Post-processing for Individual Fairness

Anonymous Author(s)
Affiliation
Address
email

Abstract

Post-processing in algorithmic fairness is a versatile approach for correcting bias in ML systems that are already used in production. The main appeal of post-processing is that it avoids expensive retraining. In this work, we propose a suite of general post-processing algorithms for individual fairness (IF). We consider a setting where the learner only has access to the predictions of the original model and a similarity graph between individuals guiding the desired fairness constraints. We cast the IF post-processing problem as a graph smoothing problem corresponding to graph Laplacian regularization that preserves the desired "treat similar individuals similarly" interpretation. Our theoretical results demonstrate the connection of the new objective function to a local relaxation of the original individual fairness. Empirically, our post-processing algorithms correct individual biases in large scale NLP models, e.g., BERT, while preserving accuracy.

1 Introduction

There are many instances of algorithmic bias in machine learning (ML) models [1, 24, 6, 3], which has led to the development of methods for *quantifying* and *correcting* algorithmic biases. To quantify algorithmic biases, researchers have proposed numerous mathematical definitions of algorithmic fairness. Broadly speaking, these definitions fall into two categories: *group fairness* [9] and *individual fairness* [14]. The former formalizes the idea that ML system should treat certain *groups* of individuals similarly, e.g., requiring the average loan approval rate for applicants of different ethnicities be similar [18]. The latter asks for similar treatment of similar *individuals*, e.g., same outcome for applicants with resumes that differ say only in names [4]. Researchers have also developed many ways of correcting algorithmic biases. These fairness interventions broadly fall into three categories: pre-processing the data, enforcing fairness during model training (also known as in-processing), and post-processing the outputs of a model.

While both group and individual fairness (IF) definitions have their benefits and drawbacks [14, 9, 15], the existing suite of algorithmic fairness solutions mostly enforces group fairness. The few prior works on individual fairness are all in-processing methods [21, 37, 36, 33]. Although in-processing is arguably the most-effective type of intervention, it has many practical limitations. First, it requires training models from scratch. Nowadays, it is more common to fine-tune publicly available models (e.g., language models such as BERT [12] and GPT-3 [5]) than to train models afresh, as many practitioners do not have the necessary computational resources. Even with enough computational resources, training large deep learning models has a significant environmental impact [32, 3]. Post-processing offers an easier path towards incorporating algorithmic fairness into deployed ML models, and has potential to reduce environmental harm from re-training with in-processing fairness techniques.

In this paper, we propose a computationally efficient suite of methods for post-processing off-the-shelf models to be *individually fair*. We consider a setting where we are given the outputs of a (possibly unfair) ML model on a set of n individuals, and side information about their similarity for the ML task

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

at hand, which can either be obtained using a fair metric on the input space or from some human annotator. Our starting point is a post-processing version of the algorithm of Dwork et al. [14] (see (2.3)). Unfortunately, this method has two drawbacks: poor scalability and a sharp trade-off with accuracy. As we shall see, the sharp trade-off is due to the restrictions imposed on dissimilar individuals by Dwork et al. [14]'s global Lipschitz continuity condition. By relaxing these restrictions on dissimilar individuals, we obtain a better trade-off between accuracy and fairness while preserving the intuition of treating *similar* individuals similarly. This leads us to consider a graph signal-processing approach to IF post-processing that only enforces similar outputs between similar individuals. The nodes in the underlying graph correspond to individuals, edges (possibly weighted) indicate similarity, and the signal on the graph is the output of the model on the corresponding node-individuals. To enforce IF, we use Laplacian regularization [8], which encourages the signal to be smooth on the graph. We illustrate this idea in Figure 1: an algorithm decides whom to show a job ad based on their CVs and chooses Bob and Joy. Alice and Bob have similar qualification, i.e., they are both Ruby experts, and should be treated similarly to satisfy IF. We represent all four candidates as nodes in a graph, where node signal (tick or cross) is the algorithm's decision for the corresponding candidate. As Alice and Bob are connected via an edge, for this graph to be smooth, their node signals need to be similar. On the contrary, directly enforcing IF constraints [14] requires certain degree of output similarity on *all* pairs of candidates. Our main contributions are summarized below.

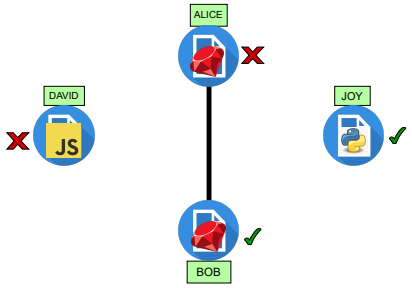

Figure 1: IF on a graph

1. We cast post-processing for individual fairness as a graph smoothing problem. We also propose a coordinate descent algorithm to scale the approach to large datasets where memory availability is a limiting factor;
2. We demonstrate theoretically and verify empirically that graph smoothing enforces individual fairness constraints *locally*, i.e., it guarantees similar treatment of *similar* individuals.
3. We empirically compare the Laplacian smoothing method to the post-processing adaptation of the algorithm of Dwork et al. [14] enforcing global Lipschitz continuity. The Laplacian smoothing method is not only computationally more efficient but also more effective in reducing algorithmic bias, and preserves accuracy of the original model.
4. We demonstrate the efficacy of Laplacian smoothing on two large-scale text datasets by reducing biases in fine-tuned BERT models.

2 Post-processing Problem Formulation

Let $\mathcal{X}$ be the feature space, $\mathcal{Y}$ be the set of possible labels/targets, and $h : \mathcal{X} \to \mathcal{Y}$ be a (possibly unfair) ML model trained for the task. Our goal is to post-process the outputs of h so that they are individually fair. Formally, the post processor is provided with a set of inputs $\{x_i\}_{i=1}^n$ and the outputs of h on the inputs $\{\widehat{y}_i \triangleq h(x_i)\}_{i=1}^n$, and its goal is to produce $\{\widehat{f}_i\}_{i=1}^n$ that is both individually fair and similar to the $\widehat{y}_i$'s. Recall that individual fairness of h is the Lipschitz continuity of h with respect to a fair metric $d_{\mathcal{X}}$ on the input space:

$$d_{\mathcal{Y}}(h(x_1), h(x_2)) \leq L d_{\mathcal{X}}(x_1, x_2) \text{ for all } x_1, x_2 \in \mathcal{X}, \tag{2.1}$$

where $L > 0$ is a Lipschitz constant. The fair metric encodes problem-specific intuition of which samples should be treated similarly by the ML model. It is analogous to the knowledge of protected attributes in group fairness needed to define corresponding fairness constraints. Recent literature proposes several practical methods for learning fair metric from data [20, 27]. We assume the post-processor is either given access to the fair metric (it can evaluate the fair distance on any pair of points in $\mathcal{X}$), or receives feedback on which inputs should be treated similarly. We encode this information in an adjacency matrix $W \in \mathbb{R}^{n \times n}$ of a graph with individuals as nodes. If the post-processor is given the fair metric, then the entries of W are

$$W_{ij} = \begin{cases} \exp(-\theta d_{\mathcal{X}}(x_i, x_j)^2) & d_{\mathcal{X}}(x_i, x_j) \leq \tau \\ 0 & \text{otherwise} \end{cases}, \tag{2.2}$$

where $\theta > 0$ is a scale parameter and $\tau > 0$ is a threshold parameter. If the post-processor is given an annotator's feedback, then W is a binary matrix with $W_{ij} = 1$ if i and j are considered to be treated similarly by the annotator and 0 otherwise. Extensions to multiple annotators are straightforward.

We start with a simple post-processing adaptation of the algorithm of Dwork et al. [14] for enforcing individual fairness, that projects the (possibly unfair) outputs of h onto a constraint set to enforce (2.1). In order words, the post-processor seeks the closest set of outputs to the $\widehat{y}_i$'s that satisfies individual fairness:

$$\{\widehat{f}_i\}_{i=1}^n \in \begin{Bmatrix} \arg\min_{f_1,\ldots,f_n} & \sum_{i=1}^n \frac{1}{2} d_{\mathcal{Y}}(f_i, \widehat{y}_i)^2 \\ \text{subject to} & d_{\mathcal{Y}}(f_i, f_j) \leq L d_{\mathcal{X}}(x_i, x_j) \end{Bmatrix}. \tag{2.3}$$

This objective function, though convex, scales poorly due to the order of n^2 constraints. Empirically we observe that (2.3) leads to post-processed outputs that are dissimilar to the $\widehat{y}_i$'s, leading to poor performance in practice. The goal of our method is to improve performance and scalability, while preserving the IF desiderata of treating *similar* individual similarly. Before presenting our method, we discuss other post-processing perspectives that differ in their applicability and input requirements.

2.1 Alternative Post-processing Formulations

We review three post-processing problem setups and the corresponding methods in the literature. First, one can fine-tune a model via an in-processing algorithm to reduce algorithmic biases. Yurochkin and Sun [36] proposed an in-processing algorithm for IF and used it to train fair models for text classification using sentence BERT embeddings. This setting is the most demanding in terms of input and computational requirements: a user needs access to the original model parameters, fair metric function, and train a predictor, e.g., a moderately deep fully connected neural network, with a non-trivial fairness-promoting objective function.

Second, it is possible to post-process by training additional models to correct the initial model's behavior. For example, Kim et al. [22] propose a boosting-based method for group fairness post-processing. This perspective can be adapted to individual fairness, however it implicitly assumes that we can train weak-learners to boost. Lohia et al. [26], Lohia [25] propose to train a bias detector to post-process for group fairness and a special, group based, notion of individual fairness. Such methods are challenging to apply to text data or other non-tabular data types.

The third perspective is the most generic: a user has access to original model outputs only, and a minimal additional feedback guiding fairness constraints. Wei et al. [34] consider such setting and propose a method to satisfy group fairness constraints, however it is not applicable to individual fairness. Our problem formulation belongs to this post-processing setup. The main benefit of this approach is its broad applicability and ease of deployment.

3 Graph Laplacian Individual Fairness

To formulate our method, we cast IF post-processing as a graph smoothing problem. Using the fair metric or human annotations as discussed in Section 2, we obtain an $n \times n$ matrix W that we treat as an adjacency matrix. As elaborated earlier, the goal of post-processing is to obtain a model f that is individually fair and accurate. The accuracy is achieved by minimizing the distance between the outputs of f and h, a pre-trained model assumed to be accurate but possibly biased. Recall that we don't have access to the parameters of h, but can evaluate its predictions. Our method enforces fairness using a graph Laplacian quadratic form [31] regularizer:

$$\widehat{\mathbf{f}} = \arg\min_f \; g_\lambda(\mathbf{f}) = \arg\min_f \; \|\mathbf{f} - \hat{\mathbf{y}}\|_2^2 + \lambda \, \mathbf{f}^\top \mathbb{L}_n \mathbf{f}, \tag{3.1}$$

where $\mathbf{f}$ is the vector of the post-processed outputs, i.e., $f_i = f(x_i)$ for $i = 1, \ldots, n$. The matrix $\mathbb{L}_n \in \mathbb{R}^{n \times n}$ is called *graph Laplacian* matrix and is a function of W. There are multiple versions of $\mathbb{L}_n$ popularized in the graph literature (see e.g., [1] or [10]). To elucidate connection to individual fairness, consider unnormalized Laplacian $\mathbb{L}_{un,n} = D - W$, where $D_{ii} = \sum_{j=1}^n W_{ij}$, $D_{ij} = 0$ for $i \neq j$ is the *degree matrix* corresponding to W. Then a known identity is:

$$\mathbf{f}^\top \mathbb{L}_{un,n} \mathbf{f} = \tfrac{1}{2} \textstyle\sum_{i \neq j} W_{ij} \left(f_i - f_j\right)^2. \tag{3.2}$$

Hence the Laplacian regularizer is small when for large W_{ij} (i.e., individuals i and j are similar), post-processed model outputs f_i and f_j (i.e., treatment) are similar, promoting the philosophy of individual fairness "treat similar individuals similarly". This observation intuitively explains the motivation for minimizing graph Laplacian quadratic form to achieve individual fairness. In Section 4 we present a more formal discussion on the connections between quadratic graph Laplacian quadratic regularization and individual fairness.

Our post-processing problem (3.1) is easy to solve: setting the gradient of g_λ to 0 implies that the optimal solution $\widehat{\mathbf{f}}$ is:

$$\widehat{\mathbf{f}} = \left(I + \lambda \left(\frac{\mathbb{L}_n + \mathbb{L}_n^\top}{2} \right) \right)^{-1} \widehat{\mathbf{y}} \,. \tag{3.3}$$

The Laplacian $\mathbb{L}_n$ is a positive semi-definite matrix ensuring that (3.1) is strongly convex and that (3.3) is a global minima. Comparing to the computationally expensive constraint optimization problem (2.3), this approach has a simple closed-form expression.

Note that the symmetry of the unnormalized Laplacian $\mathbb{L}_{un,n}$ simplifies (3.3), however there are also non-symmetric Laplacian variations. In this work, we also consider the normalized random walk Laplacian $\mathbb{L}_{nrw,n} = \left(I - \widetilde{D}^{-1} \widetilde{W} \right)$, where $\widetilde{W} = D^{-1/2} W D^{-1/2}$ is the normalized adjacency matrix and $\widetilde{D}$ is its degree matrix. We discuss its properties in the context of IF in Section 4. Henceforth, we refer to our method as Graph Laplacian Individual Fairness (GLIF) when using the unnormalized Laplacian, and GLIF-NRW when using Normalized Random Walk Laplacian.

3.1 Prior Work on Graph Laplacians

Graph based learning via a similarity matrix is prevalent in statistics and ML literature, specifically, in semi-supervised learning. The core idea is to gather information from similar unlabeled inputs to improve prediction accuracy (e.g., see [38], [2], [30] and references therein). Laplacian regularization is widely used in science engineering. We refer to Chapelle et al. [8] for a survey.

3.2 Extensions of the Basic Method

3.2.1 Multi-dimensional Output

We presented our objective function (3.1) and post-processing procedure (3.3) for the case of univariate outputs. This covers regression and binary classification. Our method readily extends to multi-dimensional output space, for example in classification $f_i, \widehat{y}_i \in \mathbb{R}^K$ can represent logits, i.e., softmax inputs, of the K classes. In this case $\mathbf{f}$ and $\hat{\mathbf{y}}$ are $n \times K$ matrices, and the term $\mathbf{f}^\top \mathbb{L}_n \mathbf{f}$ is a $K \times K$ matrix. We use the trace of it as a regularizer. The optimization problem (3.1) then becomes:

$$\widehat{\mathbf{f}} = \arg\min_f \; g_\lambda(\mathbf{f}) = \arg\min_f \; \|\mathbf{f} - \hat{\mathbf{y}}\|_F^2 + \lambda \, \mathsf{tr} \left(\mathbf{f}^\top \mathbb{L}_n \mathbf{f} \right), \tag{3.4}$$

where $\| \cdot \|_F$ is the Frobenious norm. Similar calculation as for the univariate output yields:

$$\widehat{\mathbf{f}} = \left(I + \lambda \left(\tfrac{\mathbb{L}_n + \mathbb{L}_n^\top}{2} \right) \right)^{-1} \widehat{\mathbf{y}} \,. \tag{3.5}$$

The solution is the same as (3.3), however it now accounts for the multi-dimensional outputs.

3.2.2 Coordinate Descent for Large Data

Although our method has a closed form solution, is not immediately scalable as we have to invert a $n \times n$ matrix to obtain the optimal solution. We propose a *coordinate* descent variant of our method that readily scales to any data size. The idea stems primarily from the gradient of equation (3.4), where we solve:

$$\mathbf{f} - \hat{y} + \lambda \frac{\mathbb{L}_n + \mathbb{L}_n^\top}{2} \mathbf{f} = 0 \,. \tag{3.6}$$

Fixing $\{f_j\}_{j \neq i}$, we can solve (3.6) for f_i:

$$f_i \leftarrow \frac{\hat{y}_i - \frac{\lambda}{2} \sum_{j \neq i} (\mathbb{L}_{n,ij} + \mathbb{L}_{n,ji}) f_j}{1 + \lambda \mathbb{L}_{n,ii}}. \tag{3.7}$$

This gives rise to the coordinate descent algorithm. We perform asynchronous updates over randomly selected coordinate batches until convergence. We refer the reader to Wright [35] and the references therein for the convergence properties of (asynchronous) coordinate descent.

3.2.3 Extension to Inductive Setting

This coordinate descent update is also the key to extending our approach to the inductive setting. To handle new unseen points, we assume we have a set of test points on which we have already post-processed the outputs of the ML model. To post-process new unseen points, we simply fix the outputs of the other test points and perform a single coordinate descent step with respect to the output of the new point. Similar strategies are often employed to extend transductive graph-based algorithms to the inductive setting [8].

3.2.4 Alternative Discrepancy Measures on the Output Space

So far we have considered the squared Euclidean distance as a measure of discrepancy between outputs. This is a natural choice for post-processing models with continuous-valued outputs. For models that output a probability distribution over the possible classes, we consider alternative discrepancy measures on the output space. It is possible to replace the squared Euclidean distance with a Bregman divergence with very little change to the algorithm in the case of the unnormalized Laplacian. Below, we work through the details for the KL divergence as a demonstration of the idea.

Suppose the output of the pre-trained model h is $\hat{y}_i \in \Delta^K$, where $\hat{y}_i = \{e^{\tau_{i,j}} / \sum_{k=1}^K e^{\tau_{i,k}}\}_{j=1}^K$ a K-dimensional probability vector corresponding to a K class classification problem ($\boldsymbol{\tau}_i$ is the output of the penultimate layer of the pre-trained model and $\hat{y}_i$ is obtained by passing it through the soft-max layer) and $\Delta^K = \{x \in \mathbb{R}^K : x_i \geq 0, \sum_{i=1}^K x_i = 1\}$ is the probability simplex in $\mathbb{R}^K$. Denote by P_v, the multinomial distribution with success probabilities v for any $v \in \Delta^k$. Define $\hat{\eta}_i \in \mathbb{R}^{K-1}$ (resp. η_i) as the natural parameter corresponding to $\hat{y}_i$ (resp. f_i), i.e., $\hat{\eta}_{i,j} = \log\left(\hat{y}_{i,j}/\hat{y}_{i,K}\right) = \tau_{i,j} - \tau_{i,K}$ for $1 \leq j \leq K-1$. The (unnormalized) Laplacian smoothing problem with the KL divergence is

$$\tilde{y}_i = \arg\min_{y \in \Delta^K} \left[KL\left(P_y || P_{\hat{y}_i}\right) + \frac{\lambda}{2} \sum_{j=1, j\neq i}^n W_{ij} KL\left(P_y || P_{y_j}\right) \right]. \tag{3.8}$$

The following theorem establishes that (3.5) solves the above problem in the logit space, or equivalently in the space of the corresponding natural parameters (see Appendix 1 for the proof):

Theorem 3.1. *Consider the following optimization problem on the space of natural parameters:*

$$\tilde{\eta}_i = \arg\min_{\eta} \left[\|\eta - \hat{\eta}_i\|^2 + \frac{\lambda}{2} \sum_{j=1, j\neq i}^n W_{ij} \|\eta_j - \eta_i\|^2 \right]. \tag{3.9}$$

Then, the minimizer $\tilde{\eta}_i$ of equation (3.9) *is the natural parameter corresponding to the minimizer $\tilde{y}_i$ of* (3.8)*.*

4 Local IF and Graph Laplacian Regularization

In this section we provide theoretical insights to understand why graph Laplacian regularizer enforces individual fairness. It is pointed out in section 2 that enforcing IF globally is expensive and often reduces a significant amount of accuracy of the final classifier. Here we establish that solving (3.1) is tantamount to enforcing a localized version of individual fairness, namely *Local Individual Fairness*, which is defined below:

Definition 4.1 (Local Individual Fairness)**.** *An ML model h is said to be* locally individually fair *if it satisfies:*

$$\mathbb{E}_{X \sim P} \left[\limsup_{y : d_{\mathcal{X}}(X,y) \downarrow 0} \frac{d_{\mathcal{Y}}(h(X), h(y))}{d_{\mathcal{X}}(X,y)} \right] < \infty .$$

Example 4.2. *For our theoretical analysis, we need to specify a functional form of the fair metric. A popular choice is a Mahalanobis fair metric proposed by [27], which is defined as:*

$$d^2_{\mathcal{X}}(x_1, x_2) = (x_1 - x_2)^\top \Sigma (x_1 - x_2)$$

where Σ is a dispersion matrix that puts lower weight in the directions of sensitive attributes and higher weight in the directions of relevant attributes. [27] also proposed several algorithms to

learn such a fair metric from the data. If we further assume $d_{\mathcal{Y}}(y_1, y_2) = |y_1 - y_2|$, then a simple application of Lagrange's mean value theorem yields for any realization of X:

$$\limsup_{y: d_{\mathcal{X}}(X,y)\downarrow 0} \frac{|h(X)-h(y)|}{d_{\mathcal{X}}(X,y)} \leq \|\Sigma^{-1/2}\nabla h(X)\|\,.$$

This immediately implies:

$$\mathbb{E}_{X\sim P}\left[\limsup_{y: d_{\mathcal{X}}(X,y)\downarrow 0} \frac{d_{\mathcal{Y}}(h(X),h(y))}{d_{\mathcal{X}}(X,y)}\right] \leq \mathbb{E}[\|\Sigma^{-1/2}\nabla h(X)\|]\,,$$

i.e., h satisfies local individual fairness *constraint as long as $\mathbb{E}[\|\Sigma^{-1/2}\nabla h(X)\|] < \infty$. On the other hand, the global IF constraint necessitates $\sup_{x\in\mathcal{X}} \|\Sigma^{-1/2}\nabla h(x)\| < \infty$, i.e., h is Lipschitz continuous with respect to Mahalanobis distance.*

The main advantage of this local notion of IF over its global counterpart is that the local definition concentrates on the input pairs with smaller fair distance and ignores those with larger distance. For example, in Figure 2, the edge-weight between Bob and Alice is much larger than any other pairs, therefore this local notion strongly enforces fairness constraint on that pair, while ignoring (or being less stringent on) others. This prevents over-smoothing and consequently preserves accuracy while enforce fairness as is evident from our real data experiment in Section 5.

We now present our main result, which establishes that, under certain assumptions on the underlying hypothesis class and the distribution of inputs, the graph Laplacian (both unnormalized and normalized random walk) regularizer enforces local IF constraint (as defined in definition 4.1) in the limit. For our theory, we work with $d_{\mathcal{X}}$ as the Mahalanobis distance introduced in Example 4.2 in equation (2.2) along with $\theta = 1/(2h^2)$ (h is a bandwidth parameter which goes to 0 at an appropriate rate as $n \to \infty$) and $\tau = \infty$. All our results will be thorough for any finite τ but with more tedious technical analysis. Therefore our weight matrix W becomes:

$$W_{ij} = \frac{|\Sigma|^{1/2}}{(2\pi)^{d/2}h^d} e^{-\frac{1}{2h^2}(x_i-x_j)^\top\Sigma(x_i-x_j)}$$

The constant $|\Sigma|^{1/2}/((2\pi)^{d/2}h^d)$ is for the normalization purpose and can be absorbed into the penalty parameter λ. We start by listing our assumptions:

Assumption 4.3 (Assumption on the domain)**.** *The domain of the inputs $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$ where d is the underlying dimension.*

Assumption 4.4 (Assumption on hypothesis)**.** *All functions $f \in \mathcal{F}$ of the hypothesis class satisfy the following:*

1. *The i^{th} derivative $f^{(i)}$ is uniformly bounded over the domain $\mathcal{X}$ of inputs for $i \in \{0, 1, 2\}$.*
2. *$f^{(1)}(x) = 0$ for all $x \in \partial X$, where ∂X denotes the boundary of $\mathcal{X}$.*

Assumption 4.5 (Assumption on density of inputs)**.** *The density p of X on the domain $\mathcal{X}$ satisfies the following:*

1. *There exists $p_{\max} < \infty$ and $p_{\min} > 0$ such that for all $x \in \mathcal{X}$, we have $p_{\min} \leq p(x) \leq p_{\max}$.*
2. *The derivatives of the density $p^{(i)}$ is uniformly bounded on the domain $\mathcal{X}$ for $i \in \{0, 1, 2\}$.*

Discussion on the assumptions Most of our assumptions (e.g., compactness of the domain, bounded derivatives of f or p) are for technical simplicity and are fairly common for the asymptotic analysis of graph regularization (see, e.g., Hein *et al.* [16, 1] and references therein). It is possible to relax some of the assumptions: for example, if the domain $\mathcal{X}$ of inputs is unbounded, then the target function f and the density p should decay at certain rate so that observations far away will not be able to affect the convergence (e.g., sub-exponential tails). Part (ii) of Assumption 4.4 can be relaxed if we assume $p(x)$ is 0 at boundary. However we don't pursue these extensions further in this manuscript as they are purely technical and do not add anything of significance to the main intuition of the result.

Theorem 4.6. *Under Assumptions 4.3 - 4.5, we have:*

1. *If the sequence of bandwidths $h \equiv h_n \downarrow 0$ such that $nh^2 \to \infty$ and $\mathbb{L}_{un,n}$ is unnormalized Laplacian matrix, then*

$$\tfrac{2}{n^2h^2}\mathbf{f}^\top\mathbb{L}_{un,n}\mathbf{f} \xrightarrow{P} \mathbb{E}\left[\nabla f(X)^\top\Sigma^{-1}\nabla f(X)\, p(X)\right]\,.$$

2. *If the sequence of bandwidths $h \equiv h_n \downarrow 0$ such that $(nh^{d+4})/(\log(1/h)) \to \infty$ and $\mathbb{L}_{nrw,n}$ is the normalized random walk Laplacian matrix, then:*

$$\tfrac{1}{nh^2}\mathbf{f}^\top \mathbb{L}_{nrw,n}\mathbf{f} \xrightarrow{P} \mathbb{E}\left[\nabla f(X)^\top \Sigma^{-1} \nabla f(X)\right] .$$

where $\mathbf{f} = \{f(x_i)\}_{i=1}^n$. Consequently both the Laplacian regularizers asymptotically enforce weak local IF.

The proof of the above theorem can be found in Appendix 1. If we use normalized random walk graph Laplacian matrix $\mathbb{L}_{nrw,n}$ as regularizer, then (asymptotically) it penalizes $\mathbb{E}\left[\nabla f(X)^\top \Sigma^{-1} \nabla f(X)\right] = \mathbb{E}\left[\|\Sigma^{-1/2}\nabla f(X)\|^2\right]$, which, by Example 4.2, is equivalent to enforcing local IF constraint. Similarly, the un-normalized Laplacian matrix $\mathbb{L}_{un,n}$, also enforces the same under Assumption 4.5 as:

$$\mathbb{E}\left[\|\Sigma^{-1/2}\nabla f(X)\|^2\right] \leq \tfrac{1}{p_{\min}} \mathbb{E}\left[\nabla f(X)^\top \Sigma^{-1} \nabla f(X)\, p(X)\right], \text{ where } p_{\min} = \inf_{x \in \mathcal{X}} p(x).$$

Although both the Laplacian matrices enforce local IF, the primary difference between them is that the limit of the unnormalized Laplacian involves the density $p(x)$, i.e., it upweights the high-density region (consequently stringent imposition of fairness constraint), whereas down-weights the under-represented/low-density region. On the other hand, the limit corresponding to the normalized random walk Laplacian matrix does not depend on $p(x)$ and enforces fairness constraint with equal intensity on the entire input space. It is not immediately clear in what situation one should be preferable to the other, however we used both the regularizers in our experiments to compare and contrast their performance on several practical ML problems.

5 Empirical Studies

The goals of our experimental studies are threefold: (1) Explore the trade-offs between post-processing for local individual fairness with GLIF and post-processing with individual fairness constraints following our adaptation of the Dwork et al. [14] algorithm described in (2.3). We use CVXPY [13] to solve (2.3) and call this method IF-constraints. Due to its poor scalability, we consider a smaller dataset for this study; (2) study practical implications of theoretical differences between GLIF and GLIF-NRW, i.e., different graph Laplacians, presented in Section 4; (3) evaluate the effectiveness of GLIF in its main application, i.e., computationally light debiasing of large deep learning models such as BERT.

5.1 Comparing GLIF and IF-constraints

For this experiment we consider the sentiment prediction task [19], i.e., classifying words as positive or negative. The baseline model is a neural network trained with GloVe word embeddings [29]. Yurochkin et al. [37] evaluate such classifier on a set of human names typical for Caucasian and African-American ethnic groups [7] and show that it tends to assign drastically different sentiment scores to the names. An individually fair model should assign similar sentiment scores to all names. Yurochkin et al. [37] propose a fair metric learning procedure for this task using a side dataset of names, and an in-processing technique for achieving individual fairness. We use their method to obtain the fair metric and compare post-processing of the baseline model with GLIF and IF-constraints. The test set consists of 663 sentiment words from the original task and 94 names. Post-processing methods are applied on the concatenated (unlabeled) sentiment words and the names, i.e., no problem specific knowledge is used. The resulting post-processed outputs on sentiment words are used to evaluate accuracy, and outputs on names for evaluating fairness metrics. Even for this small problem, IF-constraints, i.e., CVXPY implementation of (2.3), takes 5 minutes to run. For GLIF(-NRW), we implement the closed-form solution (3.3) that takes less than a second to run. See Appendix 2 for additional experimental details.

We evaluate fairness-accuracy trade-off for a grid of hyperparameters in Figure 2. The left figure shows standard deviation of the post-processed outputs on all names as a function of test accuracy on the original sentiment task. Lower values imply that all names received similar predictions, which is the goal of individual fairness. The center figure visualizes group fairness and accuracy, i.e., difference in average name sentiment scores for the two ethnic groups. In this problem, individual fairness is a stronger notion of fairness: achieving similar predictions for all names implies similar

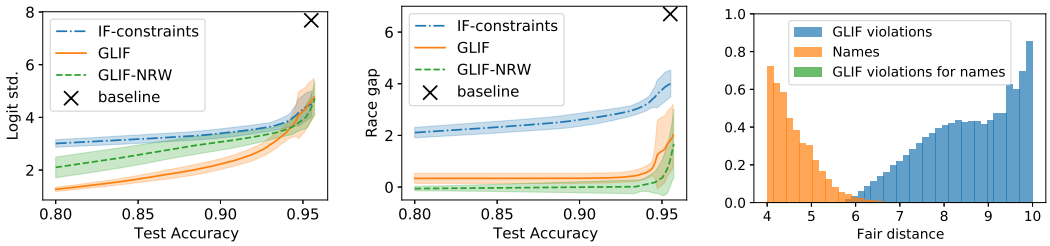

Figure 2: Sentiment experiment. Left: Trade-off between standard deviations of logits of names measuring individual fairness and accuracy. Center: Trade-off between race gap measuring group fairness and accuracy. Right: Frequencies of violations of the IF constraints, frequency of constraints corresponding to names, and frequency of violations for names; most violations are not among the names.

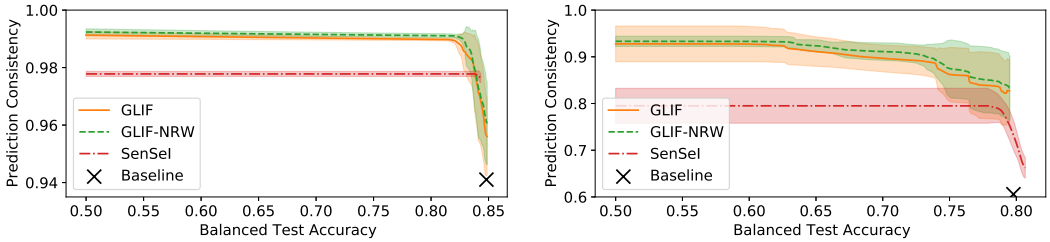

Figure 3: Accuracy-Consistency trade-offs for Bios and Toxicity.

group averages, but not vice a versa. Therefore, for this task, post-processing for individual fairness can also correct group disparities.

Both GLIF and GLIF-NRW achieve significantly better fairness metrics for the same levels of test accuracy in comparison to IF-constraints. To understand the reason, we study which IF constraints are violated by GLIF. There are $n(n-1)/2$ unique constraints in (2.3), and IF-constraints satisfies all of them by design. Each constraint corresponds to a fair metric value, which we bin and present the proportion of constraints violated by GLIF for each bin in Figure 2 (right). We set $L = 2.25$ in (2.3) corresponding to 89.4% accuracy of IF-constraints and show constraint violations of GLIF corresponding to 95% accuracy. First, note the effect of enforcing *local* individual fairness demonstrated in our theoretical analysis in Section 4. GLIF does not violate any constraints corresponding to small fair distances, i.e., it satisfies IF on similar individuals, while violating many large fair distance constraints. Figure 2 (right) also shows that majority of constraints corresponding to small fair distances correspond to pairs of names. This is expected in this task because we consider all names similar, so fair distances between them should be small. To summarize, GLIF ignores unnecessary (in the context of this problem) constraints allowing it to achieve higher accuracy, while satisfying the more relevant local IF constraints (the green area in the figure is tiny) leading to improved fairness metrics.

We comment on the practical differences between GLIF and GLIF-NRW. In Figure 2 (left) GLIF has smaller standard deviation on the name outputs, but in the center plot GLIF-NRW achieves lower race gap. In Theorem 4.6, we showed that GLIF penalizes fairness violations in high density data regions stronger. As a result, GLIF may favor enforcing similar outputs in the high density region causing lower standard deviation, while leaving outputs nearly unchanged in the lower density region, resulting in larger race gaps. GLIF-NRW weighs all data density regions equally, i.e., it is less likely to miss a small subset of names, but is less stringent in the high density regions.

5.2 Post-processing to Debias Large Language Models

Large language models have achieved impressive results on many tasks, however there is also significant evidence demonstrating that they are prone to biases [23, 28, 3]. Debiasing these models remains largely an open problem: most in-processing algorithms are not applicable or computationally prohibitive due to large and highly complex model architectures, and challenges in handling text

Table 1: Results for the Bios task.

Method	Test Acc.	Pred. Consist.
Baseline	$\mathbf{0.846} \pm 0.003$	0.942 ± 0.002
GLIF	0.830 ± 0.004	0.986 ± 0.002
GLIF-NRW	0.834 ± 0.003	$\mathbf{0.988} \pm 0.002$
SenSEI	0.843 ± 0.003	0.977 ± 0.001

Table 2: Results for the Toxicity task.

Method	Test Acc.	Pred. Consist.
Baseline	$\mathbf{0.809} \pm 0.004$	0.614 ± 0.013
GLIF	0.803 ± 0.003	0.835 ± 0.012
GLIF-NRW	0.803 ± 0.003	$\mathbf{0.844} \pm 0.013$
SenSEI	0.791 ± 0.005	0.773 ± 0.043

inputs. Even if an appropriate in-processing algorithm arises, significant environmental impact due to re-training is unavoidable [32, 3]. In our experiments we evaluate effectiveness of GLIF as a simple post-processing technique to debias BERT-based models for text classification. Another possible solution is to fine-tune BERT with an in-processing technique as was done by Yurochkin and Sun [36]. The two approaches are not directly comparable: fine-tuning with SenSeI [36] requires knowledge of the model parameters, alleviates only part of the computational burden, and has more stringent requirements on the fair metric, while post-processing with GLIF is transductive, i.e., it requires access to unlabeled test data (see extended discussion in Section 2.1).

We replicate the experiments of Yurochkin and Sun [36] on Bios [11] and Toxicity[1] datasets. They use the approach of Mukherjee et al. [27] for fair metric learning which we reproduce. We refer to the Appendix B.1 of Yurochkin and Sun [36] for details. In both tasks, following Yurochkin and Sun [36], we quantify performance with balanced accuracy due to class imbalance, and measure individual fairness via *prediction consistency*, i.e., the fraction of test points where the prediction remains unchanged when performing task-specific input modifications.

In Bios the goal is to predict occupation of a person based on their textual biography. Such models can be useful for recruiting purposes. However, due to historical gender bias in some occupations, the baseline BERT model learns to associate gender pronouns and names with the corresponding occupations. Individual fairness is measured with prediction consistency with respect to gender pronouns and names alterations. We present the fairness-accuracy trade-off in Figure 3 for a grid of hyperparameters, and compare performance based on hyperparameter values selected with a validation data in Table 1. Both GLIF and GLIF-NRW noticeably improve individual fairness measured with prediction consistency, while retaining most of the accuracy.

In Toxicity the task is to identify toxic comments—an important tool for facilitating inclusive discussions online. Baseline BERT model learns to associate certain identity words, e.g., "gay", with toxicity, because they are often abused in online conversations. Prediction consistency is with respect to changes to identity words in the inputs. There are 50 identity words, e.g., "gay", "muslim", "asian", etc. We present the trade-off results in Figure 3 and compare performance in Table 2. Our methods reduce individual biases in BERT predictions. We note that in both Toxicity and Bios experiments we observe no practical differences between GLIF and GLIF-NRW.

6 Summary and Discussion

We studied a suite of post-processing methods for enforcing individual fairness. The methods provably enforce a local form of IF and scale readily to large datasets. We hope this broadens the appeal of IF by (i) alleviating the computational costs of operationalizing IF and (ii) allowing practitioners to use off-the-shelf models for standard ML tasks. We also note that it is possible to use our objective for in-processing.

We conclude with two warnings: first, enforcing any algorithmic fairness definition does not guarantee complete fairness from the perspective of the user. The problem-specific meaning of fairness is often hard to encode exactly with a mathematical fairness definition; second, while we believe that in many applications it is reasonable to consider local individual fairness, this choice should be understood and verified by a practitioner when choosing to enforce individual fairness with our method as opposed to directly enforcing IF constraints.

[1] Based on the Kaggle "Toxic Comment Classification Challenge".

References

[1] Solon Barocas and Andrew D. Selbst. Big Data's Disparate Impact. *SSRN Electronic Journal*, 2016. ISSN 1556-5068. doi: 10.2139/ssrn.2477899.

[2] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pages 624–638. Springer, 2004.

[3] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623, 2021.

[4] Marianne Bertrand and Sendhil Mullainathan. Are Emily and Greg More Employable Than Lakisha and Jamal? A Field Experiment on Labor Market Discrimination. *American Economic Review*, 94(4):991–1013, September 2004. ISSN 0002-8282. doi: 10.1257/0002828042002561.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *arXiv:2005.14165 [cs]*, June 2020.

[6] Joy Buolamwini and Timnit Gebru. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of Machine Learning Research*, volume 87, pages 77–91, 2018.

[7] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, April 2017. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aal4230.

[8] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-03358-9.

[9] Alexandra Chouldechova and Aaron Roth. The Frontiers of Fairness in Machine Learning. *arXiv:1810.08810 [cs, stat]*, October 2018.

[10] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006. ISSN 1063-5203. doi: 10.1016/j.acha.2006.04.006.

[11] Maria De-Arteaga, Alexey Romanov, Hanna Wallach, Jennifer Chayes, Christian Borgs, Alexandra Chouldechova, Sahin Geyik, Krishnaram Kenthapadi, and Adam Tauman Kalai. Bias in Bios: A Case Study of Semantic Representation Bias in a High-Stakes Setting. *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*, pages 120–128, 2019. doi: 10.1145/3287560.3287572.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, October 2018.

[13] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

[14] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness Through Awareness. *arXiv:1104.3913 [cs]*, April 2011.

[15] Will Fleisher. What's fair about individual fairness? *Available at SSRN 3819799*, 2021.

[16] Matthias Hein, Jean-yves Audibert, and Ulrike Von Luxburg. From graphs to manifolds - weak and strong pointwise consistency of graph Laplacians. In *Proceedings of the 18th Conference on Learning Theory (COLT*, pages 470–485. Springer, 2005.

[1] Matthias Hein, Jean-Yves Audibert, and Ulrike von Luxburg. Graph Laplacians and their Convergence on Random Neighborhood Graphs. *Journal of Machine Learning Research*, 8(48): 1325–1368, 2007. ISSN 1533-7928.

[18] David C Hsia. Credit scoring and the equal credit opportunity act. *Hastings LJ*, 30:371, 1978.

[19] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 10, Seattle, WA, August 2004.

[20] Christina Ilvento. Metric Learning for Individual Fairness. *arXiv:1906.00250 [cs, stat]*, June 2019.

[21] Christopher Jung, Michael Kearns, Seth Neel, Aaron Roth, Logan Stapleton, and Zhiwei Steven Wu. Eliciting and Enforcing Subjective Individual Fairness. *arXiv:1905.10660 [cs, stat]*, May 2019.

[22] Michael P. Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-Box Post-Processing for Fairness in Classification. *arXiv:1805.12317 [cs, stat]*, May 2018.

[23] Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. Measuring bias in contextualized word representations. *arXiv preprint arXiv:1906.07337*, 2019.

[24] Jeff Larson and Julia Angwin. How We Examined Racial Discrimination in Auto Insurance.... https://www.propublica.org/article/minority-neighborhoods-higher-car-insurance-premiums-methodology, April 2017.

[25] Pranay Lohia. Priority-based post-processing bias mitigation for individual and group fairness. *arXiv preprint arXiv:2102.00417*, 2021.

[26] Pranay K Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R Varshney, and Ruchir Puri. Bias mitigation post-processing for individual and group fairness. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 2847–2851. IEEE, 2019.

[27] Debarghya Mukherjee, Mikhail Yurochkin, Moulinath Banerjee, and Yuekai Sun. Two simple ways to learn individual fairness metrics from data. In *International Conference on Machine Learning*, July 2020.

[28] Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*, 2020.

[29] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[30] Alexander J. Smola and Risi Kondor. Kernels and Regularization on Graphs. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Bernhard Schölkopf, and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777, pages 144–158. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-40720-1 978-3-540-45167-9. doi: 10.1007/978-3-540-45167-9_12.

[31] Daniel Spielman. Spectral graph theory. *Combinatorial scientific computing*, (18), 2012.

[32] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

[33] Alexander Vargo, Fan Zhang, Mikhail Yurochkin, and Yuekai Sun. Individually fair gradient boosting. *arXiv preprint arXiv:2103.16785*, 2021.

[34] Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio du Pin Calmon. Optimized Score Transformation for Fair Classification. *arXiv:1906.00066 [cs, math, stat]*, December 2019.

[35] Stephen J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, June 2015. ISSN 1436-4646. doi: 10.1007/s10107-015-0892-3.

[36] Mikhail Yurochkin and Yuekai Sun. SenSeI: Sensitive Set Invariance for Enforcing Individual Fairness. In *International Conference on Learning Representations*, September 2020.

[37] Mikhail Yurochkin, Amanda Bower, and Yuekai Sun. Training individually fair ML models with sensitive subspace robustness. In *International Conference on Learning Representations*, Addis Ababa, Ethiopia, 2020.

[38] Dengyong Zhou and Bernhard Schölkopf. A regularization framework for learning from graph data. In *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields (SRL 2004)*, pages 132–137, 2004.

Checklist

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