Individual Fairness Guarantees for Neural Networks

Elias Benussi[1*], **Andrea Patane**[1], **Matthew Wicker**[1], **Luca Laurenti**[2] and **Marta Kwiatkowska**[1]
[1]University of Oxford
[2]TU Delft
{elias.benussi, andrea.patane, matthew.wicker, marta.kwiatkowska}@cs.ox.ac.uk, l.laurenti@tudelft.nl

Abstract

We consider the problem of certifying the individual fairness (IF) of feed-forward neural networks (NNs). In particular, we work with the ϵ-δ-IF formulation, which, given a NN and a similarity metric learnt from data, requires that the output difference between any pair of ϵ-similar individuals is bounded by a maximum decision tolerance $\delta \geq 0$. Working with a range of metrics, including the Mahalanobis distance, we propose a method to over-approximate the resulting optimisation problem using piecewise-linear functions to lower and upper bound the NN's non-linearities globally over the input space. We encode this computation as the solution of a Mixed-Integer Linear Programming problem and demonstrate that it can be used to compute IF guarantees on four datasets widely used for fairness benchmarking. We show how this formulation can be used to encourage models' fairness at training time by modifying the NN loss, and empirically confirm our approach yields NNs that are orders of magnitude fairer than state-of-the-art methods.

1 Introduction

Reservations have been raised about the application of neural networks (NN) in contexts where *fairness* is of concern [Barocas and Selbst, 2016]. Because of inherent biases present in real-world data, if unchecked, these models have been found to discriminate against individuals on the basis of sensitive features, such as race or sex [Bolukbasi *et al.*, 2016; Angwin *et al.*, 2016]. Recently, the topic has come under the spotlight, with technologies being increasingly challenged for bias [Kirk *et al.*, 2021], leading to the introduction of a range of definitions and techniques for capturing the multifaceted properties of fairness.

Fairness approaches are broadly categorised into: *group fairness* [Hardt *et al.*, 2016], which inspects the model over data demographics; and *individual fairness* (IF) [Dwork *et al.*, 2012], which considers the behaviour over each individual. Despite its wider adoption, group fairness is only concerned with statistical properties of the model so that a situation may arise where predictions of a group-fair model can be perceived as unfair by a particular individual. In contrast, IF is a worst-case measure with guarantees over *every* possible individual in the input space. However, while techniques exist for group fairness of NNs [Albarghouthi *et al.*, 2017; Bastani *et al.*, 2019], research on IF has thus far been limited to designing training procedures that favour fairness [Yurochkin *et al.*, 2020; Yeom and Fredrikson, 2021; McNamara *et al.*, 2017] and verification over specific individuals [Ruoss *et al.*, 2020]. To the best of our knowledge, there is currently no work targeted at global certification of IF for NNs.

We develop an anytime algorithm with provable bounds for the certification of IF on NNs. We build on the ϵ-δ-IF formalisation employed by [John *et al.*, 2020]. That is, given $\epsilon, \delta \geq 0$ and a distance metric d_{fair} that captures the similarity between individuals, we ask that, for *every* pair of points x' and x'' in the input space with $d_{\text{fair}}(x', x'') \leq \epsilon$, the NN's output does not differ by more than δ. Although related to it, IF certification on NNs poses a different problem than adversarial robustness [Tjeng *et al.*, 2018], as both x' and x'' are here problem variables, spanning the whole space. Hence, local approximation techniques developed in the adversarial literature cannot be employed in the context of IF.

Nevertheless, we show how this global, non-linear requirement can be encoded in Mixed-Integer Linear Programming (MILP) form, by deriving a set of global upper and lower piecewise-linear (PWL) bounds over each activation function in the NN over the whole input space, and performing linear encoding of the (generally non-linear) similarity metric $d_{\text{fair}}(x', x'')$. The formulation of our optimisation as a MILP allows us to compute an *anytime*, worst-case bound on IF, which can thus be computed using standard solvers from the global optimisation literature [Dantzig, 2016]. Furthermore, we demonstrate how our approach can be embedded into the NN training so as to optimise for individual fairness at training time. We do this by performing gradient descent on a weighted loss that also accounts for the maximum δ-variation in d_{fair}-neighborhoods for each training point, similarly to what is done in adversarial learning [Goodfellow *et al.*, 2015; Wicker *et al.*, 2021].

We apply our method on four benchmarks widely employed in the fairness literature, namely, the Adult, German,

*Corresponding Author

Credit and Crime datasets[1], and an array of similarity metrics learnt from data that include ℓ_∞, Mahalanobis, and NN embeddings. We empirically demonstrate how our method is able to provide the first, non-trivial IF certificates for NNs commonly employed for tasks from the IF literature, and even larger NNs comprising up to thousands of neurons. Furthermore, we find that our MILP-based fair training approach consistently outperforms, in terms of IF guarantees, NNs trained with a competitive state-of-the-art technique by orders of magnitude, albeit at an increased computational cost.

The paper makes the following main contributions:[2]

- We design a MILP-based, anytime verification approach for the certification of IF as a global property on NNs.
- We demonstrate how our technique can be used to modify the loss function of a NN to take into account certification of IF at training time.
- On four datasets, and an array of metrics, we show how our techniques obtain non-trivial IF certificates and train NNs that are significantly fairer than state-of-the-art.

Related Work. A number of works have considered IF by employing techniques from adversarial robustness. [Yeom and Fredrikson, 2021] rely on randomized smoothing to find the highest stable per-feature difference in a model. Their method, however, provides only (weak) guarantees on model statistics. [Yurochkin *et al.*, 2020] present a method for IF training that builds on projected gradient descent and optimal transport. While the method is found to decrease model bias to state-of-the-art results, no formal guarantees are obtained. [Ruoss *et al.*, 2020] adapted the MILP formulation for adversarial robustness to handle fair metric embeddings. However, rather than tackling the IF problem globally as introduced by [Dwork *et al.*, 2012], the method only works iteratively on a finite set of data, hence leaving open the possibility of unfairness in the model. In contrast, the MILP encoding we obtain through PWL bounding of activations and similarity metrics allows us to provide guarantees over *any* possible pair of individuals. [Urban *et al.*, 2020] employ static analysis to certify causal fairness. While this method yields global guarantees, it cannot be straightforwardly employed for IF, and it is not *anytime*, making exhaustive analysis impractical. [John *et al.*, 2020] present a method for the computation of IF, though limited to linear and kernel models. MILP and linear relaxation have been employed to certify NNs in local adversarial settings [Ehlers, 2017; Tjeng *et al.*, 2018; Wicker *et al.*, 2020]. However, local approximations cannot be employed for the global IF problem. While [Katz *et al.*, 2017; Leino *et al.*, 2021] consider global robustness, their methods are restricted to ℓ_p metrics. Furthermore, they require the knowledge of a Lipschitz constant or are limited to ReLU.

2 Individual Fairness

We focus on regression and binary classification with NNs with real-valued inputs and one-hot encoded categorical features.[3] Such frameworks are often used in automated decision-making, e.g. for loan applications [Hardt *et al.*, 2016]. Formally, given a compact input set $X \subseteq \mathbb{R}^n$ and an output set $Y \subseteq \mathbb{R}$, we consider an L layer fully-connected NN $f^w : X \to Y$, parameterised by a vector of weights $w \in \mathbb{R}^{n_w}$ trained on $\mathcal{D} = \{(x_i, y_i), i \in \{1, ..., n_d\}\}$. For an input $x \in X$, $i = 1, \ldots, L$ and $j = 1, \ldots, n_i$, the NN is defined as:

$$\phi_j^{(i)} = \sum_{k=1}^{n_{i-1}} W_{jk}^{(i)} \zeta_k^{(i-1)} + b_j^{(i)}, \quad \zeta_j^{(i)} = \sigma^{(i)}\left(\phi_j^{(i)}\right) \tag{1}$$

where $\zeta_j^{(0)} = x_j$. Here, n_i is the number of units in the ith layer, $W_{jk}^{(i)}$ and $b_j^{(i)}$ are its weights and biases, $\sigma^{(i)}$ is the activation function, $\phi^{(i)}$ is the pre-activation and $\zeta^{(i)}$ the activation. The NN output is the result of these computations, $f^w(x) := \zeta^{(L)}$. In regression, $f^w(x)$ is the prediction, while for classification it represents the class probability. In this paper we focus on fully-connected NNs as widely employed in the IF literature[Yurochkin *et al.*, 2020; Urban *et al.*, 2020; Ruoss *et al.*, 2020]. However, we should stress that our framework, being based on MILP, can be easily extended to convolutional, max-pool and batch-norm layers or res-nets by using embedding techniques from the adversarial robustness literature (see e.g. [Boopathy *et al.*, 2019].

Individual Fairness. Given a NN f^w, IF [Dwork *et al.*, 2012] enforces the property that similar individuals are similarly treated. Similarity is defined according to a task-dependent pseudometric, $d_{\text{fair}} : X \times X \mapsto \mathbb{R}_{\geq 0}$, provided by a domain expert (e.g., a Mahalanobis distance correlating each feature to the sensitive one), whereas similarity of treatment is expressed via the absolute difference on the NN output $f^w(x)$. We adopt the ϵ-δ-IF formulation of [John *et al.*, 2020] for the formalisation of input-output IF similarity.

Definition 1 (ϵ-δ-IF [John *et al.*, 2020]). *Consider $\epsilon \geq 0$ and $\delta \geq 0$. We say that f^w is ϵ-δ-individually fair w.r.t. d_{fair} iff*

$$\forall x', x'' \text{ s.t. } d_{fair}(x', x'') \leq \epsilon \implies |f^w(x') - f^w(x'')| \leq \delta.$$

Here, ϵ measures similarity between individuals and δ is the difference in outcomes (class probability for classification). We emphasise that individual fairness is a *global* notion, as the condition in Definition 1 must hold for all pairs of points in X. We remark that the ϵ-δ-IF formulation of [John *et al.*, 2020] (which is more general than IF formulation typically used in the literature [Yurochkin *et al.*, 2020; Ruoss *et al.*, 2020]) is a slight variation on the Lipschitz property introduced by [Dwork *et al.*, 2012]. While introducing greater flexibility thanks to its parametric form, it makes an IF parametric analysis necessary at test time. In Section 4 we analyse how ϵ-δ-IF of NNs is affected by variations of ϵ and δ. A crucial component of IF is the similarity d_{fair}. The intuition is that sensitive features, or their sensitive combination, should not influence the NN output. While a number of metrics has been discussed in the literature [Ilvento, 2020], we focus on the following representative set of metrics which can be automatically learnt from data [John *et al.*, 2020;

[1]http://archive.ics.uci.edu/ml

[2]Proofs and additional details can be found in Appendix of an extended version of the paper available at http://www.fun2model.org/bibitem.php?key=BPW+22.

[3]Multi-class can be tackled with component-wise analyses.

Ruoss *et al.*, 2020; Mukherjee *et al.*, 2020; Yurochkin *et al.*, 2020]. Details on metric learning is given in Appendix B.

Weighted ℓ_p. In this case $d_{\text{fair}}(x', x'')$ is defined as a weighted version of an ℓ_p metric, i.e. $d_{\text{fair}}(x', x'') = \sqrt[p]{\sum_{i=1}^{n} \theta_i |x'_i - x''_i|^p}$. Intuitively, we set the weights θ_i related to sensitive features to zero, so that two individuals are considered similar if they only differ with respect to those. The weights θ_i for the remaining features can be tuned according to their degree of correlation to the sensitive features.

Mahalanobis. In this case we have $d_{\text{fair}}(x', x'') = \sqrt{(x' - x'')^T S (x' - x'')}$, for a given positive semi-definite (SPD) matrix S. The Mahalanobis distance generalises the ℓ_2 metric by taking into account the intra-correlation of features to capture latent dependencies w.r.t. the sensitive features.

Feature Embedding. The metric is computed on an embedding, so that $d_{\text{fair}}(x', x'') = \hat{d}(\varphi(x'), \varphi(x''))$, where $\hat{d}$ is either the Mahalanobis or the weighted ℓ_p metric, and φ is a feature embedding map. These allow for greater modelling flexibility, at the cost of reduced interpretability.

2.1 Problem Formulation

We aim at certifying ϵ-δ-IF for NNs. To this end we formalise two problems: computing certificates and training for IF.

Problem 1 (Fairness Certification)**.** *Given a trained NN f^w, a similarity d_{fair} and a distance threshold $\epsilon \geq 0$, compute*

$$\delta_{\max} = \max_{\substack{x', x'' \in X \\ d_{fair}(x', x'') \leq \epsilon}} |f^w(x') - f^w(x'')|.$$

Problem 1 provides a formulation in terms of optimisation, seeking to compute the maximum output change $\delta_{\max}$ for any pair of input points whose d_{fair} distance is no more than ϵ. One can then compare $\delta_{\max}$ with any threshold δ: if $\delta_{\max} \leq \delta$ holds then the model f^w has been certified to be ϵ-δ-IF.

While Problem 1 is concerned with an already trained NN, the methods we develop can also be employed to encourage IF at training time. Similarly to the approaches for adversarial learning [Goodfellow *et al.*, 2015], we modify the training loss $L(f^w(x), y)$ to balance between the model fit and IF.

Problem 2 (Fairness Training)**.** *Consider an NN f^w, a training set $\mathcal{D}$, a similarity metric d_{fair} and a distance threshold $\epsilon \geq 0$. Let $\lambda \in [0, 1]$ be a constant. Define the IF-fair loss as*

$$\begin{aligned} &L_{fair}(f^w(x_i), y_i, f^w(x_i^*), \lambda) = \\ &\qquad \lambda L(f^w(x_i), y_i) + (1 - \lambda)|f^w(x_i) - f^w(x_i^*)|, \end{aligned}$$

where $x_i^* = \arg\max_{x \in X \ s.t. \ d_{fair}(x_i, x) \leq \epsilon} |f^w(x_i) - f^w(x)|$. The ϵ-IF training problem is defined as finding w^{fair} s.t.:

$$w^{fair} = \arg\min_{w} \sum_{i=1}^{n_d} L_{fair}(f^w(x_i), y_i).$$

In Problem 2 we seek to train a NN that not only is accurate, but whose predictions are also fair according to Definition 1. Parameter λ balances between accuracy and IF. In particular, for $\lambda = 1$ we recover the standard training that does not account for IF, while for $\lambda = 0$ we only consider IF.

3 A MILP Approach For Individual Fairness

Certification of individual fairness on a NN thus requires us to solve the following global, non-convex optimisation problem:

$$\begin{aligned} \max_{x', x'' \in X} \quad & |\delta| \\ \text{subject to} \quad & \delta = f^w(x') - f^w(x'') && (2) \\ & d_{\text{fair}}(x', x'') \leq \epsilon. && (3) \end{aligned}$$

We develop a Mixed-Integer Linear Programming (MILP) over-approximation (i.e., providing a sound bound) to this problem. We notice that there are two sources of non-linearity here, one induced by the NN (Equation (2)), which we refer to as the *model constraint*, and the other by the fairness metric (Equation (3)), which we call *fairness constraint*. In the following, we show how these can be modularly bounded by piecewise-linear functions. In Section 3.3 we bring the results together to derive a MILP formulation for ϵ-δ-IF.

3.1 Model Constraint

We develop a scheme based on *piecewise-linear* (PWL) upper and lower bounding for over-approximating all commonly used non-linear activation functions. An illustration of the PWL bound is given in Figure 1. Let $\phi_j^{(i)L}$ and $\phi_j^{(i)U} \in \mathbb{R}$ be lower and upper bounds on the pre-activation $\phi_j^{(i)}$.[4] We proceed by building a discretisation grid over the $\phi_j^{(i)}$ values on M grid points: $\boldsymbol{\phi}_{\text{grid}} = [\phi_{j,0}^{(i)}, \ldots, \phi_{j,M}^{(i)}]$, with $\phi_{j,0}^{(i)} := \phi_j^{(i)L}$ and $\phi_{j,M}^{(i)} := \phi_j^{(i)U}$, such that, in each partition interval $[\phi_{j,l}^{(i)}, \phi_{j,l+1}^{(i)}]$, we have that $\sigma^{(i)}$ is either convex or concave. We then compute linear lower and upper bound functions for $\sigma^{(i)}$ in each $[\phi_{j,l}^{(i)}, \phi_{j,l+1}^{(i)}]$ as follows. If $\sigma^{(i)}$ is convex (resp. concave) in $[\phi_{j,l}^{(i)}, \phi_{j,l+1}^{(i)}]$, then an upper (resp. lower) linear bound is given by the segment connecting the two extremum points of the interval, and a lower (resp. upper) linear bound is given by the tangent through the mid-point of the interval. We then compute the values of each linear bound in each of its grid points, and select the minimum of the lower bounds and the maximum of the upper bound values, which we store in two vectors $\boldsymbol{\zeta}_j^{\text{PWL},(i),U} = [\zeta_{j,0}^{\text{PWL},(i),U}, \ldots, \zeta_{j,M}^{\text{PWL},(i),U}]$ and $\boldsymbol{\zeta}_j^{\text{PWL},(i),L} = [\zeta_{j,0}^{\text{PWL},(i),L}, \ldots, \zeta_{j,M}^{\text{PWL},(i),L}]$. The following lemma is a consequence of this construction.

Lemma 1. *Let $\phi \in [\phi_j^{(i)L}, \phi_j^{(i)U}]$. Denote with l the index associated to the partition of $\boldsymbol{\phi}_{grid}$ in which ϕ falls and consider $\eta \in [0, 1]$ such that $\phi = \eta \phi_{j,l-1}^{(i)L} + (1 - \eta)\phi_{j,l}^{(i)L}$. Then:*

$$\sigma^{(i)}(\phi) \geq \eta \zeta_{j,l-1}^{PWL,(i),L} + (1 - \eta)\zeta_{j,l}^{PWL,(i),L},$$
$$\sigma^{(i)}(\phi) \leq \eta \zeta_{j,l-1}^{PWL,(i),U} + (1 - \eta)\zeta_{j,l}^{PWL,(i),U},$$

[4] Computed by bound propagation over X [Ehlers, 2017].

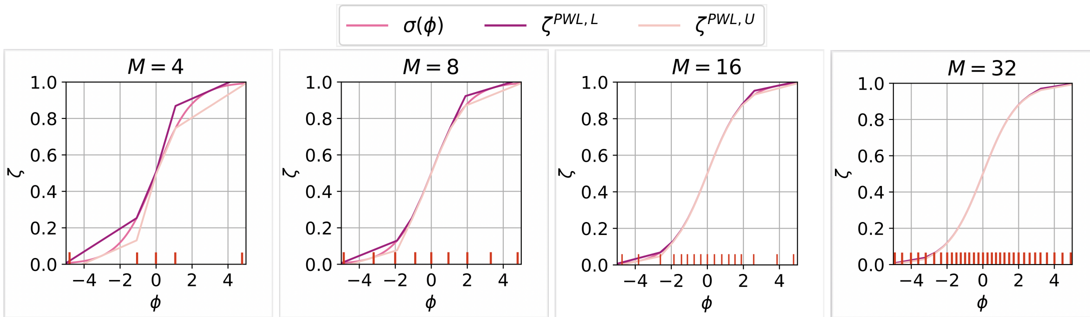

Figure 1: Upper and lower PWL functions to sigmoid for an increasing number of partition points M (marked with red ticks).

that is, $\zeta_j^{PWL,(i),L}$ and $\zeta_j^{PWL,(i),U}$ define continuous PWL lower and upper bounds for ϕ in $[\phi_j^{(i)L}, \phi_j^{(i)U}]$.

Lemma 3.1 guarantees that we can bound the non-linear activation functions using PWL functions. Crucially, PWL functions can then be encoded into the MILP constraints.

Proposition 1. *Let $y_{j,l}^{(i)}$ for $l = 1, \ldots, M$, be binary variables, and $\eta_{j,l}^{(i)} \in [0, 1]$ be continuous ones. Consider $\phi_j^{(i)} \in [\phi_j^{(i)L}, \phi_j^{(i)U}]$ then it follows that $\zeta_j^{(i)} = \sigma^{(i)}\left(\phi_j^{(i)}\right)$ implies:*

$$\sum_{l=1}^{M} y_{j,l}^{(i)} = 1, \ \sum_{l=1}^{M} \eta_{j,l}^{(i)} = 1, \phi_j^{(i)} = \sum_{l=1}^{M} \phi_{j,l}^{(i)L} \eta_{j,l}^{(i)}, \ y_{j,l}^{(i)} \leq \eta_{j,l}^{(i)} + \eta_{j,l+1}^{(i)}, \ \sum_{l=1}^{M} \zeta_{j,l}^{PWL,(i),L} \eta_{j,l}^{(i)} \leq \zeta_j^{(i)} \leq \sum_{l=1}^{M} \zeta_{j,l}^{PWL,(i),U} \eta_{j,l}^{(i)}.$$

A proof can be found in Appendix A. Proposition 1 ensures that the global behaviour of each NN neuron can be over-approximated by 5 linear constraints using $2M$ auxiliary variables. Employing Proposition 1 we can encode the model constraint of Equation (2) into the MILP form in a sound way.

The over-approximation error does not depend on the MILP formulation (which is exact), but on the PWL bounding, and is hence controllable through the selection of the number of grid points M, and becomes exact in the limit. Notice that in the particular case of ReLU activation functions the over-approximation is exact for any $M > 0$.

Proposition 2. *Assume $\sigma^{(i)}$ to be continuously differentiable everywhere in $[\phi_j^{(i)L}, \phi_j^{(i)U}]$, except possibly in a finite set. Then PWL lower and upper bounding functions of Lemma 3.1 converge uniformly to $\sigma^{(i)}$ as M goes to infinity.*

Furthermore, define $\Delta_M = (\phi_j^{(i)U} - \phi_j^{(i)L})/M$, then for finite values of M the error on the lower (resp. upper) bounding in convex (resp. concave) regions of $\sigma^{(i)}$ for $\phi \in [\phi_{j,l}^{(i)}, \phi_{j,l+1}^{(i)}]$ is given by:

$$e_1(\phi) \leq \frac{\Delta_M}{2}\left(\sigma'(\phi_{j,l+1}^{(i)}) - \sigma'\left(\phi_{j,l+1}^{(i)} - \frac{\Delta_M}{2}\right)\right)$$

and upper (resp. lower) in concave (resp. convex) regions:

$$e_2(\phi) \leq \Delta_M \left(\frac{\sigma\left(\phi_{j,l}^{(i)} + \Delta_M\right) - \sigma(\phi_{j,l}^{(i)})}{\Delta_M} + \sigma'(\phi_{j,l}^{(i)})\right).$$

A proof of Proposition 2 is given in Appendix A, alongside an experimental analysis of the convergence rate.

We remark that the PWL bound can be used over all commonly employed activation functions σ. The only assumption made is that σ has a finite number of inflection points over any compact interval of $\mathbb{R}$. For convergence (Prop. 2) we require continuous differentiability almost everywhere, which is satisfied by commonly used activations.

3.2 Fairness Constraint

The encoding of the fairness constraint within the MILP formulation depends on the specific form of the metric d_{fair}.

Weighted ℓ_p Metric: The weighted ℓ_p metric can be tackled by employing rectangular approximation regions. While this is straightforward for the ℓ_∞ metric, for the remaining cases interval abstraction can be used [Dantzig, 2016].

Mahalanobis Metric: We first compute an orthogonal decomposition of S as in $U^T S U = \Lambda$, where U is the eigenvector matrix of S and Λ is a diagonal matrix with S eigenvalues as entries. Consider the rotated variables $z' = U^T x'$ and $z'' = U^T x''$, then we have that Equation (3) can be re-written as $(z' - z'')^T \Lambda (z' - z'') \leq \epsilon^2$. By simple algebra we thus have that, for each i, $(z'_i - z''_i)^2 \leq \frac{\epsilon^2}{\Lambda_{ii}}$. By transforming back to the original variables, we obtain that Equation (3) can be over-approximated by: $-\frac{\epsilon}{\sqrt{\text{diag}(\Lambda)}} \leq U^T x' - U^T x'' \leq \frac{\epsilon}{\sqrt{\text{diag}(\Lambda)}}$.

Feature Embedding Metric We tackle the case in which φ used in the metric definition, i.e. $d_{\text{fair}}(x', x'') = \hat{d}(\varphi(x'), \varphi(x''))$, is a NN embedding. This is straightforward as φ can be encoded into MILP as for the model constraint.

3.3 Overall Formulation

We now formulate the MILP encoding for the over-approximation $\delta_* \geq \delta_{max}$ of ϵ-δ-IF. For Equation (2), we proceed by deriving a set of approximating constraints for the variables x' and x'' by using the techniques described in Section 3.1. We denote the corresponding variables as $\phi_j'^{(i)}$, $\zeta_j'^{(i)}$ and $\phi_j''^{(i)}$, $\zeta_j''^{(i)}$, respectively. The NN final output on x' and on x'' will then respectively be $\zeta'^{(L)}$ and $\zeta''^{(L)}$, so that $\delta = \zeta'^{(L)} - \zeta''^{(L)}$. Finally, we over-approximate Equation (3) as described in Section 3.2. In the case of Mahalanobis

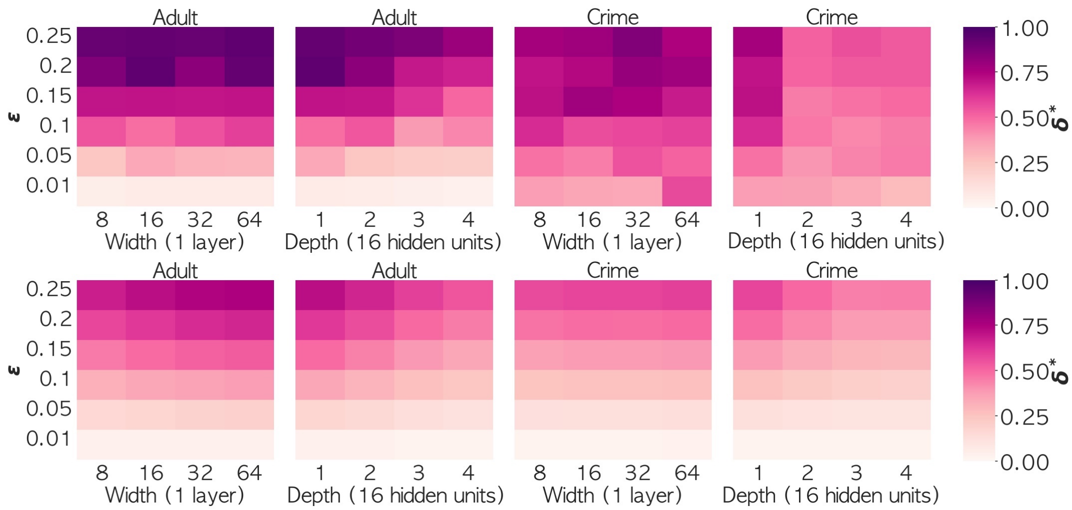

Figure 2: Certified bounds on IF (δ_*) for different architecture parameters (widths and depths) and maximum similarity (ϵ) for the Adult and the Crime datasets. **Top Row**: Mahalanobis metric used for d_{fair}. **Bottom Row**: Weighted ℓ_∞ metric used for d_{fair}.

distance, we thus obtain:

$$\max_{x',x''\in X} \quad |\delta| \tag{4}$$

$$\text{subject to} \;\; = \zeta'^{(L)} - \zeta''^{(L)}$$

$$\text{for } i = 1,\ldots,L, \;\; j = 1,\ldots,n_i, \; \dagger \in \{',''\}:$$

$$\sum_{l=1}^{M} y_{j,l}^{\dagger(i)} = 1, \quad \sum_{l=1}^{M} \eta_{j,l}^{\dagger(i)} = 1, \quad y_{j,l}^{(i)} \leq \eta_{j,l}^{(i)} + \eta_{j,l+1}^{(i)}$$

$$\phi_j^{\dagger(i)} = \sum_{k=1}^{n_{i-1}} W_{jk}^{(i)} x_k^{\dagger} + b_j^{(i)}, \; \phi_j^{\dagger(i)} = \sum_{l=1}^{M} \phi_{j,l}^{(i)L} \eta_{j,l}^{\dagger(i)}$$

$$\sum_{l=1}^{M} \zeta_{j,l}^{\text{PWL},(i),L} \eta_{j,l}^{\dagger(i)} \leq \zeta_j^{\dagger(i)} \leq \sum_{l=1}^{M} \zeta_{j,l}^{\text{PWL},(i),U} \eta_{j,l}^{\dagger(i)}$$

$$-\frac{\epsilon^2}{\sqrt{\text{diag}(\Lambda)}} \leq Ux' - Ux'' \leq \frac{\epsilon^2}{\sqrt{\text{diag}(\Lambda)}}.$$

Though similar, the above MILP is significantly different from those used for adversarial robustness (see e.g. [Tjeng *et al.*, 2018]). First, rather than looking for perturbations around a fixed a point, here we have both x' and x'' as variables. Furthermore, rather than being local, the MILP problem for ϵ-δ-IF is global, over the whole input space X. As such, local approximations of non-linearities cannot be used, as the bounding needs to be valid simultaneously over the whole input space. Finally, while in adversarial robustness one can ignore the last sigmoid layer, for IF, because of the two optimisation variables, one cannot simply map from the last preactivation value to the class probability, so that even for ReLU NNs one needs to employ bounding of non-piecewise activations for the final sigmoid.

By combining the results from this section, we have:

Theorem 1. *Consider $\epsilon \geq 0$, a similarity d_{fair} and a NN f^w. Let x'_* and x''_* be the optimal points for the optimisation problem in Equation (4). Define $\delta_* = |f^w(x'_*) - f^w(x''_*)|$. Then f^w is ϵ-δ-individually fair w.r.t. d_{fair} for any $\delta \geq \delta_*$.*

Theorem 1, whose proof can be found in Appendix A, states that a solution of the MILP problem provides us with a sound estimation of individual fairness of an NN. Crucially, it can be shown that branch-and-bound techniques for the solution of MILP problems converge in finite time to the optimal solution [Del Pia and Weismantel, 2012], while furthermore providing us with upper and lower bounds for the optimal value at each iteration step. Therefore, we have:

Corollary 1. *Let δ_k^L and δ_k^U lower and upper bounds computed by a MILP solver at step $k > 0$. Then we have that: $\delta_k^L \leq \delta_* \leq \delta_k^U$. Furthermore, given a precision, τ, there exist a finite k_* such that $\delta_{k_*}^U - \delta_{k_*}^L \leq \tau$.*

That is, our method is sound and anytime, as at each iteration step in the MILP solving we can retrieve a lower and an upper bound on δ_*, which can thus be used to provide provable guarantees while converging to δ_* in finite time.

Complexity Analysis. The encoding of the model constraint can be done in $O(LMn_{\max})$, where $n_{\max}$ is the maximum width of f^w, L is the number of layers, and M is the number of grid points used for the PWL bound. The computational complexity of the fairness constraints depends on the similarity metric employed. While for ℓ_∞ no processing needs to be done, the computational complexity is $O(n^3)$ for the Mahalanobis distance and again $O(LMn_{\max})$ for the feature embedding metric. Each iteration of the MILP solver entails the solution of a linear programming problem and is hence $O((Mn_{\max}L)^3)$. Finite time convergence of the MILP solver to δ^* with precision τ is exponential in the number of problem variables, in τ and ϵ.

3.4 Fairness Training for Neural Networks

The ϵ-δ-IF MILP formulation introduced in Section 3 can be adapted for the solution of Problem 2. The key step is the computation of x_i^* in the second component of the modified loss introduced in Problem 2, which is used to introduce fairness directly into the loss of the neural network. This computation can be done by observing that, for every training point x_i drawn from $\mathcal{D}$, the computation of $x_i^* = \arg\max_{x \in X \; s.t. \; d_x(x_i,x) \leq \epsilon} |f^w(x_i) - f^w(x)|$ is a particular case of the formulation described in Section 3, where, instead of having two variable input points, only one input point is a problem variable while the other is given and drawn from the training dataset $\mathcal{D}$. Therefore, x_i^* can be computed by solving the MILP problem, where we fix a set of the problem variables to x_i, and can be subsequently used to obtain the value of the modified loss function. Note that these constraints are not cumulative, since they are built for each mini-batch, and discarded after optimization is solved to update the weights.

Algorithm 1 Fair Training with MILP.

Input: NN architecture: f^w, Dataset: $\mathcal{D}$, Learning rate: α, Iterations: n_{epoch}, Batch Size: n_{batch}, Similarity metric: d_{fair}, Maximum similarity: ϵ, Fairness Loss Weighting: λ.
Output: w_{fair}: weight values balancing between accuracy and fairness.

1: $w_{\text{fair}} \leftarrow InitWeights(f^w)$
2: **for** $t = 1, \ldots, n_{\text{epoch}}$ **do**
3: **for** $b = 1, \ldots, \lceil |\mathcal{D}|/n_{\text{batch}} \rceil$ **do**
4: $\{X, Y\} \leftarrow \{x_i, y_i\}_{i=0}^{n_{\text{batch}}} \sim \mathcal{D}$ #Sample Batch
5: $Y_{\text{clean}} \leftarrow f^w(X)$ #Standard forward pass
6: $[\boldsymbol{\phi}', \boldsymbol{\zeta}', \boldsymbol{\phi}'', \boldsymbol{\zeta}''] \leftarrow InitMILP(f^w, d_{\text{fair}}, \epsilon)$ # Section 3
7: $X_{\text{MILP}} \leftarrow \emptyset$
8: **for** $i = 0, \ldots n_{\text{batch}}$ **do**
9: $\boldsymbol{\phi}_i', \boldsymbol{\zeta}_i' \leftarrow FixVarConst(x_i)$ #Fix constraints
10: $x_i^* \leftarrow MILP(x_i, \boldsymbol{\phi}_i', \boldsymbol{\zeta}_i')$ # Solve 'local' MILP prob.
11: $X_{\text{MILP}} \leftarrow X_{\text{MILP}} \bigcup \{x_i^*\}$
12: **end for**
13: $Y_{\text{MILP}} \leftarrow f^w(X_{\text{MILP}})$ #MILP inputs forward pass
14: $l \leftarrow L_{\text{fair}}(Y_{\text{clean}}, Y, Y_{\text{MILP}}, \lambda)$ #Fair Loss
15: $w_{\text{fair}} \leftarrow w_{\text{fair}} - \alpha \nabla_w l$ #Optimizer step (here, SGD)
16: **end for**
17: **end for**
18: return w_{fair} #Weights optimized for fairness & accuracy

We summarise our fairness training method in Algorithm 1. For each batch in each of the n_{epoch} training epochs, we perform a forward pass of the NN to obtain the output, Y_{clean} (line 5). We then formulate the MILP problem as in Section 3 (line 6), and initialise an empty set variable to collect the solutions to the various sub-problems (line 7). Then, for each training point x_i in the mini-batch, we fix the MILP constraints to the variables associated with x_i (line 9), solve the resulting MILP for x_i^*, and place x_i^* in the set that collects the solutions, i.e. X_{MILP}. Finally, we compute the NN predictions on X_{MILP} (line 13); the result is used to compute the modified loss function (line 14) and the weights are updated by taking a step of gradient descent. The resulting set of weights w_{fair} balances the empirical accuracy and fairness around the training points.

The choice of λ affects the relative importance of standard training w.r.t. the fairness constraint: $\lambda = 1$ is equivalent to standard training, while $\lambda = 0$ only optimises for fairness. In our experiments we keep $\lambda = 1$ for half of the training epochs, and then change it to $\lambda = 0.5$.

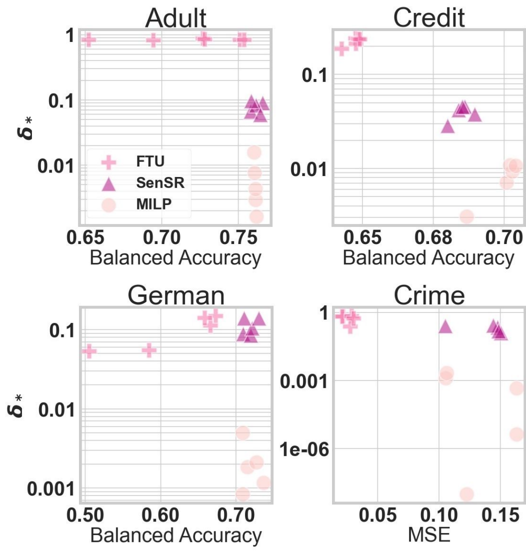

Figure 3: Balanced accuracy / individual fairness trade-off for NNs.

4 Experiments

In this section, we empirically validate the effectiveness of our MILP formulation for computing ϵ-δ-IF guarantees as well as for fairness training of NNs. We perform our experiments on four UCI datasets: the *Adult* dataset (predicting income), the *Credit* dataset (predicting payment defaults), the *German* dataset (predicting credit risk) and the *Crime* dataset (predicting violent crime). In each case, features encoding information regarding gender or race are considered sensitive. In the certification experiments we employ a precision τ for the MILP solvers of 10^{-5} and a time cutoff of 180 seconds. We compare our training approach with two different learning methods: *Fairness-Through-Unawareness* (FTU), in which the sensitive features are simply removed, and SenSR [Yurochkin *et al.*, 2020]. Exploration of the cutoff, group fairness, certification of additional NNs, scalability of the methods and additional details on the experimental settings are given in Appendix C and D.[5]

Fairness Certification. We analyse the suitability of our method in providing non-trivial certificates on ϵ-δ-IF with respect to the similarity threshold ϵ (which we vary from 0.01 to 0.25), the similarity metric d_{fair} , the width of the NN (from 8 to 64), and its number of layers (from 1 to 4). These reflect the characteristics of NNs and metrics used in the IF literature [Yurochkin *et al.*, 2020; Ruoss *et al.*, 2020; Urban *et al.*, 2020]; for experiments on larger architectures, demonstrating the scalability of our approach, see Appendix D.3. For each dataset we train the NNs by employing the FTU approach. The results for these analyses are plotted in Figure 2 for the Adult and the Crime datasets (results for Credit and German datasets can be found in Appendix D.1). Each heat map depicts the variation of δ_* as a function of ϵ and the NN architecture. The top row in the figure was computed by considering the Mahalanobis similarity metric; the

[5]An implementation of the method and of the experiments can be found at https://github.com/eliasbenussi/nn-cert-individual-fairness.

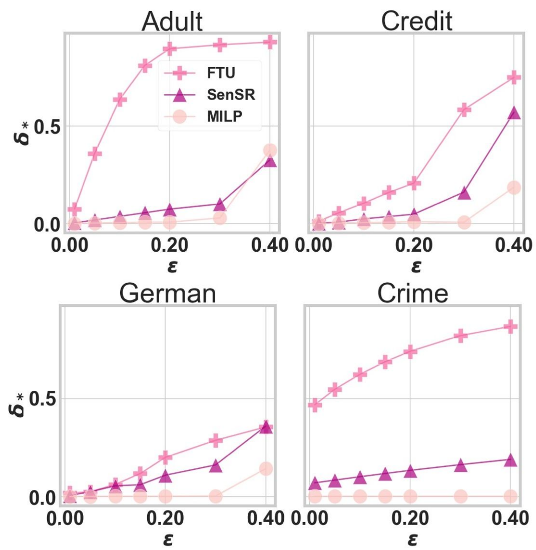

Figure 4: Certified δ_* as a function of the maximum similarity ϵ.

bottom row was computed for a weighted ℓ_∞ metric (with coefficients chosen as in [John *et al.*, 2020]) and results for the feature embedding metrics are given in Appendix D.2. As one might expect, we observe that, across all the datasets and architectures, increasing ϵ correlates with an increase in the values for δ_*, as higher values of ϵ allow for greater feature changes. Interestingly, δ_* tends to decrease (i.e., the NN becomes more fair) as we increase the number of NN layers. This is the opposite to what is observed for the adversarial robustness, where increased capacity generally implies more fragile models [Madry *et al.*, 2018]. In fact, as those NNs are trained via FTU, the main sensitive features are not accessible to the NN. A possible explanation is that, as the number of layers increases, the NN's dependency on the specific value of each feature diminishes, and the output becomes dependent on their nonlinear combination. The result suggests that overparametrised NNs could be more adept at solving IF tasks though this would come with a loss of model interpretability, and exploration would be needed to assess under which condition this holds. Finally, we observe that our analysis confirms how FTU training is generally insufficient in providing fairness on the model behaviour for ϵ-δ-IF. For each model, individuals that are dissimilar by $\epsilon \geq 0.25$ can already yield a $\delta_* > 0.5$, meaning they would get assigned to different classes if one was using the standard classification threshold.

Fairness Training. We investigate the behaviour of our fairness training algorithm for improving ϵ-δ-IF of NNs. We compare our method with FTU and SenSR [Yurochkin *et al.*, 2020]. For ease of comparison, in the rest of this section we measure fairness with d_{fair} equal to the Mahalanobis similarity metric, with $\epsilon = 0.2$, for which SenSR was developed. The results for this analysis are given in Figure 3, where each point in the scatter plot represents the values obtained for a given NN architecture. We train architectures with up to 2 hidden layers and 64 units, in order to be comparable to those trained by [Yurochkin *et al.*, 2020]. As expected, we observe that FTU performs the worst in terms of certified fairness, as simple omission of the sensitive features is unable to obfuscate latent dependencies between the sensitive and non-sensitive features. As previously reported in the literature, SenSR significantly improves on FTU by accounting for features latent dependencies. However, on all four datasets, our MILP-based training methodology consistently improves IF by orders of magnitude across all the architectures when compared to SenSR. In particular, for the architectures with more than one hidden layer, on average, MILP outperforms FTU by a factor of 78598 and SenSR by 27739. Intuitively, while SenSR and our approach have a similar formulation, the former is based on gradient optimisation so that no guarantees are provided in the worst case for the training loss. In contrast, by relying on MILP, our method optimises the worst-case behaviour of the NN at each step, which further encourages training of individually fair models. The cost of the markedly improved guarantees is, of course, a higher computational costs. In fact, the training of the models in Figure 3 with MILP had an average training time of about 3 hours. While the increased cost is significant, we highlight that this is a cost that is only paid once and may be justified in sensitive applications by the necessity of fairness at deployment time. We furthermore notice that, while our implementation is sequential, parallel per-batch solution of the MILP problems during training would markedly reduce the computational time and leave for future work the parallelisation and tensorisation of the techniques. Interestingly, we find that balanced accuracy also slightly improved with SenSR and MILP training in the tasks considered here, possibly as a result of the bias in the class labels w.r.t. sensitive features. Finally, in Figure 4 we further analyse the certified δ_*-profile w.r.t. to the input similarity ϵ, varying the value of ϵ used in for the ceritification of ϵ-δ-IF. In the experiment, both SenSR and MILP are trained with $\epsilon = 0.2$, which means that our method, based on formal IF certificates, is guaranteed to outperform SenSR up until $\epsilon = 0.2$ (as in fact is the case). Beyond 0.2, no such statement can be made, and it is still theoretically possible for SenSR to outperform MILP in particular circumstances. Empirically, however, MILP-based training still largely outperforms SenSR in terms of certified fairness obtained.

5 Conclusion

We introduced an *anytime* MILP-based method for the certification and training of ϵ-δ-IF in NNs, based on PWL bounding and MILP encoding of non-linearities and similarity metrics. In an experimental evaluation comprising four datasets, a selection of widely employed NN architectures and three types of similarity metrics, we found that our method is able to provide the first non-trivial certificates for ϵ-δ-IF in NNs and yields NNs which are orders of magnitude more fair than those obtained by a competitive techniques.

Acknowledgements

This project was funded by the ERC European Union's Horizon 2020 research and innovation programme (FUN2MODEL, grant agreement No. 834115).

References

[Albarghouthi *et al.*, 2017] Aws Albarghouthi, Loris D'Antoni, Samuel Drews, and Aditya V Nori. Fairsquare: probabilistic verification of program fairness. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA):1–30, 2017.

[Angwin *et al.*, 2016] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pages 254–264. Auerbach Publications, 2016.

[Barocas and Selbst, 2016] Solon Barocas and Andrew D Selbst. Big data's disparate impact. *Calif. L. Rev.*, 104:671–733, 2016.

[Bastani *et al.*, 2019] Osbert Bastani, Xin Zhang, and Armando Solar-Lezama. Probabilistic verification of fairness properties via concentration. *Proceedings of the ACM on Programming Languages*, 3(OOPSLA):1–27, 2019.

[Bolukbasi *et al.*, 2016] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *NeurIPS*, 29:4349–4357, 2016.

[Boopathy *et al.*, 2019] Akhilan Boopathy, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Cnn-cert: An efficient framework for certifying robustness of convolutional neural networks. In *AAAI*, volume 33, pages 3240–3247, 2019.

[Dantzig, 2016] George Dantzig. *Linear programming and extensions*. Princeton university press, 2016.

[Del Pia and Weismantel, 2012] Alberto Del Pia and Robert Weismantel. On convergence in mixed integer programming. *Mathematical programming*, 135(1):397–412, 2012.

[Dwork *et al.*, 2012] 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.

[Ehlers, 2017] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks. In *International Symposium on Automated Technology for Verification and Analysis*, pages 269–286. Springer, 2017.

[Goodfellow *et al.*, 2015] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[Hardt *et al.*, 2016] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *NeurIPS*, 29:3315–3323, 2016.

[Ilvento, 2020] Christina Ilvento. Metric learning for individual fairness. In *1st Symposium on Foundations of Responsible Computing*, 2020.

[John *et al.*, 2020] Philips George John, Deepak Vijaykeerthy, and Diptikalyan Saha. Verifying individual fairness in machine learning models. In *UAI*, pages 749–758. PMLR, 2020.

[Katz *et al.*, 2017] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017.

[Kirk *et al.*, 2021] Hannah Rose Kirk, Filippo Volpin, Haider Iqbal, Elias Benussi, Frederic Dreyer, Aleksandar Shtedritski, Yuki Asano, et al. Bias out-of-the-box: An empirical analysis of intersectional occupational biases in popular generative language models. *NeurIPS*, 34:2611–2624, 2021.

[Leino *et al.*, 2021] Klas Leino, Zifan Wang, and Matt Fredrikson. Globally-robust neural networks. In *International Conference on Machine Learning*, pages 6212–6222. PMLR, 2021.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

[McNamara *et al.*, 2017] Daniel McNamara, Cheng Soon Ong, and Robert C. Williamson. Provably fair representations. *CoRR*, abs/1710.04394, 2017.

[Mukherjee *et al.*, 2020] 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*, pages 7097–7107. PMLR, 2020.

[Ruoss *et al.*, 2020] Anian Ruoss, Mislav Balunovic, Marc Fischer, and Martin Vechev. Learning certified individually fair representations. *NeurIPS*, 33:7584–7596, 2020.

[Tjeng *et al.*, 2018] Vincent Tjeng, Kai Y Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. In *ICLR*, 2018.

[Urban *et al.*, 2020] Caterina Urban, Maria Christakis, Valentin Wüstholz, and Fuyuan Zhang. Perfectly parallel fairness certification of neural networks. *ACM on Programming Languages*, 4(OOPSLA):1–30, 2020.

[Wicker *et al.*, 2020] Matthew Wicker, Luca Laurenti, Andrea Patane, and Marta Kwiatkowska. Probabilistic safety for bayesian neural networks. In *UAI*, pages 1198–1207. PMLR, 2020.

[Wicker *et al.*, 2021] Matthew Wicker, Luca Laurenti, Andrea Patane, Zhuotong Chen, Zheng Zhang, and Marta Kwiatkowska. Bayesian inference with certifiable adversarial robustness. In *AISTATS*, pages 2431–2439. PMLR, 2021.

[Yeom and Fredrikson, 2021] Samuel Yeom and Matt Fredrikson. Individual fairness revisited: transferring techniques from adversarial robustness. In *IJCAI*, pages 437–443, 2021.

[Yurochkin *et al.*, 2020] Mikhail Yurochkin, Amanda Bower, and Yuekai Sun. Training individually fair ml models with sensitive subspace robustness. In *ICLR*, pages 1–18, 2020.