in Figure 3. We varied the number of samples n from 10^2 to 10^7

Results The result is reported in Figure 4. As we expected, we saw the accuracy and F1 score are consistently improved as the linear Monge map is recovered when n increases. Similarly, we observed that perfect fairness is achieved after a small number of samples (10^2) are obtained.

5.3 Compatibility with other fair ML methods

Claim Investigated Our method corrects LF bias at the individual LF level. We expect our methods can work independently, in a constructive way, with other fair ML methods developed for fully supervised learning settings.

Setup and Procedure We used the same real datasets, procedure, and metrics as before. We combined the optimal threshold method [25] with WS (baseline) and our approach, SBM (sinkhorn).

Results The results are shown in Table 6. As we expected, we saw the effect of optimal threshold method, which produces an accuracy-fairness (DP) tradeoff. This has the same effect upon our method. Thus, when optimal threshold

Table 7: Slice discovery (Domino) with SBM results

Dataset	Census		IMDb		Tennis	
Methods / metrics	Acc (↑)	F1 (↑)	Acc (↑)	F1 (↑)	Acc (↑)	F1 (↑)
FS	0.846	0.634	0.780	0.776	0.899	0.858
WS (Baseline)	0.602	0.512	0.753	0.755	0.813	0.763
SBM (w/o OT)	**0.732**	**0.546**	0.751	0.753	0.798	0.747
SBM (OT linear)	0.714	0.471	**0.756**	**0.760**	**0.852**	**0.799**
SBM (OT sinkhorn)	0.678	0.369	0.756	0.758	0.798	0.747

is applied to both, our method has better performance and fairness aligned with the result without optimal threshold. More experimental results with other real datasets and additional fair ML methods are reported in Appendix E.5.

5.4 Performance gain without group identities

Claim Investigated We postulated that even when group annotations are not given, still we can achieve performance improvements by finding subpopulations that have the accuracy gap and applying our method.

Setup and Procedure We used Census, IMDb, and Tennis dataset from the WRENCH benchmark [54], which is a well-known weak supervision benchmark but does not include any group information. Instead of the given group annotations, we generated group annotations by slice discovery [12, 16, 29, 49], which is a method to discover groups that share a common characteristic. To find groups with a large accuracy gap, we used Domino [16]. It discovers regions of the embedding space based on the accuracy of model. Since the WS setting does not allow access to true labels, we replaced true labels with pseudolabels obtained from the label model and model scores with label model probabilities. Snorkel [2] is used as the label model. We used the group information generated by the two discovered slices to apply our methods. We used the same weak supervision pipeline and metrics as in the other experiments, except for fairness, since group annotations are generated by accuracy.

Results The results can be seen in Table 7. As expected, even without known group divisions, we still observed improvements in accuracy and F1 score. These gains suggest that it is possible to further combine our approach with other principled methods for subpopulation discovery to substantially improve weak supervision in general settings.

6 Conclusion

Weak supervision has been successful in overcoming the manual labeling bottleneck for dataset construction. However, fairness in weak supervision has not been adequately studied. Our work has found that WS can easily induce additional bias due to unfair label functions. In order to address this issue, we have proposed a novel approach towards mitigating bias in LFs and further improving model performance. We have demonstrated the effectiveness of our approach using both synthetic and real datasets and have shown that it is compatible with traditional fair ML methods. We believe that our proposed technique can make weak supervision safer to apply in important societal settings and so encourages its wider adoption.

References

[1] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[2] Stephen H Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, et al. Snorkel drybell: A case study in deploying weak supervision at industrial scale. In *Proceedings of the 2019 International Conference on Management of Data*, pages 362–375, 2019.

[3] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. *Microsoft, Tech. Rep. MSR-TR-2020-32*, 2020.

[4] Emily Black, Samuel Yeom, and Matt Fredrikson. Fliptest: fairness testing via optimal transport. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 111–121, 2020.

[5] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pages 491–500, 2019.

[6] Maarten Buyl and Tijl De Bie. Optimal transport of classifiers to fairness. In *Advances in Neural Information Processing Systems*, 2022.

[7] Mayee F Chen, Daniel Y Fu, Dyah Adila, Michael Zhang, Frederic Sala, Kayvon Fatahalian, and Christopher Ré. Shoring up the foundations: Fusing model embeddings and weak supervision. In *Uncertainty in Artificial Intelligence*, pages 357–367. PMLR, 2022.

[8] Yongxin Chen, Tryphon T Georgiou, and Allen Tannenbaum. Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278, 2018.

[9] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[10] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied statistics*, pages 20–28, 1979.

[11] Mostafa Dehghani, Hamed Zamani, Aliaksei Severyn, Jaap Kamps, and W. Bruce Croft. Neural ranking models with weak supervision. In *Proceedings of the 40th International ACM SIGIR Conferenceon Research and Development in Information Retrieval*, 2017.

[12] Greg d'Eon, Jason d'Eon, James R Wright, and Kevin Leyton-Brown. The spotlight: A general method for discovering systematic errors in deep learning models. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1962–1981, 2022.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Tuan Dinh, Yuchen Zeng, Ruisu Zhang, Ziqian Lin, Michael Gira, Shashank Rajput, Jy-yong Sohn, Dimitris Papailiopoulos, and Kangwook Lee. Lift: Language-interfaced fine-tuning for non-language machine learning tasks. In *Advances in Neural Information Processing Systems*, 2022.

[15] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[16] Sabri Eyuboglu, Maya Varma, Khaled Saab, Jean-Benoit Delbrouck, Christopher Lee-Messer, Jared Dunnmon, James Zou, and Christopher Ré. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022.

[17] Rémi Flamary, Karim Lounici, and André Ferrari. Concentration bounds for linear monge mapping estimation and optimal transport domain adaptation. *arXiv preprint arXiv:1905.10155*, 2019.

[18] Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T.H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL `http://jmlr.org/papers/v22/20-451.html`.

[19] Daniel Y. Fu, Mayee F. Chen, Frederic Sala, Sarah M. Hooper, Kayvon Fatahalian, and Christopher Ré. Fast and three-rious: Speeding up weak supervision with triplet methods. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, 2020.

[20] Paula Gordaliza, Eustasio Del Barrio, Gamboa Fabrice, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *International Conference on Machine Learning*, pages 2357–2365. PMLR, 2019.

[21] Ozgur Guldogan, Yuchen Zeng, Jy-yong Sohn, Ramtin Pedarsani, and Kangwook Lee. Equal improvability: A new fairness notion considering the long-term impact. *arXiv preprint arXiv:2210.06732*, 2022.

[22] Sonal Gupta and Christopher D Manning. Improved pattern learning for bootstrapped entity extraction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 98–108, 2014.

[23] Vivek Gupta, Pegah Nokhiz, Chitradeep Dutta Roy, and Suresh Venkatasubramanian. Equalizing recourse across groups. *arXiv preprint arXiv:1909.03166*, 2019.

[24] Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020.

[25] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

[26] Hoda Heidari, Vedant Nanda, and Krishna P Gummadi. On the long-term impact of algorithmic decision policies: Effort unfairness and feature segregation through social learning. *arXiv preprint arXiv:1903.01209*, 2019.

[27] Wen Huan, Yongkai Wu, Lu Zhang, and Xintao Wu. Fairness through equality of effort. In *Companion Proceedings of the Web Conference 2020*, pages 743–751, 2020.

[28] David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems*, pages 1953–1961, 2011.

[29] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

[30] Martin Knott and Cyril S Smith. On the optimal mapping of distributions. *Journal of Optimization Theory and Applications*, 43:39–49, 1984.

[31] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR, 2021.

[32] Ron Kohavi et al. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd*, volume 96, pages 202–207, 1996.

[33] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

[34] Hunter Lang, Aravindan Vijayaraghavan, and David Sontag. Training subset selection for weak supervision. *arXiv preprint arXiv:2206.02914*, 2022.

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[36] Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. Hatexplain: A benchmark dataset for explainable hate speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14867–14875, 2021.

[37] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, accountability and transparency*, pages 107–118. PMLR, 2018.

[38] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[39] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[40] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[42] A. J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and C. Ré. Data programming: Creating large training sets, quickly. In *Proceedings of the 29th Conference on Neural Information Processing Systems (NIPS 2016)*, Barcelona, Spain, 2016.

[43] A. J. Ratner, B. Hancock, J. Dunnmon, F. Sala, S. Pandey, and C. Ré. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Honolulu, Hawaii, 2019.

[44] Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the 44th International Conference on Very Large Data Bases (VLDB)*, Rio de Janeiro, Brazil, 2018.

[45] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771, 2019.

[46] Changho Shin, Winfred Li, Harit Vishwakarma, Nicholas Carl Roberts, and Frederic Sala. Universalizing weak supervision. In *International Conference on Learning Representations (ICLR)*, 2022.

[47] Nian Si, Karthyek Murthy, Jose Blanchet, and Viet Anh Nguyen. Testing group fairness via optimal transport projections. In *International Conference on Machine Learning*, pages 9649–9659. PMLR, 2021.

[48] Chiappa Silvia, Jiang Ray, Stepleton Tom, Pacchiano Aldo, Jiang Heinrich, and Aslanides John. A general approach to fairness with optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3633–3640, 2020.

[49] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding failures of deep networks via robust feature extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12853–12862, 2021.

[50] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[51] Harit Vishwakarma and Frederic Sala. Lifting weak supervision to structured prediction. In *Advances in Neural Information Processing Systems*, 2022.

[52] Ziwei Wu and Jingrui He. Fairness-aware model-agnostic positive and unlabeled learning. In *ACM Conference on Fairness, Accountability, and Transparency (ACM FAccT)*, 2022.

[53] Xianli Zeng, Edgar Dobriban, and Guang Cheng. Bayes-optimal classifiers under group fairness. *arXiv preprint arXiv:2202.09724*, 2022.

[54] Jieyu Zhang, Yue Yu, Yinghao Li, Yujing Wang, Yaming Yang, Mao Yang, and Alexander Ratner. Wrench: A comprehensive benchmark for weak supervision. *arXiv preprint arXiv:2109.11377*, 2021.

[55] Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*, 2022.

[56] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.

The appendix contains additional details, proofs, and experimental results. The glossary contains a convenient reminder of our terminology (Appendix A). Appendix B provides more related works and discussion about the relationship between our work and related papers. In Appendix C, we describe the details of our algorithm and discuss their implementations. Appendix D provides the proofs of theorems that appeared in Section 4. Finally, we give more details and analysis of the experiments and provide additional experiment results.

A Glossary

The glossary is given in Table 8 below.

Symbol	Definition
$\mathcal{X}$	Feature space
$\mathcal{Y}$	Label metric space
$\mathcal{Z}$	Latent space of feature space
A	Group variable, assumed to have one of two values {0, 1} for simplicity
X_k	Inputs from group k.
X	Inputs from all groups $k \in \{1,...,l\}$
Y_k	True labels from group k
Y	True labels from all groups $k \in \{1,...,l\}$
P	Probability distribution in the latent space
P_k	Probability distribution in the group k
λ_k^j	Noisy labels of labeling function j from group k. If it is used with input (e.g. $\lambda_k^j(x)$), it denotes labeling function j from group k such that its outputs are noisy labels
λ^j	Noisy labels of labeling function j from all groups $k \in \{1,...,l\}$. If it is used with input (e.g. $\lambda^j(x)$), it denotes labeling function j such that its outputs are noisy labels
Λ_k	Collection of noisy labels from group k, $\Lambda_k = [\lambda_k^1,...,\lambda_k^m]$
Λ	Collection of noisy labels from all groups $k \in \{1,...,l\}$, $\Lambda = [\lambda^1,...,\lambda^m]$
g_k	$g_k : \mathcal{Z} \to \mathcal{X}$, k-th group transformation function
h_k	$h_k : \mathcal{X} \to \mathcal{Z}$ the inverse transformation of g_k, i.e. $h_k g_k(x) = x$ for $x \in \mathcal{Z}$
θ_y	Prior parameter for Y in label model
θ_j	Accuracy parameter for λ^j in label model
$\theta_{j,x}$	Accuracy parameter for x of λ^j in label model [7]
a_k^j	Accuracy of LF j in group k, $a_k^j = \mathbb{E}[[]\lambda_j Y \vert A = k]$
a^j	Accuracy of LF j, $a^j = \mathbb{E}[[]\lambda_j Y]$
$\hat{a}_k^j, \hat{a}^j$	Estimates of a_k^j, a^j
μ_k	Mean of features in group k
Σ_k	Covariance of features in group k
I	Identity transformation, i.e. $I(x) = x$
$tr(\Sigma)$	Trace of Σ
$\lambda_{\max}(\Sigma), \lambda_{\min}(\Sigma)$	Maximum, minimum values of Σ
$\mathbf{r}(\Sigma)$	Effective rank of Σ, i.e. $\mathbf{r}(\Sigma) = \frac{tr(\Sigma)}{\lambda_{\max}(\Sigma)}$

Table 8: Glossary of variables and symbols used in this paper.

B Extended related work

Weak supervision In weak supervision, we assume true label $Y \in \mathcal{Y}$ cannot be accessed, but labeling functions $\lambda^1, \lambda^2, \cdots, \lambda^m \in \mathcal{Y}$ which are noisy versions of true labels, are given. These weak label sources include code snippets expressing heuristics about Y, crowdworkers, external knowledge bases, pretrained models, etc [11, 22, 28, 38, 44].

Given $\lambda^1,\lambda^2,\cdots,\lambda^m$, WS often takes a two-step procedure to get an end model [10, 19, 42, 45, 46, 51]. The first step is to get fine-grained pseudolabels by modeling accuracies and correlations of label sources. The second step is training or fine-tuning the end model with pseudolabels to get better generalization than the label model. Not only our method can improve performance and fairness, but also it is compatible with the previous WS label models since it works at the label source level before modeling the label model.

Another related line of works is WS using embeddings of inputs [7, 34]. Lang et al. [34] uses embeddings for subset selection, where high-confidence subsets are selected based on the proximity to the same pseudolabels. Chen et al. [7] exploits embedding to estimate local accuracy and improve coverage. Similarly, our method uses embedding to improve fairness by mapping points in different groups on the embedded space. Embedded space by a pre-trained model typically offers better distance metrics than the input space, which provides an advantage when the relationship between data points can be useful.

Fairness in machine learning Fairness in machine learning is an active research area to detect and address biases in ML algorithms. There have been many suggested fairness (bias) notions and solutions for them [15, 21, 23, 25–27, 33]. Popular approaches include adopting constrained optimization or regularization with fairness metrics [1, 15, 23, 25–27] and postprocessing model outputs to guarantee some fairness notions [37, 53]. While they have been successful in reducing bias, those methods often have a fundamental tradeoff between accuracy and fairness.

A more recent line of works using Optimal Transport (OT) [4, 6, 20, 47, 48] has shown that OT can be used to improve fairness without tradeoff by uniformizing distributions in different groups. Our method improves fairness and performance spontaneously by utilizing OT. However, OT-based methods can suffer from additional errors induced by optimal transport mapping. In our method, OT-induced errors that occurred at the label sources can be covered by the label model, since worsened labeling functions would be down-weighted in label model fitting. Our method has a limitation in the point that fairness scores after applying our methods are unpredictable in advance, while methods having fairness-accuracy tradeoff typically come with tunable parameters regarding the tradeoff. However, as we argued in Section 5, our method is compatible with other methods, thus it has the optionality to control the fairness of the end model by adopting other methods as well.

C Algorithm details

Algorithm 2: Estimate Accuracy (Triplet) [19]

Parameters: Weak label values $\lambda^1,...,\lambda^m$
for $i,j,k \in \{1,2,...,m\}$ $(i \neq j \neq k)$ **do**
  $|\hat{a}^i| \leftarrow \sqrt{\hat{\mathbb{E}}[\lambda^i\lambda^j]\hat{\mathbb{E}}[\lambda^i\lambda^k]/\hat{\mathbb{E}}[\lambda^j\lambda^k]}$
  $|\hat{a}^j| \leftarrow \sqrt{\hat{\mathbb{E}}[\lambda^i\lambda^j]\hat{\mathbb{E}}[\lambda^j\lambda^k]/\hat{\mathbb{E}}[\lambda^i\lambda^k]}$
  $|\hat{a}^k| \leftarrow \sqrt{\hat{\mathbb{E}}[\lambda^i\lambda^k]\hat{\mathbb{E}}[\lambda^j\lambda^k]/\hat{\mathbb{E}}[\lambda^i\lambda^j]}$
end for
return ResolveSign $(\hat{a}^i)$ $\forall i \in \{1,2,...,m\}$

In this appendix section, we discuss the details of the algorithms we used. The overall WS pipeline including SBM is illustrated in Figure 5. Our method is placed at the first step, as a refinement of noisy labels. In this step, we improve the noisy labels by transporting the low accuracy group to the high accuracy group. To identify low accuracy and high accuracy groups, we estimate accuracy using Algorithm 2. After estimating accuracies, we transport the low accuracy group to the high accuracy group and get refined labels for the low accuracy group by Algorithm 3. Linear OT (Optimal Transport) is used to estimate a mapping. Sinkhorn OT approximates optimal coupling using Sinkhorn iteration. After obtaining the optimal coupling, data points from the low accuracy group are mapped to the coupled data points in the high accuracy group. If OT type is not given, data points from the low accuracy

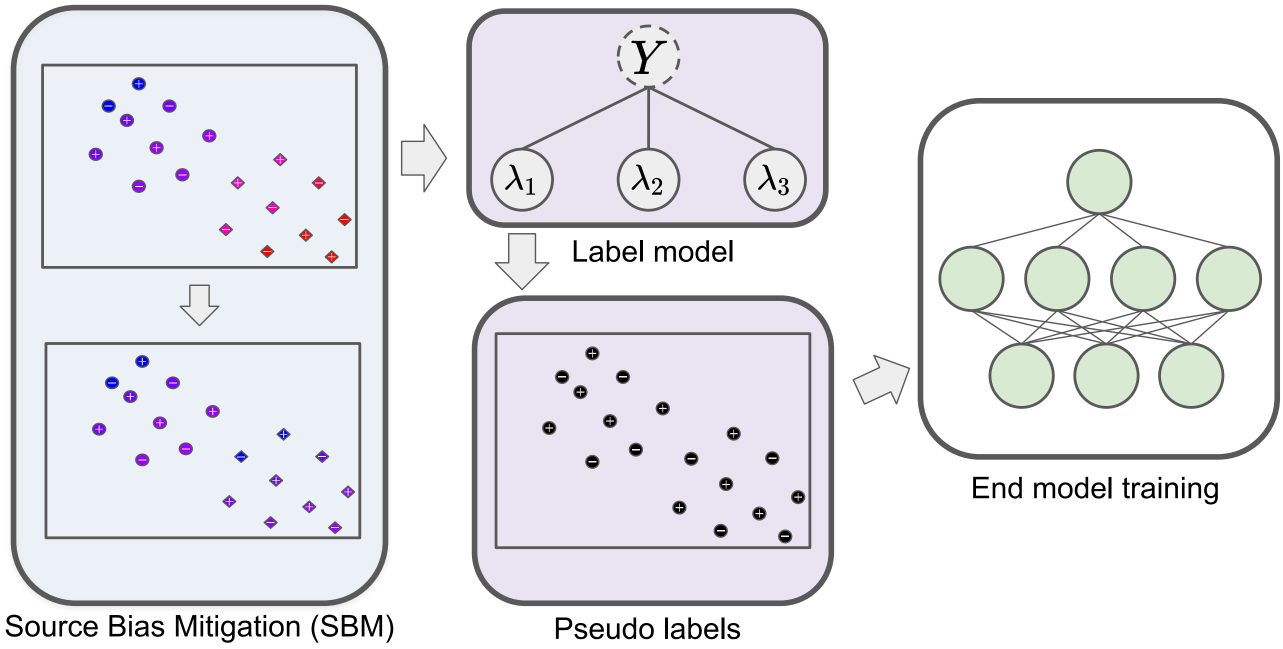


Figure 5: Overall WS pipeline involving SBM

Algorithm 3: TRANSPORT

Parameters: Source input X_{src}, destination input X_{dst}, source weak label values λ_{src}, destination weak label values λ_{dst}, Optimal Transport type O, the number of nearest neighbors $k=1$

if OT type is linear **then**

$\tilde{X}_{src} \leftarrow \text{LINEAROT}(X_{src}, X_{dst})$ [30]

else if OT type is sinkhorn **then**

$\tilde{X}_{src} \leftarrow \text{SINKHORNOT}(X_{src}, X_{dst})$ [9]

else

$\tilde{X}_{src} \leftarrow X_{src}$

end if

$\tilde{\lambda}_{src} \leftarrow kNN(\tilde{X}_{src}, X_{dst}, \lambda_{dst}, k)$

return $\tilde{\lambda}_{src}$

Algorithm 4: LINEAROT[30]

Parameters: Source input X_{src}, destination input X_{dst}

$\mu_s, \mu_t \leftarrow \text{MEAN}(X_{src}), \text{MEAN}(X_{dst})$

$\Sigma_s, \Sigma_t \leftarrow \text{Cov}(X_{src}), \text{Cov}(X_{dst})$

$A \leftarrow \Sigma_s^{-1/2}(\Sigma_s^{1/2}\Sigma_t\Sigma_s^{1/2})^{1/2}\Sigma_s^{-1/2}$

$b \leftarrow \mu_t - A\mu_s$

$B \leftarrow [b \cdots b]$

$\tilde{X}_{src} \leftarrow X_{src}A^T + B^T$

return $\tilde{X}_{src}$

---

group are mapped to the 1-nearest neighbor in the high accuracy group. After obtaining mapped points, weak labels associated with mapped points in the target group are used for the data points from the low accuracy group.

Algorithm 5: SINKHORNOT[9]

Parameters: Source input X_{src}, destination input X_{dst}, Entropic regularization parameter η=1, max_iter=10
$M = \text{PAIRWISEDISTANCE}(X_{src}, X_{dst})$
$n_{src}, n_{dst} = size(X_{src}), size(X_{dst})$
$a, b = P(X_{src}), P(X_{dst})$
$u, K, v = Sinkhorn(M, \eta, a, diag(b))$ // **Algorithm 1** of [9]
$T = uKv^T$
$\tilde{X}_{src} \leftarrow TX_{dst}$
return $\tilde{X}_{src}$

The next step is a standard weak supervision pipeline step – training a label model and running inference to obtain pseudolabels. Finally, we train the end model. We used the off-the-shelf label model Snorkel [2] for the label model and logistic regression as the end model in all experiments. For the optimal transport algorithm implementation, we used the Python POT package [18].

D Theory details

Accuracy is defined as $P(\lambda(x) = y) = P(\lambda(x) = 1, y = 1) + P(\lambda(x) = -1, y = -1)$. For ease of notation, below we set $\phi(x, x^{\text{center}_0}) = (1 + ||x - x^{\text{center}_0}||)^{-1}$.

D.1 Proof of Theorem 4.1

Theorem 4.1. *Let $g_1^{(k)}$ be an arbitrary sequence of functions such that $\lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{X})}[||x' - x^{center_0}||] \to \infty$. Suppose our assumptions above our met; in particular, that the label y is independent of the observed features $x = I(z)$ or $x' = g_1^{(k)}(z), \forall k$, conditioned on the latent features z. Then,*

$$\lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}[P(\lambda(x') = y)] = \frac{1}{2},$$

which corresponds to random guessing.

Proof of Theorem 4.1. Because $\lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}[||x' - x^{\text{center}_0}||] \to \infty$, we have

$$\lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}[\phi(x', x^{\text{center}_0})] = 0$$

and because $Z = \sum_{i \in \{0,1\}} \sum_{j \in \{0,1\}} P(\lambda(x') = i, y = j)$,

$$\begin{aligned}
\lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}[P(\lambda(x') = y)] &= \lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}[P(\lambda(x') = 1, y = 1) + P(\lambda(x') = -1, y = -1)] \\
&= \lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}\left[\frac{1}{Z}\exp(\theta_0(1)(1)\phi(x', x^{\text{center}_0})) + \frac{1}{Z}\exp(\theta_0(-1)(-1)\phi(x', x^{\text{center}_0}))\right] \\
&= \lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}\left[\frac{2}{Z}\exp(\theta_0\phi(x', x^{\text{center}_0}))\right] \\
&= \lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}\left[\frac{2\exp(\theta_0\phi(x', x^{\text{center}_0}))}{2(\exp(\theta_0\phi(x', x^{\text{center}_0})) + \exp(-\theta_0\phi(x', x^{\text{center}_0})))}\right] \\
&= \lim_{k\to\infty} \mathbb{E}_{x' \in g_1^{(k)}(\mathcal{Z})}\left[\frac{\exp(\theta_0\phi(x', x^{\text{center}_0}))}{\exp(\theta_0\phi(x', x^{\text{center}_0})) + \exp(-\theta_0\phi(x', x^{\text{center}_0}))}\right] \\
&= \frac{1}{2}.
\end{aligned}$$

□

D.2 Proof of Theorem 4.2

Theorem 4.2. *Let* $\tau = \max\left(\frac{\mathbf{r}(\Sigma_0)}{n_0}, \frac{\mathbf{r}(\Sigma_1)}{n_1}, \frac{t}{\min(n_0,n_1)}, \frac{t^2}{\max(n_0,n_1)^2}\right)$ *and* C *be a constant. Using Algorithm 1, for any* $t > 0$*, we bound the difference*

$$
\begin{aligned}
|\mathbb{E}_{x\in\mathcal{Z}}[P(\lambda(x)=y)] - \mathbb{E}_{x'\in\mathcal{X}}[P(\lambda(\hat{h}(x'))=y)]|& \\
\leq 4\theta_0 C\sqrt{\tau\mathbf{r}(\Sigma_1)}&.
\end{aligned}
$$

with probability $1-e^{-t}-\frac{1}{n_1}$.

We use the result from [17], which bounds the difference of the true h and empirical $\hat{h}$ Monge estimators between two distributions $P_0(x)$ and $P_1(x')$. Let $\mathbf{r}(\Sigma)$, $\lambda_{\min}(\Sigma)$ and $\lambda_{\max}(\Sigma)$ denote the effective rank, minimum and maximum eigenvalues of matrix Σ respectively. Then,

Lemma D.1 ([17])**.** *Let* $P_0(x)$ *and* $P_1(x')$ *be sub-Gaussian distributions on* $\mathcal{X} = \mathbb{R}^d$ *with expectations* μ_0, μ_1 *and positive-definite covariance operators* Σ_0, Σ_1 *respectively. We observe* n_0 *points from the distribution* $P_0(x)$ *and* n_1 *points from the distribution* $P_1(x')$*. We assume that*

$$
c_1 < \min_{j\in\{0,1\}}\{\lambda_{\min}(\Sigma_j)\} \leq \max_{j\in\{0,1\}}\{\lambda_{\max}(\Sigma_j)\} \leq c_2,
$$

for fixed constants $0 < c_1 \leq c_2 < \infty$*. Further, we assume* $n_0 \geq c\mathbf{r}(\Sigma_0)$ *and* $n_1 \geq d$ *for sufficiently large constant* $c > 0$*. Then, for any* $t > 0$*, we have with probability at least* $1-e^{-t}-\frac{1}{n_1}$ *that*

$$
\mathbb{E}_{x'\in\mathcal{X}}[||h(x') - \hat{h}(x')||] \leq C\sqrt{\tau\mathbf{r}(\Sigma_1)},
$$

where C *is a constant independent of* $n_0, n_1, d, \mathbf{r}(\Sigma_0), \mathbf{r}(\Sigma_1)$ *and* $\tau = \max\left(\frac{\mathbf{r}(\Sigma_0)}{n_0}, \frac{\mathbf{r}(\Sigma_1)}{n_1}, \frac{t}{\min(n_0,n_1)}, \frac{t^2}{\max(n_0,n_1)^2}\right)$.

We will also use the following:

Lemma D.2. *The probability* $P(\lambda(x)=y)$ *is L-Lipschitz with respect to* $x \in \mathcal{X}$*. Specifically,* $\forall x_1, x_2 \in \mathcal{X}$*,*

$$
|P(\lambda(x_1)=y) - P(\lambda(x_2)=y)| \leq 4\theta_0||x_1 - x_2||.
$$

Proof. We will demonstrate that, because $||\nabla_x P(\lambda(x)=y)||$ is bounded above by $4\theta_0$, $P(\lambda(x)=y)$ must be $4\theta_0$-Lipschitsz with respect to x. First,

$$
\begin{aligned}
P(\lambda(x)=y) &= \frac{\exp(\theta_0\phi(x, x^{\text{center}_0}))}{\exp(\theta_0\phi(x, x^{\text{center}_0})) + \exp(-\theta_0\phi(x, x^{\text{center}_0}))} \\
&= \frac{1}{1+\exp(-2\theta_0\phi(x, x^{\text{center}_0}))} \\
&= \sigma(2\theta_0\phi(x, x^{\text{center}_0})),
\end{aligned}
$$

where σ denotes the sigmoid function. Note that $\frac{d}{du}\sigma(u) = \frac{\exp(-u)}{(1+\exp(-u))^2}$ for $u \in \mathbb{R}$ and that $\nabla_x[2\theta_0\phi(x, x^{\text{center}_0})] = 2\theta_0\nabla_x\left[\frac{1}{1+||x-x^{\text{center}_0}||}\right] = 2\theta_0\frac{\nabla_x[||x-x^{\text{center}_0}||]}{(1+||x-x^{\text{center}_0}||)^2}$. Further, note that $\nabla_x[||x-x^{\text{center}_0}||] = \frac{x-x^{\text{center}_0}}{||x-x^{\text{center}_0}||}$ because we assume Euclidean norm. Thus,

$$
\begin{aligned}
||\nabla_x P(\lambda(x)=y)|| &= ||\nabla_x \sigma(2\theta_0\phi(x,x^{\text{center}_0}))|| \\
&= \left|\left|2\theta_0 \frac{exp(-2\theta_0\phi(x,x^{\text{center}_0}))}{(1+exp(-2\theta_0\phi(x,x^{\text{center}_0})))^2(1+||x-x^{\text{center}_0}||)^2} \cdot \nabla_x[||x-x^{\text{center}_0}||]\right|\right| \\
&= \left|\left|2\theta_0 \frac{exp(-2\theta_0\phi(x,x^{\text{center}_0}))}{(1+exp(-2\theta_0\phi(x,x^{\text{center}_0})))^2}\phi(x,x^{\text{center}_0})^2 \cdot \frac{x-x^{\text{center}_0}}{||x-x^{\text{center}_0}||}\right|\right| \\
&= \left|\left|2\theta_0\sigma(2\theta_0\phi(x,x^{\text{center}_0}))(1-\sigma(2\theta_0\phi(x,x^{\text{center}_0})))\phi(x,x^{\text{center}_0})^2 \cdot \frac{x-x^{\text{center}_0}}{||x-x^{\text{center}_0}||}\right|\right| \\
&< 2\theta_0\left|\left|1\cdot(1-(-1))\phi(x,x^{\text{center}_0})^2 \cdot \frac{x-x^{\text{center}_0}}{||x-x^{\text{center}_0}||}\right|\right| \\
&= 4\theta_0\left|\left|\phi(x,x^{\text{center}_0})^2 \cdot 1\right|\right| \\
&\leq 4\theta_0.
\end{aligned}
$$

Thus, $||\nabla_x P(\lambda(x)=y)||$ is bounded above by $4\theta_0$. We now use this fact to demonstrate that $P(\lambda(x)=y)$ is $4\theta_0$-Lipschitz with respect to x.

For $0 \leq v \leq 1$, let $s(v) = x_1 + (x_2 - x_1)v$. For x between x_1 and x_2 inclusive and for v such that $s(v) = x$, because $||\nabla_x P(\lambda(x)=y)|| = |\frac{d}{dv}P(\lambda(s(v))=y)| \leq 4\theta_0$, we have

$$
\begin{aligned}
|P(\lambda(x_1)=y) - P(\lambda(x_2)=y)| &= \left|\int_{v=0}^{1}\left[\frac{d}{dv}P(\lambda(s(v))=y)\right]||s(v)||dv\right| \\
&\leq \int_{v=0}^{1}\left|\frac{d}{dv}P(\lambda(s(v))=y)\right|||s(v)||dv \\
&\leq \int_{v=0}^{1} 4\theta_0||s(v)||dv \\
&= 4\theta_0||x_1-x_2||.
\end{aligned}
$$

□

Proof of Theorem 4.2. Now we are ready to complete our proof. We have,

$$
\begin{aligned}
&|\mathbb{E}_{x\in\mathcal{Z}}[P(\lambda(x)=y)] - \mathbb{E}_{x'\in\mathcal{X}}[P(\lambda(\hat{h}(x'))=y)]| \\
&\quad\leq \mathbb{E}_{z\in\mathcal{Z}}[|P(\lambda(I(z))=y) - P(\lambda(\hat{h}(g_1(z)))=y|] \\
&\quad= \mathbb{E}_{z\in\mathcal{Z}}[|P(\lambda(h(g_1(z)))=y) - P(\lambda(\hat{h}(g_1(z)))=y|] \\
&\quad\leq \mathbb{E}_{z\in\mathcal{Z}}[4\theta_0||h(g_1(z)) - \hat{h}(g_1(z))||] \text{ (by Lemma D.2)} \\
&\quad= \mathbb{E}_{x'\in\mathcal{X}}[4\theta_0||h(x') - \hat{h}(x')||] \\
&\quad\leq 4\theta_0 C\sqrt{\tau\mathbf{r}(\Sigma_1)},
\end{aligned}
$$

where the last line holds with probability at least $1 - e^{-t} - \frac{1}{n_1}$ by Lemma D.1.

□

E Experiment details

E.1 Dataset details

Adult The adult dataset [32] has information about the annual income of people and their demographics. The classification task is to predict whether the annual income of a person exceeds $50,000 based on demographic

	$P(Y=1\|A=1)$	$P(Y=1\|A=0)$	Δ_{DP}
Adult	0.3038	0.1093	0.1945
Bank	0.1093	0.2399	0.1306
CivilComments	0.3154	0.1057	0.2097
HateXplain	0.8040	0.4672	0.3368
CelebA	0.3244	0.6590	0.3346
UTKFace	0.4856	0.4610	0.0246

Table 9: Distribution and inherent bias in datasets

features. Demographic features include age, work class, education, marital status, occupation, relationship, race, capital gain and loss, work hours per week, and native country. The group variable is whether age is greater than 25 or not. The training data has 32561 examples, and the test data has 16281 examples.

BankMarketing The bank marketing dataset [39] was collected during the direct marketing campaigns of a Portuguese banking institution from 2008 to 2013. Features consist of demographic attributes, financial attributes, and the attributes related to the interaction with the campaign. The task is to classify whether a client will make a deposit subscription or not. The group variable is, whether age is greater than 25 or not. The total number of instances is 41188, where the training dataset has 28831 rows and the test dataset has 12357 rows.

CivilComments The CivilComments dataset [5] is constructed from data collected in Jigsaw online conversattion platform. We used the dataset downloaded from [31]. Features are extracted using BERT [13] embeddings from comments, and the task is to classify whether a given comment is toxic or not. This dataset has annotations about the content type of comments, e.g. race, age, religion, etc. We chose “black” as the group variable. The total number of instances is 402820, where the training dataset has 269038 rows and the test dataset has 133782 rows.

HateXplain The HateXplain dataset [36] is comments collected from Twitter and Gab and labeled through Amazon Mechanical Turk. Features are extracted using BERT embeddings from comments. The task is to classify whether a given comment is toxic or not. This dataset has three label types - hate, offensive, or normal. We used hate and offensive label types as “toxic” label, that is, if one of two labels types is 1 for a row, the row has label 1. The dataset has annotations about the target communities - race, religion, gender, LGBTQ. We used the race group as the group variable. The total number of instances is 17281, where the training set has 15360 rows and the test set has 1921 rows.

CelebA The CelebA dataset [35] is the image datasets that has images of faces with various attributes. Features are extracted using CLIP[41] embeddings from images. We tried gender classification task from this dataset—various other tasks can be constructed. We used “Young” as the group variable. The total number of instances is 202599, where the training set has 162770 rows and the test set has 39829 rows.

UTKFace UTKFace dataset [56] consists of faces images in the wild with annotations of age, gender, and ethnicity. Features are extracted using CLIP embeddings from images. This dataset has attributes of age, gender, race and we chose gender as label (Male:0, Female: 1), solving gender classification. We used the race as the group variable, encoding “Asian” group as 1, other race groups as 0. The total number of instances is 23705, where the training set has 18964 rows and the test set has 4741 rows.

E.2 Application of LIFT

To use smooth embeddings in the transport step of tabular datasets (adult and bank marketing), we convert tabular data rows into text sentences and then get BERT embeddings.

Adult First we decided r.pronoun for each row r from <r.sex> and then generated sentences with the following template. "<r.pronoun> is in her <r.age_range>s. <r.pronoun> is from <r.nationality>. <r.pronoun> works in <r.workclass>. Specifically, <r.pronoun> has a <r.job> job. <r.pronoun> works <r.working_hour> hours per week. <r.pronoun> total education year is <r.education_years>. <r.pronoun> is <r.marital_status>. <r.pronoun> is <r.race>".

An example sentence is "She is in her 20s. She is from United-States. She works in private sector. Specifically, she has a sales job. She works 30 hours per week. Her total education year is 6 years. She is married. She is White."

Bank marketing Similarly, text sentences for each row r are generated using the following template. "This person is <r.age> years old. This person works in a <r.job> job. This person is <r.marital_status>. This person has a <r.education_degree> degree. This person <if r.has_housing_loan "has" else "doesn't have"> a housing loan. This person <if r.has_personal_loan "has" else "doesn't have"> a personal loan. THis person's contact is <r.contact_type>. The last contact was on <r.last_contact_weekday> in <r.last_contact_month>. The call lasted for <r.duration> seconds. <r.campaign> times of contacts performed during this campaign. Before this campaign, <r.previous> times of contacts have been performed for this person. The employment variation rate of this person is <r.emp_var_rate>%. The consumer price index of this person is <r.cons_price_idx>%. The consumer confidence index of this person is <r.cons_conf_idx>%. The euribor 3 month rate of this person is <r.euribor3m>. The number of employees is <r.nr_employed>".

An example sentence is "This person is 37 years old. This person works in a management job. This person is married. This person has a university degree. This person doesn't have a housing loan. This person has a personal loan. This person's contact is a cellular. The last contact was on a Thursday in May. The call lasted for 195 seconds. 1 times of contacts performed during this campaign. Before this campaign, 0 times of contacts have been performed for this person. The employment variation rate of this person is -1.8%. The consumer price index of this person is 92.893%. The consumer confidence index of this person is -46.2%. The euribor 3 month rate of this person is 1.327%. The number of employees is 5099."

E.3 LF details

In this subsection, we describe the labeling functions used in the experiments. The performances of labeling functions, including our SBM results, are reported in Table 10 $\sim$ 21.

Adult We generated heuristic labeling functions based on the features as follows.

- LF 1 (age LF): True if the age is between 30 and 60. False otherwise.
- LF 2 (education LF): True if the person has a Bachelor, Master, PhD degree. False otherwise
- LF 3 (marital LF): True if marital status is married. False otherwise.
- LF 4 (relationship LF): True if relationship status is Wife, Own-child, or Husband, False otherwise.
- LF 5 (capital LF): True if the capital gain is >5000. False otherwise.
- LF 6 (race LF): True if Asian-Pac-Islander or race-Other. False otherwise.
- LF 7 (country LF): True if Germany, Japan, Greece, or China. False otherwise.
- LF 8 (workclass LF): True if the workclass is Self employed, federal government, local government, or state government. False otherwise.
- LF 9 (occupation LF): True if the occupation is sales, executive managerial, professional, or machine operator. False otherwise.

Bank marketing Similar to Adult dataset, we generated heuristic labeling functions based on the features as follows.

- LF 1 (loan LF): True if the person has loan previously.False otherwise.

- LF 2 (previous contact LF): True if the previous contact number is >1.1. False otherwise.
- LF 3 (duration LF): True if the duration of bank marketing phone call is > 6 min. False otherwise.
- LF 4 (marital LF): True if marital status is single. False otherwise.
- LF 5 (previous outcome LF): True if the previous campaign was successful. False otherwise.
- LF 6 (education LF): True if the education level is university degree or professional course taken. False otherwise.

CivilComments We generated heuristic labeling functions based on the inclusion of specific word lists. If a comment has a word that is included in the word list of LF, it gets True for that LF.

- LF 1 (sex LF): ["man", "men", "woman", "women", "male", "female", "guy", "boy", "girl", "daughter", "sex", "gender", "husband", "wife", "father", "mother", "rape", "mr.", "feminist", "feminism", "pregnant", "misogy", "pussy", "penis", "vagina", "butt", "dick"]
- LF 2 (LGBTQ LF): ["gay", "lesbian", "transgender", "homosexual", "homophobic", "heterosexual", "anti-gay", "same-sex", "bisexual", "biological"]
- LF 3 (religion LF): ["muslim", "catholic", "church", "christian", "god", "jesus", "christ", "jew", "islam", "israel", "bible", "bishop", "gospel", "clergy", "protestant", "islam"]
- LF 4 (race LF): ["white", "black", "racist", "trump", "supremacist", "american", "canada", "kkk", "nazi", "facist", "african", "non-white", "discrimination", "hate", "neo-nazi", "asia", "china", "chinese"]
- LF 5 (toxic LF): ["crap", "damn", "bitch", "ass", "fuck", "bullshit", "hell", "jerk"]
- LF 6 (threat LF): ["shoot", "kill", "shot", "burn", "stab", "murder", "gun", "fire", "rape", "punch", "hurt", "hunt", "bullet", "hammer"]
- LF 7 (insult LF): ["stupid", "idiot", "dumb", "liar", "poor", "disgusting", "moron", "nasty", "lack", "brain", "incompetent", "sociopath"]

HateXplain We used heuristic the same labeling functions with CivilComments. However, their performance was close to random guess (accuracy close to 0.5), so we added 5 pretrained model LFs from detoxify [24] repository. We used models listed as original, unbiased, multilingual, original-small, unbiased-small.

CelebA We used models pretrained from other datasets as LFs.

- LF 1: ResNet-18 fine-tuned on gender classification dataset [1]
- LF 2: ResNet-34 fine-tuned on FairFace dataset
- LF 3: ResNet-34 fine-tuned on UTKFace dataset
- LF 4: ResNet-34 fine-tuned on UTKFace dataset (White only)
- LF 5: ResNet-34 fine-tuned on UTKFace dataset (non-White only)
- LF 6: ResNet-34 fine-tuned on UTKFace dataset (age ≥ 50 only)
- LF 7: ResNet-34 fine-tuned on UTKFace dataset (age < 50 only)

UTKFace We used models pretrained from other datasets as LFs.

- LF 1: ResNet-18 fine-tuned on gender classification dataset
- LF 2: ResNet-34 fine-tuned on gender classification dataset
- LF 3: ResNet-34 fine-tuned on CelebA dataset
- LF 4: ResNet-34 fine-tuned on CelebA dataset (attractive only)

[1] https://www.kaggle.com/datasets/cashutosh/gender-classification-dataset

Table 10: Tabular dataset raw LF performance

Dataset	LF	Acc	F1	Δ_{DP}	Δ_{EO}
	LF 1 (age LF)	0.549	0.476	0.100	0.023
	LF 2 (education LF)	0.743	0.455	0.033	0.044
Adult	LF 3 (marital LF)	0.699	0.579	0.447	0.241
	LF 4 (relationship LF)	0.564	0.486	0.381	0.243
	LF 5 (capital LF)	0.800	0.315	0.035	0.019
	LF 6 (race LF)	0.737	0.066	0.003	0.004
	LF 7 (country LF)	0.756	0.024	0.001	0.004
	LF 8 (workclass LF)	0.678	0.399	0.066	0.003
	LF 9 (occupation LF)	0.644	0.466	0.013	0.012
	LF 1 (loan LF)	0.769	0.124	0.004	0.013
	LF 2 (previous contact LF)	0.888	0.187	0.055	0.068
Bank Marketing	LF 3 (duration LF)	0.815	0.415	0.019	0.125
	LF 4 (marital LF):	0.682	0.197	0.602	0.643
	LF 5 (previous outcome LF)	0.898	0.301	0.052	0.062
	LF 6 (education LF)	0.575	0.206	0.183	0.219

- LF 5: ResNet-34 fine-tuned on CelebA dataset (non-attractive only)
- LF 6: ResNet-34 fine-tuned on CelebA dataset (young only)
- LF 7: ResNet-34 fine-tuned on CelebA dataset (non-young only)
- LF 8: ResNet-34 fine-tuned on CelebA dataset (unfair sampling)

Table 11: Tabular dataset SBM (w/o OT) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
Adult	LF 1	0.597 (0.048)	0.594 (0.118)	0.119 (0.019)	0.106 (0.083)
	LF 2	0.601 (-0.142)	0.346 (-0.109)	0.035 (0.002)	0.019 (-0.025)
	LF 3	0.998 (0.299)	0.998 (0.419)	0.303 (-0.144)	0.001 (-0.240)
	LF 4	0.719 (0.155)	0.723 (0.237)	0.304 (-0.077)	0.091 (-0.152)
	LF 5	0.649 (-0.151)	0.183 (-0.132)	0.035 (0.000)	0.023 (0.004)
	LF 6	0.615 (-0.122)	0.082 (0.016)	0.003 (0.000)	0.029 (0.025)
	LF 7	0.621 (-0.135)	0.023 (-0.001)	0.001 (0.000)	0.005 (0.001)
	LF 8	0.612 (-0.066)	0.318 (-0.021)	0.019 (-0.047)	0.027 (0.024)
	LF 9	0.565 (-0.079)	0.460 (-0.006)	0.013 (0.000)	0.026 (0.014)
Adult (LIFT)	LF 1	0.506 (-0.043)	0.415 (-0.061)	0.309 (0.209)	0.198 (0.175)
	LF 2	0.861 (0.118)	0.691 (0.236)	0.080 (0.047)	0.126 (0.082)
	LF 3	0.691 (-0.008)	0.188 (-0.391)	0.130 (-0.317)	0.186 (-0.055)
	LF 4	0.391 (-0.173)	0.307 (-0.179)	0.455 (0.074)	0.314 (0.071)
	LF 5	0.725 (-0.075)	0.117 (-0.198)	0.014 (-0.021)	0.029 (0.010)
	LF 6	0.703 (-0.034)	0.109 (0.043)	0.003 (0.000)	0.004 (0.000)
	LF 7	0.708 (-0.048)	0.036 (0.012)	0.001 (0.000)	0.000 (-0.004)
	LF 8	0.897 (0.219)	0.800 (0.461)	0.033 (-0.033)	0.235 (0.232)
	LF 9	0.601 (-0.043)	0.446 (-0.020)	0.013 (0.000)	0.068 (0.056)
Bank	LF 1	0.706 (-0.063)	0.165 (0.041)	0.004 (0.000)	0.003 (-0.010)
	LF 2	0.824 (-0.064)	0.221 (0.034)	0.037 (-0.018)	0.113 (0.045)
	LF 3	0.931 (0.116)	0.829 (0.414)	0.019 (0.000)	0.110 (-0.015)
	LF 4	0.427 (-0.255)	0.411 (0.214)	0.100 (-0.502)	0.004 (-0.639)
	LF 5	0.833 (-0.065)	0.287 (-0.014)	0.061 (0.009)	0.195 (0.133)
	LF 6	0.624 (0.049)	0.172 (-0.034)	0.009 (-0.174)	0.046 (-0.173)
Bank (LIFT)	LF 1	0.768 (-0.001)	0.105 (-0.019)	0.000 (-0.003)	0.013 (0.000)
	LF 2	0.888 (0.000)	0.188 (0.001)	0.037 (-0.017)	0.068 (0.000)
	LF 3	0.815 (0.000)	0.415 (0.000)	0.019 (0.000)	0.125 (0.000)
	LF 4	0.317 (-0.365)	0.228 (0.031)	0.100 (-0.502)	0.065 (-0.578)
	LF 5	0.898 (0.000)	0.303 (0.001)	0.061 (0.009)	0.086 (0.023)
	LF 6	0.680 (0.105)	0.126 (-0.081)	0.009 (-0.175)	0.084 (-0.134)

Table 12: Tabular dataset SBM (OT linear) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
Adult	LF 1	0.841 (0.292)	0.846 (0.370)	0.654 (0.554)	0.733 (0.710)
	LF 2	0.334 (-0.409)	0.000 (-0.455)	0.208 (0.175)	0.000 (-0.044)
	LF 3	0.346 (-0.353)	0.000 (-0.579)	0.174 (-0.273)	0.000 (-0.241)
	LF 4	0.895 (0.331)	0.904 (0.418)	0.735 (0.354)	0.824 (0.581)
	LF 5	0.394 (-0.406)	0.000 (-0.315)	0.027 (-0.008)	0.000 (-0.019)
	LF 6	0.408 (-0.329)	0.070 (0.004)	0.003 (0.000)	0.037 (0.033)
	LF 7	0.405 (-0.351)	0.019 (-0.005)	0.001 (0.000)	0.010 (0.006)
	LF 8	0.336 (-0.342)	0.000 (-0.339)	0.202 (0.136)	0.000 (-0.003)
	LF 9	0.501 (-0.143)	0.512 (0.046)	0.013 (0.000)	0.439 (0.427)
Adult (LIFT)	LF 1	0.628 (0.079)	0.655 (0.179)	0.036 (-0.064)	0.027 (0.004)
	LF 2	0.780 (0.037)	0.664 (0.209)	0.014 (-0.019)	0.032 (-0.012)
	LF 3	0.525 (-0.174)	0.296 (-0.283)	0.097 (-0.350)	0.013 (-0.228)
	LF 4	0.497 (-0.067)	0.554 (0.068)	0.137 (-0.244)	0.150 (-0.093)
	LF 5	0.606 (-0.194)	0.177 (-0.138)	0.023 (-0.012)	0.042 (0.023)
	LF 6	0.565 (-0.172)	0.089 (0.023)	0.003 (0.000)	0.008 (0.004)
	LF 7	0.567 (-0.189)	0.029 (0.005)	0.001 (0.000)	0.000 (-0.004)
	LF 8	0.618 (-0.060)	0.409 (0.070)	0.011 (-0.055)	0.017 (0.014)
	LF 9	0.843 (0.199)	0.817 (0.351)	0.013 (0.000)	0.046 (0.034)
Bank	LF 1	0.818 (0.049)	0.062 (-0.062)	0.004 (0.000)	0.140 (0.127)
	LF 2	0.980 (0.092)	0.703 (0.516)	0.024 (-0.031)	0.542 (0.474)
	LF 3	0.777 (-0.038)	0.093 (-0.322)	0.019 (0.000)	0.265 (0.139)
	LF 4	0.047 (-0.635)	0.083 (-0.114)	0.131 (-0.471)	1.000 (0.357)
	LF 5	0.988 (0.090)	0.839 (0.538)	0.032 (-0.020)	0.723 (0.660)
	LF 6	0.950 (0.375)	0.000 (-0.206)	0.244 (0.061)	0.000 (-0.219)
Bank (LIFT)	LF 1	0.125 (-0.645)	0.197 (0.073)	0.853 (0.849)	0.867 (0.854)
	LF 2	0.888 (0.000)	0.180 (-0.007)	0.000 (-0.055)	0.036 (-0.032)
	LF 3	0.815 (0.000)	0.415 (0.000)	0.018 (-0.001)	0.137 (0.012)
	LF 4	0.876 (0.194)	0.085 (-0.112)	0.869 (0.268)	0.954 (0.311)
	LF 5	0.898 (0.000)	0.300 (-0.002)	0.047 (-0.005)	0.039 (-0.023)
	LF 6	0.124 (-0.451)	0.198 (-0.008)	0.756 (0.572)	0.717 (0.498)

Table 13: Tabular dataset SBM (OT sinkhorn) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
Adult	LF 1	0.604 (0.055)	0.595 (0.119)	0.145 (0.045)	0.125 (0.102)
	LF 2	0.601 (-0.142)	0.340 (-0.115)	0.036 (0.003)	0.015 (-0.029)
	LF 3	0.998 (0.299)	0.998 (0.419)	0.293 (-0.154)	0.002 (-0.239)
	LF 4	0.690 (0.126)	0.700 (0.214)	0.221 (-0.160)	0.050 (-0.193)
	LF 5	0.656 (-0.144)	0.181 (-0.134)	0.032 (-0.003)	0.021 (0.002)
	LF 6	0.621 (-0.116)	0.081 (0.015)	0.003 (0.000)	0.030 (0.026)
	LF 7	0.627 (-0.129)	0.023 (-0.001)	0.001 (0.000)	0.005 (0.001)
	LF 8	0.607 (-0.071)	0.256 (-0.083)	0.069 (0.003)	0.095 (0.092)
	LF 9	0.558 (-0.086)	0.446 (-0.020)	0.013 (0.000)	0.014 (0.002)
Adult (LIFT)	LF 1	0.475 (-0.074)	0.436 (-0.040)	0.031 (-0.069)	0.032 (0.009)
	LF 2	0.953 (0.210)	0.910 (0.455)	0.047 (0.014)	0.079 (0.035)
	LF 3	0.712 (0.013)	0.191 (-0.388)	0.157 (-0.290)	0.250 (0.009)
	LF 4	0.373 (-0.191)	0.425 (-0.061)	0.204 (-0.177)	0.120 (-0.123)
	LF 5	0.764 (-0.036)	0.301 (-0.014)	0.035 (0.000)	0.107 (0.088)
	LF 6	0.711 (-0.026)	0.117 (0.051)	0.003 (0.000)	0.018 (0.014)
	LF 7	0.715 (-0.041)	0.037 (0.013)	0.001 (0.000)	0.002 (-0.002)
	LF 8	0.716 (0.038)	0.272 (-0.067)	0.147 (0.081)	0.284 (0.281)
	LF 9	0.640 (-0.004)	0.495 (0.029)	0.013 (0.000)	0.213 (0.201)
Bank	LF 1	0.694 (-0.075)	0.173 (0.049)	0.004 (0.000)	0.017 (0.004)
	LF 2	0.806 (-0.082)	0.209 (0.022)	0.059 (0.004)	0.211 (0.143)
	LF 3	0.949 (0.134)	0.881 (0.466)	0.019 (0.000)	0.136 (0.011)
	LF 4	0.351 (-0.331)	0.382 (0.185)	0.040 (-0.562)	0.016 (-0.627)
	LF 5	0.814 (-0.084)	0.267 (-0.034)	0.069 (0.017)	0.247 (0.185)
	LF 6	0.583 (0.008)	0.020 (-0.186)	0.040 (-0.143)	0.103 (-0.116)
Bank (LIFT)	LF 1	0.769(-0.001)	0.087 (-0.038)	0.006 (0.003)	0.038 (0.025)
	LF 2	0.888 (0.001)	0.204 (0.017)	0.141 (0.087)	0.305 (0.237)
	LF 3	0.814 (-0.001)	0.409 (-0.006)	0.054 (0.035)	0.357 (0.231)
	LF 4	0.307 (-0.375)	0.229 (0.032)	0.085 (-0.517)	0.042 (-0.600)
	LF 5	0.898 (0.000)	0.311 (0.010)	0.133 (0.081)	0.236 (0.173)
	LF 6	0.677 (0.102)	0.105 (-0.102)	0.003 (-0.180)	0.122 (-0.096)

Table 14: NLP dataset raw LF performance

Dataset	LF	Acc	F1	Δ_{DP}	Δ_{EO}
CivilComments	LF 1 (sex LF)	0.755	0.187	0.046	0.019
	LF 2 (LGBTQ LF)	0.877	0.073	0.001	0.017
	LF 3 (religion LF)	0.861	0.049	0.012	0.013
	LF 4 (race LF)	0.847	0.234	0.634	0.574
	LF 5 (toxic LF)	0.886	0.068	0.006	0.012
	LF 6 (threat LF)	0.862	0.102	0.055	0.054
	LF 7 (insult LF)	0.872	0.176	0.028	0.035
HateXplain	LF 1 (sex LF)	0.427	0.253	0.015	0.002
	LF 2 (LGBTQ LF)	0.405	0.077	0.047	0.041
	LF 3 (religion LF)	0.437	0.197	0.001	0.003
	LF 4 (race LF)	0.419	0.327	0.139	0.168
	LF 5 (toxic LF)	0.451	0.233	0.007	0.016
	LF 6 (threat LF)	0.415	0.097	0.000	0.004
	LF 7 (insult LF)	0.427	0.100	0.015	0.005
	LF 8 (Detoxify - original)	0.645	0.704	0.165	0.086
	LF 9 (Detoxify - unbiased)	0.625	0.668	0.150	0.078
	LF 10 (Detoxify - multilingual)	0.649	0.700	0.168	0.077
	LF 11 (Detoxify - original-small)	0.644	0.705	0.152	0.076
	LF 12 (Detoxify - unbiased-small)	0.643	0.699	0.186	0.113

Table 15: NLP dataset SBM (w/o OT) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
Civil Comments	LF 1	0.790 (0.035)	0.325 (0.138)	0.048 (0.002)	0.041 (0.022)
	LF 2	0.896 (0.019)	0.268 (0.195)	0.001 (0.000)	0.053 (0.036)
	LF 3	0.868 (0.007)	0.155 (0.106)	0.012 (0.000)	0.043 (0.030)
	LF 4	0.858 (0.011)	0.252 (0.018)	0.114 (-0.520)	0.160 (-0.414)
	LF 5	0.886 (0.000)	0.137 (0.069)	0.006 (0.000)	0.003 (-0.009)
	LF 6	0.916 (0.054)	0.482 (0.380)	0.023 (-0.032)	0.002 (-0.052)
	LF 7	0.918 (0.046)	0.501 (0.325)	0.022 (-0.006)	0.012 (-0.023)
HateXplain	LF 1	0.483 (0.056)	0.273 (0.020)	0.015 (0.000)	0.001 (-0.001)
	LF 2	0.473 (0.068)	0.058 (-0.019)	0.000 (-0.047)	0.004 (-0.037)
	LF 3	0.460 (0.023)	0.163 (-0.034)	0.001 (0.000)	0.004 (0.001)
	LF 4	0.481 (0.062)	0.328 (0.001)	0.044 (-0.095)	0.039 (-0.129)
	LF 5	0.515 (0.064)	0.267 (0.034)	0.014 (0.007)	0.021 (0.005)
	LF 6	0.471 (0.056)	0.106 (0.009)	0.000 (0.000)	0.015 (0.011)
	LF 7	0.472 (0.045)	0.088 (-0.012)	0.012 (-0.003)	0.014 (0.009)
	LF 8	0.831 (0.186)	0.864 (0.160)	0.006 (-0.159)	0.000 (-0.086)
	LF 9	0.917 (0.292)	0.928 (0.260)	0.015 (-0.135)	0.000 (-0.078)
	LF 10	0.864 (0.215)	0.888 (0.188)	0.009 (-0.159)	0.000 (-0.077)
	LF 11	0.839 (0.195)	0.870 (0.165)	0.015 (-0.137)	0.000 (-0.076)
	LF 12	0.837 (0.194)	0.868 (0.169)	0.014 (-0.172)	0.000 (-0.113)

Table 16: NLP dataset SBM (OT linear) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
Civil Comments	LF 1	0.791 (0.036)	0.321 (0.134)	0.003 (-0.043)	0.022 (0.003)
	LF 2	0.898 (0.021)	0.272 (0.199)	0.001 (0.000)	0.020 (0.003)
	LF 3	0.870 (0.009)	0.156 (0.107)	0.012 (0.000)	0.041 (0.028)
	LF 4	0.860 (0.013)	0.244 (0.010)	0.017 (-0.617)	0.017 (-0.557)
	LF 5	0.887 (0.001)	0.139 (0.071)	0.006 (0.000)	0.027 (0.015)
	LF 6	0.917 (0.055)	0.484 (0.382)	0.012 (-0.043)	0.026 (-0.028)
	LF 7	0.919 (0.047)	0.501 (0.325)	0.007 (-0.021)	0.013 (-0.022)
HateXplain	LF 1	0.457 (0.030)	0.279 (0.026)	0.015 (0.000)	0.001 (-0.001)
	LF 2	0.433 (0.028)	0.062 (-0.015)	0.002 (-0.045)	0.006 (-0.035)
	LF 3	0.429 (-0.008)	0.171 (-0.026)	0.001 (0.000)	0.001 (-0.002)
	LF 4	0.457 (0.038)	0.323 (-0.004)	0.011 (-0.128)	0.009 (-0.159)
	LF 5	0.479 (0.028)	0.263 (0.030)	0.017 (0.010)	0.029 (0.013)
	LF 6	0.436 (0.021)	0.111 (0.014)	0.000 (0.000)	0.011 (0.007)
	LF 7	0.434 (0.007)	0.091 (-0.009)	0.012 (-0.003)	0.017 (0.012)
	LF 8	0.845 (0.200)	0.882 (0.178)	0.790 (0.093)	0.000 (-0.086)
	LF 9	0.923 (0.298)	0.938 (0.270)	0.883 (0.179)	0.000 (-0.078)
	LF 10	0.875 (0.226)	0.903 (0.203)	0.823 (0.111)	0.000 (-0.077)
	LF 11	0.848 (0.204)	0.884 (0.179)	0.792 (0.098)	0.000 (-0.076)
	LF 12	0.845 (0.202)	0.882 (0.183)	0.789 (0.090)	0.000 (-0.113)

Table 17: NLP dataset SBM (OT sinkhorn) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
Civil Comments	LF 1	0.791 (0.036)	0.320 (0.133)	0.015 (-0.031)	0.082 (0.063)
	LF 2	0.897 (0.020)	0.271 (0.198)	0.001 (0.000)	0.029 (0.012)
	LF 3	0.870 (0.009)	0.156 (0.107)	0.012 (0.000)	0.043 (0.030)
	LF 4	0.860 (0.013)	0.245 (0.011)	0.017 (-0.617)	0.016 (-0.558)
	LF 5	0.887 (0.001)	0.138 (0.070)	0.006 (0.000)	0.020 (0.008)
	LF 6	0.917 (0.055)	0.482 (0.380)	0.011 (-0.044)	0.006 (-0.048)
	LF 7	0.919 (0.047)	0.504 (0.328)	0.019 (-0.009)	0.039 (0.004)
HateXplain	LF 1	0.444 (0.017)	0.277 (0.024)	0.015 (0.000)	0.002 (0.000)
	LF 2	0.417 (0.012)	0.056 (-0.021)	0.001 (-0.046)	0.001 (-0.040)
	LF 3	0.418 (-0.019)	0.172 (-0.025)	0.001 (0.000)	0.002 (-0.001)
	LF 4	0.448 (0.029)	0.332 (0.005)	0.032 (-0.107)	0.045 (-0.123)
	LF 5	0.459 (0.008)	0.256 (0.023)	0.030 (0.023)	0.041 (0.025)
	LF 6	0.422 (0.007)	0.110 (0.013)	0.000 (0.000)	0.011 (0.007)
	LF 7	0.418 (-0.009)	0.091 (-0.009)	0.018 (0.003)	0.025 (0.020)
	LF 8	0.851 (0.206)	0.889 (0.185)	0.056 (-0.109)	0.000 (-0.086)
	LF 9	0.927 (0.302)	0.942 (0.274)	0.061 (-0.089)	0.000 (-0.078)
	LF 10	0.884 (0.235)	0.911 (0.211)	0.052 (-0.116)	0.000 (-0.077)
	LF 11	0.853 (0.209)	0.890 (0.185)	0.056 (-0.096)	0.000 (-0.076)
	LF 12	0.847 (0.204)	0.886 (0.187)	0.063 (-0.123)	0.000 (-0.113)

Table 18: Vision dataset raw LF performance

Dataset	LF	Acc	F1	Δ_{DP}	Δ_{EO}
CelebA	LF 1 (ResNet-18 fine-tuned on gender classification dataset)	0.798	0.794	0.328	0.284
	LF 2 (ResNet-34 fine-tuned on FairFace dataset)	0.890	0.901	0.314	0.105
	LF 3 (ResNet-34 fine-tuned on UTKFace dataset)	0.826	0.831	0.309	0.195
	LF 4 (ResNet-34 fine-tuned on UTKFace dataset (White only))	0.825	0.832	0.277	0.131
	LF 5 (ResNet-34 fine-tuned on UTKFace dataset (non-White only))	0.818	0.832	0.271	0.134
	LF 6 (ResNet-34 fine-tuned on UTKFace dataset (age ≥ 50 only)	0.764	0.750	0.279	0.194
	LF 7 (ResNet-34 fine-tuned on UTKFace dataset (age < 50 only))	0.830	0.845	0.299	0.175
UTKFace	LF 1 (ResNet-18 fine-tuned on gender classification dataset)	0.869	0.856	0.060	0.039
	LF 2 (ResNet-34 fine-tuned on gender classification dataset)	0.854	0.842	0.060	0.060
	LF 3 (ResNet-34 fine-tuned on CelebA dataset)	0.742	0.758	0.158	0.032
	LF 4 (ResNet-34 fine-tuned on CelebA dataset (attractive only))	0.580	0.692	0.065	0.002
	LF 5 (ResNet-34 fine-tuned on CelebA dataset (non-attractive only))	0.687	0.608	0.129	0.034
	LF 6 (ResNet-34 fine-tuned on CelebA dataset (young only))	0.664	0.729	0.116	0.012
	LF 7 (ResNet-34 fine-tuned on CelebA dataset (non-young only))	0.619	0.429	0.136	0.081
	LF 8 (ResNet-34 fine-tuned on CelebA dataset (unfair sampling))	0.631	0.676	0.113	0.053

Table 19: Vision dataset SBM LF (w/o OT) performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
CelebA	LF 1	0.847 (0.049)	0.832 (0.038)	0.328 (0.000)	0.267 (-0.017)
	LF 2	0.890 (0.000)	0.895 (-0.006)	0.314 (0.000)	0.101 (-0.004)
	LF 3	0.926 (0.100)	0.923 (0.092)	0.309 (0.000)	0.097 (-0.098)
	LF 4	0.914 (0.089)	0.912 (0.080)	0.277 (0.000)	0.027 (-0.104)
	LF 5	0.899 (0.081)	0.900 (0.068)	0.271 (0.000)	0.030 (-0.104)
	LF 6	0.705 (-0.059)	0.629 (-0.121)	0.177 (-0.102)	0.052 (-0.142)
	LF 7	0.913 (0.083)	0.915 (0.070)	0.299 (0.000)	0.056 (-0.119)
UTKFace	LF 1	0.929 (0.060)	0.924 (0.068)	0.102 (0.042)	0.011 (-0.028)
	LF 2	0.939 (0.085)	0.935 (0.093)	0.102 (0.042)	0.007 (-0.053)
	LF 3	0.631 (-0.111)	0.678 (-0.080)	0.078 (-0.080)	0.034 (0.002)
	LF 4	0.549 (-0.031)	0.681 (-0.011)	0.017 (-0.048)	0.002 (0.000)
	LF 5	0.740 (0.053)	0.679 (0.071)	0.129 (0.000)	0.040 (0.006)
	LF 6	0.694 (0.030)	0.755 (0.026)	0.116 (0.000)	0.037 (0.025)
	LF 7	0.694 (0.075)	0.541 (0.112)	0.061 (-0.075)	0.033 (-0.048)
	LF 8	0.591 (-0.040)	0.654 (-0.022)	0.071 (-0.042)	0.054 (0.001)

Table 20: Vision dataset SBM (OT linear) LF performance. Δs are obtained by comparison with raw LF performance.

Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
CelebA	LF 1	0.847 (0.049)	0.832 (0.038)	0.328 (0.000)	0.268 (-0.016)
	LF 2	0.890 (0.000)	0.894 (-0.007)	0.314 (0.000)	0.102 (-0.003)
	LF 3	0.926 (0.100)	0.922 (0.091)	0.309 (0.000)	0.098 (-0.097)
	LF 4	0.915 (0.090)	0.913 (0.081)	0.277 (0.000)	0.029 (-0.102)
	LF 5	0.898 (0.080)	0.900 (0.068)	0.271 (0.000)	0.031 (-0.103)
	LF 6	0.648 (-0.116)	0.498 (-0.252)	0.059 (-0.220)	0.217 (0.023)
	LF 7	0.914 (0.084)	0.916 (0.071)	0.299 (0.000)	0.058 (-0.117)
UTKFace	LF 1	0.931 (0.062)	0.925 (0.069)	0.026 (-0.034)	0.012 (-0.027)
	LF 2	0.945 (0.091)	0.940 (0.098)	0.017 (-0.043)	0.006 (-0.054)
	LF 3	0.599 (-0.143)	0.667 (-0.091)	0.004 (-0.154)	0.001 (-0.031)
	LF 4	0.523 (-0.057)	0.667 (-0.025)	0.008 (-0.057)	0.002 (0.000)
	LF 5	0.738 (0.051)	0.674 (0.066)	0.129 (0.000)	0.029 (-0.005)
	LF 6	0.691 (0.027)	0.752 (0.023)	0.116 (0.000)	0.037 (0.025)
	LF 7	0.690 (0.071)	0.525 (0.096)	0.000 (-0.136)	0.016 (-0.065)
	LF 8	0.572 (-0.059)	0.655 (-0.021)	0.001 (-0.112)	0.014 (-0.039)

Table 21: Vision dataset SBM (OT sinkhorn) LF performance. Δs are obtained by comparison with raw LF performance.

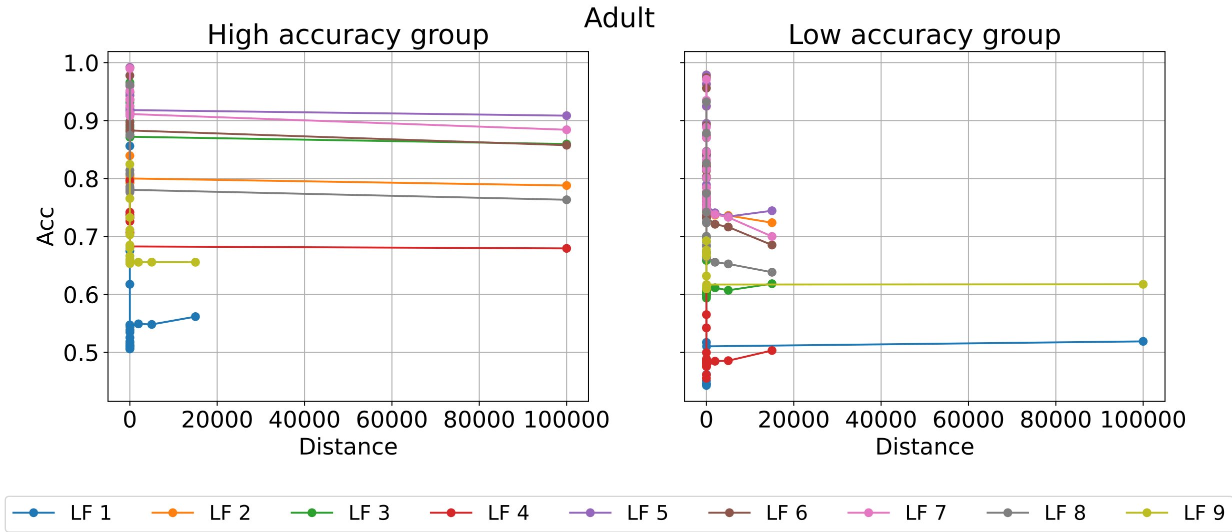
Dataset	LF	Acc (Δ)	F1 (Δ)	Δ_{DP} (Δ)	Δ_{EO} (Δ)
CelebA	LF 1	0.850 (0.052)	0.835 (0.041)	0.328 (0.000)	0.266 (-0.018)
	LF 2	0.893 (0.003)	0.897 (-0.004)	0.314 (0.000)	0.101 (-0.004)
	LF 3	0.923 (0.097)	0.920 (0.089)	0.309 (0.000)	0.098 (-0.097)
	LF 4	0.913 (0.088)	0.910 (0.078)	0.277 (0.000)	0.029 (-0.102)
	LF 5	0.901 (0.083)	0.903 (0.071)	0.271 (0.000)	0.031 (-0.103)
	LF 6	0.618 (-0.146)	0.430 (-0.320)	0.016 (-0.263)	0.282 (0.088)
	LF 7	0.911 (0.081)	0.913 (0.068)	0.299 (0.000)	0.059 (-0.116)
UTKFace	LF 1	0.931 (0.062)	0.926 (0.070)	0.010 (-0.050)	0.001 (-0.038)
	LF 2	0.949 (0.095)	0.945 (0.103)	0.015 (-0.045)	0.001 (-0.059)
	LF 3	0.577 (-0.165)	0.652 (-0.106)	0.003 (-0.155)	0.013 (-0.019)
	LF 4	0.517 (-0.063)	0.664 (-0.028)	0.006 (-0.059)	0.007 (0.005)
	LF 5	0.730 (0.043)	0.669 (0.061)	0.129 (0.000)	0.018 (-0.016)
	LF 6	0.694 (0.030)	0.756 (0.027)	0.116 (0.000)	0.036 (0.024)
	LF 7	0.683 (0.064)	0.523 (0.094)	0.010 (-0.126)	0.007 (-0.074)
	LF 8	0.562 (-0.069)	0.652 (-0.024)	0.009 (-0.104)	0.012 (-0.041)

E.4 Identification of centers

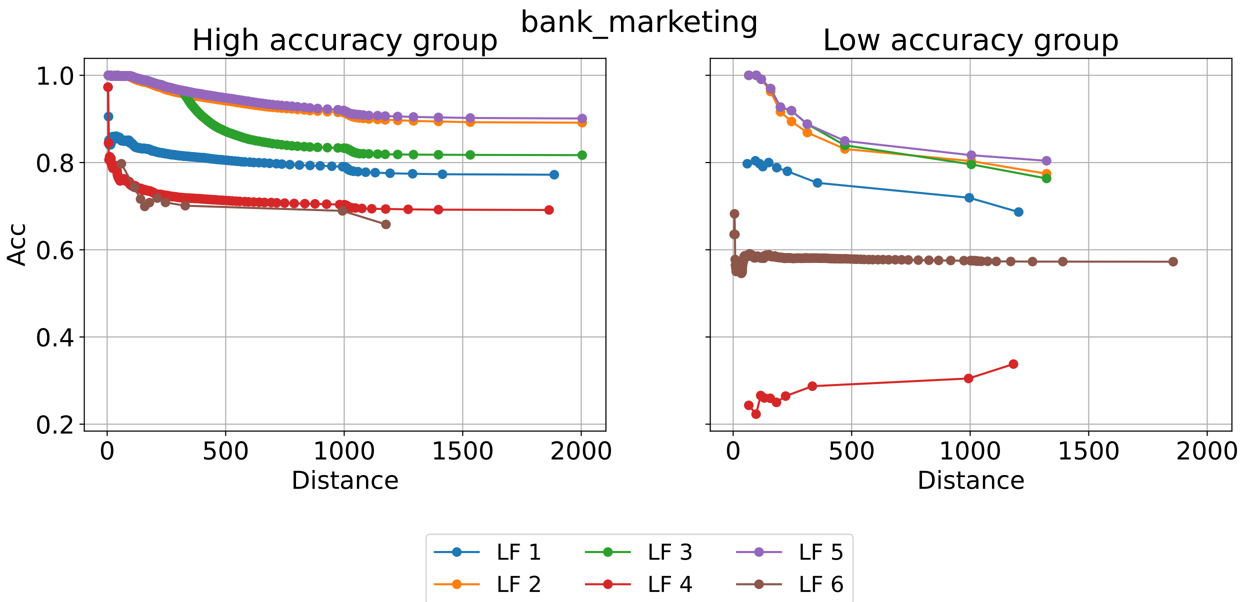
While our method improves performance by matching one group distribution and attempting to make these uniform, it does not imply accuracy improvement. A presumption of our method is that the group with high (estimated) accuracy possesses the high accuracy regime and our method can transport data points to this high accuracy regime while keeping their structure, which results in accuracy improvements. To empirically support this hypothesis, we used the following procedure and visualized the results in Figure 6, 8, 9.

1. Find the best accuracy center by evaluating the accuracy of the nearest 10% of data points for each center candidate point.
2. Expand 2% percent of data points closest to the center each time, compute their accumulated average accuracy (y-axis) and the farthest distance (x-axis) from the neighborhood
3. Find the group with better accuracy group and visualize their accumulated average accuracy (y-axis) and the farthest distance (x-axis) in each group.

We are able to obtain two insights from the visualization. First, the high accuracy regime typically exists in the group with the high estimated accuracy, which supports our hypothesis. Thus accuracy improvement by optimal transport can be justified. Secondly, the groups actually show the distributional difference in the input space $\mathcal{X}$. Given center points, lines in the high accuracy group start with a smaller distance to the center than the low accuracy group.

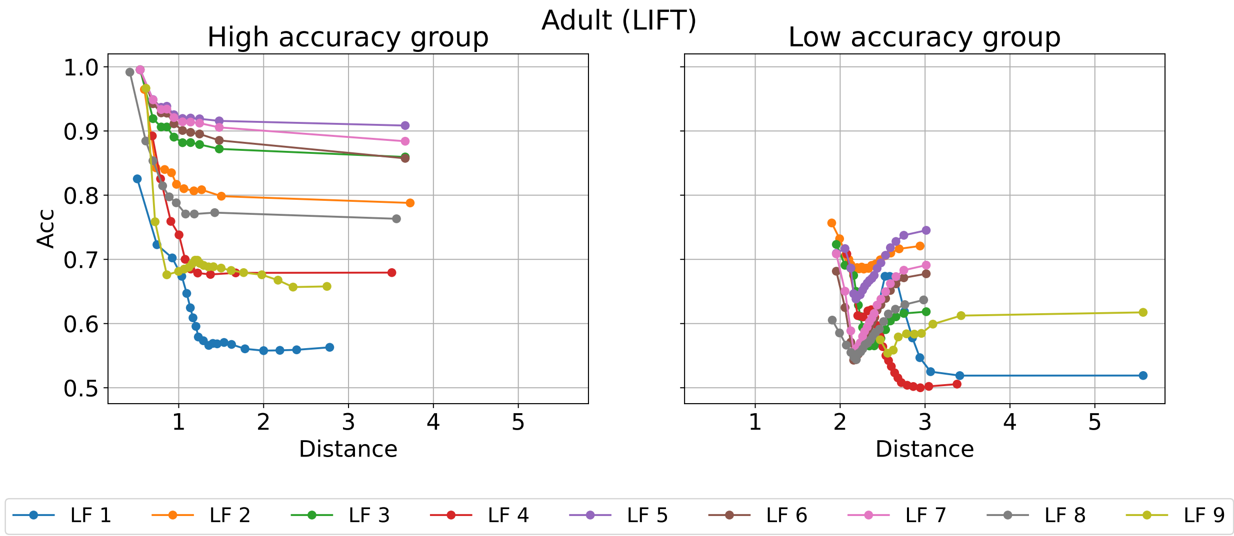


(a) Adult (raw)

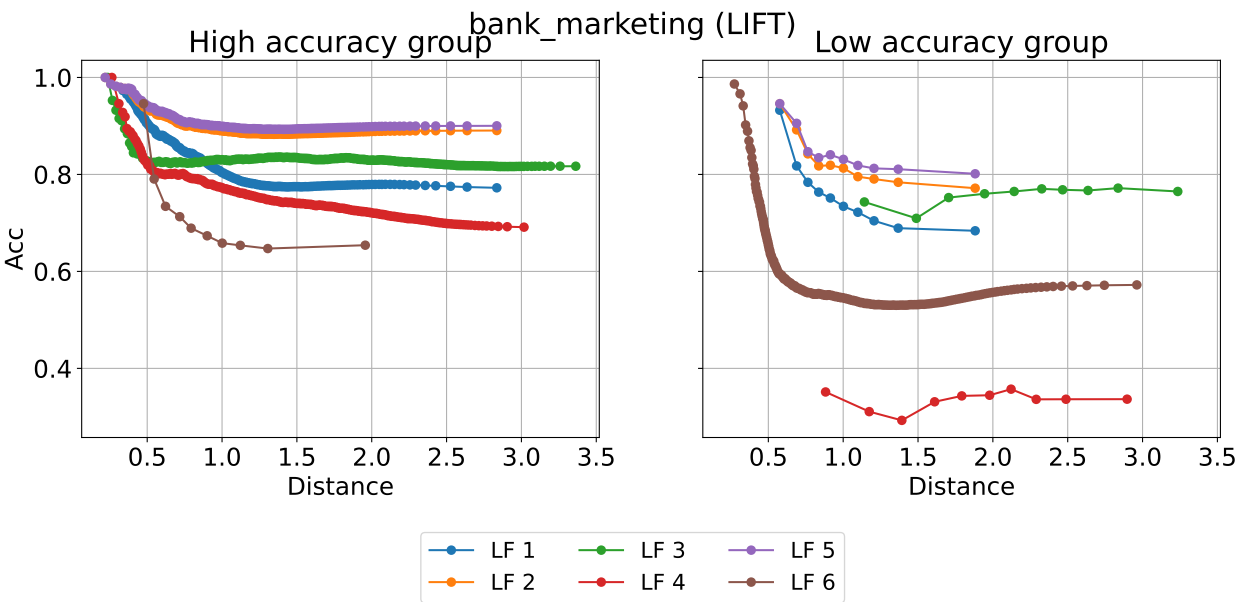


(b) Bank Marketing (raw)

Figure 6: Identification of high accuracy regimes for tabular datasets.

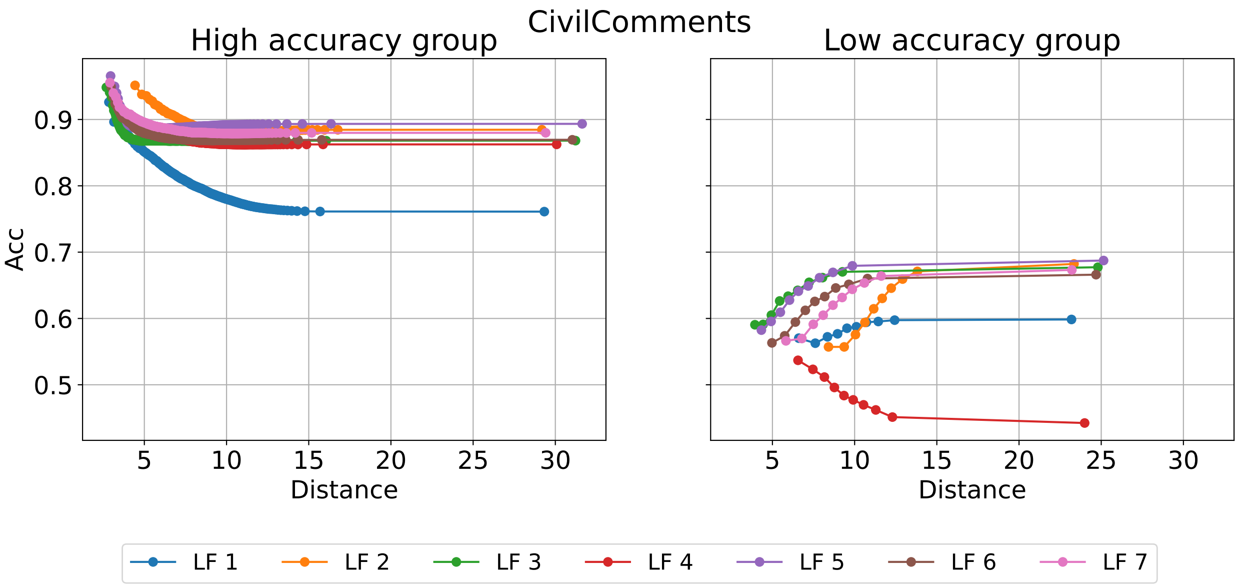


(a) Adult (LIFT)

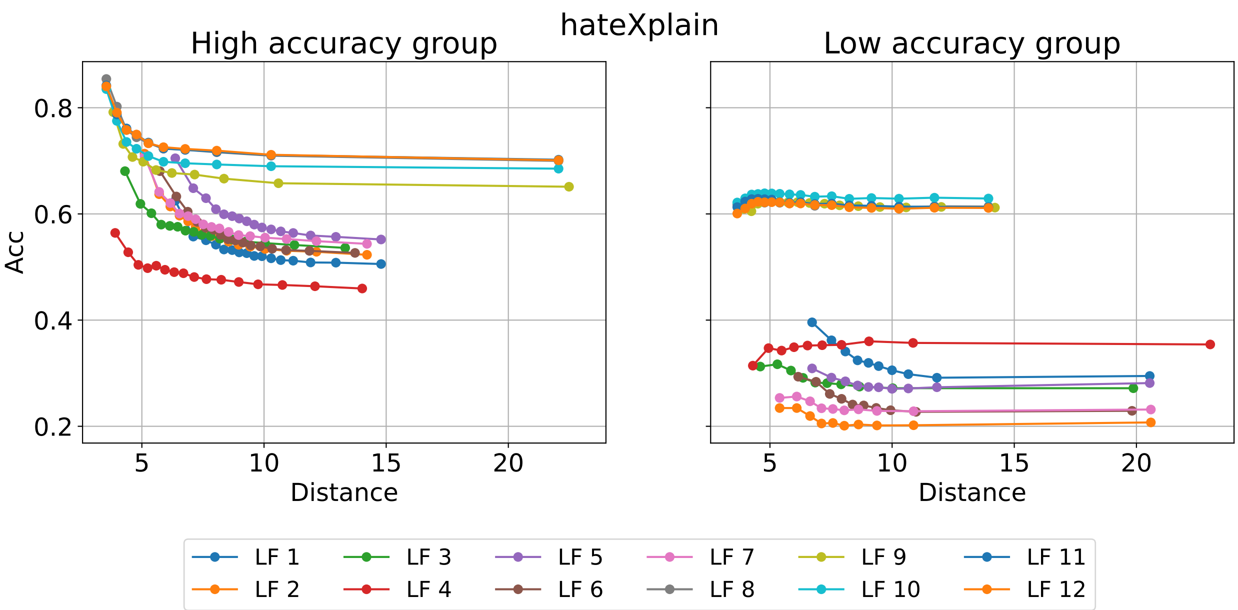


(b) Bank Marketing (LIFT)

Figure 7: Identification of high accuracy regimes for tabular datasets (LIFT).

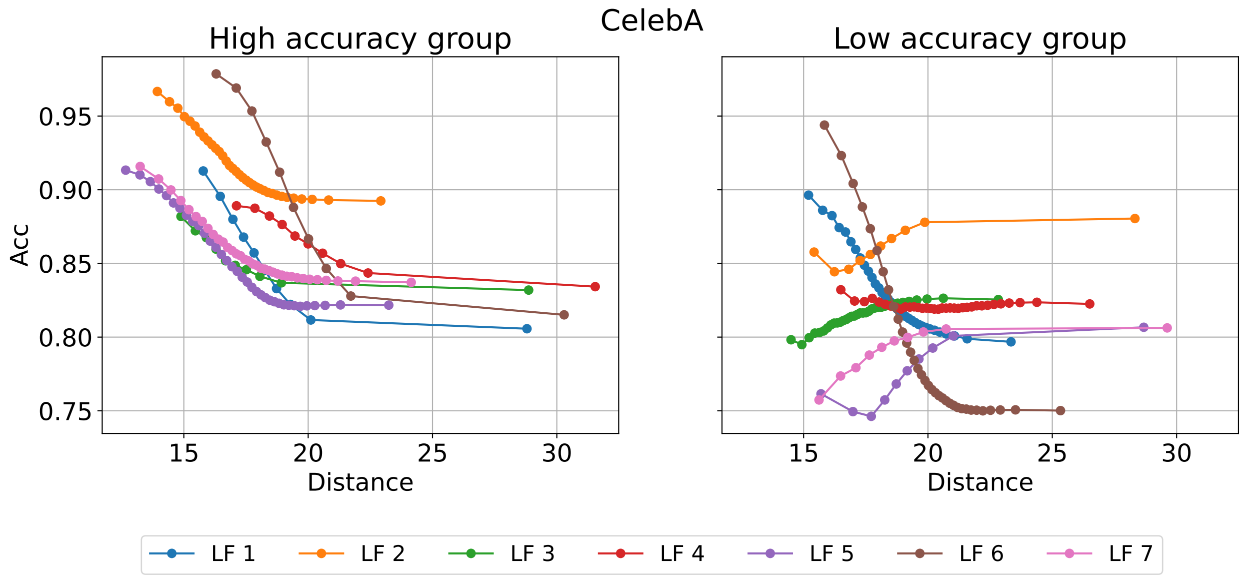


(a) CivilComments

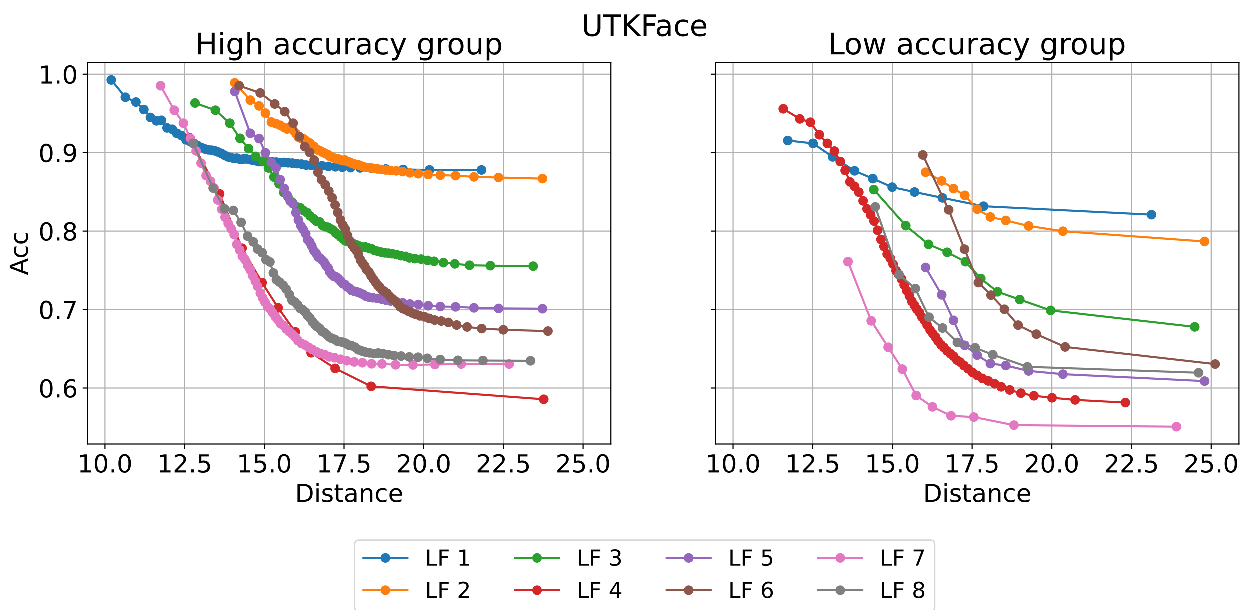


(b) HateXplain

Figure 8: Identification of high accuracy regimes for NLP datasets.



(a) CelebA



(b) UTKFace

Figure 9: Identification of high accuracy regimes for vision datasets.

E.5 Compatibility with other fair ML methods

One advantage of our method is that we can use other successful fair ML methods in a supervised learning setting on top of SBM, since our method works in weak label sources while traditional fair ML methods work in the preprocessing/training/postprocessing steps, which are independent of the label model. To make this point, we tried traditional fair ML methods from fairlearn [3] with each of WS settings. We used CorrelationRemover, ExponentiatedGradient [1], ThresholdOptimizer [25] with the demographic parity (DP), equal opportunity (EO) as parity criteria, and accuracy as the performance criteria. The results are reported in Table 22 - 29. As expected, combining with other methods yields an accuracy-fairness tradeoff given weak label sources. Typically, SBM yields additional gains upon traditional fair ML methods. One another observation is that fair ML methods to modify equal opportunity typically fail to achieve the reduction of Δ_{EO}. This can be interpreted as the result of the noise in the training set labels.

Table 22: SBM combined with other fair ML methods in Adult dataset

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.717	0.587	0.475	0.325
	correlation remover	0.716	0.587	0.446	0.287
	optimal threshold (DP)	0.578	0.499	0.002	0.076
	optimal threshold (EO)	0.721	0.563	0.404	0.217
	exponentiated gradient (DP gap = 0)	0.582	0.502	0.002	0.066
	exponentiated gradient (EO gap = 0)	0.715	0.585	0.445	0.284
SBM (w/o OT)	N/A	0.720	0.592	0.439	0.273
	correlation remover	0.717	0.586	0.437	0.264
	optimal threshold (DP)	0.591	0.507	0.003	0.059
	optimal threshold (EO)	0.722	0.571	0.387	0.189
	exponentiated gradient (DP gap = 0)	0.693	0.525	0.014	0.052
	exponentiated gradient (EO gap = 0)	0.722	0.586	0.404	0.233
SBM (OT linear)	N/A	0.560	0.472	0.893	0.980
	correlation remover	0.460	0.443	0.084	0.005
	optimal threshold (DP)	0.324	0.404	0.006	0.089
	optimal threshold (EO)	0.300	0.399	0.103	0.015
	exponentiated gradient (DP gap = 0)	0.345	0.414	0.002	0.016
	exponentiated gradient (EO gap = 0)	0.558	0.479	0.861	0.792
SBM (OT sinkhorn)	N/A	0.722	0.590	0.429	0.261
	correlation remover	0.729	0.595	0.408	0.249
	optimal threshold (DP)	0.596	0.507	0.003	0.050
	optimal threshold (EO)	0.723	0.571	0.382	0.184
	exponentiated gradient (DP gap = 0)	0.687	0.527	0.011	0.045
	exponentiated gradient (EO gap = 0)	0.728	0.587	0.390	0.218

Table 23: SBM combined with other fair ML methods in Adult dataset (LIFT)

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.711	0.584	0.449	0.290
	correlation remover	0.716	0.587	0.446	0.287
	optimal threshold (DP)	0.578	0.499	0.002	0.076
	optimal threshold (EO)	0.721	0.563	0.404	0.217
	exponentiated gradient (DP gap = 0)	0.582	0.502	0.002	0.066
	exponentiated gradient (EO gap = 0)	0.715	0.585	0.445	0.284
SBM (w/o OT)	N/A	0.704	0.366	0.032	0.192
	correlation remover	0.686	0.351	0.006	0.155
	optimal threshold (DP)	0.707	0.363	0.007	0.133
	optimal threshold (EO)	0.713	0.362	0.022	0.079
	exponentiated gradient (DP gap = 0)	0.682	0.350	0.011	0.163
	exponentiated gradient (EO gap = 0)	0.701	0.369	0.019	0.134
SBM (OT linear)	N/A	0.700	0.520	0.015	0.138
	correlation remover	0.686	0.504	0.011	0.105
	optimal threshold (DP)	0.701	0.520	0.008	0.124
	optimal threshold (EO)	0.712	0.521	0.060	0.025
	exponentiated gradient (DP gap = 0)	0.673	0.504	0.005	0.071
	exponentiated gradient (EO gap = 0)	0.691	0.516	0.058	0.035
SBM (OT sinkhorn)	N/A	0.782	0.448	0.000	0.178
	correlation remover	0.772	0.435	0.002	0.180
	optimal threshold (DP)	0.782	0.447	0.001	0.176
	optimal threshold (EO)	0.790	0.427	0.087	0.104
	exponentiated gradient (DP gap = 0)	0.784	0.452	0.000	0.171
	exponentiated gradient (EO gap = 0)	0.747	0.380	0.107	0.049

Table 24: SBM combined with other fair ML methods in Bank Marketing dataset

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.674	0.258	0.543	0.450
	correlation remover	0.890	0.057	0.002	0.006
	optimal threshold (DP)	0.890	0.058	0.002	0.007
	optimal threshold (EO)	0.890	0.066	0.030	0.040
	exponentiated gradient (DP gap = 0)	0.889	0.039	0.000	0.009
	exponentiated gradient (EO gap = 0)	0.890	0.070	0.033	0.051
SBM (w/o OT)	N/A	0.876	0.550	0.106	0.064
	correlation remover	0.874	0.547	0.064	0.095
	optimal threshold (DP)	0.876	0.547	0.031	0.208
	optimal threshold (EO)	0.876	0.548	0.053	0.171
	exponentiated gradient (DP gap = 0)	0.877	0.525	0.037	0.182
	exponentiated gradient (EO gap = 0)	0.872	0.531	0.066	0.124
SBM (OT linear)	N/A	0.892	0.304	0.095	0.124
	correlation remover	0.890	0.290	0.011	0.111
	optimal threshold (DP)	0.891	0.280	0.008	0.163
	optimal threshold (EO)	0.881	0.313	0.841	0.678
	exponentiated gradient (DP gap = 0)	0.891	0.263	0.003	0.106
	exponentiated gradient (EO gap = 0)	0.895	0.296	0.097	0.136
SBM (OT sinkhorn)	N/A	0.847	0.515	0.122	0.080
	correlation remover	0.847	0.512	0.072	0.122
	optimal threshold (DP)	0.846	0.511	0.043	0.236
	optimal threshold (EO)	0.847	0.515	0.113	0.104
	exponentiated gradient (DP gap = 0)	0.843	0.487	0.052	0.143
	exponentiated gradient (EO gap = 0)	0.848	0.512	0.114	0.088

Table 25: SBM combined with other fair ML methods in Bank Marketing dataset (LIFT)

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.674	0.258	0.543	0.450
	correlation remover	0.890	0.057	0.002	0.006
	optimal threshold (DP)	0.890	0.058	0.002	0.007
	optimal threshold (EO)	0.890	0.066	0.030	0.040
	exponentiated gradient (DP gap = 0)	0.889	0.039	0.000	0.009
	exponentiated gradient (EO gap = 0)	0.890	0.070	0.033	0.051
SBM (w/o OT)	N/A	0.698	0.255	0.088	0.137
	correlation remover	0.836	0.358	0.025	0.114
	optimal threshold (DP)	0.699	0.252	0.002	0.019
	optimal threshold (EO)	0.698	0.253	0.006	0.028
	exponentiated gradient (DP gap = 0)	0.687	0.262	0.014	0.053
	exponentiated gradient (EO gap = 0)	0.654	0.226	0.096	0.107
SBM (OT linear)	N/A	0.892	0.305	0.104	0.121
	correlation remover	0.891	0.304	0.079	0.000
	optimal threshold (DP)	0.891	0.289	0.016	0.094
	optimal threshold (EO)	0.892	0.305	0.103	0.121
	exponentiated gradient (DP gap = 0)	0.893	0.265	0.001	0.093
	exponentiated gradient (EO gap = 0)	0.892	0.305	0.100	0.109
SBM (OT sinkhorn)	N/A	0.698	0.080	0.109	0.072
	correlation remover	0.699	0.081	0.230	0.106
	optimal threshold (DP)	0.697	0.083	0.028	0.011
	optimal threshold (EO)	0.695	0.089	0.174	0.205
	exponentiated gradient (DP gap = 0)	0.681	0.113	0.032	0.063
	exponentiated gradient (EO gap = 0)	0.691	0.124	0.041	0.036

Table 26: SBM combined with other fair ML methods in CivilComments dataset

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.854	0.223	0.560	0.546
	correlation remover	0.886	0.000	0.000	0.000
	optimal threshold (DP)	0.886	0.000	0.000	0.000
	optimal threshold (EO)	0.886	0.000	0.000	0.000
	exponentiated gradient (DP gap = 0)	0.886	0.000	0.000	0.000
	exponentiated gradient (EO gap = 0)	0.886	0.000	0.000	0.000
SBM (w/o OT)	N/A	0.879	0.068	0.048	0.047
	correlation remover	0.878	0.062	0.010	0.030
	optimal threshold (DP)	0.880	0.054	0.001	0.015
	optimal threshold (EO)	0.880	0.061	0.019	0.010
	exponentiated gradient (DP gap = 0)	0.881	0.046	0.002	0.008
	exponentiated gradient (EO gap = 0)	0.880	0.059	0.018	0.010
SBM (OT linear)	N/A	0.880	0.070	0.042	0.039
	correlation remover	0.879	0.056	0.011	0.028
	optimal threshold (DP)	0.880	0.060	0.001	0.017
	optimal threshold (EO)	0.880	0.063	0.013	0.002
	exponentiated gradient (DP gap = 0)	0.882	0.043	0.002	0.008
	exponentiated gradient (EO gap = 0)	0.882	0.039	0.006	0.001
SBM (OT sinkhorn)	N/A	0.882	0.047	0.028	0.026
	correlation remover	0.879	0.057	0.011	0.029
	optimal threshold (DP)	0.882	0.040	0.000	0.011
	optimal threshold (EO)	0.882	0.042	0.010	0.000
	exponentiated gradient (DP gap = 0)	0.881	0.045	0.001	0.008
	exponentiated gradient (EO gap = 0)	0.880	0.056	0.016	0.005

Table 27: SBM combined with other fair ML methods in HateXplain dataset

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.584	0.590	0.171	0.133
	correlation remover	0.555	0.557	0.007	0.031
	optimal threshold (DP)	0.539	0.515	0.005	0.047
	optimal threshold (EO)	0.573	0.573	0.129	0.090
	exponentiated gradient (DP gap = 0)	0.562	0.561	0.006	0.055
	exponentiated gradient (EO gap = 0)	0.579	0.586	0.130	0.093
SBM (w/o OT)	N/A	0.592	0.637	0.159	0.138
	correlation remover	0.563	0.616	0.033	0.053
	optimal threshold (DP)	0.586	0.660	0.013	0.006
	optimal threshold (EO)	0.538	0.561	0.034	0.074
	exponentiated gradient (DP gap = 0)	0.581	0.638	0.039	0.013
	exponentiated gradient (EO gap = 0)	0.580	0.630	0.047	0.095
SBM (OT linear)	N/A	0.606	0.670	0.120	0.101
	correlation remover	0.587	0.657	0.057	0.087
	optimal threshold (DP)	0.600	0.683	0.010	0.004
	optimal threshold (EO)	0.563	0.615	0.039	0.071
	exponentiated gradient (DP gap = 0)	0.600	0.673	0.029	0.011
	exponentiated gradient (EO gap = 0)	0.593	0.669	0.044	0.087
SBM (OT sinkhorn)	N/A	0.612	0.687	0.072	0.037
	correlation remover	0.587	0.668	0.073	0.105
	optimal threshold (DP)	0.607	0.694	0.002	0.031
	optimal threshold (EO)	0.572	0.696	0.201	0.182
	exponentiated gradient (DP gap = 0)	0.598	0.683	0.005	0.020
	exponentiated gradient (EO gap = 0)	0.585	0.672	0.070	0.093

Table 28: SBM combined with other fair ML methods in CelebA dataset

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.866	0.879	0.308	0.193
	correlation remover	0.845	0.862	0.099	0.066
	optimal threshold (DP)	0.816	0.845	0.009	0.035
	optimal threshold (EO)	0.789	0.793	0.196	0.033
	exponentiated gradient (DP gap = 0)	0.781	0.814	0.008	0.006
	exponentiated gradient (EO gap = 0)	0.838	0.854	0.205	0.025
SBM (w/o OT)	N/A	0.870	0.883	0.309	0.192
	correlation remover	0.849	0.865	0.095	0.066
	optimal threshold (DP)	0.819	0.848	0.009	0.038
	optimal threshold (EO)	0.792	0.798	0.194	0.030
	exponentiated gradient (DP gap = 0)	0.783	0.818	0.006	0.007
	exponentiated gradient (EO gap = 0)	0.841	0.857	0.206	0.029
SBM (OT linear)	N/A	0.870	0.883	0.306	0.185
	correlation remover	0.849	0.866	0.096	0.066
	optimal threshold (DP)	0.819	0.848	0.010	0.034
	optimal threshold (EO)	0.792	0.798	0.193	0.023
	exponentiated gradient (DP gap = 0)	0.783	0.818	0.007	0.008
	exponentiated gradient (EO gap = 0)	0.841	0.857	0.203	0.026
SBM (OT sinkhorn)	N/A	0.872	0.885	0.306	0.184
	correlation remover	0.851	0.867	0.097	0.062
	optimal threshold (DP)	0.821	0.850	0.010	0.035
	optimal threshold (EO)	0.795	0.801	0.193	0.023
	exponentiated gradient (DP gap = 0)	0.784	0.819	0.008	0.010
	exponentiated gradient (EO gap = 0)	0.840	0.857	0.200	0.022

Table 29: SBM combined with other fair ML methods in UTKFace dataset

	Fair ML method	Acc	F1	Δ_{DP}	Δ_{EO}
WS (Baseline)	N/A	0.791	0.791	0.172	0.073
	correlation remover	0.787	0.786	0.034	0.051
	optimal threshold (DP)	0.774	0.767	0.040	0.215
	optimal threshold (EO)	0.788	0.786	0.114	0.005
	exponentiated gradient (DP gap = 0)	0.769	0.762	0.029	0.078
	exponentiated gradient (EO gap = 0)	0.792	0.790	0.126	0.026
SBM (w/o OT)	N/A	0.797	0.790	0.164	0.077
	correlation remover	0.791	0.793	0.006	0.091
	optimal threshold (DP)	0.764	0.791	0.033	0.202
	optimal threshold (EO)	0.789	0.791	0.165	0.077
	exponentiated gradient (DP gap = 0)	0.760	0.791	0.024	0.069
	exponentiated gradient (EO gap = 0)	0.791	0.792	0.158	0.076
SBM (OT linear)	N/A	0.800	0.793	0.135	0.043
	correlation remover	0.799	0.791	0.004	0.098
	optimal threshold (DP)	0.785	0.772	0.034	0.195
	optimal threshold (EO)	0.800	0.793	0.128	0.038
	exponentiated gradient (DP gap = 0)	0.779	0.764	0.024	0.069
	exponentiated gradient (EO gap = 0)	0.797	0.789	0.123	0.030
SBM (OT sinkhorn)	N/A	0.804	0.798	0.130	0.041
	correlation remover	0.799	0.794	0.012	0.087
	optimal threshold (DP)	0.789	0.777	0.036	0.195
	optimal threshold (EO)	0.799	0.794	0.168	0.046
	exponentiated gradient (DP gap = 0)	0.776	0.764	0.023	0.068
	exponentiated gradient (EO gap = 0)	0.801	0.796	0.141	0.044