Improving Fairness Generalization Through a Sample-Robust Optimization Method

Ulrich Aïvodji[1], Julien Ferry[2*†], Sébastien Gambs[3], Marie-José Huguet[2] and Mohamed Siala[2]

[1]École de Technologie Supérieure, Montréal, Canada.
[2*]LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France.
[3]Université du Québec à Montréal, Montréal, Canada.

*Corresponding author(s). E-mail(s): jferry@laas.fr;
Contributing authors: ulrich.aivodji@etsmtl.ca; gambs.sebastien@uqam.ca; huguet@laas.fr; msiala@laas.fr;
†First author.

Abstract

Unwanted bias is a major concern in machine learning, raising in particular significant ethical issues when machine learning models are deployed within high-stakes decision systems. A common solution to mitigate it is to integrate and optimize a statistical fairness metric along with accuracy during the training phase. However, one of the main remaining challenges is that current approaches usually generalize poorly in terms of fairness on unseen data. We address this issue by proposing a new robustness framework for statistical fairness in machine learning. The proposed approach is inspired by the domain of Distributionally Robust Optimization and works in ensuring fairness over a variety of samplings of the training set. Our approach can be used to quantify the robustness of fairness but also to improve it when training a model. We empirically evaluate the proposed method and show that it effectively improves fairness generalization. In addition, we propose a simple yet powerful heuristic application of our framework that can be integrated into a wide range of existing fair classification techniques to enhance fairness generalization. Our extensive empirical study using two existing fair classification methods demonstrates the efficiency and scalability of the proposed heuristic approach.

Keywords: Supervised Learning, Fairness, Generalization, Distributionally Robust Optimization

1 Introduction

The growing integration of machine learning models in high-stakes decision systems raises several ethical, legal and philosophical issues. Among them, fairness is often a desired property in addition to being a legal requirement. Machine learning models extract and exploit correlations from their given training data. However, such correlations may not be relevant because of the data collection, processing, sampling, or historical discrimination (Tommasi et al, 2017; Torralba and Efros, 2011).

To mitigate such negative biases, several fairness notions have emerged (Verma and Rubin, 2018). In a nutshell, individual fairness (Dwork et al, 2012) consists in ensuring that similar individuals receive similar treatment. In contrast, statistical fairness requires that a given metric's value does not differ between specified subgroups of the population. The key idea here is that individuals should not receive different treatment based on their membership to a protected group. Finally, causal fairness analyzes the relationships between the different attributes and the decision to find (and possibly eliminate) correlations that can be a source of discrimination.

Many methods were proposed in the literature to enhance the fairness of machine learning models (Caton and Haas, 2020; Barocas et al, 2019). However, models that are fair with respect to their training data may still exhibit unfairness when applied to previously unseen data. Indeed, *fairness constraint overfitting* (Cotter et al, 2018, 2019a) can occur, and fairness generalization has been identified as an open challenge for trustworthy machine learning (Chuang and Mroueh, 2021; Cotter et al, 2018, 2019a; Huang and Vishnoi, 2019; Mandal et al, 2020). Our objective in this paper is precisely to address this issue.

Recent work on fairness generalization targets integrating different techniques for improving robustness into existing fair learning algorithms. While such methods have been shown (theoretically and empirically) to improve fairness generalization, they often induce a considerable computational overhead (*e.g.*, solving an additional problem to determine a worst-case unfairness (Mandal et al, 2020)), and thus have limited scalability. Some methods do not suffer from this drawback but instead require additional splitting of the data (Cotter et al, 2018, 2019a), hence possibly penalizing utility, as the amount of data used to update the model is reduced. Finally, other approaches have limited applicability, as they are designed for a particular algorithm or hypothesis class (Taskesen et al, 2020; Wang et al, 2021), or require some special property of the underlying algorithm (*e.g.*, access to a cost-sensitive classification oracle (Mandal et al, 2020)). To tackle these issues, we propose a new framework for statistical fairness robustness. Intuitively, our approach consists in ensuring fairness over a variety of samplings of the training set. We show that

this notion can be quantified precisely, and leveraged to audit or train fair and robust machine learning models in practice. We additionally design a flexible and efficient heuristic method for learning robust and fair models, which can easily be integrated into existing fair classification methods, formulated as constrained optimization problems. More precisely, our contributions can be summarized as follows.

- We propose and study a sample-robust formulation of the fair learning problem, based on the Jaccard distance and inspired by Distributionally Robust Optimization (Ben-Tal et al, 2013; Duchi et al, 2021; Rahimian and Mehrotra, 2019). The main idea of our method is that we want to meet a given fairness constraint, even if the training set sampling were somehow different (*i.e.* if some examples were not part of it).
- We show how this exact formulation can be used to quantify statistical fairness robustness of machine learning models. Our exact method is model-agnostic and can be applied to any type of hypothesis classes.
- We show that our exact method can be used for sample-robust fair learning and highlight its practical computational and integrability limitations.
- We design a simple, efficient and flexible heuristic application of our proposed formulation and illustrate its versatility by integrating it into two fair learning algorithms of the literature.
- We empirically evaluate both our exact and heuristic approaches and compare them on different datasets and statistical fairness metrics.
- We empirically demonstrate the effectiveness and performance of the proposed heuristic approach on various datasets and metrics and compare it to a state-of-the-art method for improving statistical fairness generalization.

The paper is organized as follows. First, in Section 2, we describe the necessary background and review the relevant literature on statistical fairness generalization in machine learning. Afterwards, in Section 3, we formulate our notion of sample-robustness for fairness and study its implications and practical limitations. We motivate and introduce a heuristic application of our approach in Section 4 and show how to integrate it into state-of-the-art fair learning techniques. Then, we empirically evaluate our proposed approaches in Section 5 before concluding in Section 6.

2 Background & Related Work

In this section, we first present a high-level background on supervised machine learning, fairness and distributionally robust optimization. Then, we review existing methods addressing the problem of statistical fairness generalization.

2.1 Preliminaries

2.1.1 Supervised Machine Learning - Classification

Let $\mathcal{X}$ denote the feature space, $\mathcal{A}$ the sensitive attributes and $\mathcal{Y}$ the label set. In addition, $\mathcal{P}$ will denote the true distribution over $\mathcal{X} \times \mathcal{A} \times \mathcal{Y}$ and $\mathcal{D} = (X, A, Y)$ a dataset drawn from $\mathcal{P}$. Given such a dataset and an hypothesis class of models $\mathcal{H}$, the objective of a supervised learning algorithm $\mathcal{L}$ is to build a model $\mathcal{L}(\mathcal{D}) = h \in \mathcal{H}$ such that h minimizes a given objective function f_{obj}.

For a specific training dataset $\mathcal{D}$ drawn from some distribution $\mathcal{P}$, the desired model h is the solution to the following problem, in which $f_{obj}(h, \mathcal{P})$ is the expected objective function under distribution $\mathcal{P}$:

$$\arg\min_{h \in \mathcal{H}} \; f_{obj}(h, \mathcal{P}) \tag{1}$$

In practice, $\mathcal{P}$ is often unknown, and we only get a limited number of observations from it, contained in the dataset $\mathcal{D}$. Then, the optimal solution of Problem (1) is commonly approximated solving Problem (2).

$$\arg\min_{h \in \mathcal{H}} \; f_{obj}(h, \mathcal{D}) \tag{2}$$

2.1.2 Fairness in Machine Learning

In this work, we focus on *statistical metrics for fairness*. Such metrics aim at equalizing a given statistical measure (*e.g.*, the True Positive Rate) between several (possibly overlapping) protected groups (m being the number of protected groups), defined by the sensitive attributes. For each example e_i in $\mathcal{D}$, we denote by $a_i \in A$ the list of sensitive attributes. Each coordinate $k \in [1..m]$ of a_i indicates the membership of example e_i to the protected group k. Intuitively, the objective is to ensure that examples (*e.g.*, profiles of individuals) receive similar treatment independently of the protected group they belong to. Depending on the particular value being equalized across groups, many metrics have been proposed such as statistical parity (Dwork et al, 2012), predictive equality (Chouldechova, 2017), equal opportunity (Hardt et al, 2016) and equalized odds (Hardt et al, 2016). These notions, as well as the statistical measure they equalize, are summarized in Table 1. In this paper, we denote by $\mathsf{unf}(h, \mathcal{D})$ an oracle quantifying the unfairness of a classifier h over a dataset $\mathcal{D}$. The value of $\mathsf{unf}(h, \mathcal{D})$ is in $[0, 1]$. The lower the value of $\mathsf{unf}(h, \mathcal{D})$, the more fair is h over $\mathcal{D}$. In practice, we consider a maximum acceptable unfairness value $\epsilon \in [0, 1]$ (or, equivalently, a minimum acceptable fairness value $1 - \epsilon$), and say that h is fair over $\mathcal{D}$ when $\mathsf{unf}(h, \mathcal{D}) \leq \epsilon$.

Several fairness-enhancement algorithms have been proposed in the literature. They can be categorized into three categories, depending on the stage of the machine learning pipeline in which they intervene. *Preprocessing* techniques (Kamiran and Calders, 2012) remove undesired biases from the training

Table 1 Summary of some statistical fairness measures

Fairness notion	Equalized statistical measure
Statistical parity	Probability of being assigned the positive class
Predictive equality	False positive rate
Equal opportunity	False negative rate
Equalized odds	False negative rate and False positive rate

data before applying regular learning algorithms on the sanitized dataset. *Post-processing* algorithms (Hardt et al, 2016) modify the predictions of a (possibly unfair) classifier to achieve fairness. Finally, *algorithmic modification* (also called *in-processing*) techniques (Zafar et al, 2017) directly modify the learning algorithm to ensure that the model built is fair. These in-processing approaches naturally define a bi-objective optimization problem: minimizing error while maximizing fairness (or, equivalently, maximizing accuracy while minimizing unfairness). Several methods can then be used to solve this problem, including optimizing directly or indirectly a measure of fairness or enforcing fairness constraints while learning an accurate model. In this paper, we are interested in fair learning methods formulated as constrained optimization problems, as described in Problem (3).

$$\begin{aligned} \arg\min_{h \in \mathcal{H}} \quad & f_{obj}(h, \mathcal{D}) \\ \text{s.t.} \quad & \mathsf{unf}(h, \mathcal{D}) \leq \epsilon. \end{aligned} \tag{3}$$

2.1.3 Distributionally Robust Optimization in Supervised Machine Learning

As stated in Section 2.1.1, an important challenge in machine learning is that we usually do not know the true underlying distribution $\mathcal{P}$. Instead, we often have access to a limited training set $\mathcal{D}$, whose distribution $\mathcal{P}'$ may differ from $\mathcal{P}$. To take into account this uncertainty, Distributionally Robust Optimization (DRO) techniques can be leveraged. Instead of minimizing an objective function f_{obj} for a given distribution $\mathcal{P}'$, DRO (Ben-Tal et al, 2013; Duchi et al, 2021; Rahimian and Mehrotra, 2019) consists in minimizing f_{obj} for a worst-case distribution, among a set of perturbed versions of $\mathcal{P}'$ (Sagawa et al, 2020). More precisely, the objective is to build a model h minimizing f_{obj} for a set of neighbouring distributions of $\mathcal{P}'$. Such neighbouring distributions are contained in a *perturbation set* (also called *ambiguity set*) $\mathcal{B}(\mathcal{P}')$. In the DRO setting, the supervised machine learning problem becomes:

$$\arg\min_{h \in \mathcal{H}} \max_{\mathcal{Q} \in \mathcal{B}(\mathcal{P}')} f_{obj}(h, \mathcal{Q}). \tag{4}$$

Distributionally Robust Optimization has been used in many domains (Rahimian and Mehrotra, 2019), and has been applied widely in machine learning (Kang, 2017).

2.2 Related Work on Improving Statistical Fairness Generalization

To improve the generalization of statistical fairness, several approaches have been designed based on the method proposed by Agarwal et al (2018), who formulated the problem of learning an accurate classifier under fairness constraints as a two-player zero-sum game. Considering the Lagrangian relaxation of this constrained optimization problem, the first player (θ-player) optimizes the model's parameters for the objective function with current Lagrange multipliers, while the second player (λ-player) approximates the strongest Lagrangian relaxation by updating the Lagrangian multipliers. In their original contribution, Agarwal et al (2018) analyzed the fairness generalization error of the models trained using this framework. In order to avoid the *fairness constraints overfitting*, in Cotter et al (2018, 2019a) the λ-player updates the Lagrangian multipliers based on fairness violations measured on a separate validation set (instead of the training set itself). In Mandal et al (2020), the λ-player uses linear programming to compute the worst-case fairness violation among a set of re-weightings of the training set. This approach falls into the category of DRO techniques.

Other methods also leverage DRO approaches. For instance in Sagawa et al (2020), a model is learnt while minimizing the maximum error over a set of protected groups defined by the value of some biased attributes. Several approaches have been proposed to tackle this worst-group error minimization problem. In particular, different methods do not require the full training set protected groups knowledge. Indeed, annotating protected groups membership for each training point can be costly in real-world settings (Duchi et al, 2020; Nam et al, 2020; Liu et al, 2021). Such methods do not reach the performances levels of the standard DRO approach with groups knowledge but constitute interesting alternatives. For example, Nam et al (2020) and Liu et al (2021) use two-stage approaches, in which they first train a model before leveraging its errors to train another more robust one. Duchi et al (2020) applies a DRO technique to approximate and optimize for a worst-case subpopulation above a certain size, without any group annotations.

In Taskesen et al (2020), distributionally robust and fair logistic regression models are trained by optimizing the fairness-regularized objective function for a worst-case distribution. This most adversarial distribution is considered within an ambiguity set characterized as a Wasserstein distance-based ball around the original training distribution. Rezaei et al (2020) also leverages the principles of DRO to optimize a robust logarithmic loss under fairness constraints. Their approach uses a minimax formulation, in which a fair predictor minimizes the training loss while a worst-case approximator of the population

distribution (subject to statistic-matching constraints) maximizes it. In a similar line of work, Wang et al (2021) proposes a distributionally robust measure of unfairness for the Equality of Opportunity metric. Robustness is achieved by computing the worst-case unfairness over a set of neighbouring distributions, within a type-∞ Wasserstein ambiguity set. Taking into account this measure enables the training of distributionally robust fair Support Vector Machines (SVM).

Du and Wu (2021) proposes two algorithms for fair and robust learning under sample selection bias. These two methods aim at estimating the sample selection probabilities, by leveraging (or not) the availability of unlabeled unbiased data. The key point is that knowledge of these biased sample selection probabilities can be used to re-weight the training dataset to make it representative of the true distribution. As an approximation error exists, a minimax approach is used to optimize the objective function for the worst-case sample selection probabilities in a given radius around the estimated ones. The proposed method can only handle the statistical parity metric, which is approximated using decision boundary fairness and included as a regularization term to the objective function. One consequence is that robustness is enforced jointly for error and fairness. Nonetheless, the fairness constraints may not be strictly satisfied.

Measuring prediction stability on the training set, Huang and Vishnoi (2019) proposes the addition of a regularisation term to the objective function of a fair learning algorithm. This regularisation term aims at ensuring that the predictions of the built model do not vary too much when the training dataset is perturbed. In addition, this method theoretically bounds the generalization error. This work is closely related to ours, as we seek to improve fairness robustness on samplings of the training set (which can be viewed as a form of training fairness stability).

In a different line of work, Slack et al (2020) have studied the scenario in which a model trained to be fair may behave unfairly on related but slightly different tasks. This paper introduces two contributions, namely `Fairness Warnings` and `Fair-MAML`. On the one side, `Fairness Warnings` aims at predicting whether shifts in the features' distributions may result in violating fairness. This is achieved by generating perturbed versions of the training set (they only consider mean-shifting of the features), measuring the resulting fairness violation and training an interpretable model to predict such violation given the features' shifts. On the other side, `Fair-MAML` has for objective to learn a fair model that can be adapted to particular new tasks using minimal (and possibly biased) task-specific data. This is done by adding a fairness regularizer (for either the Statistical Parity or Equal Opportunity metrics) to the loss of the Model Agnostic Meta Learning (`MAML`) framework.

More recently, Chuang and Mroueh (2021) proposed a data augmentation strategy improving the generalization of fair classifiers. This method leverages existing data augmentation strategies to generate interpolated distributions between two given sensitive groups. During training, a regularisation term

penalizes changes in the model's predictions between the different interpolated distributions. The goal here is to ensure that the model has a smooth behavior along the "path" formed by the interpolated distributions between the two sensitive groups. This approach theoretically and empirically improves the fairness generalization of the models built.

Furthermore, fairness robustness has also been studied in other settings, such as multi-source learning (Iofinova et al, 2021), or for other notions of fairness such as individual fairness (Yurochkin et al, 2020). Both are out of the scope of this paper and thus we do not further detail these approaches.

Finally, the approach that is the more closely related to ours is that of Mandal et al (2020), which is based on a similar intuition, namely ensuring fairness on a set of neighbouring distributions of the training set, called re-weightings versions, can improve its generalization. However, we consider different definitions for such neighbouring distributions and in addition we propose a heuristic approach variant exhibiting practical advantages compared to the exact one.

3 Sample-Based Robustness for Statistical Fairness

In this section, we present our sample-based approach of robustness for statistical fairness. In a nutshell, it aims at improving fairness generalization by enforcing the fairness constraints over particular samplings of the training set. More precisely, we first introduce a dataset sampling technique based on the Jaccard distance, which is used to define our perturbation sets, before characterizing the structure of such perturbation sets. Second, we define the sample-robust fair learning problem on a given perturbation set and show how unfairness metrics increase through neighbouring subsets. Then, we discuss conditions for ensuring perfect fairness sample-robustness and their implications over the resulting models. Afterwards, we introduce an approach to quantify this notion based on the resolution of an integer optimization problem. Finally, we show how to integrate this robustness formulation into existing learning algorithms solving Problem (3) and discuss the practical limitations of the approach.

3.1 Jaccard Distance-based Perturbation Sets

Following the principles of DRO, it has been shown (Mandal et al, 2020; Sagawa et al, 2020; Taskesen et al, 2020; Wang et al, 2021) that enforcing fairness over a set of distributions that are neighbours to the training one is an efficient way to improve its generalization. While DRO was formalized using distributions, practical machine learning applications usually deal with finite training sets that are sampled from an underlying distribution. Indeed, instead of considering fairness robustness over perturbed underlying distributions (which, in practice, are unknown), we enforce robustness with respect to the training set sampling. For this reason, we propose to use the Jaccard distance J_δ as the distance metric measuring similarity between sample sets.

Definition 1 (***Jaccard distance***) Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two sample sets. The Jaccard distance between $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined as follows: $J_\delta(\mathcal{D}_1, \mathcal{D}_2) = 1 - \frac{|\mathcal{D}_1 \cap \mathcal{D}_2|}{|\mathcal{D}_1 \cup \mathcal{D}_2|}$

The Jaccard distance is a very popular measure, used to quantify (dis)similarity between sample sets in a wide range of applications. For example, it has been used in Machine Learning for feature ranking stability (Khoshgoftaar et al, 2013; Saeys et al, 2008) and feature selection (Zou et al, 2016). Intuitively, two sample sets $\mathcal{D}_1$ and $\mathcal{D}_2$ that have a large intersection are close (*i.e.*, $J_\delta(\mathcal{D}_1, \mathcal{D}_2)$ is small and in particular $J_\delta(\mathcal{D}, \mathcal{D}) = 0$) while two sample sets $\mathcal{D}_3$ and $\mathcal{D}_4$ with empty intersection are far from each other (*i.e.*, $J_\delta(\mathcal{D}_3, \mathcal{D}_4)$ is 1).

We now define the perturbation sets of a given dataset $\mathcal{D}$ and highlight some consequences.

Definition 2 (***Perturbation sets***) Let $d \in [0, 1]$, we define a perturbation set $\mathcal{B}(\mathcal{D}, d)$ as the set of subsets of $\mathcal{D}$ whose Jaccard distance from $\mathcal{D}$ is less than or equal to d. That is, $\mathcal{B}(\mathcal{D}, d) = \{\mathcal{D}' \mid J_\delta(\mathcal{D}, \mathcal{D}') \leq d \wedge (\mathcal{D}' \subseteq \mathcal{D})\}$.

Definition 2 states that $\mathcal{B}(\mathcal{D}, d)$ contains all subsets of $\mathcal{D}$ of size at least $|\mathcal{D}| \times (1 - d)$. A special case arises if $d = 0$, as $\mathcal{B}(\mathcal{D}, d)$ is $\mathcal{D}$ itself. Then, because the perturbation sets defined in Definition 2 contain only subsets of $\mathcal{D}$, the Jaccard distance between such subsets and $\mathcal{D}$ is necessarily of the form $\frac{i}{|\mathcal{D}|}$, in which i is an integer between 0 and $|\mathcal{D}|$ (as the union between $\mathcal{D}$ and any of its subsets is $\mathcal{D}$ itself).

Notice that extending a perturbation set consists in adding new subsets of the original training set. An immediate consequence of Definition 2 is that a perturbation set defined with Jaccard distance d is included in perturbation sets with higher distance d'. This is stated in the following proposition:

Proposition 1 *(**Perturbation sets inclusion**) Consider a dataset $\mathcal{D}$ and two Jaccard distances $d, d' \in [0, 1]$. Then, $d \leq d' \implies \mathcal{B}(\mathcal{D}, d) \subseteq \mathcal{B}(\mathcal{D}, d')$*

Proof Consider $\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)$. Based on Definition 2, $\mathcal{D}' \subseteq \mathcal{D}$ and $J_\delta(\mathcal{D}, \mathcal{D}') \leq d \leq d'$. Hence, $\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d')$ and $\mathcal{B}(\mathcal{D}, d) \subseteq \mathcal{B}(\mathcal{D}, d')$.

□

In the following definition, we introduce an additional notation to facilitate the study of the space of our perturbation sets. We then formalize the notion of neighbouring datasets, that will be useful in the remainder of our analysis.

Definition 3 (***Sets of equidistant subsets***) Let i be an integer between 0 and $|\mathcal{D}|$. Then, for any d of the form $\frac{i}{|\mathcal{D}|}$, we define $\Gamma(\mathcal{D}, d)$ as the set of subsets of $\mathcal{D}$ whose Jaccard distance from $\mathcal{D}$ is **exactly** d:

$$\Gamma(\mathcal{D}, d) = \{\mathcal{D}' \mid J_\delta(\mathcal{D}, \mathcal{D}') = d \wedge (\mathcal{D}' \subseteq \mathcal{D})\}$$

As an immediate consequence of Definition 3, $\{\Gamma(\mathcal{D}, d') \mid d' \leq d\}$ constitutes a partition of $\mathcal{B}(\mathcal{D}, d)$.

Definition 4 (***Neighbouring datasets***) Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two sample sets. $\mathcal{D}_1$ and $\mathcal{D}_2$ are called neighbouring datasets (in the Jaccard sense) if and only if $J_\delta(\mathcal{D}_1, \mathcal{D}_2) = \frac{1}{|\mathcal{D}_1 \cup \mathcal{D}_2|}$. This means that $\mathcal{D}_1$ and $\mathcal{D}_2$ differ by exactly one element.

In a nutshell, the subsets of $\mathcal{D}$ contained in $\mathcal{B}(\mathcal{D}, d)$ can be seen as points in a metric space equipped with the Jaccard distance[1], contained within a ball centered around $\mathcal{D}$ whose radius is d. This ball is itself contained within all sets $\mathcal{B}(\mathcal{D}, d')$ with $d' \geq d$. Again, because we restrict our attention to subsets of the training set, $\Gamma(\mathcal{D}, d)$ is a sphere centered in $\mathcal{D}$ with radius d. The ball $\mathcal{B}(\mathcal{D}, d)$ thus includes all spheres $\Gamma(\mathcal{D}, d')$ with radius $d' \leq d$.

An interesting representation of our perturbation sets space can be done using a nearest neighbours (in the Jaccard sense) graph. In such graph, each vertex represents a subset of the training set, and each edge between two vertices means that their associated sets are neighbours. Figure 1 uses such representation to illustrate our perturbation sets structure on a toy dataset with two protected groups for the statistical fairness measure. In the remainder of the paper, the perturbation sets are considered for insuring fairness constraints. Thus in this graph at least one example of each protected group must be present in each subset [2]. Based on this representation, we could also derive a directed graph $\mathcal{G}'$ in which each edge from $\mathcal{D}_1$ to $\mathcal{D}_2$ means that $\mathcal{D}_2 \subseteq \mathcal{D}_1$, thus representing a superset relationship.

We now formulate a recursive definition of $\mathcal{B}(\mathcal{D}, d)$. First, observe that the smallest perturbation set $\mathcal{B}(\mathcal{D}, d)$ not restricted to $\mathcal{D}$ itself is $\mathcal{B}(\mathcal{D}, \frac{1}{|\mathcal{D}|})$. It contains all subsets of $\mathcal{D}$ formed by removing at most one example from $\mathcal{D}$. This is a particular case of Proposition 2, which generalizes this observation.

Proposition 2 (***Perturbation sets structure***) *Consider a dataset $\mathcal{D}$ and a Jaccard distance $d \in [0, 1 - \frac{1}{|\mathcal{D}|}]$. We can formulate recursive definitions for $\Gamma(\mathcal{D}, d)$ and $\mathcal{B}(\mathcal{D}, d)$ as follows:*

- $\Gamma(\mathcal{D}, d + \frac{1}{|\mathcal{D}|}) = \{\mathcal{D}'' \mid \exists \mathcal{D}' \in \Gamma(\mathcal{D}, d) \mid \mathcal{D}'' \subset \mathcal{D}' \wedge |\mathcal{D}''| = |\mathcal{D}'| - 1\}$
- $\mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|}) = \mathcal{B}(\mathcal{D}, d) \cup \Gamma(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$

Proof

- By construction following the definition of $\Gamma(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$ (Definition 3).

[1] The Jaccard distance satisfies all required properties to equip a metric space, and in particular the triangle inequality (Kosub, 2019).

[2] Indeed, statistical fairness with respect to protected groups a and b can only be measured on datasets containing examples from both protected groups. For instance, the Equal Opportunity metric measures the difference between the True Positive Rates (TPR) of the two protected groups. Hence, if there are no examples from one protected group, its TPR is undefined, and so is the unfairness measure.

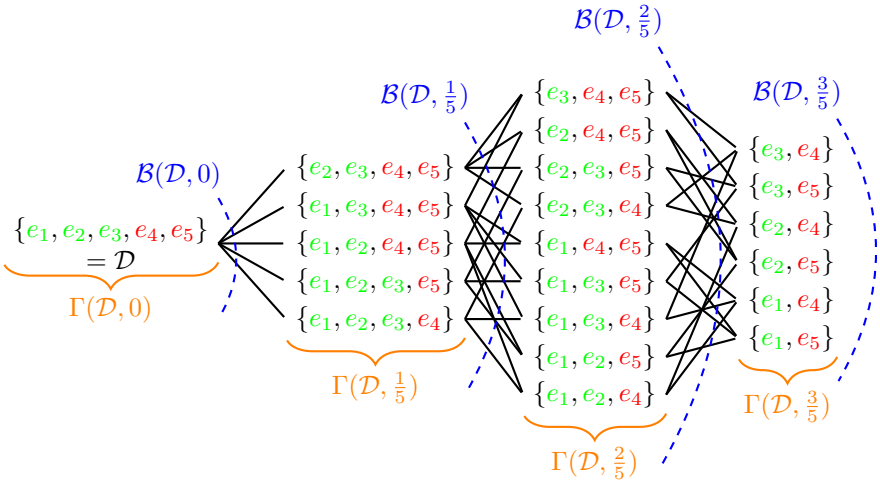

Fig. 1 Example of perturbation sets for a dataset $\mathcal{D}$ with 5 examples and two protected groups a ($\{e_1, e_2, e_3\}$) and b ($\{e_4, e_5\}$). Subsets that can not be used to audit a model's fairness with respect to protected groups a and b are not represented.

- Definition 2 states that $\mathcal{B}(\mathcal{D}, d)$ contains all subsets of $\mathcal{D}$ up to a Jaccard distance d. Definition 3 states that $\Gamma(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$ contains all subsets of $\mathcal{D}$ whose Jaccard distance from $\mathcal{D}$ is exactly $d + \frac{1}{|\mathcal{D}|}$. Thus, the union of these two sets define exactly $\mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$, which, according to Definition 2, contains all subsets of $\mathcal{D}$ up to a Jaccard distance of $d + \frac{1}{|\mathcal{D}|}$.

□

Proposition 2 states that the smallest set that is strictly bigger than $\mathcal{B}(\mathcal{D}, d)$ is $\mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$. By construction, it contains $\mathcal{B}(\mathcal{D}, d)$. The sets outside $\mathcal{B}(\mathcal{D}, d)$ and inside $\mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$ are exactly those in $\Gamma(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$. The later includes all sets formed by removing exactly one element from the (smallest) sets in $\mathcal{B}(\mathcal{D}, d)$ (which form $\Gamma(\mathcal{D}, d)$). Again, this can be visualized in Figure 1.

One may remark that instead of considering only subsets of the training set, we could take into account all neighbouring sets (as stated in Definition 4). This would require considering that examples can be added to our subsets. Even though this formulation can seem theoretically appealing, it does not have the interesting structure that we studied in this section and quantifying it is computationally harder as the denominator of the Jaccard distance would no longer be a constant.

3.2 Sample-Robust Fair Learning with our Perturbation Sets

Similar to DRO, the proposed approach consists in ensuring a given property (*e.g.*, fairness) over a set of elements contained in a perturbation set. For

DRO, such elements are distributions while we rather consider sample sets. By doing so, our objective is to improve the fairness generalization on unseen data. Hence, our perturbation sets contain different samplings of the original training set. Thus, we propose to study a sample-robust fair learning problem to reach fairness generalization.

We formulate our sample-robust fair learning problem before characterizing the evolution of unfairness through the considered subsets. Then, we investigate the conditions and implications for perfectly satisfying our proposed fairness sample-robustness criterion.

3.2.1 Robust Fair Learning for a Given Perturbation Set

By considering the perturbation set $\mathcal{B}(\mathcal{D}, d)$ as a set of samplings of the dataset $\mathcal{D}$, we aim at building a model that is fair on all sets of $\mathcal{B}(\mathcal{D}, d)$, including $\mathcal{D}$ itself. This leads to the formulation of our sample-robust version of Problem (3).

The sample-robust fair learning problem on a perturbation set $\mathcal{B}(\mathcal{D}, d)$ is formulated as follows:

$$\begin{aligned} \arg\min_{h \in \mathcal{H}} \quad & f_{obj}(h, \mathcal{D}) \\ \text{s.t.} \quad & \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)} \mathsf{unf}(h, \mathcal{D}') \leq \epsilon. \end{aligned} \tag{5}$$

This formulation is a particular instantiation of the general DRO formulation of Problem (4), in which robustness is applied only on the enforced fairness constraints rather than on the objective function. An optimal solution to Problem (5) corresponds to a model h that minimizes the objective function f_{obj} on $\mathcal{D}$, among those of $\mathcal{H}$ that exhibit unfairness at most ϵ over all sets contained in $\mathcal{B}(\mathcal{D}, d)$, including $\mathcal{D}$ itself.

With the proposed perturbation sets definition, we observe that augmenting the distance d increases the number of subsets being considered, as stated in Proposition 1. As a consequence, considering higher values of d can only raise the worst-case fairness violation, thus hardening the problem. This is formalized in Proposition 3. Hence, the parameter d directly controls the strength of the enforced robustness of the fairness constraint.

Proposition 3 *(**Worst-case fairness violation monotonicity with respect to** d) Consider a dataset $\mathcal{D}$ and a classifier h.*

$$\forall d, d' \in [0, 1], d \leq d' \implies \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)} \mathsf{unf}(h, \mathcal{D}') \leq \max_{\mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d')} \mathsf{unf}(h, \mathcal{D}'').$$

Proof According to Proposition 1, if $d \leq d'$ then $\mathcal{B}(\mathcal{D}, d) \subseteq \mathcal{B}(\mathcal{D}, d')$. Thus, the maximum unfairness over all sets in $\mathcal{B}(\mathcal{D}, d)$ is less than or equal to the maximum unfairness over all sets in $\mathcal{B}(\mathcal{D}, d')$. □

3.2.2 Unfairness Increase through Neighbouring Subsets

In order to characterize the possible increase of fairness violation (*i.e.,* the strength of the fairness constraints applied in Problem (5)) induced by increasing the size of the considered perturbation set, we first need to introduce the notion of sensitivity, which is directly taken from the differential privacy framework (Dwork and Roth, 2014) in Definition 5.

Definition 5 (***Unfairness*** l_1***-sensitivity***) The unfairness measure l_1-sensitivity γ quantifies the maximum contribution of a single example to the unfairness value of any classifier, for any pair of neighbouring datasets:

$$\gamma = \max_{\substack{h\in\mathcal{H} \\ \mathcal{D}_1,\mathcal{D}_2 \\ J_\delta(\mathcal{D}_1,\mathcal{D}_2)=\frac{1}{|\mathcal{D}_1\cup\mathcal{D}_2|}}} |\mathsf{unf}(h,\mathcal{D}_2) - \mathsf{unf}(h,\mathcal{D}_1)|$$

The increase of the worst-case fairness violation induced by extending a perturbation set $\mathcal{B}(\mathcal{D},d)$ to the next one $\mathcal{B}(\mathcal{D},d+\frac{1}{|\mathcal{D}|})$ can then be upper bounded as shown in the next proposition. This is due to the bounded sensitivity (as stated in Definition 5) of the unfairness measure $\mathsf{unf}(.)$ at hand, which has been proved for common statistical fairness metrics in the context of differentially private and fair learning (Cummings et al, 2019). The formalization of the bound goes as follows.

Proposition 4 ***(Bounded worst-case fairness violation increase between consecutive perturbation sets (general case))*** *Consider a dataset $\mathcal{D}$ and a classifier h. Let γ be the l_1-sensitivity of the unfairness measure and a Jaccard distance $d \in [0, 1-\frac{1}{|\mathcal{D}|}]$, we have:*

$$\max_{\mathcal{D}'\in\mathcal{B}(\mathcal{D},d)} \mathsf{unf}(h,\mathcal{D}') \leq \max_{\mathcal{D}''\in\mathcal{B}(\mathcal{D},d+\frac{1}{|\mathcal{D}|})} \mathsf{unf}(h,\mathcal{D}'') \leq \max_{\mathcal{D}'\in\mathcal{B}(\mathcal{D},d)} \mathsf{unf}(h,\mathcal{D}') + \gamma.$$

Proof

- Left inequality: Follows from Proposition 3 and the fact that $d \leq d + \frac{1}{|\mathcal{D}|}$.
- Right inequality: Consider $\mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d+\frac{1}{|\mathcal{D}|})$. By Proposition 2, we know that $\mathcal{D}''$ is either in $\mathcal{B}(\mathcal{D},d)$ or in $\Gamma(\mathcal{D}, d+\frac{1}{|\mathcal{D}|})$.
  - In the first case, $\mathcal{D}'' \in \mathcal{B}(\mathcal{D},d)$. The maximum unfairness measure across the perturbation set is not worsened and we have:

$$\mathsf{unf}(h,\mathcal{D}'') \leq \max_{\mathcal{D}'\in\mathcal{B}(\mathcal{D},d)} \mathsf{unf}(h,\mathcal{D}')$$

  - In the second case, $\mathcal{D}'' \in \Gamma(\mathcal{D}, d+\frac{1}{|\mathcal{D}|})$. By definition of Γ, we know that there exists some set $\mathcal{D}' \in \mathcal{B}(\mathcal{D},d)$ such that $\mathcal{D}''$ is formed by removing

exactly one element from $\mathcal{D}'$. Hence, $\mathcal{D}'$ and $\mathcal{D}''$ are neighbouring datasets, and following Definition 5, we know that:

$$\begin{aligned} |\mathsf{unf}(h,\mathcal{D}'') - \mathsf{unf}(h,\mathcal{D}')| \leq \gamma \implies & \mathsf{unf}(h,\mathcal{D}'') \leq \mathsf{unf}(h,\mathcal{D}') + \gamma \\ \implies & \mathsf{unf}(h,\mathcal{D}'') \leq \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D},d)} \mathsf{unf}(h,\mathcal{D}') + \gamma. \end{aligned}$$

□

This property can be visualized using the Jaccard neighbouring graph of Figure 1. It states that extending the perturbation set $\mathcal{B}(\mathcal{D}, d)$ by adding one more layer of the graph (the sets contained in $\Gamma(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$) cannot worsen the worst-case fairness violation by more than γ. This is due to the fact that the unfairness measure cannot be increased by more than γ between any two neighbouring datasets (*i.e.*, represented by two connected vertices in the Jaccard neighbours graph).

While Proposition 4 does not rely on any specific fairness formulation, depending on the metric at hand, tight values of γ can be computed. For instance, considering common statistical fairness metrics, we demonstrate that an exact finite value of γ can be computed for each subset, given the classifier's predictions. To ease the readibility, hereafter we consider the binary sensitive attribute setting in which there are only two non-overlapping protected groups a and b. However, note that the generalisation of the following proposition is valid for any number of protected groups, by considering all pairs of protected groups, as the resulting unfairness can be measured as the pairwise maximum unfairness.

Common statistical unfairness metrics can be defined using expressions of the form $\mathsf{unf}(h, \mathcal{D}_1) = |\frac{S_a^{\mathcal{D}_1}}{X_a^{\mathcal{D}_1}} - \frac{S_b^{\mathcal{D}_1}}{X_b^{\mathcal{D}_1}}|$, in which $\frac{S_i^{\mathcal{D}_1}}{X_i^{\mathcal{D}_1}}$ is the chosen statistical measure of classifier h for group $i \in \{a, b\}$. As it will always be clear from context, we do not include index h in the notations S and X.

For instance, for the Statistical Parity metric, $S_i^{\mathcal{D}_1}$ (respectively $X_i^{\mathcal{D}_1}$) is the number of positive predictions (respectively number of examples) among group i in the dataset $\mathcal{D}_1$, given classifier h's predictions. For the Equal Opportunity metric, $S_i^{\mathcal{D}_1}$ (respectively $X_i^{\mathcal{D}_1}$) is the number of true positive predictions (respectively number of positively labeled examples) among group i, for the dataset $\mathcal{D}_1$, given classifier h's predictions.

While the particular measures of $S_i^{\mathcal{D}_1}$ and $X_i^{\mathcal{D}_1}$ depend on the fairness metric at hand, our proposed formulation covers all common statistical fairness metrics (in particular, those considered in this paper and listed in Table 1). Additionally, we always have $S_i^{\mathcal{D}} \leq X_i^{\mathcal{D}}$, because the examples counted within $S_i^{\mathcal{D}}$ correspond to a subset of those counted within $X_i^{\mathcal{D}}$ that satisfy a given condition.

We assume without loss of generality that $\frac{S_a^{\mathcal{D}_1}}{X_a^{\mathcal{D}_1}} > \frac{S_b^{\mathcal{D}_1}}{X_b^{\mathcal{D}_1}}$. We also assume $X_a^{\mathcal{D}_1} > 0$ and $X_b^{\mathcal{D}_1} > 0$, which means that the (sub)set $\mathcal{D}_1$ contains examples from both protected groups that can be used to quantify fairness. Otherwise,

$\mathcal{D}_1$ cannot be used to audit a model's fairness with respect to protected groups a and b. An exact value of γ can then be obtained, that depends on the particular (sub)set $\mathcal{D}_1$ considered. Indeed, common statistical fairness metrics' sensitivity are data-dependent, as discussed in Cummings et al (2019) in the context of differentially-private fair learning. By a slight abuse of notation, we define $\gamma(h, \mathcal{D}_1)$ as the local l_1-sensitivity (as opposed to the global sensitivity γ) of the unfairness measure, given a classifier h, considering a dataset $\mathcal{D}_1$ and any neighbouring dataset $\mathcal{D}_2$ (see Definition 4) such that $\mathcal{D}_2 \subseteq \mathcal{D}_1$. In other words, $\gamma(h, \mathcal{D}_1)$ quantifies the maximum unfairness increase made possible by removing at most one element from $\mathcal{D}_1$. It upper bounds the fairness violation increase between $\mathcal{D}_1$ and any of its neighbours $\mathcal{D}_2$ such that $\mathcal{D}_2 \subseteq \mathcal{D}_1$:

$$\begin{aligned}\gamma(h, \mathcal{D}_1) &= \max_{\substack{\mathcal{D}_2 \subseteq \mathcal{D}_1 \\ J_\delta(\mathcal{D}_1, \mathcal{D}_2) = \frac{1}{|\mathcal{D}_1 \cup \mathcal{D}_2|}}} ||\mathsf{unf}(h, \mathcal{D}_2) - \mathsf{unf}(h, \mathcal{D}_1)||_1 \\ &= \max_{\substack{\mathcal{D}_2 \subseteq \mathcal{D}_1 \\ J_\delta(\mathcal{D}_1, \mathcal{D}_2) = \frac{1}{|\mathcal{D}_1 \cup \mathcal{D}_2|}}} |\mathsf{unf}(h, \mathcal{D}_2) - \mathsf{unf}(h, \mathcal{D}_1)|\end{aligned}$$

We can illustrate this notion with Figure 1. On one hand, the global sensitivity γ (Proposition 4) was previously able to upper bound the increase of the fairness violation among all edges of the graph. On the other hand, by leveraging the statistical fairness metrics' formulation, we can compute the (tighter) value of the local sensitivity $\gamma(h, \mathcal{D}_1)$, upper-bounding the unfairness increase through all edges outgoing the vertex associated to $\mathcal{D}_1$, in $\mathcal{G}'$ (the directed graph derived from Figure 1, where each edge from $\mathcal{D}_1$ to $\mathcal{D}_2$ means that $\mathcal{D}_2 \subseteq \mathcal{D}_1$).

Proposition 5 ***(Bounded worst-case fairness violation increase between consecutive perturbation sets (statistical fairness metrics))*** *Consider a dataset $\mathcal{D}$, a classifier h and a Jaccard distance $d \in [0, 1 - \frac{1}{|\mathcal{D}|}]$. We have:*

1. *Given a subset $\mathcal{D}'$ of $\mathcal{D}$, the value of $\gamma(h, \mathcal{D}')$ can be computed explicitly and has finite value.*
2. $\max_{\mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})} \mathsf{unf}(h, \mathcal{D}'') \leq \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)} \mathsf{unf}(h, \mathcal{D}') + \gamma(h, \mathcal{D}')$.
3. $\exists \mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$ *such that* $\mathsf{unf}(h, \mathcal{D}'') = \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)} \mathsf{unf}(h, \mathcal{D}') + \gamma(h, \mathcal{D}')$.

Proof We sketch the proof here and give the details in Appendix A. In this Proposition, $\gamma(h, \mathcal{D}')$ is the maximum increase of the unfairness measure made possible by removing at most one example from $\mathcal{D}'$. For statistical metrics, $\mathsf{unf}(h, \mathcal{D}') = |\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}|$ and we observe that there are exactly four ways of modifying $\mathsf{unf}(h, \mathcal{D}')$: removing an example of group a satisfying or not satisfying the measure, or removing an example of group b satisfying or not satisfying the measure. Note that this makes the complexity of computing $\gamma(h, \mathcal{D}')$ independent of $|\mathcal{D}'|$ (effectively resulting in $\gamma(h, \mathcal{D}')$ being computed in constant time for any $\mathcal{D}'$). □

Proposition 5 shows that the worst-case unfairness increase induced by extending our perturbation set with minimal change can be bounded and that this upper-bound can be reached (*i.e.*, the corresponding subset can be build by carefully selecting the element to be removed from $\mathcal{D}'$). The analysis of this bound (explicitly stated in Appendix A) demonstrates that the higher both groups' size are, the smoother $\mathsf{unf}(.)$ is. This observation naturally holds for any value of the Jaccard distance d, theoretically highlighting the advantage of working with sufficiently large protected groups. Indeed, too small protected groups cause unfairness to have higher sensitivity, hence being less stable to punctual changes in the data.

3.2.3 Conditions for Perfect Statistical Fairness Metrics Sample-Robustness

One important implication of Proposition 5 is that the worst-case fairness violation increase induced by minimal extension of any perturbation set is, in the general case, greater than 0 and can be computed exactly. Hence, it is in general not possible for a classifier h to be perfectly fair over all our perturbation sets. Formally, we say that h achieves perfect fairness sample-robustness given unfairness tolerance ϵ if and only if $\forall d \in [0,1], \forall \mathcal{D}' \in \mathcal{B}(\mathcal{D}, d),\ \mathsf{unf}(h, \mathcal{D}') \leq \epsilon$. In other terms, h achieves perfect fairness sample-robustness on $\mathcal{D}$ if and only if it meets the desired fairness constraint over all possible subsets of $\mathcal{D}$. In the general case, for common statistical fairness metrics, it is not possible for h to be fair for all $0 < d \leq 1$. Indeed, it is possible to build a subset of $\mathcal{D}$ on which unfairness is exactly 1.0 (hence violating any fairness constraint $\epsilon < 1$). Given the values of $S_i^{\mathcal{D}}$ and $X_i^{\mathcal{D}}$ for all protected groups i (here, $i \in \{a, b\}$), we can easily check whether the corresponding classifier h verifies perfect fairness sample robustness without building any subsets. The following proposition gives necessary and sufficient conditions that both imply the impossibility for h to satisfy perfect fairness sample robustness.

Proposition 6 ***(Necessary and sufficient conditions for perfect fairness sample-robustness infeasibility)*** *Consider a dataset $\mathcal{D}$ and a classifier h, as well as a maximum acceptable unfairness $\epsilon < 1$. Perfect fairness sample-robustness of h on $\mathcal{D}$ is infeasible, that is $\exists \mathcal{D}' \subseteq \mathcal{D}$ such that $\mathsf{unf}(h, \mathcal{D}') = 1.0$, if and only if one of the following two conditions holds:*

1. *$S_a^{\mathcal{D}} > 0$ and $S_b^{\mathcal{D}} < X_b^{\mathcal{D}}$. In this case, $\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)$, for all $d \geq \frac{1}{|\mathcal{D}|}(X_a^{\mathcal{D}} - S_a^{\mathcal{D}} + S_b^{\mathcal{D}})$.*
2. *$S_b^{\mathcal{D}} > 0$ and $S_a^{\mathcal{D}} < X_a^{\mathcal{D}}$. In this case, $\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)$, for all $d \geq \frac{1}{|\mathcal{D}|}(X_b^{\mathcal{D}} - S_b^{\mathcal{D}} + S_a^{\mathcal{D}})$.*

Proof The gist of the proof is that we can, by removing a sufficiently important number of examples from $\mathcal{D}$, get a subset $\mathcal{D}'$ which exhibits the worst possible unfairness $\mathsf{unf}(h, \mathcal{D}') = 1.0$. Indeed, to reach this value, one of the two rate measures $\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}}$ or

$\frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}$ has to be brought to 0 while the other reaches 1. To achieve value 0 for the ratio $\frac{S_i^{\mathcal{D}'}}{X_i^{\mathcal{D}'}}$, we need to remove $S_i^{\mathcal{D}}$ instances from $\mathcal{D}$ (those satisfying the measure), and to reach value 1.0, we must remove $X_i^{\mathcal{D}} - S_i^{\mathcal{D}}$ examples (those not satisfying the measure). Finally, we can either remove $X_a^{\mathcal{D}} - S_a^{\mathcal{D}} + S_b^{\mathcal{D}}$ or $X_b^{\mathcal{D}} - S_b^{\mathcal{D}} + S_a^{\mathcal{D}}$ carefully chosen examples from $\mathcal{D}$. In Case 1, we bring the a-ratio to 1 and the b-ratio to 0. In Case 2, it is the contrary. If neither 1 nor 2 can be applied, then necessarily either $S_i^{\mathcal{D}} = 0$ for all protected groups or $S_i^{\mathcal{D}} = X_i^{\mathcal{D}}$ for all protected groups. In that case, perfect fairness sample-robustness is achieved (which is discussed later in Proposition 8). □

Proposition 6 shows that a minimum finite value of d can be easily computed to define the smallest $\mathcal{B}(\mathcal{D}, d)$ containing a subset for which unfairness is exactly 1.0 (*i.e.*, fairness is 0.0 - the worst possible value). This illustrates the fact that considering too large values of d may not make sense and that the constraints defined in Problem (5) are very strong. For example, for the Statistical Parity metric, this implies that any classifier with non-constant prediction across protected groups a and b (hence, $X_a^{\mathcal{D}} > S_a^{\mathcal{D}} > 0$ or $X_b^{\mathcal{D}} > S_b^{\mathcal{D}} > 0$ for this metric) may be exactly fair on $\mathcal{D}$ but exactly unfair for some subset of $\mathcal{D}$ contained in $\mathcal{B}(\mathcal{D}, d)$. Additionally, Proposition 6 shows how this value of d can be computed. Based on the principles of this proposition, we infer in Proposition 7 a simple, yet powerful, sufficient condition for perfect fairness sample-robustness infeasibility based on the unfairness measure over $\mathcal{D}$.

Proposition 7 ***(Sufficient condition for perfect fairness sample-robustness infeasibility)*** *Consider a dataset $\mathcal{D}$ and a classifier h. If $\mathsf{unf}(h, \mathcal{D}) > 0$ then perfect fairness sample-robustness of h on $\mathcal{D}$ is infeasible for any maximum acceptable unfairness value $\epsilon < 1$.*

Proof Assume that $\mathsf{unf}(h, \mathcal{D}) > 0$, which means that $|\frac{S_a^{\mathcal{D}}}{X_a^{\mathcal{D}}} - \frac{S_b^{\mathcal{D}}}{X_b^{\mathcal{D}}}| > 0$, hence either $S_a^{\mathcal{D}} > 0$ or $S_b^{\mathcal{D}} > 0$ (or both). We also observe that either $S_a^{\mathcal{D}} < X_a^{\mathcal{D}}$ or $S_b^{\mathcal{D}} < X_b^{\mathcal{D}}$ (or both). If $S_a^{\mathcal{D}} > 0$, two cases are possible:

- On the one hand, it may be that $S_a^{\mathcal{D}} = X_a^{\mathcal{D}}$. Then, necessarily, $S_b^{\mathcal{D}} < X_b^{\mathcal{D}}$ and Case 1 of Proposition 6 is applicable.
- On the other hand, $S_a^{\mathcal{D}} < X_a^{\mathcal{D}}$
  - If $S_b^{\mathcal{D}} = X_b^{\mathcal{D}}$, then Case 2 of Proposition 6 is applicable.
  - If $S_b^{\mathcal{D}} = 0$, then Case 1 of Proposition 6 is applicable.
  - If $0 < S_b^{\mathcal{D}} < X_b^{\mathcal{D}}$ then either Case 1 or Case 2 of Proposition 6 are applicable.

A similar disjunction can be conducted over possible values of $S_b^{\mathcal{D}}$. Observe that if $\mathsf{unf}(h, \mathcal{D}) = 0$ then we can not conclude without looking at $S_a^{\mathcal{D}}$ and $S_b^{\mathcal{D}}$.

□

We now formulate necessary and sufficient conditions for guaranteeing perfect fairness sample-robustness.

Proposition 8 ***(Necessary and sufficient conditions for perfect fairness sample-robustness)*** *Consider a dataset $\mathcal{D}$ and a classifier h as well as a maximum acceptable unfairness $\epsilon < 1$. Perfect fairness sample-robustness is guaranteed if and only if one of the following conditions holds:*

1. $S_i^{\mathcal{D}} = X_i^{\mathcal{D}}$ *for all protected groups* i *(here,* $i \in \{a, b\}$*).*
2. $S_i^{\mathcal{D}} = 0$ *for all protected groups* i *(here,* $i \in \{a, b\}$*).*

Proof If $S_i^{\mathcal{D}} = X_i^{\mathcal{D}}$ for all protected groups i, then the ratio associated to all protected groups is exactly 1.0 and cannot be modified by removing examples. Similarly, if $S_i^{\mathcal{D}} = 0$ for all protected groups i, then the ratio associated to all protected groups is exactly 0.0 and cannot be modified by removing examples. Finally, in any other case, either Points 1 or 2 of Proposition 6 can be applied to prove the infeasibility of perfect fairness sample-robustness. □

The different possible cases mentioned in Propositions 6 and 8 are summarized in Figure 2. Observe that, as excepted, the conditions established in Proposition 8 are equivalent to the negation of those formulated in Proposition 6.

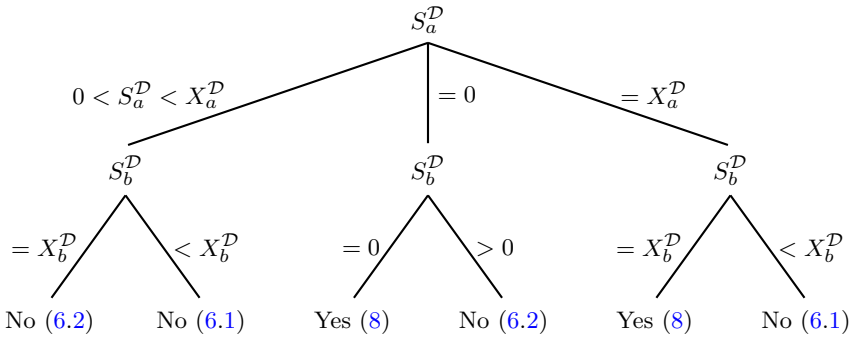

Fig. 2 The different possible situations to establish perfect fairness sample-robustness (Yes) or its impossibility (No)

3.2.4 Implications of Perfect Statistical Fairness Metrics Sample-Robustness

Based on the conditions established in Proposition 8, we now study the implications for h of being exactly fair for all our perturbation sets for statistical fairness metrics.

Consider a dataset $\mathcal{D}$ and a classifier h satisfying perfect fairness sample-robustness. On the one hand, it means that h is perfectly fair on $\mathcal{D}$ and all its

subsets, which can be desirable. However, depending on the fairness metric at hand, such a model may not be interesting as we show below.

Statistical Parity

For the Statistical Parity metric, it means that h's predictions are constant for all instances of groups a and b. This conflict strongly with utility and may result in h being a trivial model.

Predictive Equality

For the Predictive Equality metric, it means that either all negative samples of groups a and b are well classified (True Negative Rates are 1.0), or they are all misclassified (False Positive Rates are 1.0). Hence, a perfectly robust-fair model for this metric would either be 100% accurate over negative samples, or 100% inaccurate over such examples. Observe that the first case is desirable, but also easily reachable by a trivial classifier constantly predicting the negative class.

Equal Opportunity

For the Equal Opportunity metric, it means that either all positive samples of groups a and b are well classified (True Positive Rates are 1.0) or they are all misclassified (False Negative Rates are 1.0). Hence, a perfectly robust-fair model for this metric would also be either 100% accurate over positive samples, or 100% inaccurate over such examples. Observe that the first case is desirable, but also easily reachable by a trivial classifier constantly predicting the positive class.

Equalized Odds

The Equalized Odds metric is the conjunction of Predictive Equality and Equal Opportunity. Hence, a perfectly robust-fair model for this metric would be 100% accurate over its training set (or 100% inaccurate over its training set - in which case inverting its predictions would be sufficient).

These results illustrate the strength of our robustness notion. However, they also suggest that a direct application may not be possible. Indeed, a training accuracy of 100% cannot be reached in general. In addition, it may not be desirable to reach such accuracy as it usually indicates overfitting. In the next subsection, we show how our framework can be used to quantify the sample-robustness of statistical fairness.

3.3 Maximal Perturbation Set Ensuring Statistical Fairness Constraint

We showed in Proposition 8 that in the special situations in which both $S_a^{\mathcal{D}} = 0$ and $S_b^{\mathcal{D}} = 0$, or both $S_a^{\mathcal{D}} = X_a^{\mathcal{D}}$ and $S_b^{\mathcal{D}} = X_b^{\mathcal{D}}$, h is perfectly fair over all the perturbation sets defined with respect to $\mathcal{D}$. However, as discussed earlier, perfect fairness sample-robustness may not be desirable, nor achievable.

Instead of trying to enforce perfect sample-robustness of statistical fairness for a classifier, we will use our new framework to quantify a classifier's fairness sample-robustness, as defined in the following definition.

Definition 6 (***Quantifying sample-robustness for fairness***) Consider a dataset $\mathcal{D}$, a classifier h and an acceptable unfairness tolerance ϵ. The unfairness sample-robustness of h on $\mathcal{D}$ for constraint ϵ, denoted by $\mathcal{SR}(h, \mathcal{D}, \epsilon)$, is the Jaccard distance ($\mathcal{SR}(h, \mathcal{D}, \epsilon) \in [0, 1]$) such that:

1. $\forall d \geq \mathcal{SR}(h, \mathcal{D}, \epsilon), \exists \mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)$ such that $\mathsf{unf}(h, \mathcal{D}') > \epsilon$.
2. $\forall d < \mathcal{SR}(h, \mathcal{D}, \epsilon), \forall \mathcal{D}' \in \mathcal{B}(\mathcal{D}, d), \mathsf{unf}(h, \mathcal{D}') \leq \epsilon$.

In other words, $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ is the largest possible value of the Jaccard distance d such that h is fair over all sets in $\mathcal{B}(\mathcal{D}, d'), \forall d' < d$.

Consider that $\mathcal{D}$ and all its subsets are points into a metric space equipped with the Jaccard distance. Intuitively, $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ is the radius of the largest ball centered around $\mathcal{D}$ such that h if fair over all sample sets strictly contained within this ball. In simple words, h is fair on $\mathcal{D}$ and on subsets of $\mathcal{D}$ up to a (Jaccard) distance of $\mathcal{SR}(h, \mathcal{D}, \epsilon)$. The bigger $\mathcal{SR}(h, \mathcal{D}, \epsilon)$, the more sample-robust h's fairness is. The a- and b-ratios evolve non-linearly. Hence, it is not possible to compute a simple bound as in Proposition 6 for values of ϵ such that $0 < \epsilon < 1$. Therefore, we propose to consider a simple constrained optimization problem, denoted by $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$, to compute the minimal number of examples that need to be removed from $\mathcal{D}$ to build a subset of examples such that h is not ϵ-fair over it. Note, however, that the bounds proposed in Proposition 6 still hold, but are not tight (*i.e.,* using these bounds we get an over-estimation of $\mathcal{SR}(h, \mathcal{D}, \epsilon)$).

Definition 7 (***The integer program for quantifying sample-robustness for fairness***) A solution of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ is a tuple (x_a, x_b, y_a, y_b), in which these four decision variables represent the number of examples to be removed from $\mathcal{D}$ to form a subset on which the unfairness constraint is violated.

More precisely, x_i represents the number of examples of group i satisfying the given measure (hence counted within both $S_i^{\mathcal{D}}$ and $X_i^{\mathcal{D}}$), while y_i represents the number of examples of group i not satisfying the given measure (hence counted only within $X_i^{\mathcal{D}}$). The optimal solution of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ is the one minimizing the total number of examples to be removed (7) to build the closest (in the Jaccard sense) subset of $\mathcal{D}$.

$$\mathcal{IPSR}(h, \mathcal{D}, \epsilon): \tag{6}$$

$$\min_{x_a, x_b, y_a, y_b} \quad x_a + x_b + y_a + y_b \tag{7}$$

$$\text{s.t.} \quad \left| \frac{S_a^{\mathcal{D}} - x_a}{X_a^{\mathcal{D}} - x_a - y_a} - \frac{S_b^{\mathcal{D}} - x_b}{X_b^{\mathcal{D}} - x_b - y_b} \right| > \epsilon \tag{8}$$

$$0 \leq x_a \leq S_a^{\mathcal{D}} \tag{9}$$

$$0 \leq x_b \leq S_b^{\mathcal{D}} \tag{10}$$

$$0 \leq y_a \leq X_a^{\mathcal{D}} - S_a^{\mathcal{D}} \tag{11}$$
$$0 \leq y_b \leq X_b^{\mathcal{D}} - S_b^{\mathcal{D}} \tag{12}$$
$$x_a + y_a < X_a^{\mathcal{D}} \tag{13}$$
$$x_b + y_b < X_b^{\mathcal{D}}. \tag{14}$$

Constraint (8) encodes the fact that the fairness constraint must be violated on the resulting subset. Constraints (9) to (12) capture the variables' domains. Finally, constraints (13) and (14) enforce that at least one example of each group is kept (otherwise unfairness is undefined).

Illustration of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ ***for an example metric:*** For the Equal Opportunity metric, recall that $S_i^{\mathcal{D}}$ is the number of positively labelled examples belonging to group i that are positively predicted by h (true positives). For this metric, $X_i^{\mathcal{D}}$ is the total number of positively labelled examples belonging to group i. Then, x_i represents the number of examples removed from $\mathcal{D}$ that belong to group i and are positively labelled and positively predicted by h. Removing x_i such examples decrements both $S_i^{\mathcal{D}}$ and $X_i^{\mathcal{D}}$. On the other side, y_i is the number of examples removed from $\mathcal{D}$ that belong to group i and are positively labelled and negatively predicted by h. Removing y_i such examples decrements only $X_i^{\mathcal{D}}$.

In the next proposition, we show that $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ can be used to exactly compute a classifier's fairness sample-robustness $\mathcal{SR}(h, \mathcal{D}, \epsilon)$.

Proposition 9 *(**Quantifying Sample-Robustness for fairness using** $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$) Let* $(x_a^*, x_b^*, y_a^*, y_b^*)$ *be the optimal solution of* $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$*. Then:*

$$\mathcal{SR}(h, \mathcal{D}, \epsilon) = \frac{x_a^* + x_b^* + y_a^* + y_b^*}{|\mathcal{D}|}.$$

Proof To prove this equality, we will need to prove the two conditions of Definition 6.

Let $z^* = x_a^* + x_b^* + y_a^* + y_b^*$ be the value of the objective function of the optimal solution of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$. We define $d^* = \frac{z^*}{|\mathcal{D}|} = \frac{x_a^* + x_b^* + y_a^* + y_b^*}{|\mathcal{D}|}$. Then:

1. Consider $\mathcal{D}^*$, the subset of $\mathcal{D}$ formed by removing x_a^* (respectively x_b^*) examples of group a (respectively b) satisfying the statistical criterion, and y_a^* (respectively y_b^*) examples of group a (respectively b) not satisfying the statistical criterion. The bounds of the decision variables of Problem (6) enforce that $\mathcal{D}^*$ exists. We have: $J_\delta(\mathcal{D}, \mathcal{D}^*) = \frac{x_a^* + x_b^* + y_a^* + y_b^*}{|\mathcal{D}|} = d^*$. Additionally, we know that $\mathsf{unf}(h, \mathcal{D}^*) > \epsilon$, because $(x_a^*, x_b^*, y_a^*, y_b^*)$ is a solution of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ and then necessarily satisfies Constraint (8). Hence, $\forall d \geq d^*, \exists \mathcal{D}' = \mathcal{D}^* \in \mathcal{B}(\mathcal{D}, d)$ such that $\mathsf{unf}(h, \mathcal{D}') > \epsilon$.
2. Assume that $\exists \mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d)$ with $d < d^*$ such that $\mathsf{unf}(h, \mathcal{D}'') > \epsilon$. Then, $\mathcal{D}''$ is formed by removing $z'' < z^*$ examples from $\mathcal{D}$. In addition, $\mathcal{D}''$ is a solution to Problem (6) as $\mathsf{unf}(h, \mathcal{D}'') > \epsilon$. This contradicts the fact that z^* is the optimal objective value of Problem (6). Hence, $\forall d < d^*, \forall \mathcal{D}' \in \mathcal{B}(\mathcal{D}, d), \mathsf{unf}(h, \mathcal{D}') \leq \epsilon$.

Finally, by (1) and (2), $d^* = \mathcal{SR}(h, \mathcal{D}, \epsilon)$. □

$\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ can be solved using any Mixed-Integer Programming (MIP) solver (more implementation details are provided in Section 5.1.2). The main computational challenge resides in the fact that Constraint (8) is non-linear. However, due to the modest size of the model, common solvers are able to solve the problem to optimum within fractions of seconds.

Additionally, based on the principles described in Proposition 5, we have designed a simple greedy algorithm $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ that can be used to approximate $\mathcal{SR}(h, \mathcal{D}, \epsilon)$. Its pseudo-code is depicted in Appendix B.

Intuitively, $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ starts with the entire dataset $\mathcal{D}$, and successively removes examples to build subsets of $\mathcal{D}$ until fairness is violated. At each step, it removes exactly one example from the current subset $\mathcal{D}^c$. This example is chosen to maximize the fairness violation increase. Indeed, this value, as well as the associated example to be removed, can be computed in constant time using $\gamma(h, \mathcal{D}^c)$ as defined in Proposition 5. This is due to the fact that, given $\mathcal{D}^c$, only four possible operations can be considered to modify fairness: remove an example of protected group a (respectively b) that satisfies (respectively does not satisfy) the fairness requirement.

$\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ comes with no optimality guarantee and we can easily craft instances on which it does not achieve optimality. However, it provides an upper-bound on $\mathcal{SR}(h, \mathcal{D}, \epsilon)$, and has polynomial $\mathcal{O}(|\mathcal{D}|)$ complexity, where $|\mathcal{D}|$ is the number of examples in $\mathcal{D}$. We show empirically in Section 5.1.4 that it can approximate $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ well in practice.

We formally defined our sample-robustness criterion for fairness, as well as an Integer Programming model to precisely quantify it. One of the strengths of the proposed approach is that it can be used to quantify the fairness robustness of any classifier h, given only an access to its predictions. In particular, the approach is agnostic to the hypothesis class of the classifier h, no additional assumptions are necessary and a black-box access to the model is sufficient. In the next subsection, we present the resulting learning problem statement. Afterwards, we discuss the practical issues with this formulation and show how it can be integrated within existing learning algorithms.

3.4 Integration with Fair Learning Algorithms and Practical Challenges

As discussed previously, the use of the Jaccard distance to define the perturbation sets around $\mathcal{D}$ has some theoretical advantages. However, carefully calibrating the parameter d to avoid over-constraining the problem is necessary. In addition, the resulting problem may still be hard to solve in practice and/or penalize utility too much.

A possible use would be to solve $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ and directly use the resulting $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ to quantify the fairness sample-robustness of classifier h. Integrating this term directly within the objective function of a learning

algorithm might appear to be suitable. However, this would require solving $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ at each model update to be able to audit the fairness sample-robustness. For instance, this would not be trivial for gradient-based learning techniques, as $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ is not a differentiable value. In addition, sample-robustness values found by solving $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ depend on the dataset considered and its structure (in particular, through the influence of the cardinalities of the protected groups). This implies that a particular sample-robustness value may be satisfactory for a given task, but may not be meaningful for another dataset or another pair of protected groups. Furthermore, there is often an important gap between realistic, task-useful models' sample-robustness and that of any constant classifier (which is 1.0). These observations make the integration of our robustness quantification notion into learning algorithms more difficult. Then, we formulate the sample-robust fair learning problem as a multi-objective problem, using an ϵ-constraint method. In other words, considering the fair learning problem (3), we include our fairness sample-robustness term as a constraint:

$$\begin{aligned} \underset{h \in \mathcal{H}}{\arg\min} \quad & f_{obj}(h, \mathcal{D}) \\ \text{s.t.} \quad & \mathsf{unf}(h, \mathcal{D}) \leq \epsilon \\ & \mathcal{SR}(h, \mathcal{D}, \epsilon) \geq \mu \end{aligned} \tag{15}$$

Note that Problem (15) is indeed equivalent to Problem (5), reformulated to use the fairness sample-robustness quantification notion introduced in Section 3.3 (*i.e.*, $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ and the discussed tools to measure it). In particular, the μ parameter of Problem (15) corresponds to the d parameter of Problem (5).

As discussed earlier, an important difficulty with Problem (15) is the calibration of the μ parameter. More precisely, as a meaningful value of μ depends on the dataset at hand, on the considered sensitive attributes, on the unfairness metric and on the unfairness constraint ϵ, determining a good value for μ is difficult. For this reason, we propose to build a Pareto frontier between utility ($f_{obj}(h, \mathcal{D})$) and fairness sample-robustness ($\mathcal{SR}(h, \mathcal{D}, \epsilon)$), for a fixed value of ϵ. To realize this, we first solve Problem (15) with no constraint on $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ ($\mu = 0$). Then, we measure the resulting model's h_0 sample robustness and solve Problem (15) again, using this value ($\mathcal{SR}(h_0, \mathcal{D}, \epsilon)$) for the μ parameter. We iterate this process until reaching perfect sample-robustness of 1.0 (which can always be reached by building a trivial constant classifier). However, remark that this approach has two drawbacks. First, it is not obvious which solution of the Pareto frontier should be kept. Second, as the process is done sequentially, it may be long to finish and this time is not predictable.

To address the first challenge, once a model's training is finished, we propose to audit its fairness on a separate validation set. If the validation unfairness meets a given criterion (*e.g.,* is lower than the training unfairness

or lower than ϵ), we return this model. Otherwise, we strengthen the sample-robustness constraint and iterate. The second difficulty remains (even though the validation step stops the process earlier instead of building the entire Pareto frontier).

Finally, we propose a sample-robustness framework for statistical fairness. After characterizing our perturbation sets structure and the resulting learning problem, we show how it can be integrated within existing algorithms. We further conduct an experimental evaluation of this approach in Section 5.1. However, practical difficulties remain, such as an important computational overhead and practical integration challenges (solving a MIP within a learning algorithm). These challenges motivate a heuristic formulation of the problem.

4 A Heuristic Method to Improve Fairness Sample-Robustness

In this section, we propose a heuristic method designed to improve fairness sample-robustness, without exhibiting some of the practical limitations of the exact approach proposed in the previous section. First, we introduce this heuristic method before showing how it can be integrated into two state-of-the-art fair learning algorithms.

4.1 Approximating the Perturbation Sets

We have showed that an exact application of our proposed formulation is possible, but challenging. Indeed, in practice, a heuristic application of our proposed principle can be beneficial, even if no formal guarantees hold. The approach we propose consists in computing n random subsets of the training set using n random binary masks. Each mask $\mathcal{M}_i$ is a vector of size $|\mathcal{D}|$, in which each coordinate $\mathcal{M}_{ij} \in \{0, 1\}$ ($i \in \{1 \dots n\}$ and $j \in \{1 \dots |\mathcal{D}|\}$) is a random binary value. We denote by $\mathcal{D}_i$ the subset associated with mask $\mathcal{M}_i$ as follows: $\mathcal{D}_i = \{e_j \in \mathcal{D} \mid \mathcal{M}_{i,j} = 1\}$. This is used in Definition 8 to define the heuristic perturbation set.

Definition 8 (***Heuristic perturbation sets***)
Consider a dataset $\mathcal{D}$ and a set of n binary masks $M_1 \dots M_n$ of size $|\mathcal{D}|$. The heuristic perturbation set, denoted by $\mathcal{B}_\omega(\mathcal{D}, n)$, is defined as: $\mathcal{B}_\omega(\mathcal{D}, n) = \{\mathcal{D}, \mathcal{D}_1, \mathcal{D}_2, \dots \mathcal{D}_n\}$.

In a nutshell, instead of considering the entire previously defined perturbation set $\mathcal{B}(\mathcal{D}, d)$, we only enforce fairness on some randomly generated subsets (belonging to $\mathcal{B}(\mathcal{D}, d)$ by construction). Intuitively, $\mathcal{B}(\mathcal{D}, d)$ considers all subsets of $\mathcal{D}$ whose Jaccard distance from $\mathcal{D}$ is at most d. In contrast, $\mathcal{B}_\omega(\mathcal{D}, n)$ only considers n random subsets of $\mathcal{D}$ (along with $\mathcal{D}$ itself). In the graph

representation of Figure 1, our heuristic perturbation sets contain randomly selected vertices.

By replacing $\mathcal{B}(\mathcal{D}, d)$ by $\mathcal{B}_\omega(\mathcal{D}, n)$ in Problem (5), we get the heuristic formulation of our sample-robust fair learning problem. The intuition behind this heuristic approach is that the randomly sampled subsets of $\mathcal{D}$ have slightly different distributions. Hence, enforcing fairness for such subsets effectively leads to a form of heuristic distributionally robust optimization. It is possible to draw a parallel with the Bagging (Bootstrap AGGregatING) ensemble learning method (Zhou, 2012). Indeed, the idea underlying bagging is that training different models using different samplings of the training set may improve robustness by reducing the variance. This happens because such samplings have slightly different distributions, neighbouring the original one. While bagging leverages the different samplings to learn a set of models that will reduce the variance of the accuracy, we use them to enforce fairness in a robust manner.

This heuristic formulation does not have the theoretical appeal of our exact sample-robustness quantification framework, but exhibits considerable practical advantages. Indeed, it does not require calibrating the μ parameter of Problem (15), which explains why we no longer need a separate validation set. In addition, computing unfairness over a finite set of subsets defined with masks can be done in linear time with respect to the input size, which is considerably simpler than solving $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$. It is also easier to integrate within existing algorithms (and in particular, gradient-based techniques - as we show later).

Compared to the approach of Mandal et al (2020), our heuristic method for robust fair learning does not come with theoretical guarantees, but its simplicity provides practical advantages in terms of scalability and applicability. First, it can be easily integrated into most fair classification techniques and it does not require access to a cost-sensitive learning algorithm (in cost-sensitive learning, the instances of the training set are associated to weights that define their contribution to the objective function value). Second, unlike Cotter et al (2018, 2019a), we do not require a prior split of the data. Finally, as we show later in the experiments section, our heuristic method can be efficiently integrated within fair learning algorithms and allows an empirical improvement of the generalization of fairness.

In the next two sections, we show how to include our heuristic method into two state-of-the-art fair classification techniques (solving Problem (3)) that have different characteristics. The first one is an exact branch-and-bound algorithm that builds inherently interpretable models. It works with binary data and binary protected group membership. As interpretability is becoming a key property for machine learning models (Freitas, 2014; Rudin, 2019), we believe that our method could be applicable in a wide range of contexts.

The second one is based on a two-player game formulation of constrained optimization. It uses gradient-based techniques without necessarily binarizing

the data and handles fairness for any number of protected groups. Both methods are metric-agnostic and could be used to enforce any statistical fairness measure.

4.2 Integration with FairCORELS

FairCORELS (Aïvodji et al, 2019; Aïvodji et al, 2021) is an extension of the CORELS (Angelino et al, 2017, 2018) algorithm that builds fair rule lists on binary datasets (attributes and labels) given two protected groups. A rule list (Rivest, 1987) is a classification model defined by an ordered list of **if-then** rules (with a default prediction if non of the rules applies). Given a collection of rules (consisting in any combination of attributes mined as preprocessing), FairCORELS certifiably builds a rule list with the highest objective function among those meeting a given statistical fairness constraint. It is a branch-and-bound algorithm that represents the search space of the rule lists $\mathcal{R}$ using a prefix tree. In this prefix tree, each node is a rule and each path from the root to a node is a prefix (ordered set of rules), that can be extended with a default decision to form a potential solution. Leveraging a collection of bounds, FairCORELS explores this search space using a given search heuristic and updates the current best solution only when the candidate model has an unfairness value at most equal to a given ϵ. Let $\mathsf{misc}(.)$ be the misclassification error, $\mathsf{unf}(.)$ the unfairness oracle, K_r the length of prefix of rule list r and λ a regularization parameter penalizing longer rule lists, FairCORELS solves the following constrained optimization problem:

$$\begin{aligned} \underset{r\in\mathcal{R}}{\arg\min}\ & f_{obj_{\texttt{FairCORELS}}} = \mathsf{misc}(r,\mathcal{D}) + \lambda.K_r \\ \text{s.t.}\quad & \mathsf{unf}(r,\mathcal{D}) \leq \epsilon \end{aligned}$$

Integrating the proposed heuristic perturbation sets into the FairCORELS algorithm is quite simple. Whenever an evaluated rule list improves over the current best objective function, it is accepted only if it has an unfairness value lower than ϵ on the training set and on each of its subsets defined by the n masks. Finally, the modified algorithm searches for the rule list solution to the following problem:

$$\begin{aligned} \underset{r\in\mathcal{R}}{\arg\min}\ & f_{obj_{\texttt{FairCORELS}}} = \mathsf{misc}(r,\mathcal{D}) + \lambda.K_r \\ \text{s.t.}\quad & \max_{\mathcal{D}'\in\mathcal{B}_\omega(\mathcal{D},n)} \mathsf{unf}(h,\mathcal{D}') \leq \epsilon \end{aligned}$$

In practice, it is often not necessary to compute $\mathsf{unf}(.)$ for each subset. We only compute these quantities if the candidate prefix improves over the current best one and meets the fairness constraint on the training set. In this case, subsets unfairness are computed sequentially and stopped early if the constraint is violated on any of the subsets. This efficient implementation leads to no significant computational overhead compared to the original FairCORELS.

4.3 Integration with TFCO

TensorFlow Constrained Optimization[3] (TFCO) is a Python library for optimizing inequity-constrained problems in TensorFlow to produce machine learning models (not restricted to the fair learning problem). It implements the method of Cotter et al (2019b), formulating the constrained optimization problem as a two-player game. Considering the Lagrangian relaxation of the problem, the first player (θ-player) optimizes a model's parameters to minimize the objective function while the second player (λ-player) updates the Lagrangian-multipliers to approximate the strongest Lagrangian relaxation. While original fairness constraints are non-differentiable proportions (linear combinations of indicators), TFCO allows for the computation of objective and proxy constraints as hinge upper bounds of the real quantities, which allows for the use of gradient-based techniques. In this setting, Cotter et al (2019b) proposes the Proxy Lagrangian framework. The latter reduces the constrained optimization problem to a two-player non-zero sum game, in which the "learner" optimizes the model's parameters to minimize objective function including proxy constraints while the "auditor" updates the Lagrangian multipliers based on the true constraints' violations.

In the general context of constrained optimization, the proxy Lagrangians associated to the two players optimizing objective function $g_0(\theta)$ under m constraints $g_{i,i\in[m]}$ are:

$$\mathcal{L}_\theta(\theta,\lambda) = \lambda_1 g_0(\theta) + \sum_{i=1}^{m} \lambda_{i+1}\tilde{g}_i(\theta) \qquad (16)$$

$$\mathcal{L}_\lambda(\theta,\lambda) = \sum_{i=1}^{m} \lambda_{i+1} g_i(\theta)$$

in which λ_j are the Lagrange multipliers, g_i measures violation of constraint i and $\tilde{g}_i$ is its differentiable proxy.

Integrating our heuristic perturbation sets into the TFCO framework does not require modifying the library. Indeed, it simply consists in including additional constraints to the declared optimization problem, to enforce the fairness constraints on the subsets of the training set defined by the n masks. Formally, we add one constraint per protected group per mask. When dealing with m protected groups, the original fair formulation includes m constraints (bounding a statistical measure's difference between each protected group and the overall training set). Our heuristic sample-robust method declares $m.(n+1)$ fairness constraints (enforcing the m fairness constraints on the $n+1$ sets of $\mathcal{B}_\omega(\mathcal{D},n)$) that will be included in the objective function and weighted with Lagrange-multipliers, following the approach of Cotter et al (2019b), as in Equation (16). Finally, integrating our proposed heuristic approach within TFCO is quite straightforward. One may observe that it would not be the case

[3] https://github.com/google-research/tensorflow_constrained_optimization

for the exact method proposed in Section 3, as the output of $\mathcal{IPSR}$ is not a differentiable value.

5 Experiments

We now empirically evaluate the proposed sample-robustness approaches for statistical fairness, over a variety of datasets, sensitive attributes, and statistical fairness metrics, and two fair learning algorithms of the literature.

In a first subsection, we compare the exact and heuristic formulations, both integrated within the `FairCORELS` algorithm. Afterwards in a second subsection, we demonstrate the effectiveness of our heuristic formulation within `TFCO` and compare it to a state-of-the art technique for improving statistical fairness generalization (Cotter et al, 2018, 2019a).

5.1 Integration into `FairCORELS` and Comparison between Exact and Heuristic Methods

In this section, we integrate and evaluate our exact and heuristic methods within the `FairCORELS` algorithm. We first introduce the considered setup and define the different methods implemented. Then, we show that the exact formulation effectively improves unfairness generalization through the performed iterations. In the fourth and fifth subsections, we show that our heuristic method improves fairness sample-robustness and statistical fairness generalization. Finally, we compare the exact and heuristic approaches in terms of fairness sample-robustness and learning quality (trade-offs between accuracy and fairness at test time).

5.1.1 Setup

Our experiments cover four binarized datasets widely used in the fair learning literature. For each dataset, the mined rules are single- and two-clause antecedents (*i.e.*, conjunctions of at most two attributes or their negation).

- Adult Income dataset[4] (Frank and Asuncion, 2010). This dataset gathers records of more than 45,000 individuals from the 1994 U.S. census. We consider the binary classification task of predicting whether an individual earns more than 50,000$ per year. We use the same preprocessing as Aïvodji et al (2019); Aïvodji et al (2021), considering gender to be the binary sensitive attribute (`Male` or `Female`).
- COMPAS dataset[5](analyzed by Angwin et al (2016)). We consider the same discretized dataset used to evaluate `CORELS` in Angelino et al (2017). The associated label/decision is whether the person will re-offend (recidivate) within 2 years (yes or no) while the binary sensitive attribute is race

[4]https://archive.ics.uci.edu/ml/datasets/adult
[5]https://raw.githubusercontent.com/propublica/compas-analysis/master/compas-scores-two-years.csv

(`African-American` or `Caucasian`). Rule mining is done similarly to Aïvodji et al (2019); Aïvodji et al (2021).

- Defaut of Credit Card Clients dataset[6] (Yeh and hui Lien, 2009). The dataset is discretized using quantiles. The discrete attributes were used to generate single and two clause rules, and only rules with support higher than 0.5 were kept. The associated decision is whether the person will default in payment (the next time they use their credit card), with the sensitive attribute being gender (`Male` or `Female`). The resulting dataset contains 189 rules and 29,986 examples.
- Bank Marketing dataset[7] (Moro et al, 2014). The dataset is discretized using quantiles. The discrete attributes are used to generate single and two clause rules and only rules with support higher than 0.5 were kept. The associated decision is whether the person will subscribe to a term deposit. The resulting dataset contains 41,175 examples and 179 rules, among which the 2 protected attributes : `age:30-60` and `not_age:30-60`.

For each dataset, we prevent the use of the sensitive attributes in the model built to avoid disparate treatment. For all experiments, we set the maximum number of nodes in `FairCORELS`' prefix tree to 2.5×10^6 along with some fixed parameters such as the branching heuristic after a preprocessing step.

5.1.2 Methods

Exact Method. We modified `FairCORELS` to solve Problem (15)[8]. Solving $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ is costly (even though it can be done in fractions of seconds in practice) and we should avoid doing it at each iteration of the learning algorithm. Within `FairCORELS`, $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ is solved only when the current best solution update subroutine is called (just like mask-related constraints are verified in the integration of our heuristic approach). In other words, we only audit a model's sample-robustness when it is about to become the new current best solution. This guarantees that the final solution meets the desired fairness sample-robustness constraint, while in practice performing a small number of calls to the solver. For our experiments, $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ is solved using the `OR-Tools` CP-SAT solver (Perron and Furnon, 2019). Implementation of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ necessitates some reformulation, in particular regarding Constraint (8), which is non-linear and requires products computation. Indeed, we can get rid of the divisions by multiplying both sides of the inequality by the (positive) product of the denominator variables, and further using intermediate variables. We consider the four fairness metrics presented in Table 1 and an unfairness tolerance $\epsilon \in \{0.02, 0.015, 0.01, 0.005\}$. Results for the different values of ϵ show similar trends, hence we only report those for $\epsilon = 0.01$ for conciseness reasons. We compare five variants based on our exact method:

[6] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients
[7] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[8] Our source code will be publicly released upon acceptance.

- We solve Problem (15) iteratively and update the fairness sample-robustness constraint μ at each step, without validation set, until $\mathcal{SR}(h, \mathcal{D}, \epsilon) = 1$. More precisely, we build the entire Pareto frontier between accuracy and fairness sample-robustness, for a fixed value of ϵ. We denote this set of models the `sample robust fair frontier (no validation)`. While selecting a particular model within the built sequence remains an open problem, this enables to visualize the different trade-offs that are obtained during the iterations. Among these built models, we select the non-constant one with higher fairness sample robustness. We call this the `no validation (before-constant)` method.
- We solve Problem (15) iteratively with a validation set. We then obtain a Pareto frontier between train accuracy and fairness sample-robustness. This set of models is the `sample robust fair frontier (validation)`. We leverage the validation set to define two stopping criteria. The first one, called `validation (`ϵ `criterion)`, stops iterating when the validation unfairness is under ϵ (*i.e.*, when the fairness constraint enforced on the training set is also met on the validation set). The second one is called `validation (train unf. criterion)`. It stops iterating when the validation unfairness is smaller or equal to the training one.

Heuristic Method. The integration of our heuristic approach within `FairCORELS` is depicted in Section 4.2[9]. For a number of masks $n \in \{0, 10, 30\}$, we compute the training Pareto frontier (between accuracy and fairness) of `FairCORELS`, with a fixed list of 147 values for the unfairness tolerance ϵ (ranging non-linearly with a higher density for higher fairness constraints). By evaluating each model obtained on its test set, we obtain approximations of the Pareto frontier of `FairCORELS` in test. We compute such frontier for the four different statistical notions of fairness presented in Table 1. Based on this setup, we assess `FairCORELS`' fairness generalization ability, with ($n \in \{10, 30\}$) or without ($n = 0$) our proposed approach.

All reported values are averaged using 5-folds cross-validation, with all methods are trained and evaluated on the same data splits. For the methods `sample robust fair frontier (validation)`, `validation (`ϵ `criterion)` and `validation (train unf. criterion)`, part of the original training set is used for validation (and not used for training the model).

5.1.3 Exact Method: Effects of Fairness Sample-Robustness on Fairness Generalization and Accuracy

In this subsection, we show that our exact method effectively improves the statistical fairness generalization. We also visualize the resulting trade-offs between accuracy and fairness sample-robustness.

Figures 3 and 4 illustrate the obtained results on the four datasets, for the Statistical Parity fairness metric ($\epsilon = 0.01$). Results for the three other metrics ($\epsilon = 0.01$), and for the four datasets, are given in Appendices C.1

[9] Our source code will be publicly released upon acceptance.

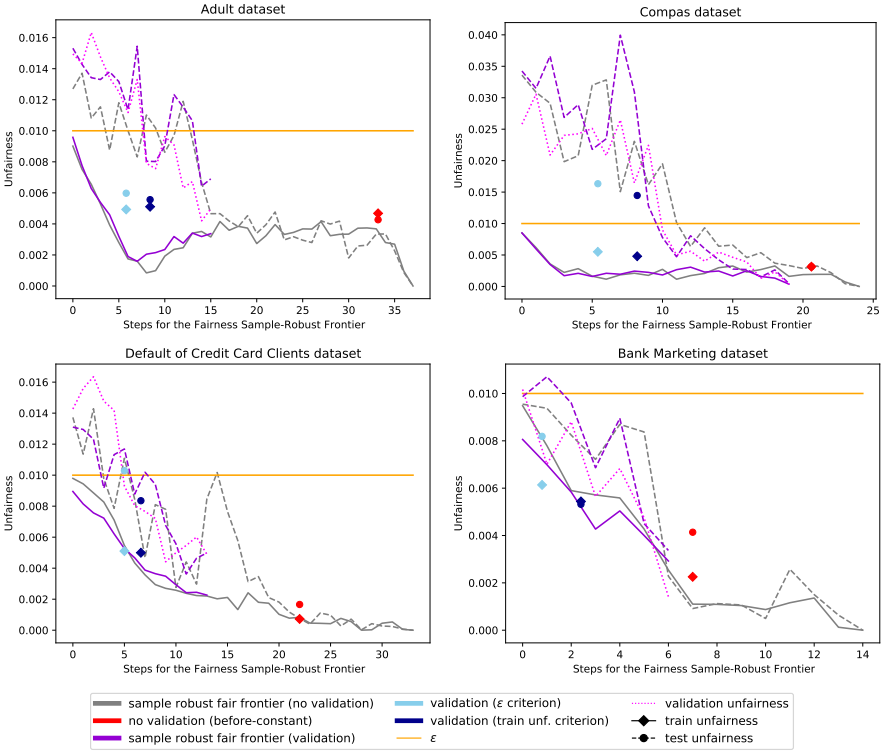

Fig. 3 Training, test and validation (when applicable) unfairness of models generated by `FairCORELS` through the iterations of our exact method (Statistical Parity metric, $\epsilon = 0.01$).

and C.2. Detailed results for the Statistical Parity metric (including standard deviation) are included in Appendix C.3. In both figures, the `sample-robust fair frontier` (with or without validation set) presents all obtained models, sorted by their index in the sequence. This means that the leftmost point corresponds to the unconstrained model, while the rightmost point exhibits 1.0 fairness sample-robustness and is obtained after the last iteration of the process. Note that this last model is usually a constant classifier, as discussed in Section 3.2.4. However, this is not always the case, and for the Predictive Equality and Equal Opportunity metrics, some models achieve perfect sample-robustness without reaching trivial accuracy. This is possible if the built rule lists only make mistakes on the positively (respectively negatively) labelled instances. This is the case, for example, if the model's prefix rules only capture positively (respectively negatively) labelled examples, while a default decision classifies negatively (respectively positively) the uncaptured ones.

Intuitively, we strengthen the fairness sample-robustness constraint of Problem (15) through the iterations of the method (numbered in the x-axis). This increases the fairness sample-robustness sequentially, as shown in

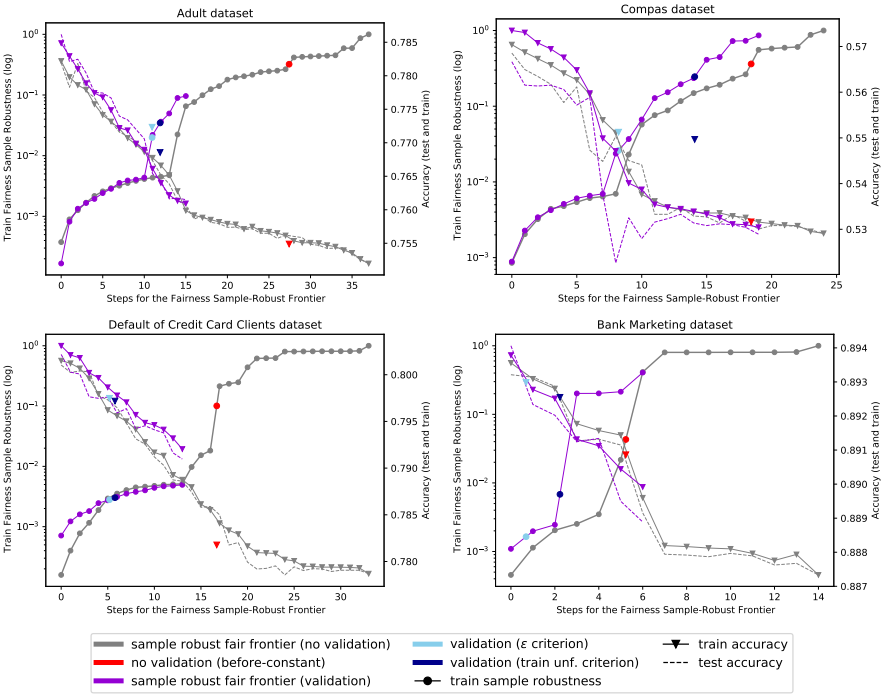

Fig. 4 Fairness sample-robustness and accuracy of models generated by `FairCORELS` through the iterations of our exact method (Statistical Parity metric, $\epsilon = 0.01$)

Figure 4. This effectively lowers the test unfairness as expected (Figure 3), but degrades the training and test accuracy (Figure 4). This suggests that a trade-off between accuracy and fairness sample-robustness exists. In Figure 3, we see that considering the non-constant classifier with higher fairness sample-robustness (`no validation (before-constant)` method) satisfies the fairness requirement at test time, which was precisely the aim of our method. However, we see in Figure 4 that this fairness generalization improvement comes at a high cost in terms of accuracy (both at training and test times).

While using a separate validation set, we see that we usually require fewer iterations and get models closer to the unfairness tolerance ϵ. Such models do not always meet the fairness constraint at test time, but still lead to a reduction of the test unfairness.

One may observe that the points associated to `no validation (before-constant)`, `validation (`ϵ` criterion)` and `validation (train unf. criterion)` do not lie on their associated curves. The reason for this is that the stopping criteria are applied separately on each fold (with the fold's validation set). Hence, the models obtained with these methods are learnt with different number of steps on different folds. The x-positioning of the points is then performed based on the average sample-robustness value within the

corresponding frontier. Finally, we see that when the fairness constraint is already satisfied (as for the Bank Marketing dataset), the method still allows a reduction of the test unfairness, which may strengthen unfairness robustness.

The evaluation of our exact methods' test accuracy as well as comparison with our heuristic method is performed later in Section 5.1.6. In the next two subsections, we evaluate our heuristic method.

5.1.4 Heuristic Method: Fairness Sample-Robustness Improvement Results

In this part of our experiments, we use our heuristic sample-robust method within `FairCORELS` and empirically show that it effectively improves the fairness sample-robustness. As stated previously, we compare three variants: the original `FairCORELS` (`no mask`) as well as `FairCORELS` modified with our heuristic sample-robust method for `10 masks` and `30 masks`.

We use $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ and $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ to audit *a posteriori* the fairness sample-robustness of the models built with each of the three methods. As discussed earlier, $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ provides no optimality guarantee, but it has polynomial complexity and can be used to upper-bound $\mathcal{SR}(h, \mathcal{D}, \epsilon)$. On the other side, $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ computes the exact value of $\mathcal{SR}(h, \mathcal{D}, \epsilon)$ but is computationally more expensive.

To empirically demonstrate that our heuristic approach is suitable to improve sample-robustness, we report in Figure 5 results for the four metrics (Statistical Parity, Predictive Equality, Equal Opportunity and Equalized Odds), for the Defaut of Credit Card Clients dataset. Results for the remaining datasets are given in Appendix C.4. When the fairness constraint enforced is strengthened ($1-\epsilon$ grows), fairness sample-robustness decreases. We explain this by the fact that, as the fairness constraint becomes tighter, it is met on a lower number of subsets of the training set. In particular, the radius of the ball (measured using the Jaccard distance) around the training dataset in which the fairness constraint is met everywhere ($\mathcal{SR}(h, \mathcal{D}, \epsilon)$) diminishes. However, we observe in Figure 5 that our heuristic method is able to mitigate the decrease of the fairness sample-robustness. Additionally, there seems to be a correlation between the number of masks used and the resulting fairness sample-robustness. We note that $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ actually performs very well and proposes a close over-approximation of $\mathcal{SR}(h, \mathcal{D}, \epsilon)$. For these experiments, $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ found the optimal solution (*i.e.*, the value returned by $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$) 88% of the time. For the remaining 12%, the average gap to optimality is less than 5%. Considering the four datasets and the four fairness metrics, $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ found the optimal solutions for 78% of the executions. For the remaining 22%, the average gap to optimality is around 36%. This means that, for some executions, the value found by $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ is considerably higher than the exact value of $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$, resulting in a high upper-bound.

We have showed that our heuristic method empirically improves fairness sample-robustness. In the next subsection, we show that as a result it also

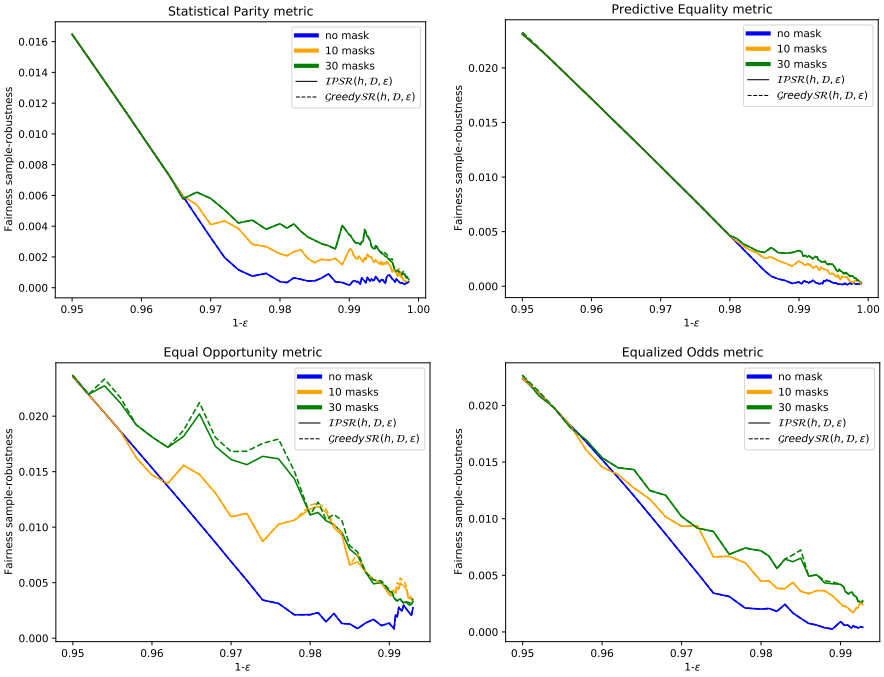

Fig. 5 Fairness sample-robustness of models generated by `FairCORELS` using our heuristic method (Default of Credit Card Clients dataset)

improves the fairness generalization, hence allowing the construction of models with better test accuracy/fairness trade-offs in regimes of low unfairness.

5.1.5 Heuristic Method: Statistical Fairness Generalization Improvement Results

In this part of our experiments, we use our heuristic sample-robust method within `FairCORELS` and empirically show that it effectively enhances the statistical fairness generalization.

Figures 6, 7, 8 and 9 present respectively the performances of the three variants on the Adult Income, COMPAS, Defaut of Credit Card Clients and Bank Marketing datasets, for the Equal Opportunity metric (which is widely used in the literature). Results for the four datasets using all fairness metrics, which display similar trends, are included in Appendix C.5. Each figure has three graphs, with each point of a graph corresponding to a solution (averaged with the 5-folds cross validation). Note that we focus on solutions whose unfairness is at most 0.05, because this part of the trade-offs (medium to strong fairness constraints) is the most interesting one in our experiments. It allows us to investigate unfairness generalization under meaningful constraints. To

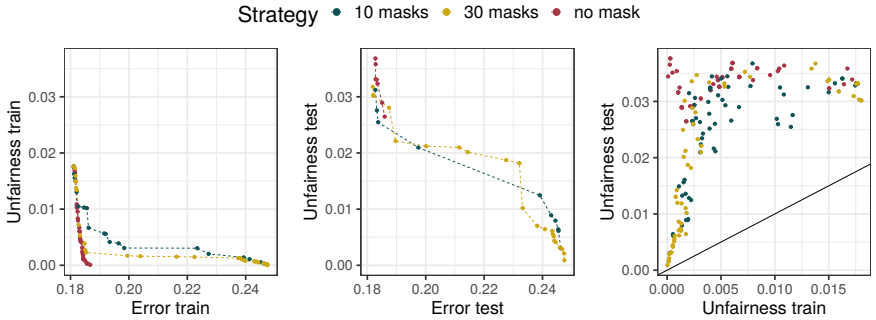

Fig. 6 Results of our heuristic method (Adult Income dataset, Equal Opportunity metric).

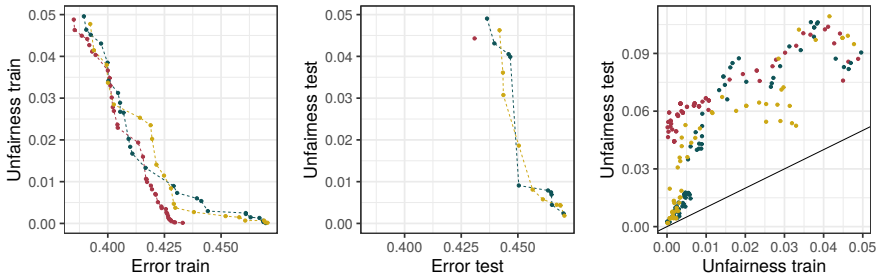

Fig. 7 Results of our heuristic method (COMPAS dataset, Equal Opportunity metric).

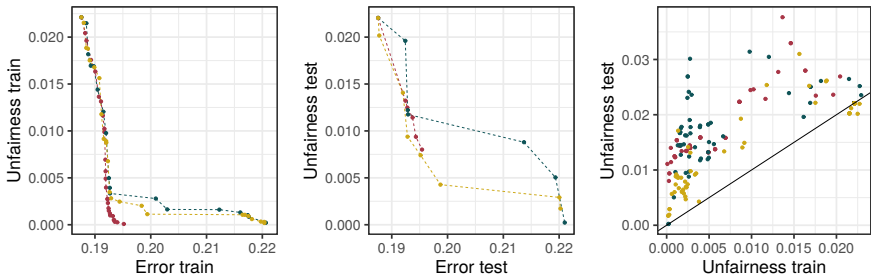

Fig. 8 Results of our heuristic method (Default Credit dataset, Equal Opportunity metric).

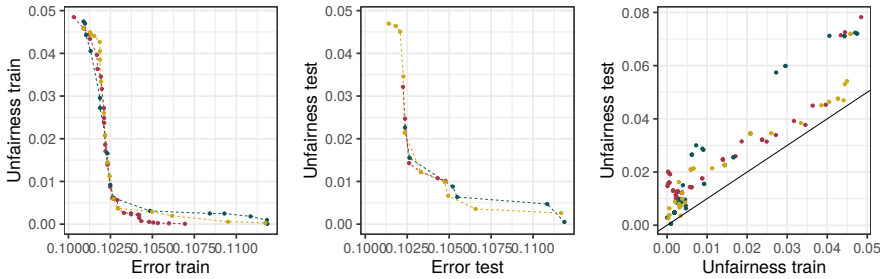

Fig. 9 Results of our heuristic method (Marketing dataset, Equal Opportunity metric).

ease train/test comparison, the ranges of both axis of the first two plots are set identically.

The first graph is a Pareto frontier built on the training set, which displays the set of non-dominated solutions (in terms of unfairness and error) on the training set. Solutions closer to the lower left corner are preferable, as they correspond to lower error and lower unfairness. We observe that, in all cases, our heuristic method leads to a lower trade-off on the training set. This suggests that robustness comes at the cost of a lower training performance, which is not really problematic as we shall prefer solutions exhibiting worse accuracy/fairness trade-offs at training time but generalizing better.

The second graph is the Pareto frontier built on the test set, which illustrates the effectiveness of our proposed approach. In all cases, using our heuristic method (with either 10 or 30 masks) leads to a denser Pareto frontier, exhibiting better accuracy/fairness tradeoffs. This is particularly the case for tight fairness constraints, in which the standard `FairCORELS` fails to generate solutions exhibiting such low test unfairness.

The third graph illustrates the generalization of unfairness, presenting the test unfairness as a function of the training one. The ideal generalization scenario, in which the test unfairness is exactly equal to the training unfairness, is represented by the diagonal line. We observe that for tight fairness constraints (left side of the figure), solutions generated by the original `FairCORELS` generalize badly: they exhibit low training unfairness but considerably higher test unfairness. In contrast, solutions generated using our approach generalize considerably better as the corresponding points are closer to the ideal scenario.

Finally, all the presented results show that the use of our heuristic method leads to the generation of models trading some training performances for fairness robustness, and presenting better accuracy/fairness trade-offs on unseen data (at test time). In particular, models learnt using our heuristic method have a considerably smaller unfairness generalization error, which allows for populating areas of the test Pareto frontier that the original `FairCORELS` failed to fill in.

5.1.6 Comparing the Exact and Heuristic Approaches

We now compare the proposed exact and heuristic approaches on similar problems.

Figure 10 displays the built models' sample-robustness, for the Statistical Parity metric ($\epsilon = 0.01$). Results for the three other metrics ($\epsilon = 0.01$) and for the four datasets, are given in Appendix C.6. We observe that both the heuristic and exact methods are able to improve fairness sample-robustness over the original `FairCORELS`. Overall, as expected the exact method can lead to higher sample-robustness values as the models are learnt with constraints over this precise value.

Figure 11 shows the obtained test error/test unfairness trade-offs for the Statistical Parity metric ($\epsilon = 0.01$). Results for the three other metrics ($\epsilon = 0.01$), for the four datasets, are given in Appendix C.7. We observe that all

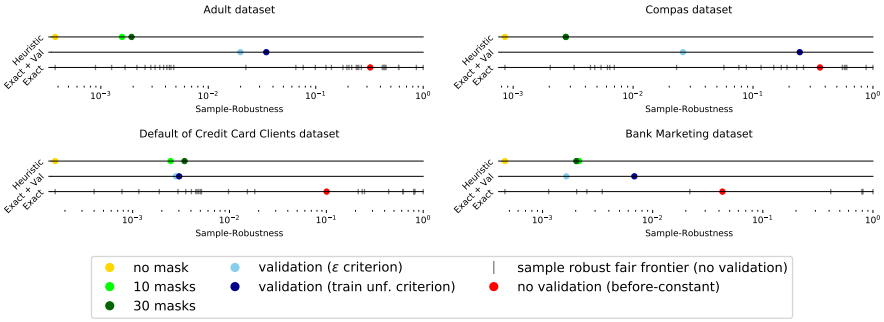

Fig. 10 Fairness sample-robustness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Statistical Parity metric, $\epsilon = 0.01$)

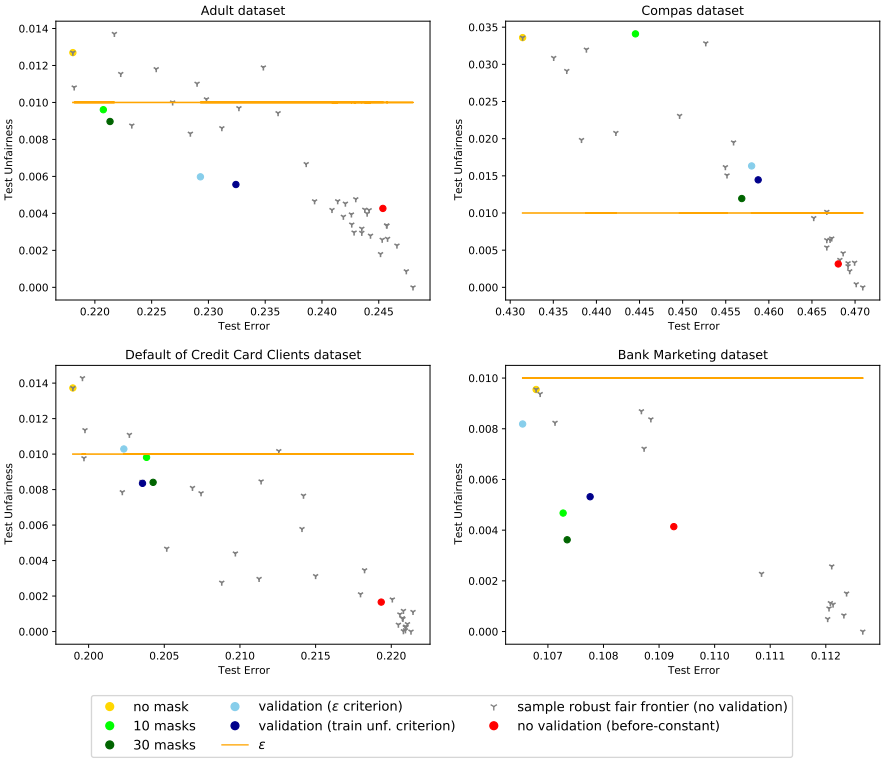

Fig. 11 Test error and unfairness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Statistical Parity metric, $\epsilon = 0.01$)

the proposed methods usually diminish fairness violation at test time (the associated points are either under ϵ or closer to it). This improvement on fairness generalization induces a cost on the model's error. As a general trend, we see that the greater the fairness generalization improvement, the greater the error incurred. However, the generated solutions often propose interesting trade-offs between error and unfairness. In particular, in the Bank Marketing experiment the robustness enforced for fairness can sometimes benefit the error generalization as well.

Finally, we showed that the integration of our exact or heuristic methods within `FairCORELS` practically improve fairness generalization. While both approaches successfully enforce fairness robustness, the exact method is computationally more expensive, because it consists in a sequence of trainings (while the heuristic method only trains once). In addition, each training is more expensive as it requires solving an integer programming model. Overall, the heuristic approach seems more appealing for practical applications, thanks to its flexibility, computational efficiency and empirical effectiveness.

5.2 Heuristic Method: Integration into `TFCO` and Comparison to a State-of-the-Art Method

The objective of this section is two-fold. First, we show that the use of our heuristic sample-robust method improves fairness generalization over the standard fair learning formulation using `TFCO`. Second, we compare these results with a state-of-the-art method improving fairness generalization (Cotter et al, 2018, 2019a). This approach possesses scalability and applicability properties similar to ours. It is also implemented using `TensorFlow Constrained Optimization`, which enables a direct comparison with our heuristic method.

5.2.1 Setup

For these experiments, we build on the setup of Cotter et al (2019a) and compare the following approaches:

- `unconstrained` trains a model without enforcing fairness constraints (hence only minimizing training error).
- `baseline` is the fair learning approach based on the implementation of Cotter et al (2019b) for non-convex constrained optimization (minimizing training error subject to fairness constraints).
- `validation` is the approach described in Cotter et al (2018, 2019a), which is proposed to improve fairness generalization over the `baseline` approach. In a nutshell, to avoid *constraints overfitting*, the training set is split between two distinct sets: *train* and *validation*. Then, the Lagrangians of Equation (16) are computed on these two different sets. On the one side, the θ-player optimizes the model parameters over *train*. On the other side, the λ-player measures fairness constraints violations on *validation*.
- `dromasks-n` is the integration of our method into `baseline`, using n masks (in practice, we use $\mathtt{n} \in \{10, 30, 50\}$).

All methods are implemented using the `TensorFlow Constrained Optimization` library (Cotter et al, 2019b). Similar to Cotter et al (2019a), we evaluate all methods using two formulations: the Proxy Lagrangian described in Equation (16) (Algorithm 2 of Cotter et al (2019b)) and the usual Lagrangian (Algorithm 3 of Cotter et al (2019b)).

For `baseline`, `validation` and `dromasks-n`, the result of the training is a sequence of iterations. Following Cotter et al (2019a), we use their "shrinking" procedure to find the *best* stochastic classifier supported on the sequence of iterates. We average all the results obtained over 100 runs. For `unconstrained` and `baseline`, the runs differ by the random seed used to generate the mini-batches. For `validation`, the runs are different due to the seed used to generate the training/validation split, and for `dromasks-`n, they differ by the seed used to generate the random binary masks. All methods see exactly the same training data at each run and are evaluated over the same testing set.

We extend the setup of Cotter et al (2019a), considering four experiments using different datasets and notions of fairness[10]. We train the neural network models with one hidden layer containing 50 ReLU neurons. All models are trained using minibatches of 100 instances. For each experiment, we measure the training and testing errors and maximum fairness constraint violations. The latter increases as the reported values increase (values ≤ 0 correspond to no fairness violation). For the `validation` method, training error is computed on the training subset while training constraint violation is computed on the validation subset similarly to Cotter et al (2019a).

Experimentation 1 uses the UCI Adult dataset[11] (Frank and Asuncion, 2010), preprocessed to include only binary attributes, with the classification task being to predict whether a person's yearly income is greater than \$50,000, subject to the 80% rule for Statistical Parity. This means that for each of four overlapping protected classes (Black, White, Female and Male), the Positive Prediction Rate must be at least 80% of the overall Positive Prediction Rate. We use the designated training/testing split, with $32,561$ instances for training and $16,281$ for testing. The dataset has 122 attributes with the model being trained for 40 epochs.

Experimentation 2 uses the ProPublica COMPAS dataset[12] (analyzed by Angwin et al (2016)), preprocessed to include only binary attributes, with the classification task being to predict recidivism, subject to Equal Opportunity fairness constraints. This means that for each of four overlapping protected classes (Black, White, Female and Male), the Positive Prediction Rate on the positively-labeled examples (True Positive Rate) must be at most 5% higher than the overall Positive Prediction Rate on positively-labeled examples. We use a designated training/testing split, with $4,134$ instances for training and

[10] The third and fourth experiments differ respectively because we could not reproduce their results and because the data is not publicly available

[11] https://archive.ics.uci.edu/ml/datasets/adult

[12] https://raw.githubusercontent.com/propublica/compas-analysis/master/compas-scores-two-years.csv

2,038 for testing. The dataset has 32 attributes with all models being trained for 100 epochs.

Experimentation 3 uses the UCI Bank Marketing dataset[13] (Moro et al, 2014), with the classification task being to predict subscription to a term deposit, subject to Predictive Equality fairness constraints. We use a version of the dataset with 16 attributes, among which 10 are categorical and 6 are numerical. We form four protected groups based on the quartiles of the real-valued `age` attribute and constrain each group's false positive rate to be no larger than that of the full dataset. We use a designated training/testing split, with 30,292 instances for training and 14,919 for testing. While the 6 real attributes are left unchanged (*i.e.*, they are not discretized, as both `TFCO` and our proposed framework can handle them directly), we one-hot encode categorical ones. We also apply a standard preprocessing (centering and scaling) to all features. The resulting dataset contains 51 attributes. All models are trained for 200 epochs.

Experimentation 4 uses the Defaut of Credit Card Clients dataset[14] (Yeh and hui Lien, 2009), which contains 23 numerical attributes. Among them, there are 9 integer attributes representing categories, and 14 real attributes. We do not discretize them, as `TFCO` is able to handle continuous features, and our proposed framework imposes no limitation on the attributes' nature. Here also, we apply a standard preprocessing (centering and scaling) to all features. There are 30,000 examples in the dataset. We generate 100 random splits, using two thirds of the dataset for training, and one third for testing. We form four overlapping protected groups, based on the values of the attributes `gender` (Male or Female) and `age` (Young or Old, based on the median value). The classification task is to predict whether a client will default in payment. However, the dataset is highly unbalanced, with about 78% negative examples. Hence, machine learning models can reach a high predictive accuracy without accurately detecting positive examples. For this reason, we enforce that the True Positive rates among each protected group is at least 50%. This may result in slightly increasing the overall error (because we might increase the False Positive rates), but detecting more positive examples. Remark that such constraints do not follow the traditional statistical fairness formulation but nevertheless, we show that our approach is still able to improve their generalization. The models are trained for 100 epochs.

5.2.2 Results

Table 2 summarizes the results obtained and shows that our method effectively improves fairness generalization while not penalizing accuracy significantly. Overall, the method is competitive to the state-of-the-art `validation` method without requiring prior split of the data. Results of Experimentation 1 on Adult Income demonstrate that the fairness constraints violations on the test set are the smallest using our method. In addition, only the `dromasks-n` techniques

[13] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing
[14] https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

Table 2 Error rates and maximum fairness constraints violations for all compared methods, for our four experiments (all values are averaged over 100 runs as described in the setup). Best test results are shown in **bold**, second best in *italics*.

	Proxy Lagrangian				**Lagrangian**			
	Train		Test		Train		Test	
Method	Error	Viol.	Error	Viol.	Error	Viol.	Error	Viol.
Adult Income Dataset								
`unconstrained`	.122	.072	**.144**	.071	.122	.072	**.144**	.071
`baseline`	.141	0	*.154*	.009	.141	0	*.155*	.006
`validation`	.132	-.002	.158	.004	.134	0	.157	*.004*
`dromasks-10`	.14	-.003	.156	*.003*	.143	-.001	*.155*	**-.003**
`dromasks-30`	.14	-.004	.157	**-.001**	.148	-.002	.156	**-.003**
`dromasks-50`	.14	-.003	.157	**-.001**	.151	-.002	.157	**-.003**
COMPAS Dataset								
`unconstrained`	.265	.043	**.33**	.064	.265	.043	.33	.064
`baseline`	.263	-.004	**.33**	.019	.264	-.003	.328	.025
`validation`	.235	.001	.353	**.005**	.235	-.002	.352	.001
`dromasks-10`	.261	-.008	*.336*	.014	.295	-.007	*.326*	-.006
`dromasks-30`	.261	-.009	.337	.015	.307	-.009	*.326*	*-.011*
`dromasks-50`	.262	-.009	.337	*.013*	.31	-.011	**.322**	**-.012**
Bank Marketing Dataset (unfairness violations are $\times 10^{-1}$)								
`unconstrained`	.058	.071	.102	.161	.058	.071	.102	.161
`baseline`	.073	.001	**.099**	.096	.081	0	**.099**	.073
`validation`	.078	.005	.102	**.041**	.075	0	.105	.042
`dromasks-10`	.074	.001	**.099**	.091	.089	0	*.101*	.057
`dromasks-30`	.076	0	**.099**	.083	.114	0	.115	*.009*
`dromasks-50`	.078	0	*.1*	*.082*	.117	0	.117	**.003**
Default of Credit Card Clients Dataset								
`unconstrained`	.171	.141	**.181**	.164	.171	.141	**.181**	.164
`baseline`	.18	-.001	*.192*	*.035*	.183	-.002	*.193*	.03
`validation`	.18	.001	.203	**.012**	.182	-.001	.204	.011
`dromasks-10`	.18	-.002	*.192*	*.035*	.185	-.017	.197	.016
`dromasks-30`	.18	-.002	.193	*.035*	.188	-.035	.202	*.001*
`dromasks-50`	.18	-.002	.193	*.035*	.19	-.041	.205	**-.005**

are able to meet the fairness constraints on the test set. Furthermore, increasing the number of masks seems to improve the fairness generalization while penalizing accuracy, which suggest a fairness robustness / accuracy trade-off. While the `validation` method also proposes an important reduction of the test fairness violation, `dromasks-n` gives more interesting results on these experiments while less conflicting with accuracy (which was expected as in the `validation` approach, each player only sees half of the data during training). Results for Experimentation 2 (on the COMPAS dataset) suggest that in some situations the fact that our approach does not use a separate validation set (but subsets of the same training data) can limit its generalization improvement abilities. However, compared to `validation`, it has a considerably smaller impact on accuracy, and the resulting trade-offs appear competitive overall. Additionally, we observe that enforcing fairness constraints in a robust manner can improve error generalization due to the metric used (*i.e.*, Equal Opportunity) being

aligned with accuracy. Hence, ensuring fairness robustness may also benefit to accuracy. This is also observed is the third experiment on the Bank Marketing dataset, in which some fair models are also more accurate at testing time. Overall, the Lagrangian algorithm appears more suitable for our method. Indeed, we observe that enforcing our proxy constraints (as done with the Proxy Lagrangian algorithm) may not always lead to significant generalization improvements using our masks method. This is particularly clear in the fourth experiment using the Default of Credit Card Clients dataset.

Overall, our method, combined with the Lagrangian algorithm, leads to the most important constraints violation generalization improvements, while having limited impact on accuracy.

6 Conclusion

We proposed a novel formulation of robustness for fair learning aimed at enhancing the statistical fairness generalization in machine learning. Our framework is metric-agnostic and based on the idea that one wants to learn a model whose fairness is verified, even if the training dataset sampling is somehow different. Our formulation is designed to be widely applicable, as many real-world machine learning applications consider finite training sets. In addition, the proposed method can be used both to audit any classifier's fairness robustness without any knowledge of the classifier's structure but also for robust fair learning, although it has some practical limitations. To deal with this issue, we proposed an effective and efficient heuristic method, exhibiting practical advantages while still improving fairness sample-robustness and fairness generalization.

A limitation of our framework is that it considers only subsets of the training set (and not all possible sample sets within a given Jaccard distance). This prevents the creation of unrealistic sample sets, which could result in over-constraining the problem. It also gives an interesting structure to our perturbation sets, allowing the derivation of several theoretical properties. Additionally, it leads to an important computational advantage. Indeed, fairness sample robustness audit can be performed solving an integer programming model whose objective function is linear in the decision variables. However, in a more general formulation of sample robustness, this would not be the case, as the denominator of the Jaccard distance would no longer be a constant. Formulating and solving this problem efficiently is a promising direction as well as studying the theoretical and empirical privacy implications of our sample-robustness formulation for fairness.

Finally, automatically determining the best parameters for our heuristic method (*i.e.,* distribution and number of the binary masks, and cardinalities of the defined subsets) is also a research avenue that we want to pursue in the future.

Appendix A Proof of Proposition (5)

Proposition 5 ***(Bounded worst-case fairness violation increase between consecutive perturbation sets (statistical fairness metrics))*** *Consider a dataset $\mathcal{D}$, a classifier h and a Jaccard distance $d \in [0, 1 - \frac{1}{|\mathcal{D}|}]$. We have:*

1. *Given a subset $\mathcal{D}'$ of $\mathcal{D}$, the value of $\gamma(h, \mathcal{D}')$ can be computed explicitly and has finite value.*
2. $\max_{\mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d+\frac{1}{|\mathcal{D}|})} \mathsf{unf}(h, \mathcal{D}'') \leq \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)} \mathsf{unf}(h, \mathcal{D}') + \gamma(h, \mathcal{D}')$
3. $\exists \mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$ *such that* $\mathsf{unf}(h, \mathcal{D}'') = \max_{\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)} \mathsf{unf}(h, \mathcal{D}') + \gamma(h, \mathcal{D}')$.

Proof 1. We consider a dataset $\mathcal{D}'$, whose associated unfairness is $\mathsf{unf}(h, \mathcal{D}')$. In the context of this Proposition, $\gamma(h, \mathcal{D}')$ is then the maximum increase of the unfairness measure made possible by removing at most one example from $\mathcal{D}'$. We will denote the resulting dataset $\mathcal{D}''$ throughout this proof. Recall that $\mathsf{unf}(h, \mathcal{D}') = |\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}|$. Observe that there are exactly four ways of modifying $\mathsf{unf}(h, \mathcal{D}')$.

- In the first case, we remove an example of group a not satisfying the measure. The a-ratio becomes $\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}-1}$. Then, $\mathsf{unf}(h, \mathcal{D}'') = \frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}-1} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}$ and we have:

$$\begin{aligned}
\mathsf{unf}(h, \mathcal{D}'') - \mathsf{unf}(h, \mathcal{D}') &= \left(\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'} - 1} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}\right) - \left(\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}\right) \\
&= \frac{S_a^{\mathcal{D}'}.X_a^{\mathcal{D}'} - S_a^{\mathcal{D}'}.(X_a^{\mathcal{D}'} - 1)}{(X_a^{\mathcal{D}'} - 1).X_a^{\mathcal{D}'}} \\
&= \frac{S_a^{\mathcal{D}'}}{(X_a^{\mathcal{D}'} - 1).X_a^{\mathcal{D}'}} \\
&= \alpha_1(\mathcal{D}')
\end{aligned}$$

  We note that this change is possible only if $S_a^{\mathcal{D}'} < X_a^{\mathcal{D}'}$ and $X_a^{\mathcal{D}'} > 1$. We define $\mathbb{1}(\xi)$ as the indicator function, which evaluates to 1 if ξ is True, and to 0 otherwise. Finally, we note:

$$\beta_1(\mathcal{D}') = \mathbb{1}(S_a^{\mathcal{D}'} < X_a^{\mathcal{D}'} \wedge X_a^{\mathcal{D}'} > 1).\alpha_1(\mathcal{D}')$$

- In the second case, we remove an example of group b satisfying the measure. The b-ratio becomes $\frac{S_b^{\mathcal{D}'}-1}{X_b^{\mathcal{D}'}-1}$. Then, $\mathsf{unf}(h, \mathcal{D}'') = \frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}-1}{X_b^{\mathcal{D}'}-1}$ and

we have:

$$\begin{aligned}\mathsf{unf}(h,\mathcal{D}'') - \mathsf{unf}(h,\mathcal{D}') &= (\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}-1}{X_b^{\mathcal{D}'}-1}) - (\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}) \\ &= \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}-1}{X_b^{\mathcal{D}'}-1} \\ &= \frac{S_b^{\mathcal{D}'}.(X_b^{\mathcal{D}'}-1) - (S_b^{\mathcal{D}'}-1).X_b^{\mathcal{D}'}}{(X_b^{\mathcal{D}'}-1).X_b^{\mathcal{D}'}} \\ &= \frac{X_b^{\mathcal{D}'} - S_b^{\mathcal{D}'}}{(X_b^{\mathcal{D}'}-1).X_b^{\mathcal{D}'}} \\ &= \alpha_2(\mathcal{D}')\end{aligned}$$

We note that this change is possible only if $S_b^{\mathcal{D}'} > 0$ and $X_b^{\mathcal{D}'} > 1$. Finally, we note:

$$\beta_2(\mathcal{D}') = \mathbb{1}(S_b^{\mathcal{D}'} > 0 \wedge X_b^{\mathcal{D}'} > 1).\alpha_2(\mathcal{D}')$$

- In a third case, we remove an example of group a satisfying the measure. This will decrease the a-ratio, which becomes $\frac{S_a^{\mathcal{D}'}-1}{X_a^{\mathcal{D}'}-1}$. However, because we consider the absolute value of the difference, this may result in increasing unfairness overall. Then, $\mathsf{unf}(h,\mathcal{D}'') = |\frac{S_a^{\mathcal{D}'}-1}{X_a^{\mathcal{D}'}-1} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}|$. If $\frac{S_a^{\mathcal{D}'}-1}{X_a^{\mathcal{D}'}-1} \geq \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}$, unfairness is not increased and this modification should not be considered. We hence only consider the case where $\frac{S_a^{\mathcal{D}'}-1}{X_a^{\mathcal{D}'}-1} < \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}$. Then, we have:

$$\begin{aligned}\mathsf{unf}(h,\mathcal{D}'') - \mathsf{unf}(h,\mathcal{D}') &= (\frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}} - \frac{S_a^{\mathcal{D}'}-1}{X_a^{\mathcal{D}'}-1}) - (\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}) \\ &= \frac{2.S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}} - \frac{2.S_a^{\mathcal{D}'}.X_a^{\mathcal{D}'} - X_a^{\mathcal{D}'} - S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}.(X_a^{\mathcal{D}'}-1)} \\ &= \alpha_3(\mathcal{D}')\end{aligned}$$

We note that this change is possible only if $S_a^{\mathcal{D}'} > 0$ and $X_a^{\mathcal{D}'} > 1$. Finally, we note:

$$\beta_3(\mathcal{D}') = \mathbb{1}(S_a^{\mathcal{D}'} > 0 \wedge X_a^{\mathcal{D}'} > 1 \wedge \frac{S_a^{\mathcal{D}'}-1}{X_a^{\mathcal{D}'}-1} < \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}).\alpha_3(\mathcal{D}')$$

- In a fourth case, we remove an example of group b not satisfying the measure. This will increase the b-ratio, which becomes $\frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}-1}$. However,

because we consider the absolute value of the difference, this may result in increasing unfairness overall. Then, $\mathsf{unf}(h, \mathcal{D}'') = |\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}-1}|$. If $\frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}-1} \leq \frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}}$, unfairness is not increased and this modification should not be considered. We hence only consider the case where $\frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}-1} > \frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}}$. Then, we have:

$$\begin{aligned}\mathsf{unf}(h, \mathcal{D}'') - \mathsf{unf}(h, \mathcal{D}') &= (\frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}-1} - \frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}}) - (\frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} - \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}}) \\ &= \frac{2.S_b^{\mathcal{D}'}.X_b^{\mathcal{D}'} - S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}.(X_b^{\mathcal{D}'}-1)} - \frac{2.S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}} \\ &= \alpha_4(\mathcal{D}')\end{aligned}$$

We note that this change is possible only if $S_b^{\mathcal{D}'} < X_b^{\mathcal{D}'}$ and $X_b^{\mathcal{D}'} > 1$. Finally, we note:

$$\beta_4(\mathcal{D}') = \mathbb{1}(S_b^{\mathcal{D}'} < X_b^{\mathcal{D}'} \wedge X_b^{\mathcal{D}'} > 1 \wedge \frac{S_b^{\mathcal{D}'}}{X_b^{\mathcal{D}'}-1} > \frac{S_a^{\mathcal{D}'}}{X_a^{\mathcal{D}'}}).\alpha_4(\mathcal{D}')$$

By picking the option that worsens fairness the most, we build $\mathcal{D}''$ such that

$$\begin{aligned}\mathsf{unf}(h, \mathcal{D}'') - \mathsf{unf}(h, \mathcal{D}') &= \gamma(h, \mathcal{D}') \\ &= \max(\beta_1(\mathcal{D}'), \beta_2(\mathcal{D}'), \beta_3(\mathcal{D}'), \beta_4(\mathcal{D}'))\end{aligned}$$

An interesting observation is that $\gamma(h, \mathcal{D}')$ depends on the protected groups sizes. In particular, the bigger such groups are, the smaller $\gamma(h, \mathcal{D}')$ is (because $X_j^{\mathcal{D}'}, j \in \{a, b\}$ terms appear at the denominators of all $\alpha_i, i \in \{1..4\}$).

2. Consequence of Proposition 4 and of the fact that $\gamma(h, \mathcal{D}')$ is the l_1-sensitivity of the unfairness measure of h over dataset $\mathcal{D}'$, in the particular case where we remove at most one example from $\mathcal{D}'$. Intuitively, observe that for all $\mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$, there exists a superset $\mathcal{D}' \in \mathcal{B}(\mathcal{D}, d)$ such that $\mathcal{D}''$ is formed by removing exactly one element from $\mathcal{D}'$. Hence, $\mathsf{unf}(h, \mathcal{D}'') \leq \mathsf{unf}(h, \mathcal{D}') + \gamma(h, \mathcal{D}')$.
3. In 1, we showed that $\gamma(h, \mathcal{D}')$ can be reached be carefully selecting the element to be removed from $\mathcal{D}'$ to build dataset $\mathcal{D}'' \in \mathcal{B}(\mathcal{D}, d + \frac{1}{|\mathcal{D}|})$.

□

Appendix B Greedy Algorithm for Quantifying Sample Robustness

The pseudo-code of $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ is depicted in Algorithm 1. Intuitively, our greedy algorithm starts from $\mathcal{D}$ and successively removes elements to build a subset of $\mathcal{D}$. The process stops when the fairness constraint is violated over the current subset or when there are no more examples that can be removed to increase unfairness. At each iteration of the main loop, we compute the unfairness increase induced by the four possible moves (removing an element from protected group i that satisfies (or not) the statistical criterion). Details on the computation of these values can be found in the proof of Proposition 5 in Appendix A. We then execute the move associated to the higher unfairness increase. Finally, the algorithm returns the Jaccard distance from $\mathcal{D}$ to the (implicitly) built subset.

The objective of this greedy strategy is to find the closest subset (in the Jaccard sense) on which the fairness constraint is violated. This algorithm comes with no optimality guarantee (as the subset found by the greedy strategy may not be the closest from the original dataset). In other terms, $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$ returns an upper-bound on $\mathcal{SR}(h, \mathcal{D}, \epsilon)$, and this upper-bound may be not be tight. Its has $\mathcal{O}(|\mathcal{D}|)$ worst-case complexity, as the operations performed within the While loop are constant-time, and this loop is executed at most $|\mathcal{D}| - 2$ times (we keep at least one example from each group - which is guaranteed by the conditions of the indicator functions). Such appealing complexity can be achieved because we do not need to explicitly build the corresponding subsets (as, for the unfairness metric, only subgroups cardinalities S_i and X_i matter).

Algorithm 1 $\mathcal{G}reedy\mathcal{SR}(h, \mathcal{D}, \epsilon)$

Input: classifier h's predictions, dataset $\mathcal{D}$, unfairness tolerence ϵ

$S_a, X_a, S_b, X_b \Leftarrow S_a^{\mathcal{D}}, X_a^{\mathcal{D}}, S_b^{\mathcal{D}}, X_b^{\mathcal{D}}$
while $|\frac{S_a}{X_a} - \frac{S_b}{X_b}| \leq \epsilon$ **do**
  $\alpha_1 \Leftarrow \frac{S_a}{(X_a-1).X_a}$
  $\beta_1 \Leftarrow \mathbb{1}(S_a < X_a \wedge X_a > 1).\alpha_1$
  $\alpha_2 \Leftarrow \frac{X_b-S_b}{(X_b-1).X_b}$
  $\beta_2 \Leftarrow \mathbb{1}(S_b > 0 \wedge X_b > 1).\alpha_2$
  $\alpha_3 \Leftarrow \frac{2.S_b}{X_b} - \frac{2.S_a.X_a-X_a-S_a}{X_a.(X_a-1)}$
  $\beta_3 \Leftarrow \mathbb{1}(S_a > 0 \wedge X_a > 1 \wedge \frac{S_a-1}{X_a-1} < \frac{S_b}{X_b}).\alpha_3$
  $\alpha_4 \Leftarrow \frac{2.S_b.X_b-S_b}{X_b.(X_b-1)} - \frac{2.S_a}{X_a}$
  $\beta_4 \Leftarrow \mathbb{1}(S_b < X_b \wedge X_b > 1 \wedge \frac{S_b}{X_b-1} > \frac{S_a}{X_a}).\alpha_4$
  switch $\max(\beta_1, \beta_2, \beta_3, \beta_4)$ **do**
    case β_1
      $X_a \Leftarrow X_a - 1$
    case β_2
      $S_b, X_b \Leftarrow S_b - 1, X_b - 1$
    case β_3
      $S_a, X_a \Leftarrow S_a - 1, X_a - 1$
    case β_4
      $X_b \Leftarrow X_b - 1$
    case 0 ▷ No operation can be done anymore
      return 1
end while
return $\frac{|\mathcal{D}|-S_a-S_b-X_a-X_b}{|\mathcal{D}|}$

Appendix C Integration of our Methods within FairCORELS: Experimental Results

This appendix section contains the experimental results of the integration of our exact and heuristic methods within the FairCORELS algorithm.

C.1 Exact Method: Unfairness Generalization Improvements Results

This subsection contains experimental results (training, test and validation unfairness functions of the number of steps performed by our method) for the integration of our exact method with FairCORELS. Results for the Statistical Parity metric are presented in Section 5.1.3. Results for the remaining metrics, for $\epsilon = 0.01$, are detailed here.

Figures 12, 13 and 14 summarize the experimental results using the Predictive Equality, Equal Opportunity, and Equalized Odds metrics (respectively).

C.2 Exact Method: Sample Robustness and Accuracy Evolution

This subsection contains experimental results (fairness sample-robustness along with training and test accuracy, functions of the number of steps performed by our method) for the integration of our exact method with `FairCORELS`. Results for the Statistical Parity metric are presented in Section 5.1.3. Results for the remaining metrics, for $\epsilon = 0.01$, are detailed here.

Figures 15, 16 and 17 summarize the experimental results using the Predictive Equality, Equal Opportunity, and Equalized Odds metrics (respectively).

C.3 Exact Method: Detailed Results

Table 3 summarizes all results obtained (and presented in Section 5.1.3) using our exact method within `FairCORELS`, for the Statistical Parity metric. It shows that accuracy is rather stable between the different folds, and its variation is not significantly affected by our method. Considering all datasets, the unfairness variation that exists between the different folds is not significantly affected by our method, and when the enforced fairness sample-robustness is strong enough (*e.g.,* with the `no validation (before-constant)` method) it tends to be reduced. Note that this variation can be explained by the fact that, for the `validation (`ϵ `criterion)` and `validation (train unf. criterion)` methods, the models are built performing different numbers of steps (*i.e.,* enforcing different fairness sample-robustness levels) - depending on the unfairness measured on a different validation set. Then, because different fairness sample-robustness levels are needed to reach the stopping criterion on the validation sets, unfairness measures are reduced for all folds, but in different magnitudes. As mentioned earlier, the fact that the number of performed steps is not known in advance and may vary is one of the drawbacks of our exact method (motivating our heuristic formulation), especially when dealing with small datasets such as COMPAS.

C.4 Heuristic Method: Fairness Sample-Robustness Audit Results

This appendix section contains experimental results for the integration of our heuristic method with `FairCORELS`. Sample-Robustness audit performed on the built models using $\mathcal{IPSR}(h, \mathcal{D}, \epsilon)$ and $\mathcal{GreedySR}(h, \mathcal{D}, \epsilon)$ are presented in Section 5.1.4 for the Default of Credit Card Clients dataset. Results for the Adult Income, COMPAS, and Bank Marketing datasets are presented in Figures 18, 19, and 20 (respectively).

C.5 Heuristic Method: Performances Results

This appendix section contains experimental results for the integration of our heuristic method with `FairCORELS`. Performances results for the equal opportunity metric are presented in Section 5.1.5. Results for the remaining metrics are detailed here.

Figures 21, 22, 23, 24 summarize the experimental results using the Statistical Parity metric. Figures 25, 26, 27, 28 summarize the experimental results using the Predictive Equality metric. Figures 29, 30, 31, 32 summarize the experimental results using the Equalized Odds metric. Note that for all these Figures, the ranges of both axis of the first two plots are set identically in order to ease train/test comparison.

C.6 Comparison between the Exact and Heuristic Methods: Fairness Sample-Robustness

This appendix section contains the experimental comparison of the fairness sample-robustness of our exact and heuristic methods with `FairCORELS`. Results for the Statistical Parity metric are presented in Section 5.1.6. Results for the remaining metrics, for $\epsilon = 0.01$, are detailed here. Figures 33, 34 and 35 summarize the experimental results using the Predictive Equality, Equal Opportunity, and Equalized Odds metrics (respectively).

C.7 Comparison between the Exact and Heuristic Methods: Performances

This appendix section contains the experimental comparison of the performances (test error/test unfairness tradeoffs) of our exact and heuristic methods with `FairCORELS`. Results for the statistical parity metric are presented in Section 5.1.6. Results for the remaining metrics, for $\epsilon = 0.01$, are detailed here. Figures 36, 37 and 38 summarize the experimental results using the Predictive Equality, Equal Opportunity, and Equalized Odds metrics (respectively).

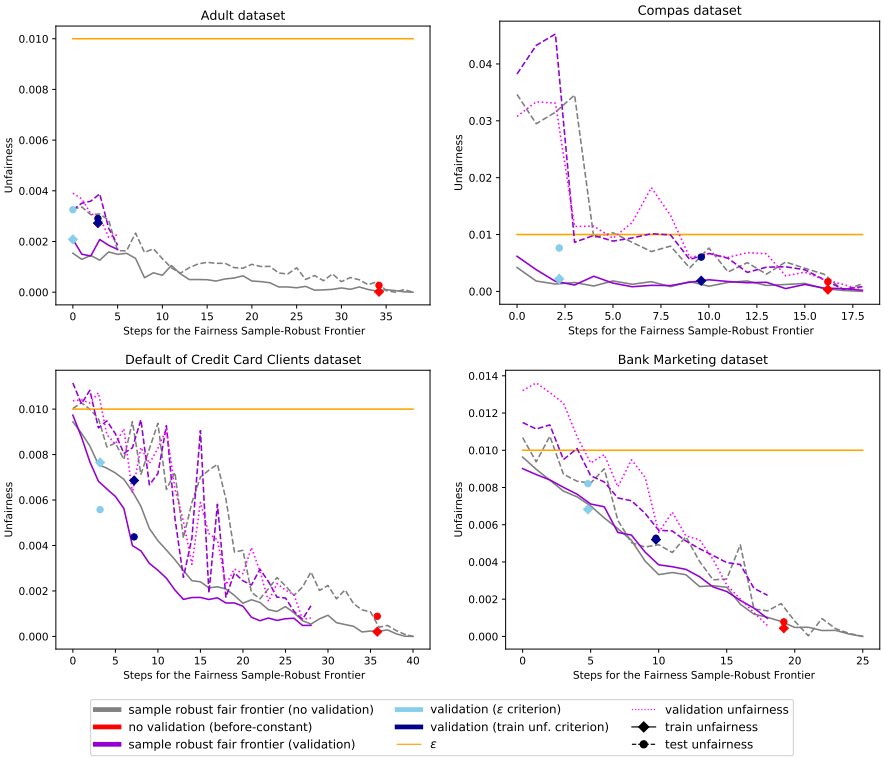

Fig. 12 Training, test and validation (when applicable) unfairness of models generated by `FairCORELS` through the iterations of our exact method (Predictive Equality metric, $\epsilon = 0.01$)

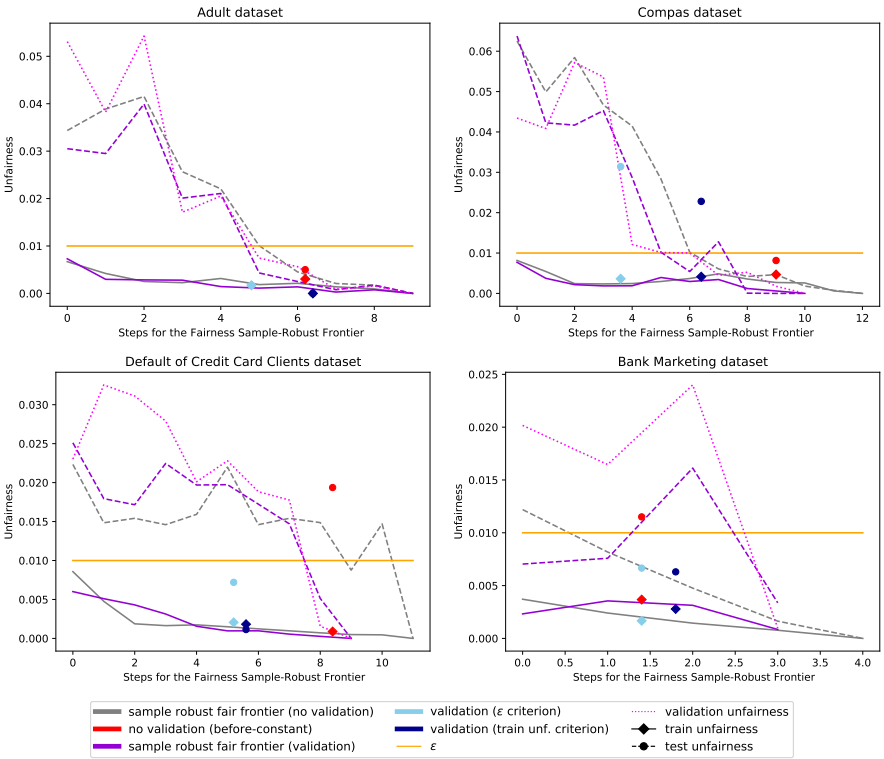

Fig. 13 Training, test and validation (when applicable) unfairness of models generated by `FairCORELS` through the iterations of our exact method (Equal Opportunity metric, $\epsilon = 0.01$)

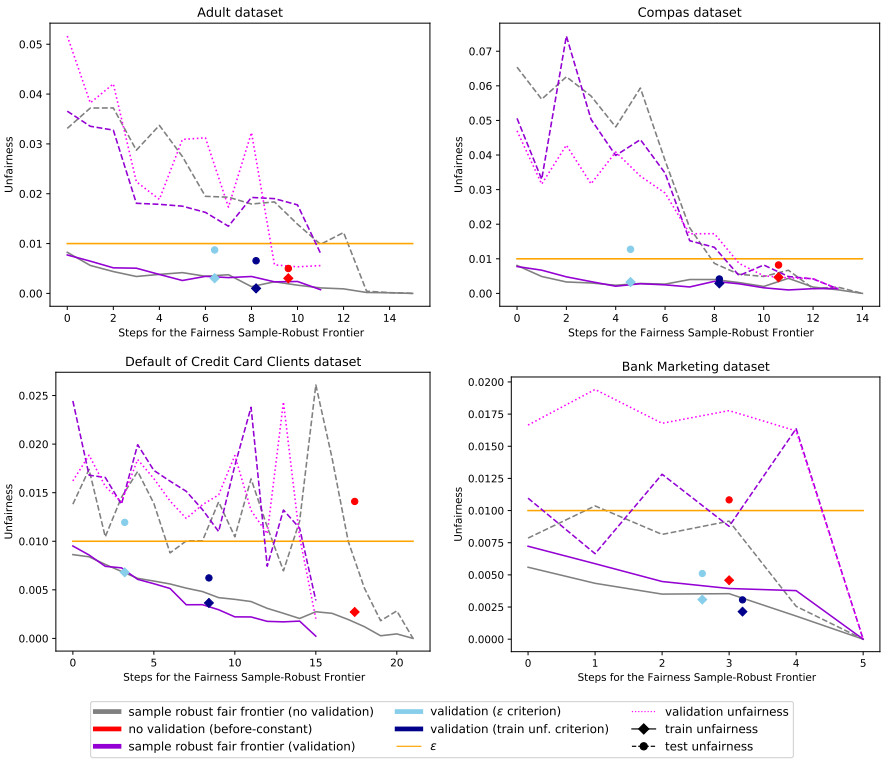

Fig. 14 Training, test and validation (when applicable) unfairness of models generated by `FairCORELS` through the iterations of our exact method (Equalized Odds metric, $\epsilon = 0.01$)

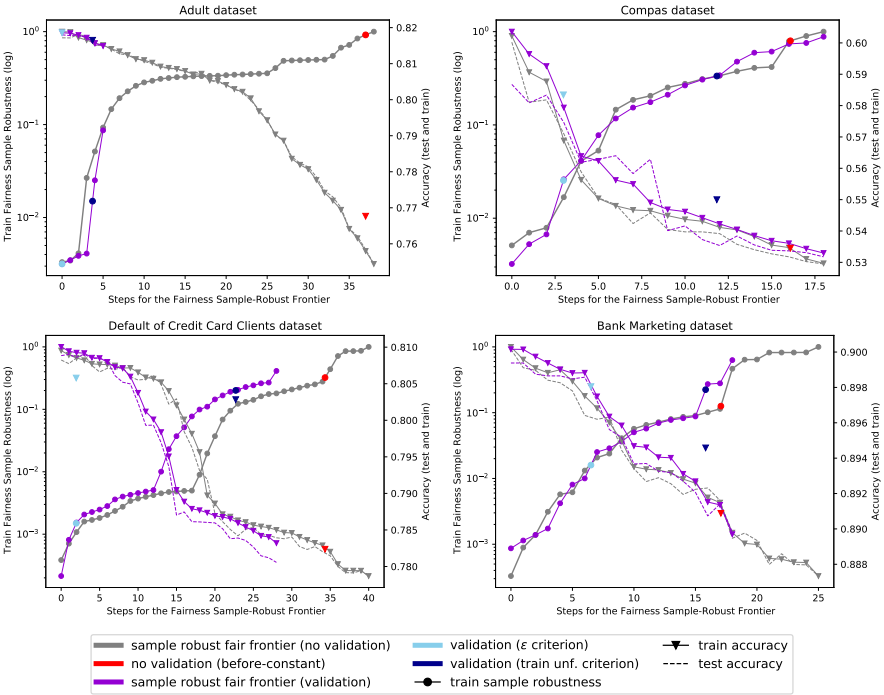

Fig. 15 Fairness sample-robustness and accuracy of models generated by `FairCORELS` through the iterations of our exact method (Predictive Equality metric, $\epsilon = 0.01$)

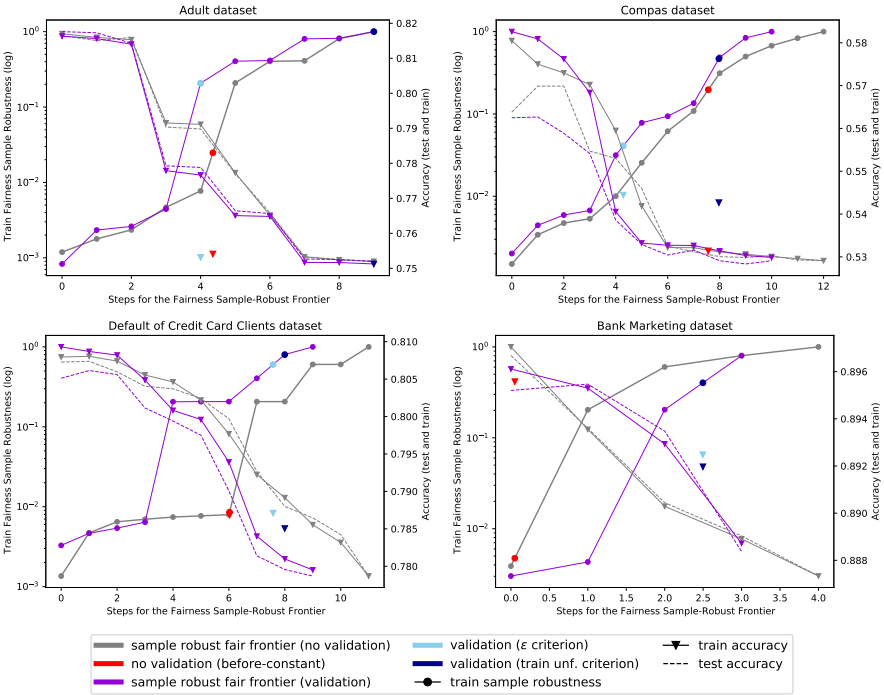

Fig. 16 Fairness sample-robustness and accuracy of models generated by `FairCORELS` through the iterations of our exact method (Equal Opportunity metric, $\epsilon = 0.01$)

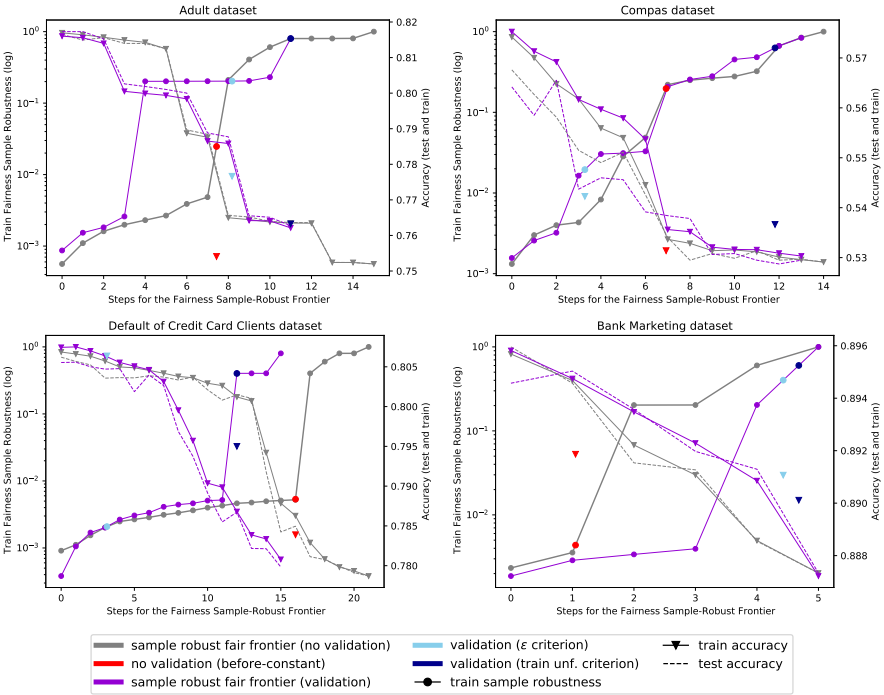

Fig. 17 Fairness sample-robustness and accuracy of models generated by `FairCORELS` through the iterations of our exact method (Equalized Odds metric, $\epsilon = 0.01$)

Table 3 Summary of the experimental results using our exact sample-robustness method within `FairCORELS`, for the Statistical Parity metric. We report training accuracy (Train Acc.), test accuracy (Test Acc.), training unfairness (Train Unf.) and test unfairness (Test Unf.). For each measure, we report its average value and standard deviation. The results are reported for the different methods introduced in Section 5.1.2: the original `FairCORELS` (original), `validation (ϵ criterion)` (val. (ϵ)), `validation (train unf. criterion)` (val. (train. unf.)) and `no validation (before-constant)` (before-constant).

Method	Train Acc.	Test Acc.	Train Unf.	Test Unf.
Adult Income				
original	.7822 ± .0037	.7819 ± .0073	.0090 ± .0011	.0127 ± .0087
val. (ϵ)	.7723 ± .0060	.7707 ± .0077	.0049 ± .0033	.0060 ± .0055
val. (train. unf.)	.7685 ± .0056	.7676 ± .0054	.0051 ± .0029	.0056 ± .0035
before-constant	.7549 ± .0008	.7546 ± .0037	.0047 ± .0004	.0043 ± .0012
COMPAS				
original	.5704 ± .0038	.5686 ± .0125	.0085 ± .0011	.0336 ± .0130
val. (ϵ)	.5512 ± .0190	.5420 ± .0266	.0055 ± .0028	.0163 ± .0214
val. (train. unf.)	.5497 ± .0197	.5412 ± .0281	.0048 ± .0034	.0145 ± .0224
before-constant	.5316 ± .0045	.5319 ± .0184	.0031 ± .0003	.0031 ± .0016
Default of Credit Card Clients				
original	.8015 ± .0024	.8010 ± .0016	.0098 ± .0002	.0137 ± .0040
val. (ϵ)	.7975 ± .0028	.7977 ± .0042	.0051 ± .0030	.0103 ± .0034
val. (train. unf.)	.7972 ± .0031	.7964 ± .0053	.0050 ± .0029	.0084 ± .0056
before-constant	.7818 ± .0010	.7807 ± .0044	.0007 ± .0006	.0017 ± .0012
Bank Marketing				
original	.8936 ± .0009	.8932 ± .0036	.0095 ± .0003	.0095 ± .0055
val. (ϵ)	.8930 ± .0006	.8935 ± .0031	.0061 ± .0026	.0082 ± .0072
val. (train. unf.)	.8925 ± .0004	.8922 ± .0042	.0054 ± .0024	.0053 ± .0054
before-constant	.8909 ± .0008	.8907 ± .0033	.0023 ± .0010	.0041 ± .0021

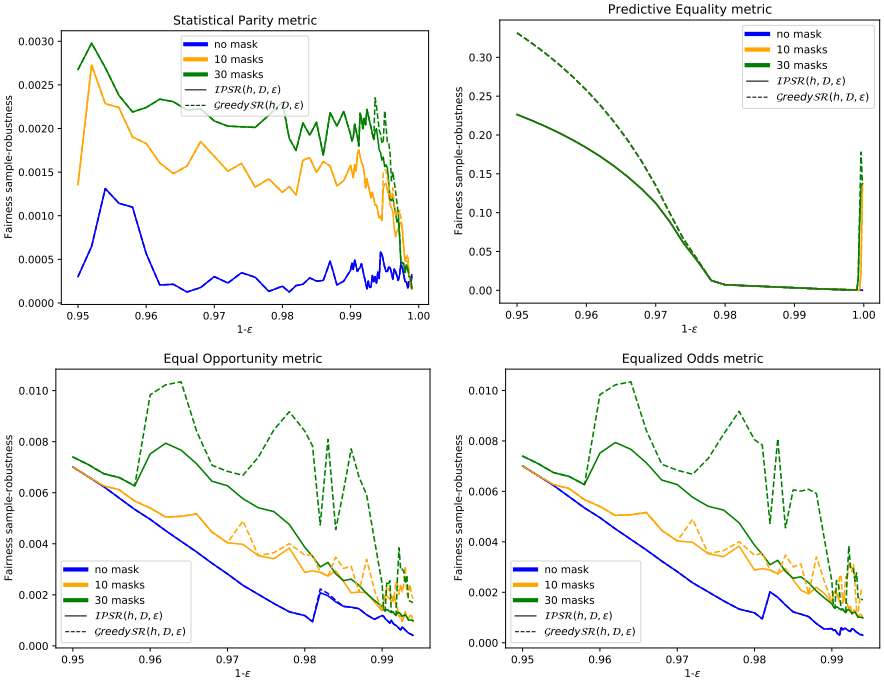

Fig. 18 Fairness sample-robustness of models generated by `FairCORELS` using our heuristic method (Adult Income dataset)

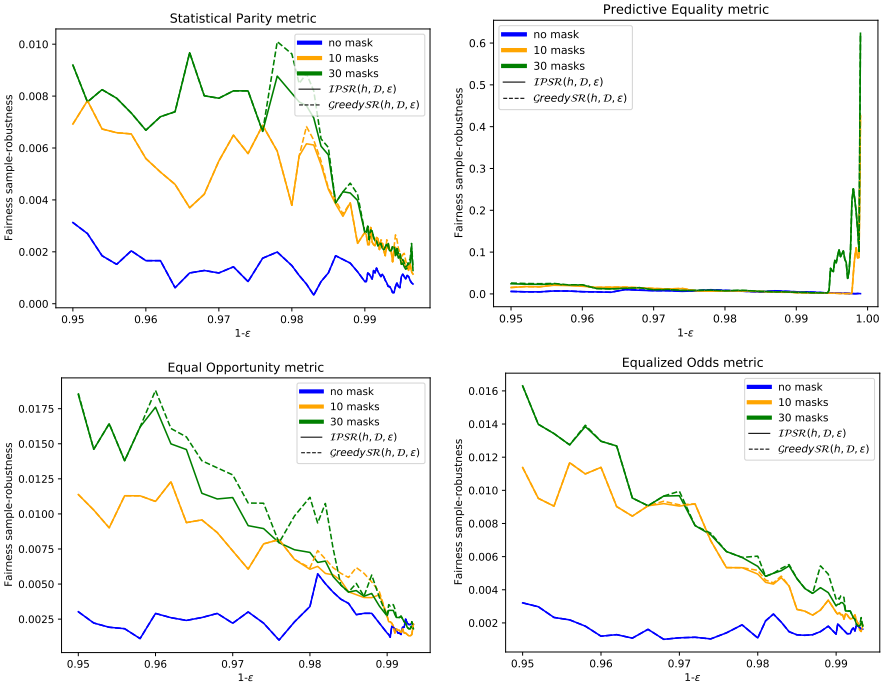

Fig. 19 Fairness sample-robustness of models generated by `FairCORELS` using our heuristic method (COMPAS dataset)

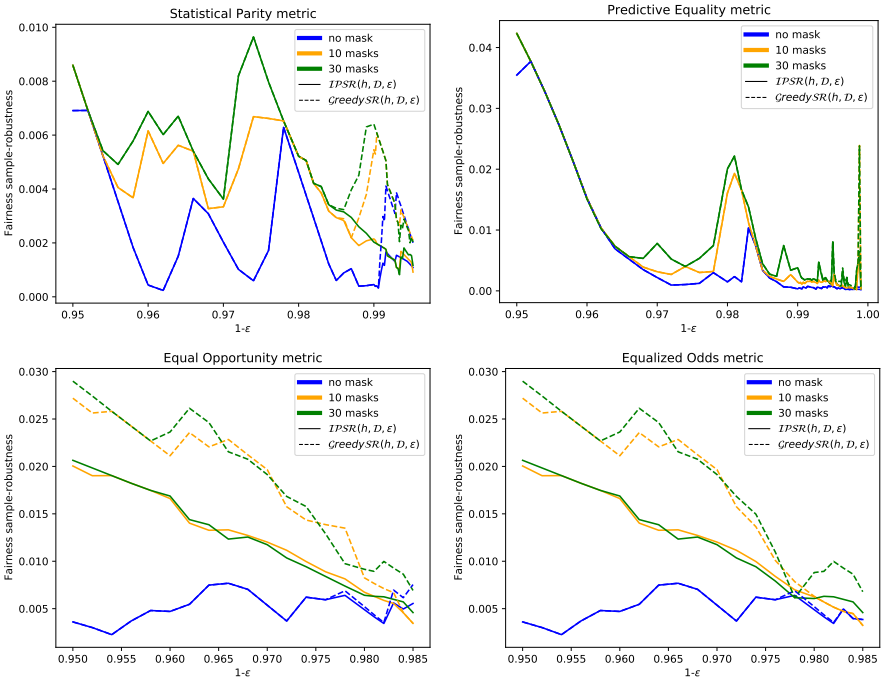

Fig. 20 Fairness sample-robustness of models generated by `FairCORELS` using our heuristic method (Bank Marketing dataset)

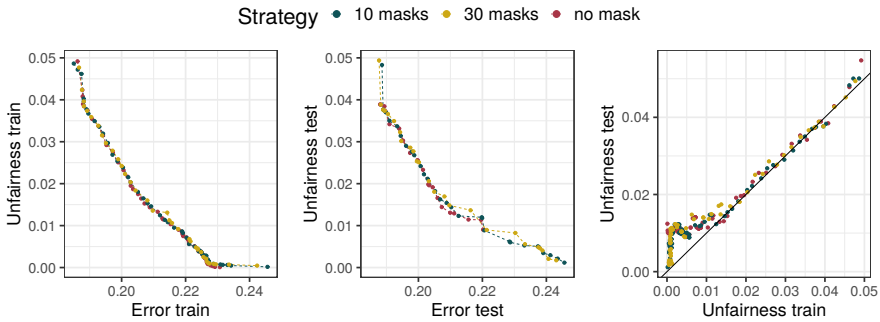

Fig. 21 Results of our heuristic method (Adult Income dataset, Statistical Parity metric).

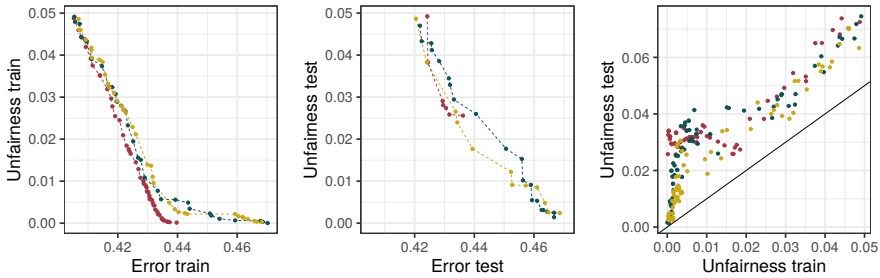

Fig. 22 Results of our heuristic method (COMPAS dataset, Statistical Parity metric).

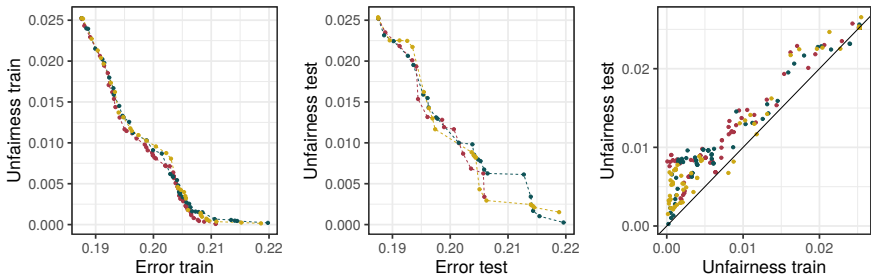

Fig. 23 Results of our heuristic method (Default Credit dataset, Statistical Parity metric).

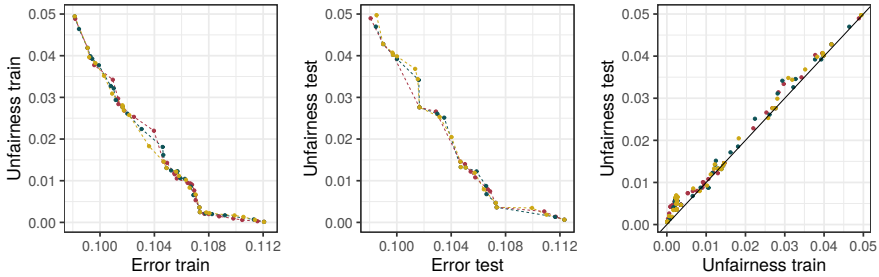

Fig. 24 Results of our heuristic method (Marketing dataset, Statistical Parity metric).

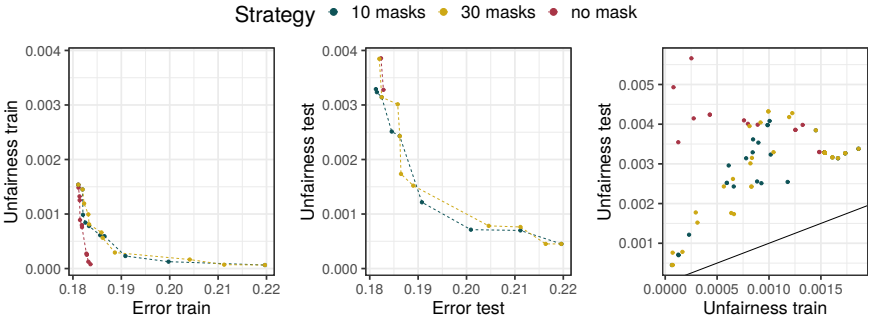

Fig. 25 Results of our heuristic method (Adult Income dataset, Predictive Equality metric).

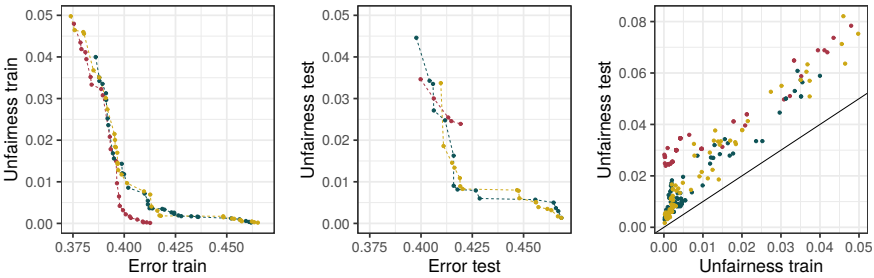

Fig. 26 Results of our heuristic method (COMPAS dataset, Predictive Equality metric).

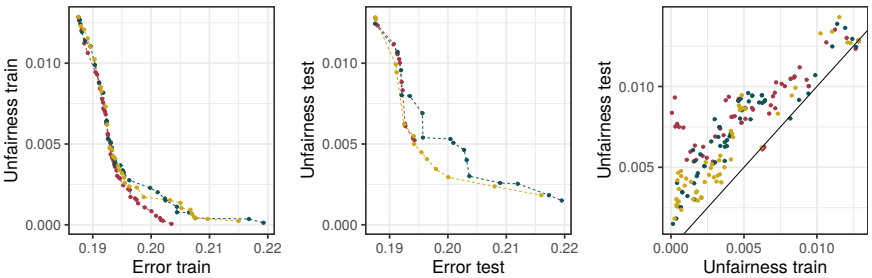

Fig. 27 Results of our heuristic method (Default Credit dataset, Predictive Equality metric).

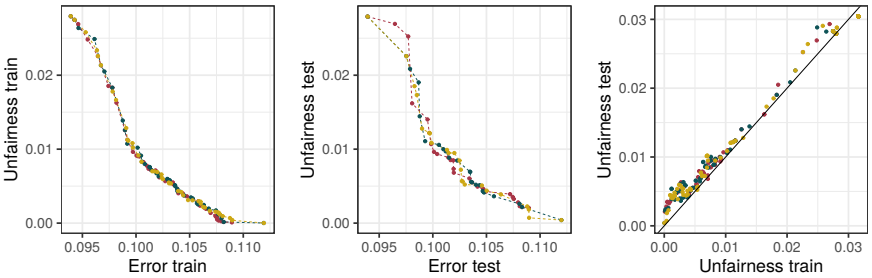

Fig. 28 Results of our heuristic method (Marketing dataset, Predictive Equality metric).

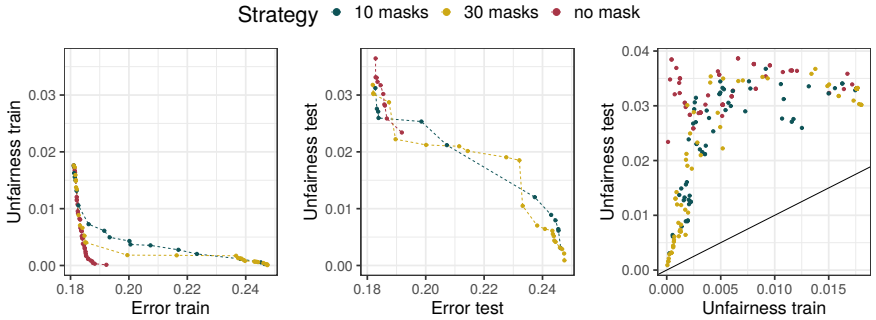

Fig. 29 Results of our heuristic method (Adult Income dataset, Equalized Odds metric).

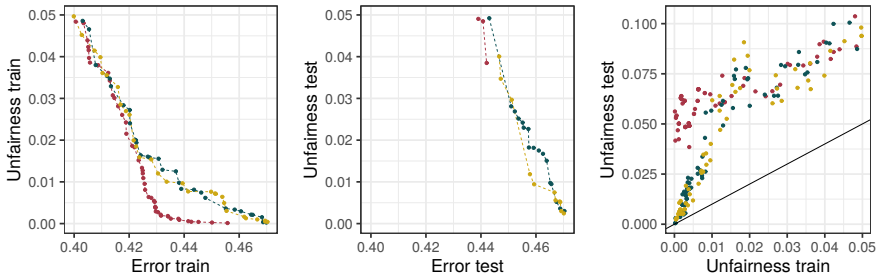

Fig. 30 Results of our heuristic method (COMPAS dataset, Equalized Odds metric).

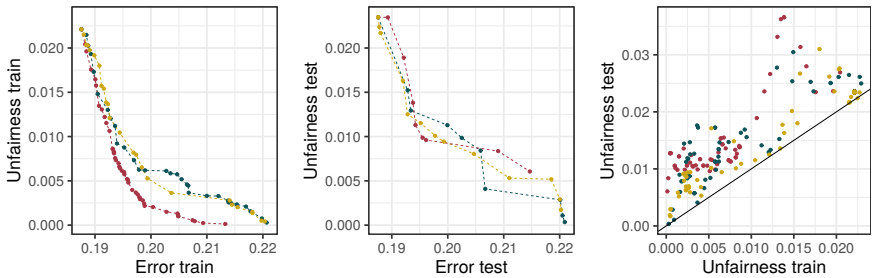

Fig. 31 Results of our heuristic method (Default Credit dataset, Equalized Odds metric).

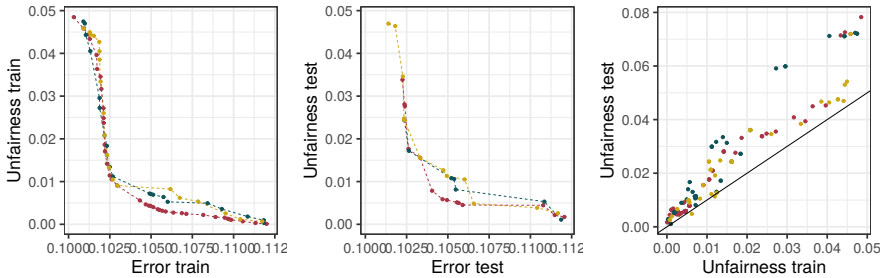

Fig. 32 Results of our heuristic method (Marketing dataset, Equalized Odds metric).

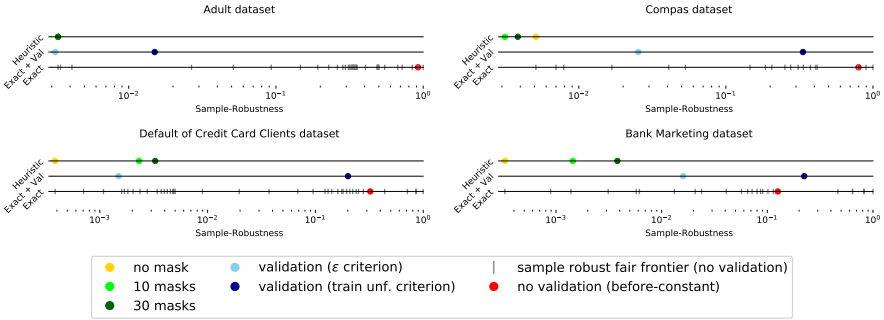

Fig. 33 Fairness sample-robustness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Predictive Equality metric, $\epsilon = 0.01$)

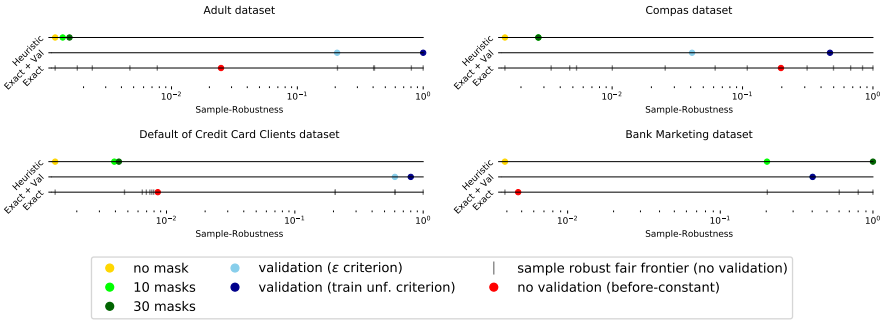

Fig. 34 Fairness sample-robustness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Equal Opportunity metric, $\epsilon = 0.01$)

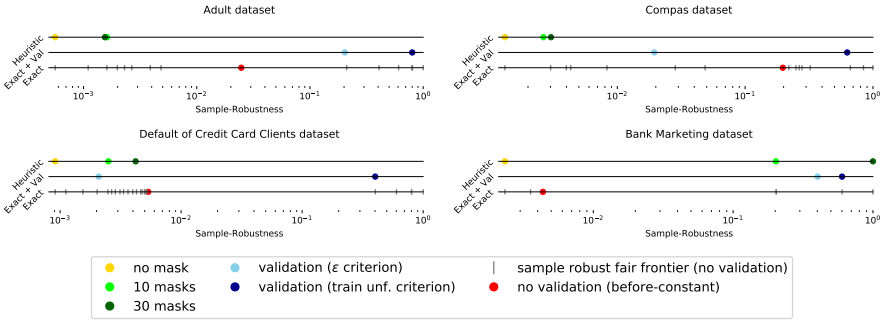

Fig. 35 Fairness sample-robustness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Equalized Odds metric, $\epsilon = 0.01$)

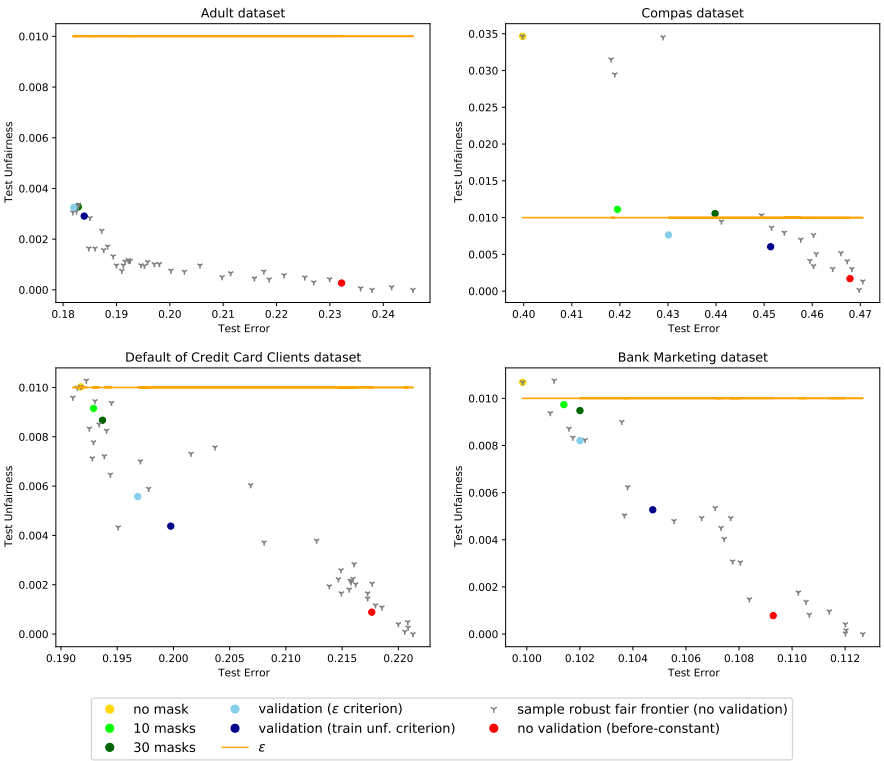

Fig. 36 Test error and unfairness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Predictive Equality metric, $\epsilon = 0.01$)

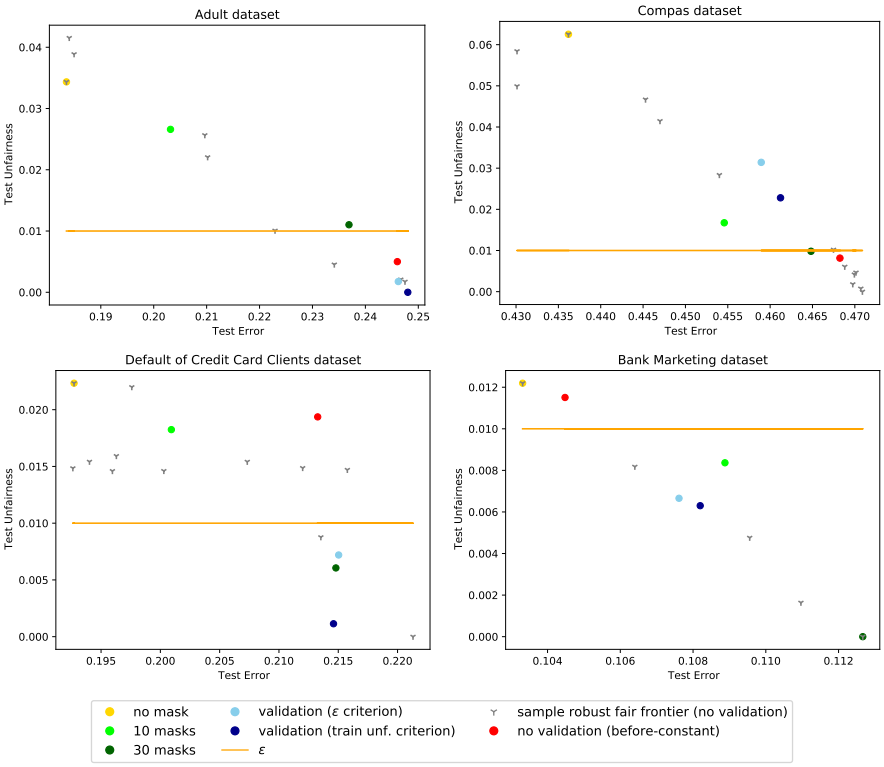

Fig. 37 Test error and unfairness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Equal Opportunity metric, $\epsilon = 0.01$)

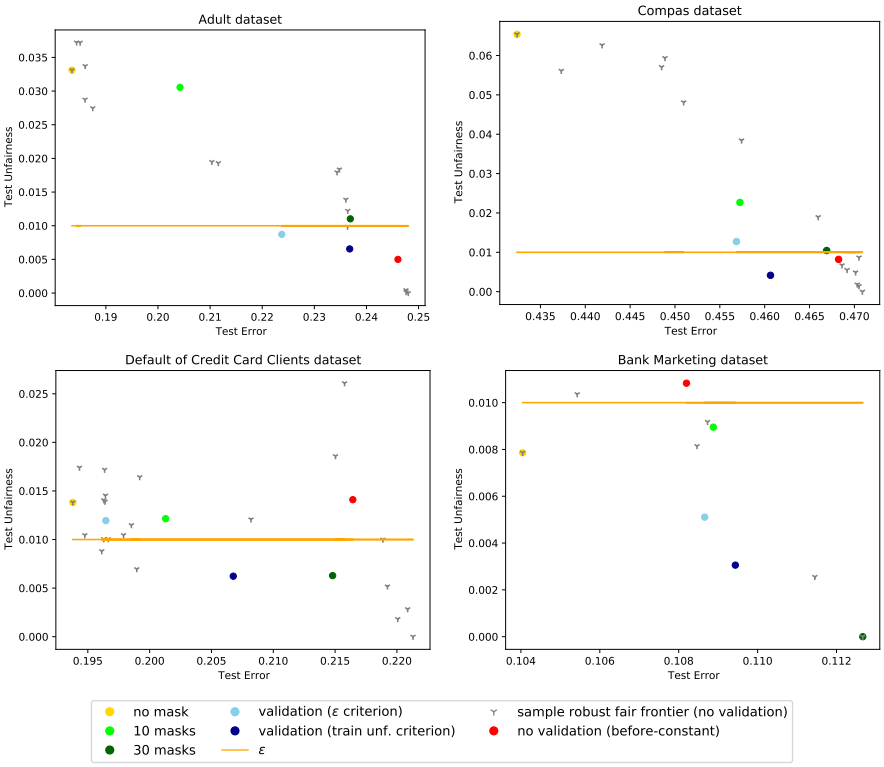

Fig. 38 Test error and unfairness of models generated by `FairCORELS` using our exact and heuristic sample-robust fair methods (Equalized Odds metric, $\epsilon = 0.01$)

Declarations

- Funding - This work is partially supported by the Canada Research Chairs (Privacy-preserving and ethical analysis of Big Data chair) and by the LabEx CIMI (ANR-11-LABX-0040).
- Conflict of interest/Competing interests - All the authors declared that they have no conflict of interest.
- Ethics approval - Not applicable.
- Consent to participate - Not applicable.
- Consent for publication - Not applicable.
- Availability of data and materials - All datasets used in the experiments are publicly available (links are provided within the paper).
- Code availability - Our source code will be publicly released upon acceptance, along with detailed files and instructions to reproduce our results.
- Authors' contributions - All authors contributed equally to this work.

References

Agarwal A, Beygelzimer A, Dudik M, et al (2018) A reductions approach to fair classification. In: *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, vol 80. PMLR, pp 60–69, URL https://proceedings.mlr.press/v80/agarwal18a.html

Aïvodji U, Ferry J, Gambs S, et al (2019) Learning fair rule lists. *arXiv preprint arXiv:190903977*

Aïvodji U, Ferry J, Gambs S, et al (2021) Faircorels, an open-source library for learning fair rule lists. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, New York, NY, USA, CIKM '21, p 4665–4669, https://doi.org/10.1145/3459637.3481965

Angelino E, Larus-Stone N, Alabi D, et al (2017) Learning certifiably optimal rule lists. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, KDD '17, p 35–44, https://doi.org/10.1145/3097983.3098047

Angelino E, Larus-Stone N, Alabi D, et al (2018) Learning certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research* 18(234):1–78. URL http://jmlr.org/papers/v18/17-716.html

Angwin J, Larson J, Mattu S, et al (2016) Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. propublica (2016). *ProPublica, May* 23

Barocas S, Hardt M, Narayanan A (2019) *Fairness and Machine Learning*. fairmlbook.org, http://www.fairmlbook.org

Ben-Tal A, Den Hertog D, De Waegenaere A, et al (2013) Robust solutions of optimization problems affected by uncertain probabilities. *Management Science* 59(2):341–357. https://doi.org/10.1287/mnsc.1120.1641

Caton S, Haas C (2020) Fairness in machine learning: A survey. *arXiv preprint arXiv:201004053*

Chouldechova A (2017) Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5(2):153–163. https://doi.org/10.1089/big.2016.0047

Chuang CY, Mroueh Y (2021) Fair mixup: Fairness via interpolation. In: *9th International Conference on Learning Representations, ICLR*, URL https://openreview.net/forum?id=DNl5s5BXeBn

Cotter A, Gupta M, Jiang H, et al (2018) Training fairness-constrained classifiers to generalize. *FATML*

Cotter A, Gupta M, Jiang H, et al (2019a) Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, vol 97. PMLR, pp 1397–1405, URL http://proceedings.mlr.press/v97/cotter19b.html

Cotter A, Jiang H, Sridharan K (2019b) Two-player games for efficient non-convex constrained optimization. In: *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, Illinois, USA*, Proceedings of Machine Learning Research, vol 98. PMLR, pp 300–332, URL http://proceedings.mlr.press/v98/cotter19a.html

Cummings R, Gupta V, Kimpara D, et al (2019) On the compatibility of privacy and fairness. In: *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*. Association for Computing Machinery, New York, NY, USA, UMAP'19 Adjunct, p 309–315, https://doi.org/10.1145/3314183.3323847

Du W, Wu X (2021) Fair and robust classification under sample selection bias. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, New York, NY, USA, CIKM '21, p 2999–3003, https://doi.org/10.1145/3459637.3482104

Duchi JC, Hashimoto T, Namkoong H (2020) Distributionally robust losses for latent covariate mixtures. *arXiv preprint arXiv:200713982*

Duchi JC, Glynn PW, Namkoong H (2021) Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research* 46(3):946–969. https://doi.org/10.1287/moor.2020.1085

Dwork C, Roth A (2014) The algorithmic foundations of differential privacy. *Found Trends Theor Comput Sci* 9(3–4):211–407. https://doi.org/10.1561/0400000042

Dwork C, Hardt M, Pitassi T, et al (2012) Fairness through awareness. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. Association for Computing Machinery, New York, NY, USA, ITCS '12, p 214–226, https://doi.org/10.1145/2090236.2090255

Frank A, Asuncion A (2010) UCI machine learning repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California. *School of information and computer science* 213:2–2

Freitas AA (2014) Comprehensible classification models: A position paper. *SIGKDD Explor Newsl* 15(1):1–10. https://doi.org/10.1145/2594473.2594475

Hardt M, Price E, Price E, et al (2016) Equality of opportunity in supervised learning. In: *Advances in Neural Information Processing Systems*, vol 29. Curran Associates, Inc., URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf

Huang L, Vishnoi NK (2019) Stable and fair classification. In: Chaudhuri K, Salakhutdinov R (eds) *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, vol 97. PMLR, pp 2879–2890, URL http://proceedings.mlr.press/v97/huang19e.html

Iofinova E, Konstantinov N, Lampert CH (2021) Flea: Provably fair multisource learning from unreliable training data. *arXiv preprint arXiv:210611732*

Kamiran F, Calders T (2012) Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems* 33(1):1–33. https://doi.org/10.1007/s10115-011-0463-8

Kang Y (2017) Distributionally robust optimization and its applications in machine learning. PhD thesis, Columbia University

Khoshgoftaar TM, Fazelpour A, Wang H, et al (2013) A survey of stability analysis of feature subset selection techniques. In: *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*, pp 424–431, https://doi.org/10.1109/IRI.2013.6642502

Kosub S (2019) A note on the triangle inequality for the jaccard distance. *Pattern Recognition Letters* 120:36–38. https://doi.org/10.1016/j.patrec.2018.12.007

Liu EZ, Haghgoo B, Chen AS, et al (2021) Just train twice: Improving group robustness without training group information. In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, vol 139. PMLR, pp 6781–6792, URL http://proceedings.mlr.press/v139/liu21f.html

Mandal D, Deng S, Jana S, et al (2020) Ensuring fairness beyond the training data. In: *Advances in Neural Information Processing Systems*, vol 33. Curran Associates, Inc., pp 18,445–18,456, URL https://proceedings.neurips.cc//paper/2020/file/d6539d3b57159babf6a72e106beb45bd-Paper.pdf

Moro S, Cortez P, Rita P (2014) A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62:22–31. https://doi.org/10.1016/j.dss.2014.03.001

Nam J, Cha H, Ahn S, et al (2020) Learning from failure: Debiasing classifier from biased classifier. In: *Advances in Neural Information Processing Systems*, vol 33. Curran Associates, Inc., pp 20,673–20,684, URL https://proceedings.neurips.cc/paper/2020/file/eddc3427c5d77843c2253f1e799fe933-Paper.pdf

Perron L, Furnon V (2019) Or-tools. URL https://developers.google.com/optimization/

Rahimian H, Mehrotra S (2019) Distributionally robust optimization: A review. *arXiv preprint arXiv:190805659*

Rezaei A, Fathony R, Memarrast O, et al (2020) Fairness for robust log loss classification. In: , pp 5511–5518, https://doi.org/10.1609/aaai.v34i04.6002

Rivest RL (1987) Learning decision lists. *Machine learning* 2(3):229–246. https://doi.org/10.1007/BF00058680

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5):206–215. https://doi.org/10.1038/s42256-019-0048-x

Saeys Y, Abeel T, Van de Peer Y (2008) Robust feature selection using ensemble feature selection techniques. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, pp 313–325, https://doi.org/10.1007/978-3-540-87481-2_21

Sagawa S, Koh PW, Hashimoto TB, et al (2020) Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, URL https://openreview.net/forum?id=ryxGuJrFvS

Slack D, Friedler SA, Givental E (2020) Fairness warnings and fair-maml: learning fairly with minimal data. In: Hildebrandt M, Castillo C, Celis LE, et al (eds) *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*. ACM, pp 200–209, https://doi.org/10.1145/3351095.3372839

Taskesen B, Nguyen VA, Kuhn D, et al (2020) A distributionally robust approach to fair classification. *arXiv preprint arXiv:200709530*

Tommasi T, Patricia N, Caputo B, et al (2017) A deeper look at dataset bias. In: *Domain Adaptation in Computer Vision Applications. Advances in Computer Vision and Pattern Recognition*. Springer, p 37–55, https://doi.org/10.1007/978-3-319-58347-1_2

Torralba A, Efros AA (2011) Unbiased look at dataset bias. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA, pp 1521–1528, https://doi.org/10.1109/CVPR.2011.5995347

Verma S, Rubin J (2018) Fairness definitions explained. In: *Proceedings of the International Workshop on Software Fairness*. Association for Computing Machinery, New York, NY, USA, FairWare '18, p 1–7, https://doi.org/10.1145/3194770.3194776

Wang Y, Nguyen VA, Hanasusanto GA (2021) Wasserstein robust support vector machines with fairness constraints. *arXiv preprint arXiv:210306828*

Yeh IC, hui Lien C (2009) The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications* 36(2, Part 1):2473–2480. https://doi.org/10.1016/j.eswa.2007.12.020

Yurochkin M, Bower A, Sun Y (2020) Training individually fair ml models with sensitive subspace robustness. In: *8th International Conference on Learning Representations, ICLR Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, URL https://openreview.net/forum?id=B1gdkxHFDH

Zafar MB, Valera I, Gomez Rodriguez M, et al (2017) Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In: *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering

Committee, Republic and Canton of Geneva, CHE, WWW '17, p 1171–1180, https://doi.org/10.1145/3038912.3052660

Zhou ZH (2012) *Ensemble Methods: Foundations and Algorithms*, 1st edn. Chapman & Hall/CRC, https://doi.org/10.1201/b12207

Zou Q, Zeng J, Cao L, et al (2016) A novel features ranking metric with application to scalable visual and bioinformatics data classification. *Neurocomputing* 173:346–354. https://doi.org/10.1016/j.neucom.2014.12.123