Optimizing and Tuning Fairness in Machine Learning: An Augmented Lagrangian Method with a Performance Budget

Michele Fontana[1,2] (✉), Francesca Naretto[1], and Anna Monreale[1]

[1] University of Pisa, Italy,
{michele.fontana@phd.unipi.it,francesca.naretto@unipi.it,anna.monreale@unipi.it}
[2] ISTI-CNR, Pisa, Italy

Abstract. Fairness in Machine Learning has become a concern, particularly if models are deployed in high-stakes decision-making. Most existing approaches aim to enforce fairness during training, but they face significant challenges for the scalability and the effectiveness of fairness enforcement. To address these limitations, we propose a method for training fair classifiers under multiple group and intersectional fairness constraints with high predictive performance. We combine an Augmented Lagrangian learning procedure with a tunable *performance budget*, which regulates the trade-off between fairness and utility. Experiments demonstrate that our method mitigates bias while scaling efficiently with increasing problem complexity. By adjusting the performance budget, we provide a flexible mechanism to balance fairness enforcement and predictive performance, offering a solution for real-world applications.

Keywords: Fairness · Machine Learning · Ethical AI.

1 Introduction

In recent years, Machine Learning (ML) models have been developed and widely applied across various domains without posing any particular attention on the model trustworthiness but only optimizing the model utility for the specific task to be addressed. However, with the advent of new EU AI legislation[3], there is now an increased emphasis on the legal and ethical requirements of ML models, including, but not limited to, fairness, privacy and transparency [22,21,20]. Fairness seeks to reduce biases in model predictions. Although achieving fairness and model utility simultaneously is ideal, practitioners often face challenges in balancing the two, as improving one aspect can often undermine the other. In the literature, to prevent the amplification of unfair behavior of ML models Multi-Objective Optimization approaches have been proposed considering that fair ML has the goal of simultaneously minimizing classification error while also optimizing for one or more fairness criteria [24]. Nevertheless, the state-of-the-art approaches present some limits related to their scalability and their ability to

[3] The AI Act

find an acceptable trade-off between fairness and model performance, especially when addressing intersectional fairness.

In this paper, we propose `FairLAB` (Fairness via Lagrangian Augmented and performance Budget), a method for learning a neural network (`NN`) model that balances predictive performance while satisfying a collection of fairness constraints, including intersectional ones. Our method integrates fairness directly into the learning process by ensuring that fairness violations are addressed dynamically while optimizing the predictive objective. It exploits a *performance budget*, which enables control over the performance-fairness trade-off. Intersectional fairness further increases the complexity of this task, as multiple sensitive attributes combine to form a large number of subgroups, making fairness enforcement computationally challenging. To overcome the scalability issue, `FairLAB` exploits a *divide-et-impera* strategy which splits the fairness problem in subtasks addressed by multiple learners. A wide experimentation on three datasets demonstrates that `FairLAB` successfully mitigates fairness requirements also in case of challenging settings while maintaining under control the model utility and outperforms the state-of-the-art methods.

2 Related Work

Group Fairness. Several studies have explored the group fairness problem, proposing different mitigation strategies: pre-processing, in-processing, and post-processing methods [3]. Among them we focus on the *in-processing* approaches, which incorporate fairness requirements directly into the model's training. A common tactic involves regularization schemes that penalize correlations between predicted outcomes and sensitive attributes, balancing fairness against predictive performance [14]. Constraint-based optimization methods similarly aim to enforce fairness while maintaining overall accuracy, by coupling a performance metric with fairness constraints [4]. Cotter et al. [5] propose a primal-dual Lagrangian approach that can incorporate multiple, potentially non-differentiable constraints. Lokhande et al. [17] exploit the Augmented Lagrangian Method (ALM), though their approach is limited to a single fairness constraint and one binary sensitive attribute. Agarwal et al. [1] introduce Exponentiated Gradient (`ExpGrad`), which reduces fair classification to a sequence of cost-sensitive subproblems. Another category is adversarial debiasing, where an adversarial network tries to predict sensitive attributes from the model's outputs. The main model is trained to defeat the adversary, thus mitigating bias [27,6]. Another research direction addresses fairness under a Multi-Objective Optimization (MOO) framework, which aims to jointly optimize multiple fairness metrics along with predictive performance. MOO approaches capture various trade-offs by constructing a *Pareto front* of equivalent models [7], either with gradient-based or evolutionary algorithms [28,26]. Among the gradient-based methods, Ruchte et al. [24] introduce `COSMOS`, an efficient algorithm to approximate the Pareto front in high-dimensional settings, reducing the computational costs often associated with naive MOO approaches.

Intersectional Fairness. These algorithms must address two key challenges: *data sparsity*, where certain demographic subgroups contain very few instances, and *computational complexity*, which grows exponentially with the number of protected attributes. As a result, standard fairness metrics may become computationally intractable [11]. Hence, alternative approaches have been developed, including *subgroup fairness* [15] and *differential fairness* [10], that handle numerous subgroups more efficiently. A prominent line of work adopts an *auditing* paradigm: an auditor identifies subgroups exhibiting high unfairness under a given metric, and a learner then reduces prediction error subject to fairness constraints [16]. For instance, in [15], it is proposed a zero-sum game that leverages a cost-sensitive classification oracle. Another approach combines and extends group fairness methods, to incorporate *differential fairness* in intersectional settings, employing a tailored loss function to balance fairness and accuracy, relying on the correlations between protected and unprotected features [19].

To the best of our knowledge, in the literature there are no methods that can handle the performance-fairness trade-off directly in the learning process, addressing simultaneously group and intersectional fairness efficiently.

3 Background

Fairness Fundamentals. Here we describe the group and intersectional fairness metrics used throughout our work. Consider a dataset with a sensitive attribute a taking values in $\{v_1, v_2, \ldots, v_n\}$ and a binary classifier $\hat{Y} \in \{0, 1\}$. The *Demographic Parity* (`DP`) [2] assesses whether the probability of receiving a positive prediction is independent of the sensitive attribute. Formally, the disparity in positive prediction rates across subgroups of the attribute a is quantified as $DP(\hat{Y}) = \max_{1 \leq i < j \leq n} \left| \Pr(\hat{Y} = 1 \mid a = v_i) - \Pr(\hat{Y} = 1 \mid a = v_j) \right|$. The *Equal Opportunity* metric [13] measures the maximum gap in true positive rates across subgroups, while *Predictive Equality* [13] measures the maximum gap in false positive rates. *Equalized Odds* (`EOD`) [13] enforces fairness simultaneously in both measures and is defined as the maximum between them. We say that a classifier $\hat{Y}$ is fair with respect to metric F if $F^a(\hat{Y}) \leq \tau$, where τ is a given threshold (often set to 0.2).

In practice, fairness concerns often involve not just single sensitive attributes (e.g., gender or race), but *intersections* of multiple attributes (e.g., race *and* gender). This is known as *intersectional fairness*, and it aims to protect subgroups that may be disadvantaged at the intersection of multiple identities [10]. To evaluate fairness in this setting, the same group fairness metrics (e.g., `DP`, `EOD`) are applied to the cross-product of sensitive attributes, treating each intersectional group (e.g., `Black Woman`, `White Man`, etc.) as a distinct subgroup. This allows for a more fine-grained assessment of disparities that may be hidden when attributes are considered in isolation. However, this also increases the number of groups to monitor, raising statistical and optimization challenges in both measurement and mitigation.

In the following, we use $F^a(m)$ to denote the value of fairness metric F evaluated on model m w.r.t. to attribute a.

Augmented Lagrangian Method (ALM) [9] is a constrained optimization technique that extends the Lagrangian formulation. Consider a problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \quad \text{subject to} \quad g_i(\mathbf{x}) \leq 0\,(i=1,\ldots,r), \quad h_j(\mathbf{x}) = 0\,(j=1,\ldots,m)$$

ALM introduces a penalty parameter $\sigma > 0$ and Lagrange multipliers $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^r$ to enforce equality and inequality constraints, respectively. The Augmented Lagrangian $\mathcal{L}_A(\mathbf{x}, \lambda, \mu, \sigma)$ is defined as:

$$f(\mathbf{x}) \;+\; \sum_{j=1}^{m} \lambda_j\, h_j(\mathbf{x}) \;+\; \frac{\sigma}{2} \sum_{j=1}^{m} h_j(\mathbf{x})^2 + \frac{1}{2\sigma} \sum_{i=1}^{r} \Big(\max\{0,\, \mu_i + \sigma\, g_i(\mathbf{x})\}^2 - \mu_i^2 \Big)$$

At each iteration e, the variable $\mathbf{x}$ is updated by minimizing $\mathcal{L}_A$, followed by multiplier updates that correct constraint violations. The multipliers for equality and inequality constraints are adjusted as

$$\lambda^{(e+1)} = \lambda^{(e)} + \rho\, h\big(\mathbf{x}^{(e+1)}\big), \quad \mu^{(e+1)} = \max\Big\{0,\, \mu^{(e)} + \rho\, g\big(\mathbf{x}^{(e+1)}\big)\Big\} \tag{1}$$

When applied to deep learning, $\mathcal{L}_A$ is treated as a loss function that accounts for both predictive objectives and constraints satisfaction. For classification tasks, the function to be minimized is usually the Cross Entropy. The model parameters are updated through gradient-based methods. The multipliers are refreshed via Eq. 1 after each epoch, reducing constraint violations as training proceeds [18].

4 `FairLAB` method

Our objective is to train a neural network that balances predictive performance and fairness while satisfying a collection of fairness constraints, even intersectional ones. Our method integrates fairness directly into the learning process by ensuring that fairness violations are addressed dynamically while optimizing the predictive objective. We exploit a *performance budget* that controls the performance-fairness trade-off. Intersectional fairness increases the computational complexity, as multiple sensitive attributes create a large number of subgroups. To mitigate this issue, we adopt a *divide-et-impera* strategy, wherein a central orchestrator divides the fairness problem in sub-tasks among multiple learners by partitioning the fairness constraints, reducing the need to handle all intersectional subgroups simultaneously and improving scalability. Fig. 1 provides a schematic representation of the process, which consists of two main phases: a *setup phase*, where constraints are assigned to the learners, and a *global learning phase*, where fairness violations are iteratively identified and mitigated.

Preliminaries. Before detailing the algorithmic process, we describe its inputs and we define the key concepts. The algorithm takes as input (i) a set of fairness requirements $\mathcal{R}$, (ii) a performance metric P (e.g. accuracy or F_1 score)

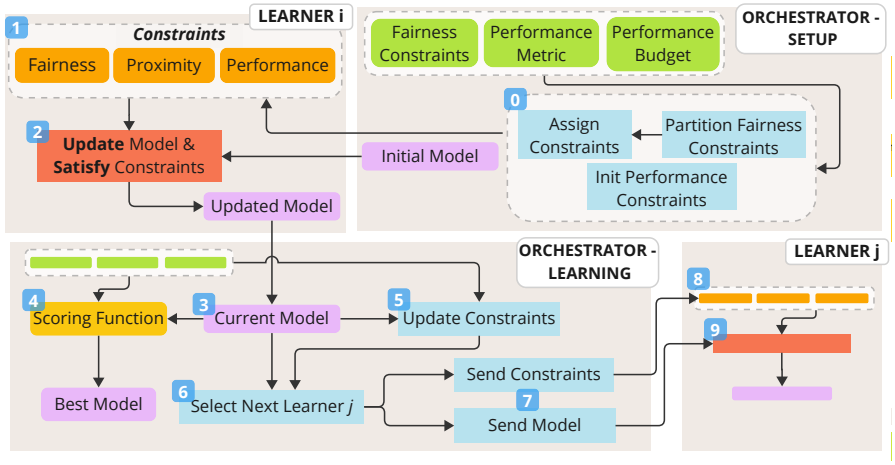

Fig. 1. Overview of `FairLAB`'s main steps in the setup and global learning phases.

Algorithm 1: $\texttt{FairLAB}_{\text{Orchestrator}}(w^1, E, P, \mathcal{R}, \beta)$

Input: w^1, E : the initial parameters, the number of iterations.
Input: P, $\mathcal{R}$: the performance metric to optimize, the fairness requirements.
Input: β : the performance budget.
Output: w_{best}: the last best parameters.

$\tilde{\mathcal{R}} = BinarizeConstraints(\mathcal{R})$;
$L = InitLocalLearners(\tilde{\mathcal{R}})$;
$w_{\text{best}} = w^1; s_{\text{best}} = EvaluateModel(w^1, \texttt{S}); p^* = P(w^1)$;
for $e \leftarrow 1$ **to** E **do**
  $\mathcal{P}^e, \mathcal{D}^e = BuildPerformance\&ProximityConstraints(w^e, \beta, p^*)$;
  $l^e = SelectLocalLearner(w^e, L)$;
  $w^{e+1}, p^* = UpdateModel_{Local}(l^e, w^e, \mathcal{P}^e, \mathcal{D}^e)$;
  $s^{e+1} = EvaluateModel(w^{e+1}, \texttt{S})$;
  if $s^{e+1} > s_{best}$ **then**
    $w_{\text{best}} = w^{e+1}$; $s_{\text{best}} = s^{e+1}$
return w_{best}

to be maximized and (iii) a performance budget $\beta \geq 0$. The fairness requirement is defined as $\mathcal{R} = \{(F_i, a_i, \tau_i)\}_{i=1}^{|\mathcal{R}|}$, where F_i is a fairness metric (e.g. `DP` or `EOD`), $a_i \in \mathcal{A}$ is a sensitive attribute (either binary or non-binary) belonging to the set of sensitive attributes $\mathcal{A}$ involved in the fairness requirements $\mathcal{R}$, $\tau_i \geq 0$ is a threshold defining the maximum acceptable violation. Note that, in case of intersectional fairness an attribute a_i might be both a simple attribute e.g., race *or* gender, and a combined attribute, e.g., race *and* gender. Given a `ML` model m each constraint takes the form $F_i^{a_i}(m) \leq \tau_i$, and $F_i^{a_i}(m)$ denotes the evaluation of the fairness metric F_i on the attribute a_i for the `ML` model m. To quantify

constraint satisfaction, we define the *scoring function* S, expressed as in Eq. 2:

$$\mathtt{S}(m;\mathcal{R},P) \;=\; \lambda_P\, P(m) \;-\; (1-\lambda_P)\sum_{i=1}^{|\mathcal{R}|}\max\!\big\{0,\; F_i^{a_i}(m)-\tau_i\big\}\,\big(F_i^{a_i}(m)-\tau_i\big) \quad (2)$$

where $\lambda_P \in [0,1]$ determines the trade-off between performance and fairness. Since fairness constraints are often defined over categorical attributes that may have multiple possible values, we introduce a *binary partition* process that allows us to compare fairness metrics between pairs of attribute values. Given a sensitive attribute a with domain $\mathrm{dom}(a) = \{v_1, v_2, \ldots, v_n\}$, we define the binary partition on an attribute as an operator, that restricts the domain values to any pair $\{v_i, v_j\}$, denoted as $\pi(a \mid v_i, v_j)$. This binary partition process enables also a strategy of fairness constraint partition among multiple learners.

Method Description.

To provide a clear understanding of the main steps performed by `FairLAB`, we present its pseudo-code in Algorithm 1. The proposed approach operates in a *setup phase* (lines 1-3) followed by a *global learning phase*. During *setup*, the orchestrator converts the fairness requirement $\mathcal{R}$ into a set of pairwise binary constraints, using the binary partition process, which we denote as $\tilde{\mathcal{R}}$ (line 1). Specifically, for each constraint $(F_i, a_i, \tau_i) \in \mathcal{R}$, the orchestrator generates a set of constraints of the form $\big(F_i, \pi(a_i \mid v_j, v_k), \tau_i\big)\ \forall (v_j, v_k) \in \mathrm{dom}(a_i)$. The resulting set $\tilde{\mathcal{R}}$ contains all the derived binary constraints and replaces the original multi-valued fairness constraints in the subsequent optimization process. This transformation ensures that fairness constraints are enforced, reducing the complexity of handling multi-valued sensitive attributes. Once $\tilde{\mathcal{R}}$ is constructed, the orchestrator initializes a set of learners $L = \{l_1, \ldots, l_{|\mathrm{dom}(a)|}\}$ (line 2), where each learner is made responsible for enforcing the fairness constraints that involve a specific value v of attribute a, explicitly assigned by the orchestrator, i.e., $(F_i, \pi(a|v, v_j), \tau_i) \in \tilde{\mathcal{R}}$. As a consequence, we have a number of learners equal to the $|dom(a)|$.

After partitioning the fairness constraints, the orchestrator sets up the parameters for the next phase (line 3). Given the initial parameters of the network w^1 the orchestrator assigns it to w_{best}. Moreover, given the scoring function to evaluate the constraint satisfaction S, the orchestrator evaluates the current model to assign the initial scoring value to s_{best} and its performance as the best ones to p^*. Next, `FairLAB` starts the *global learning phase*, which is an optimization procedure consisting of a maximum of E iterations (line 4). A generic iteration e involves the following steps: (i) *constraint updates* (line 5), (ii) *learner selection* (line 6), (iii) *model updates* (line 7), and (iv) *model evaluation* (line 8). In the following, we detail each of these steps.

Constraints updates. In each iteration e, given the model m characterized by the parameters w^e, the orchestrator updates two sets of constraints: (i) the *performance* constraints $\mathcal{P}^e$ and (ii) the *proximity* constraints $\mathcal{D}^e$.

The *performance* constraints $\mathcal{P}^e$ ensure that the model maintains high predictive performance throughout the training process, preventing excessive degradation due to fairness enforcement. However, for converging to acceptable balance

between fairness and performance we introduce the *performance budget* β, a key factor that enables the control of how much performance can be sacrificed to satisfy fairness constraints. A larger β permits a greater reduction in utility, allowing stronger fairness enforcement, while a smaller β prioritizes performance at the possible expense of fairness improvements. As a consequence the constraints are defined as: $P(m) \geq p^* + \rho_{\text{step}}$ and $P(m) \geq p^* - \beta$. The first constraint incentivizes performance improvement at each iteration by requiring the model to exceed the best observed performance p^* by at least a small step size ρ_{step}. This helps prevent stagnation and ensures that fairness constraints do not lead to a complete neglect of predictive performance. The second constraint prevents excessive deterioration of predictive performance by ensuring that performance does not drop more than β units below the highest recorded performance p^*. The parameter β acts as a performance safeguard, allowing some flexibility for fairness improvements while preserving overall utility.

The *proximity* constraints $\mathcal{D}^e$ are designed to ensure that fairness properties already achieved at a given iteration are preserved in subsequent updates. Without such constraints, fairness improvements obtained in earlier iterations could be undone as the model continues to optimize for other objectives. To prevent this, given the subset of attributes for which we have the fairness constraints almost or fully satisfied, we propose to measure how much the model's prediction distribution shifts for each subgroup between different iterations using the *1-Wasserstein distance* (also known as the Earth Mover's Distance) [25]. This distance quantifies the minimal amount of probability mass that must be transferred to transform one distribution into another. The proximity constraints then enforce an upper bound on this shift, ensuring that the model's decision distribution remains stable across training updates:

$$\mathcal{D}^e = \{W_1(w^{e-1}, w^e \mid v) \leq \delta \mid \forall a_i \in \mathcal{A}\ \forall v \in \mathcal{AS}(a_i)\},$$

where $W_1(w^{e-1}, w^e \mid v)$ represents the 1-Wasserstein distance between the distribution of model outputs for subgroup v at iteration e and iteration $e-1$, while δ is a small tolerance threshold that controls the maximum allowed change in subgroup-level decision distributions. Note that, given a sensitive attribute a, such distance is conditioned to its values, named *active set* $\mathcal{AS}(a)$, which either satisfy or are close to satisfying fairness constraints. More formally, given a sensitive attribute $a \in \mathcal{A}$, the value v belongs to $\mathcal{AS}(a)$ if and only if there is no fairness violation or the violation between v and any $v_j \neq v$ does not exceed a given threshold, i.e, $\tau_i - F_i^{\pi(a|v,v_j)}(m) \leq \nu_{tot}$. We highlight that $\mathcal{D}^e$ constraints serve to (i) *reduce the risk of fairness violations re-emerging* by discouraging large shifts in subgroup-level decision distributions, which could lead to previously satisfied fairness constraints being violated in subsequent iterations; and (ii) *regularize subgroup-level decision changes*, preventing overfitting to specific fairness constraints while maintaining consistency in model predictions.

Learner selection. At each iteration, the orchestrator selects a learner to perform the next model update. To make this selection, each learner $l_i \in L$ first reports its total fairness violation ν_i, which quantifies how much the fair-

ness constraints assigned to l_i are currently being violated. The orchestrator then assigns a selection probability to each learner, ensuring that learners with higher fairness violations are more likely to be chosen for updates. The selection probability follows a softmax-like distribution, defined as $\frac{e^{-T\nu_i}}{\sum_j e^{-T\nu_j}}$, where $T > 0$ is a temperature parameter that controls the sharpness of the probability distribution. In particular, when T is small, the probability distribution is concentrated around the learners with the highest fairness violations, strongly prioritizing them for updates. This mechanism ensures that the optimization process dynamically focuses on the fairness constraints with higher violation, while still allowing occasional updates from other learners to maintain overall stability in the model training process.

Model update. In this phase, the selected learner l^e receives the updated model parameters w^e along with the two sets of constraints $\mathcal{P}^e, \mathcal{D}^e$. The learner first constructs the full set of constraints that will be optimized in the current iteration: $\mathcal{K}_{l^e} = \mathcal{R}_{l^e} \cup \mathcal{P}^e \cup \mathcal{D}^e$, where $\mathcal{R}_{l^e}$ is the subset of fairness constraints assigned to the learner l^e during the setup phase. The learner then constructs its scoring function `LS`, which follows a similar formulation to the scoring function `S` defined in Eq.2. The key difference is that, while `S` is computed over the initial set of fairness constraints $\mathcal{R}$, the local scoring function `LS` is defined specifically for the constraints $\mathcal{K}_{l^e}$ assigned to the selected learner l^e in the current iteration. At this point l^e proceeds with a learning process based on the ALM described in Sec. 3, to update the model parameters w^e. In this process, we employ a standard task-specific loss function (e.g., cross-entropy for classification tasks) as the primary objective function in ALM. We precise that, since all constraints in $\mathcal{K}_{l^e}$ are formulated as inequality constraints, the Augmented Lagrangian function is defined exclusively over this type of constraint, without including equality constraints. The goal of this optimization process is to refine the network parameters w^e and obtain a new set of parameters w^{e+1} that satisfies the constraints in $\mathcal{K}_{l^e}$, while maintaining predictive performance. Since fairness and performance metrics are often non-differentiable, the learner l^e approximates them using soft confusion-matrices. To achieve this, the learner queries the model m to obtain the logits, which are then processed through the *Entmax* function to produce a sparse probability distribution [23]. This probability distribution is subsequently used to construct soft confusion matrices, from which fairness and performance metrics are computed. These soft approximations allow for the application of gradient-based optimization techniques within the Augmented Lagrangian framework. To ensure numerical stability and prevent fairness constraints from dominating the optimization process, the learner employs an early-stopping criterion in ALM. Specifically, the Lagrangian multipliers are updated only if a constraint violation does not improve over multiple consecutive epochs. This prevents excessive growth of the multipliers relative to the primary optimization objective and mitigates numerical instability.

Model evaluation. At the end of the model update, the learner l^e returns the updated model w^{e+1}, corresponding to the one that maximizes `LS`. If the new model achieves a higher performance than any previously recorded one, the

learner also updates the reference performance value p^* that will be used in the next performance constraints $\mathcal{P}^{e+1}$. The orchestrator then evaluates w^{e+1} using its function `S`. If the new model w^{e+1} achieves a higher score than the best recorded so far, the orchestrator updates its reference model, marking w^{e+1} as the new best model.

This iterative process continues until a stopping criterion is met. Training stops either when the maximum number of iterations is reached or when an early-stopping condition is triggered based on `S`. At that point, the algorithm outputs the model that attains the highest value of `S`.

5 Experimental Settings

We present the experimental setup for validating `FairLAB`[4], considering three benchmarking datasets: `FolkTables`, `Compas`, and `MEPS`[5]. For each dataset we remove duplicates, standardize numerical features by removing the mean and scaling to unit variance, and encode categorical features using one-hot encoding.

`FolkTables` dataset contains census data from California in 2014, with age, education level and so on. The task is to classify whether an individual's income exceeds $50K$. We use $183,380$ samples with 20 features. The sensitive attributes considered are *Job*, *Race* and *MaritalStatus*.

`Compas` dataset contains criminal records from Broward County, Florida. It includes demographic information, criminal history, and risk scores for $6,172$ defendants. The task is to predict whether an arrestee was convicted of violence within two years. After pre-processing, the dataset consists of 34 features. The sensitive attributes considered are *Gender*, *Race* and *Age*.

`MEPS` dataset contains information on healthcare expenditures, medical service utilization, and patient demographics from the United States in 2015. It contains $33,400$ records and over 200 features. After feature selection and one-hot encoding, we reduce to 132. The classification task is to predict whether an individual's total medical expenses exceed the third quantile. The sensitive attributes considered are *Gender*, *Race* and *MaritalStatus*.

For the hyper-parameters and the other implementation details, we refer the interested reader to the Supplementary Material.

Competitors We compare `FairLAB` against: (i) `Vanilla`, a `ML` approach that optimizes only for predictive performance without incorporating fairness constraints, serving as a reference to estimate the initial algorithmic bias present in the data; (ii) FFVAE (`Adversarial`), an adversarial debiasing method designed to mitigate unfairness in scenarios involving non-binary sensitive attributes [6]; (iii) Exponentiated Gradient (`ExpGrad`), a constrained learning approach that dynamically adjusts constraint weights to balance fairness and accuracy, producing a final model by combining multiple reweighted classifiers [1]; and (iv)

[4] We performed the experiments on a server having an Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz, 16 cores and 64GB of RAM. The code is available at: `https://github.com/michelefontana92/FairLAB`

[5] Folk, Compas, Meps

Table 1. Results of the `2-Ways Intersections`. In F_1 we highlight in bold the best performance, while for DP and EOD we report in bold the values below the fairness threshold (0.20) and underline the lowest value.

Debiasing results on `FolkTables`												
Algorithm	Demopraphic Parity (DP)						Equalized Odds (EOD)					
	$JobRace$		$JobMarital$		$RaceMarital$		$JobRace$		$JobMarital$		$RaceMarital$	
	F_1	DP	F_1	DP	F_1	DP	F_1	EOD	F_1	EOD	F_1	EOD
`Vanilla`	**0.77**	0.39	**0.77**	0.38	**0.77**	0.36	**0.77**	0.37	**0.77**	0.37	**0.77**	0.48
`Adversarial`	0.73	0.28	0.74	0.25	0.71	0.28	0.75	0.28	0.70	**0.14**	0.70	**0.18**
`ExpGrad`	0.75	0.34	0.70	**0.19**	0.67	**0.20**	0.73	0.25	0.75	0.24	0.74	0.28
`COSMOS`	0.70	0.23	0.73	0.29	0.73	0.32	0.74	0.28	0.73	**0.20**	0.73	0.33
`FairLAB` (0.05)	0.72	**0.20**	0.74	**0.19**	0.72	**0.20**	0.74	**0.20**	0.73	**0.20**	0.74	**0.19**

Debiasing results on `Compas`												
Algorithm	Demopraphic Parity (DP)						Equalized Odds (EOD)					
	$GenderRace$		$GenderAge$		$RaceAge$		$GenderRace$		$GenderAge$		$RaceAge$	
	F_1	DP	F_1	DP	F_1	DP	F_1	EOD	F_1	EOD	F_1	EOD
`Vanilla`	**0.69**	0.49	**0.69**	0.62	**0.69**	0.56	**0.69**	0.45	**0.69**	0.64	**0.69**	0.53
`Adversarial`	0.65	0.27	0.67	0.30	0.60	0.22	0.65	**0.18**	0.55	**0.09**	0.61	**0.20**
`ExpGrad`	0.61	**0.12**	0.65	0.26	0.63	0.26	0.65	**0.20**	0.65	**0.11**	0.64	0.27
`COSMOS`	0.66	0.32	0.64	0.31	0.68	0.38	0.68	0.36	0.69	0.44	0.62	0.29
`FairLAB` (0.05)	0.66	**0.19**	0.65	**0.20**	0.64	**0.19**	0.66	**0.19**	0.65	**0.20**	0.65	**0.20**

Debiasing results on `MEPS`												
Algorithm	Demopraphic Parity (DP)						Equalized Odds (EOD)					
	$GenderRace$		$GenderMarital$		$RaceMarital$		$GenderRace$		$GenderMarital$		$RaceMarital$	
	F_1	DP	F_1	DP	F_1	DP	F_1	EOD	F_1	EOD	F_1	EOD
`Vanilla`	**0.81**	0.29	**0.81**	0.45	**0.81**	0.53	**0.81**	0.34	**0.81**	0.57	**0.81**	0.62
`Adversarial`	0.78	0.22	0.70	**0.10**	0.78	0.40	0.78	0.35	0.80	0.42	0.75	0.45
`ExpGrad`	0.80	0.26	0.79	0.33	0.79	0.33	0.80	0.32	0.79	0.40	0.78	0.52
`COSMOS`	0.80	0.21	0.79	0.29	0.79	0.37	0.78	0.25	0.80	0.30	0.80	0.40
`FairLAB` (0.05)	0.80	**0.14**	0.79	**0.18**	0.79	**0.20**	0.75	**0.19**	0.78	**0.15**	0.76	**0.20**

`COSMOS`, a MOO method that efficiently learns a Pareto front of models with equivalent trade-offs between fairness and accuracy [24].

6 Experiments

In this section we provide our experimental evaluation of `FairLAB`, against the competitors across multiple fairness settings. In the evaluation we use `DP` and `EOD` as fairness metrics, and F_1 score for the predictive performance. Our analysis aims to assess (i) how effectively `FairLAB` balances fairness and predictive performance, studying key factors such as the performance budget β and the proximity threshold δ that influence its behavior, and (ii) its scalability.

In particular, we analyze from simpler to more complex fairness scenarios, exploring the intersectional fairness with two sensitive attributes in Sec. 6.1 and with three in Sec. 6.2, highlighting the increasing difficulty of enforcing fairness as dimensionality grows. Following, Sec. 6.3 examines cases where multiple fairness constraints are applied simultaneously, incorporating both group-level and intersectional constraints. Sec. 6.4 explores the impact of the proximity thresh-

old δ on the trade-off between preserving the fairness properties and improving the performance. Finally, Sec. 6.5 proves that `FairLAB` is scalable.

6.1 Experiments with Two-Attribute Intersections

We evaluate fairness constraints by enforcing either `DP` or `EOD` individually, considering all intersections of two sensitive attributes for each dataset. We impose $F^{a_i}(m) \leq 0.2$ on subgroups defined by a_i which intersects two attributes. Here F is `DP` or `EOD`. Table 1 reports the results.

Demographic Parity. `FairLAB` is the only method that meets fairness objectives in all cases. While `Adversarial` often achieves a substantial bias reduction (as seen by comparing fairness values to `Vanilla`, the baseline), in some scenarios it fails (e.g., *JobRace* in `FolkTables` and $RaceMarital$ in `MEPS`). Similarly, `ExpGrad` succeeds for certain intersections (e.g., $JobMarital$ in `FolkTables`) but struggles elsewhere (e.g., *JobRace* in `FolkTables`). `COSMOS` has mixed outcomes, even exceeding fairness by more than 0.18 (e.g., *RaceAge* in `Compas`).
Equalized-Odds. `FairLAB` meets the fairness constraint in every configuration, demonstrating strong bias mitigation. For `FolkTables`, all competitors achieve acceptable fairness and F_1 score. For `Compas` and `MEPS`, results are mixed: while `Adversarial` performs well, the other methods mitigate bias only for certain attributes, often exceeding the desired bias threshold of 20 by at least 0.10.
Performance Analysis. All fairness-aware approaches have lower F_1 than the unconstrained `Vanilla`, which achieves the highest performance but significantly violates fairness. Among the debiasing methods, `FairLAB` strikes a strong balance between fairness and utility, maintaining competitive F_1 while meeting all fairness targets. For instance, on `MEPS` under `EOD`, `FairLAB` reduces bias below 0.20 with a slight decrease in F_1 (at most 0.06). By contrast, `Adversarial` sometimes suffers substantial utlity drops (e.g., $F_1 = 0.55$ on `Compas`, compared to 0.69 for `Vanilla` and 0.65 for `FairLAB`). Although `ExpGrad` and `COSMOS` often retain good F_1 scores, they do not always meet the fairness constraints.

6.2 Experiments with Three-Attribute Intersections

In this experiment, we evaluate `FairLAB` by creating attributes combining three sensitive factors to test its performance in a more complex setting. Specifically, we consider $JobRaceMarital$ for `FolkTables`, $GenderRaceAge$ for `Compas`, and $GenderRaceMarital$ for `MEPS`. We enforce `DP` and `EOD` separately, each with a threshold of 0.20. In addition, given the more challenging setting, we evaluate `FairLAB` under three different values of β to analyze the trade-off between fairness and F_1 score. Table 2 reports the F_1 and fairness values for all methods.
Demographic Parity. None of the competitors reduce `DP` below 0.20 on any dataset. `Adversarial` shows significant violations (e.g., 0.65 in `Compas`), indicating that it becomes less effective as the dimensionality of the sensitive attribute increases. Although `ExpGrad` and `COSMOS` achieve moderate improvements, with `DP` between 0.33 and 0.43, these remain well above the threshold. By contrast, `FairLAB` achieves the fairness objectives with bigger values for the performance

Table 2. Results `3-Ways Intersections`. The F_1 column highlights in bold the best predictive performance, while the DP and EOD columns highlight in bold the values below the fairness threshold (0.20) and underline the lowest value.

Algorithm	FolkTables				Compas				MEPS			
	DP		EOD		DP		EOD		DP		EOD	
	F_1	DP	F_1	EOD	F_1	DP	F_1	EOD	F_1	DP	F_1	EOD
`Vanilla`	**0.77**	0.50	**0.77**	0.84	**0.69**	0.75	**0.69**	0.82	**0.81**	0.59	**0.81**	0.69
`Adversarial`	0.76	0.45	0.75	0.80	0.67	0.65	0.63	0.67	0.79	0.56	0.80	0.60
`ExpGrad`	0.74	0.36	0.66	0.50	0.60	0.43	0.60	0.71	0.79	0.36	0.78	0.56
`COSMOS`	0.71	0.33	0.75	0.67	0.61	0.42	0.67	0.73	0.80	0.43	0.80	0.53
`FairLAB` (0.05)	0.72	0.30	0.70	0.35	0.64	0.31	0.66	0.42	0.76	**0.20**	0.75	0.25
`FairLAB` (0.10)	0.68	**0.20**	0.68	0.31	0.62	0.25	0.60	0.34	0.75	**0.17**	0.73	**0.20**
`FairLAB` (0.15)	0.65	**0.17**	0.65	0.28	0.58	**0.19**	0.56	0.24	0.71	**0.18**	0.71	**0.18**

budget β. In addition, with β, it offers tunable trade-offs. For $\beta = 0.05$, it attains $DP \leq 0.20$ only in `MEPS`, implying that a minimal sacrifice in F_1 is sufficient there but not in `FolkTables` or `Compas` (where DP remains around 0.30). Increasing β to 0.10 narrows this gap, dropping `DP` to 0.20 in `FolkTables` at the cost of an F_1 reduction from 0.72 to 0.68. Achieving $DP \leq 0.20$ in `Compas` requires $\beta = 0.15$, which drives F_1 as low as 0.58. This highlights the *data-dependent* nature of intersectional fairness: some datasets (e.g. `Compas`) demand a larger performance budget to meet the tighter fairness constraint.

Equalized Odds. A similar pattern emerges for `EOD`. While `ExpGrad` and `COSMOS` exceed 0.50 in every scenario, `FairLAB` lowers `EOD` below 0.20 in `MEPS` with $\beta \geq 0.10$. However, `FairLAB` still registers `EOD` values of 0.28 and 0.24 in `FolkTables` and `Compas`, respectively, suggesting that meeting intersectional `EOD` thresholds for all subgroups may require even higher budgets.

Performance Analysis. In all three datasets, the unconstrained model `Vanilla` achieves the highest F_1 scores (e.g., 0.77 in `FolkTables` and 0.69 in `Compas`), albeit with severe fairness violations. Methods like `Adversarial`, `ExpGrad`, and `COSMOS` typically retain F_1 close to `Vanilla`, but their debiasing effect is insufficient for three-attribute intersections. The pivotal element of `FairLAB` 's performance is the parameter β, which balances accuracy and fairness, explaining why `FairLAB` with $\beta = 0.05$ can satisfy `DP` in `MEPS` but not in `FolkTables` or `Compas`: the latter datasets require more significant adjustments to model predictions to mitigate intersectional biases. For `EOD`, the same reasoning applies, further amplified by the metric's dependence on both true and false positive rates, which increases data fragmentation when three sensitive attributes intersect. Another factor is the *data distribution* within each intersectional subgroup. With three attributes, certain subgroups may contain relatively few samples, so applying `EOD` constraints forces the model to adapt its decision boundary more drastically. As β grows, `FairLAB` can impose these constraints more effectively, albeit with a noticeable drop in F_1.

Table 3. Results of `Multiple Constraints`, with DP. For F_1, the best performance is in bold, while for DP we highlight the values below the fairness threshold (0.20 for the intersectional and 0.10 for the single attribute) and underline the lowest value.

Debiasing results on `FolkTables`								
Algorithm	F_1	$J*R*M$	$J*R$	$J*M$	$R*M$	Job	$Race$	$Marital$
		DP	DP	DP	DP	DP	DP	DP
`Vanilla`	**0.77**	0.50	0.39	0.38	0.36	0.18	0.21	0.22
`COSMOS`	0.73	0.36	0.34	0.23	0.27	0.09	0.15	0.18
`FairLAB` (0.05)	0.72	0.32	**0.20**	**0.16**	0.28	**0.04**	0.12	0.15
`FairLAB` (0.10)	0.67	**0.20**	**0.14**	**0.14**	**0.17**	**0.03**	**0.10**	**0.09**
`FairLAB` (0.15)	0.67	**0.19**	**0.17**	**0.20**	**0.18**	**0.02**	**0.08**	**0.09**

Debiasing results on `Compas`								
Algorithm	F_1	$G*R*A$	$G*R$	$G*A$	$R*A$	$Gender$	$Race$	Age
		DP	DP	DP	DP	DP	DP	DP
`Vanilla`	**0.69**	0.75	0.49	0.62	0.56	0.31	0.24	0.39
`COSMOS`	0.66	0.46	0.33	0.27	0.34	0.13	0.20	0.14
`FairLAB` (0.05)	0.62	0.32	**0.17**	**0.20**	**0.20**	**0.09**	**0.08**	**0.07**
`FairLAB` (0.10)	0.60	0.27	**0.16**	**0.18**	**0.18**	**0.07**	**0.10**	**0.08**
`FairLAB` (0.15)	0.61	0.23	**0.16**	**0.14**	**0.18**	**0.07**	**0.07**	**0.05**

Debiasing results on `MEPS`								
Algorithm	F_1	$G*R*M$	$G*R$	$G*M$	$R*M$	$Gender$	$Race$	$Marital$
		DP	DP	DP	DP	DP	DP	DP
`Vanilla`	**0.81**	0.59	0.29	0.45	0.53	**0.09**	0.22	0.39
`COSMOS`	0.76	0.29	0.25	**0.15**	0.28	**0.01**	**0.09**	**0.10**
`FairLAB` (0.05)	0.76	**0.17**	**0.09**	**0.12**	**0.14**	**0.01**	**0.07**	**0.09**
`FairLAB` (0.10)	0.76	**0.17**	**0.12**	**0.10**	**0.17**	**0.02**	**0.09**	**0.06**
`FairLAB` (0.15)	0.75	**0.12**	**0.08**	**0.07**	**0.19**	**0.01**	**0.05**	**0.07**

6.3 Experiments with Mixed Constraints

We also evaluate `FairLAB` in a scenario where multiple intersectional fairness constraints are enforced *simultaneously*. Specifically, each dataset is subject to seven constraints which express all the possible intersectional attributes (combining two or three attributes) and the single sensitive attributes (e.g., *Gender*, *Race*). We set the threshold to 0.20 for the intersectional attributes and to 0.10 for the single ones. We compare `FairLAB` exclusively to `COSMOS` because it is the only method capable of handling multiple fairness constraints simultaneously[6].

Demographic Parity. Table 3 shows that `FairLAB` successfully meets all `DP` constraints on both `FolkTables` and `MEPS`. As already mentioned, the trade-off between fairness and accuracy depends on the performance budget β, which value varies across datasets to achieve the required fairness level. On `FolkTables`, reducing `DP` below 0.20 at the three-way attribute $JobRaceMarital$ requires $\beta \geq 0.10$, which lowers F_1 from 0.77 to 0.67. In contrast, `MEPS` requires a lower β, with `FairLAB` satisfying every constraint at $\beta = 0.05$ while maintaining a good F_1 of 0.76. The behavior on `Compas` is more challenging, as `FairLAB` meets six out of seven constraints with β in $[0.10, 0.15]$, while $GenderRaceAge$ remains slightly above 0.20. However, we consider this result acceptable, as the deviation

[6] Columns with $*$ are intersectional attributes, where initials indicate their features.

Table 4. Results of `Multiple Constraints`, with EOD. For F_1 we highlight the best performance, for EOD we report in bold the values below the fairness threshold (0.20 for intersectional and 0.10 for single attributes) and underline the lowest value.

Debiasing results on `FolkTables`								
Algorithm	F_1	$J*R*M$	$J*R$	$J*M$	$R*M$	Job	$Race$	$Marital$
		EOD	EOD	EOD	EOD	EOD	EOD	EOD
`Vanilla`	**0.77**	0.84	0.37	0.37	0.48	0.18	0.27	0.36
`COSMOS`	0.75	0.80	0.34	0.33	0.21	0.12	0.19	**0.09**
`FairLAB` (0.05)	0.73	0.38	0.24	**0.18**	**0.19**	0.18	0.12	**0.09**
`FairLAB` (0.10)	0.60	0.29	**0.20**	**0.20**	**0.19**	**0.08**	**0.07**	**0.08**
`FairLAB` (0.15)	0.59	0.24	**0.15**	**0.10**	**0.16**	**0.06**	**0.10**	**0.03**

Debiasing results on `Compas`								
Algorithm	F_1	$G*R*A$	$G*R$	$G*A$	$R*A$	$Gender$	$Race$	Age
		EOD	EOD	EOD	EOD	EOD	EOD	EOD
`Vanilla`	**0.69**	0.82	0.45	0.64	0.53	0.30	0.22	0.30
`COSMOS`	0.67	0.71	0.26	0.39	0.43	0.11	0.19	0.16
`FairLAB` (0.05)	0.63	0.35	0.27	**0.20**	0.25	0.14	**0.07**	0.12
`FairLAB` (0.10)	0.59	0.26	**0.18**	**0.15**	**0.16**	**0.07**	**0.05**	**0.09**
`FairLAB` (0.15)	0.54	0.22	**0.14**	**0.10**	**0.12**	**0.02**	**0.01**	**0.06**

Debiasing results on `MEPS`								
Algorithm	F_1	$G*R*M$	$G*R$	$G*M$	$R*M$	$Gender$	$Race$	$Marital$
		EOD	EOD	EOD	EOD	EOD	EOD	EOD
`Vanilla`	**0.81**	0.69	0.34	0.57	0.62	**0.10**	0.25	0.49
`COSMOS`	0.69	**0.15**	**0.10**	**0.12**	**0.13**	**0.02**	**0.08**	**0.07**
`FairLAB` (0.05)	0.77	0.29	**0.20**	**0.18**	**0.20**	**0.07**	**0.09**	**0.09**
`FairLAB` (0.10)	0.70	**0.19**	**0.07**	**0.09**	**0.17**	**0.03**	**0.05**	**0.05**
`FairLAB` (0.15)	0.69	**0.20**	**0.12**	**0.14**	**0.19**	**0.05**	**0.02**	**0.04**

is minimal given that it is very close to the target and occurs in a highly complex setting where all other constraints are successfully satisfied. Regarding the competitor, `COSMOS` generally preserves higher F_1 scores but fails most intersectional constraints. On `FolkTables`, for instance, it surpasses 0.30 for $JobRaceMarital$, indicating difficulty in handling finer demographic partitions.

Equalized Odds. Table 4 confirms a similar pattern when `EOD` constraints are imposed. `FairLAB` once again reduces the three-way attribute well below the unconstrained baseline in `FolkTables` and `Compas`, but cannot always push `EOD` under 0.20 for every subgroup, even at higher performance budgets (e.g., 0.24 on `FolkTables` and 0.22 on `Compas`). Nevertheless, these values are quite close to the target and represent notable improvements over `COSMOS`, which reaches 0.70 or 0.80 on the same intersectional group. On `MEPS`, `FairLAB` achieves or nears the threshold in all subgroups when $\beta \geq 0.10$. `COSMOS` exhibits partial success, meeting some constraints on `MEPS` 's single attributes more easily, yet it remains less effective for the higher-dimensional partitions.

Performance Analysis. As observed in Sections 6.1 and 6.2, the parameter β in `FairLAB` governs the balance between fairness and accuracy. Larger budgets allow the model to target smaller `DP`/`EOD` values but can reduce F_1 by up to 10–15 points relative to the unconstrained baseline `Vanilla`. On `FolkTables`,

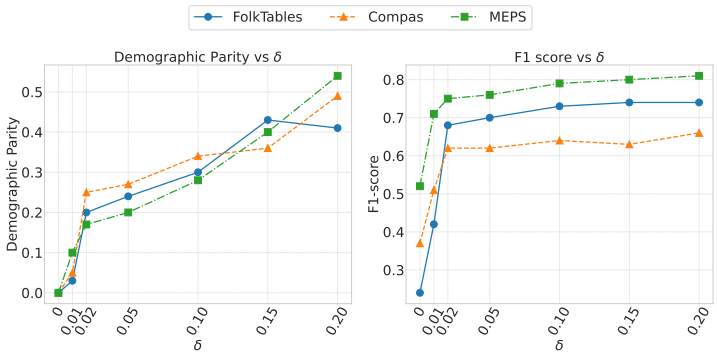

Fig. 2. Impact of the proximity parameter δ on the fairness and F_1 of the final model.

for instance, lowering `EOD` to around 0.24 at $JobRaceMarital$ requires $\beta = 0.15$, which yields an F_1 of 0.59 compared to 0.77 for `Vanilla`. However, `COSMOS`, while retaining a higher F_1, remains above 0.70 in `EOD` for the same attribute, indicating that the debiasing strategy is not working. On `MEPS`, `FairLAB` faces a more moderate trade-off and meets nearly all constraints with β as low as 0.05 or 0.10, incurring only a slight drop in F_1. Overall, these experiments demonstrate that meeting multiple constraints—especially at the intersectional level—can demand a significant performance budget in some datasets, but `FairLAB` consistently outperforms `COSMOS` in reducing unfairness across diverse subgroup partitions.

6.4 Impact of the Proximity Parameter

We also investigate how the *proximity threshold* δ influences the final model learned under the three-attribute scenario discussed in Section 6.2. By design, δ dictates how closely the new model must mimic the orchestrator's conditional distributions on specified subgroups, effectively limiting how much the updated model's predictions can deviate. We vary δ and record the resulting `DP` and F_1 scores, keeping the performance budget fixed at $\beta = 0.05$.

Figure 2 illustrates that increasing δ grants more freedom for the updated model to diverge from the orchestrator, yielding higher F_1 but at the expense of fairness. When δ becomes large, the new model no longer inherits the orchestrator's fairness properties from previous iterations and instead optimizes primarily for learner constraints. Consequently, once $\delta \geq 0.05$, `DP` surpasses the 0.20 threshold in `FolkTables`, `Compas`, and `MEPS`, indicating that intersectional fairness can no longer be maintained. In contrast, with smaller values (e.g., $\delta = 0.01$ or 0.02), the updated model remains sufficiently close to the orchestrator's distribution, keeping `DP` below 0.20 without incurring a pronounced performance penalty. This trade-off underscores the importance of calibrating δ to preserve the orchestrator's fairness characteristics while allowing for potential improvements in F_1.

Table 5. Average training times in: (1) Int2 (`2-Ways Intersections`) (2) Int3 (`3-Ways Intersections`) (3) Mixed (`Multiple Constraints`).

Algorithm	FolkTables			Compas			MEPS		
	Int2	Int3	Mixed	Int2	Int3	Mixed	Int2	Int3	Mixed
`Vanilla`	30.3 min	30.3 min	30.3 min	3.5 min	3.5 min	3.5 min	27.2 min	27.2 min	27.2 min
`Adversarial`	3.7 hrs	2 day	-	1.2 hrs	3.5 hrs	-	5.2 hrs	1.1 day	-
`ExpGrad`	**20.8 min**	**1.2 hrs**	-	**5.4 min**	**7.1 min**	-	**14.6 min**	**40.3 min**	-
`COSMOS`	10.7 hrs	2.2 day	4.5 day	3.3 hrs	1.5 day	2.8 day	13.4 hrs	1.7 day	3.9 day
`FairLAB` (0.05)	1.3 hrs	2.4 hrs	**2.7 hrs**	6.3 min	9.7 min	**12.1 min**	30.8 min	45.2 min	**48.9 min**

6.5 Runtime Analysis

We assess the scalability of `FairLAB` by measuring its training times in the experimental settings in Sec. 6.1, 6.2, and 6.3. Each setting involves an increasing number of fairness constraints and/or more complex intersectional attributes, allowing us to observe how computational costs scale with the problem complexity. Table 5 shows that `FairLAB` balances efficiency and fairness constraints, maintaining moderate training times even in complex settings. Compared to `COSMOS`, it achieves faster convergence, while `ExpGrad`, though faster, often fails to meet fairness requirements. `Adversarial` remains feasible in simpler cases but becomes impractical for larger-scale constraints. Overall, `FairLAB` offers a good trade-off between runtime and debiasing performance, making it suitable for real-world applications.

7 Conclusions

We introduced `FairLAB`, a method for training fair and high-performing `NN` under both group and intersectional fairness constraints, exploiting ALM. With the *performance budget*, we provide explicit control to the trade-off between fairness and performance, making `FairLAB` adaptive. It is also scalable thanks to a divide-et-impera strategy for decomposing the fairness problem, particularly when multiple sensitive attributes interact. Experimental results show that `FairLAB` mitigates bias across challenging real-world scenarios, where multiple fairness constraints, must be satisfied simultaneously. Also, our empirical evaluation shows that `FairLAB` outperforms state-of-the-art fairness mitigation methods, striking a balance between fairness and performance, even in high-dimensional settings. These results highlight the applicability of `FairLAB` for fairness-aware real-world applications. As future work, we plan to extend it to decentralized learning scenarios, such as Federated Learning[8,12], with the challenge of ensuring fairness across distributed nodes, while preserving user privacy and model performance.

Acknowledgments. This research was partially supported by: SoBigData.it – "Strengthening the Italian RI for Social Mining and Big Data Analytics" – Prot. IR0000013 – Avviso n. 3264 del 28/12/2021; Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI"; The European Union's Horizon Europe research and innovation programme for the project FINDHR (g.a.

No. 101070212); The European Union Horizon 2020 program under grant agreement No. 101120763 *TANGO* and No. 101070416 *GREEN.DAT.AI*; Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., Wallach, H.M.: A reductions approach to fair classification. In: ICML. vol. 80, pp. 60–69. PMLR (2018)
2. Calders, T., Kamiran, F., Pechenizkiy, M.: Building classifiers with independency constraints. In: ICDM Workshops. pp. 13–18. IEEE Computer Society (2009)
3. Caton, S., Haas, C.: Fairness in machine learning: A survey. ACM Comput. Surv. **56**(7), 166:1–166:38 (2024)
4. Celis, L.E., Huang, L., Keswani, V., Vishnoi, N.K.: Classification with fairness constraints: A meta-algorithm with provable guarantees. In: FAT (2019)
5. Cotter, A., Jiang, H., Sridharan, K.: Two-player games for efficient non-convex constrained optimization. In: Algorithmic Learning Theory. pp. 300–332 (2019)
6. Creager, E., Madras, D., Jacobsen, J.H., Weis, M., Swersky, K., Pitassi, T., Zemel, R.: Flexibly fair representation learning by disentanglement. In: ICML (2019)
7. Désidéri, J.A.: Multiple-gradient descent algorithm (mgda) for multiobjective optimization. Comptes Rendus Mathematique **350**(5-6), 313–318 (2012)
8. Fontana, M., Naretto, F., Monreale, A.: A new approach for cross-silo federated learning and its privacy risks. In: 18th International Conference on Privacy, Security and Trust, PST 2021, Auckland, New Zealand, December 13-15, 2021. pp. 1–10. IEEE (2021). https://doi.org/10.1109/PST52912.2021.9647753
9. Fortin, M., Glowinski, R.: Augmented Lagrangian methods: applications to the numerical solution of boundary-value problems, vol. 15. Elsevier (2000)
10. Foulds, J.R., Islam, R., Keya, K.N., Pan, S.: An intersectional definition of fairness. In: ICDE. pp. 1918–1921 (2020)
11. Gohar, U., Cheng, L.: A survey on intersectional fairness in machine learning: Notions, mitigation, and challenges. arXiv preprint arXiv:2305.06969 (2023)
12. Haffar, R., Naretto, F., Sánchez, D., Monreale, A., Domingo-Ferrer, J.: Glorflex: Local to global rule-based explanations for federated learning. In: 2024 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). pp. 1–9 (2024). https://doi.org/10.1109/FUZZ-IEEE60900.2024.10611878
13. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: NIPS. pp. 3315–3323 (2016)
14. Kamishima, T., Akaho, S., Sakuma, J.: Fairness-aware learning through regularization approach. In: Data Mining Workshops (ICDMW) (2011)
15. Kearns, M., Neel, S., Roth, A., Wu, Z.S.: Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In: ICML. pp. 2564–2572 (2018)
16. Kim, M., Reingold, O., Rothblum, G.: Fairness through computationally-bounded awareness. Advances in neural information processing systems **31** (2018)

17. Lokhande, V.S., Akash, A.K., Ravi, S.N., Singh, V.: Fairalm: Augmented lagrangian method for training fair models with little regret. In: European Conference on Computer Vision. pp. 365–381. Springer (2020)
18. Lu, S., Zeng, S., Cui, X., Squillante, M., Horesh, L., Kingsbury, B., Liu, J., Hong, M.: A stochastic linearized augmented lagrangian method for decentralized bilevel optimization. NeurIPS **35**, 30638–30650 (2022)
19. Martinez, N.L., Bertran, M.A., Papadaki, A., Rodrigues, M., Sapiro, G.: Blind pareto fairness and subgroup robustness. In: International Conference on Machine Learning. pp. 7492–7501. PMLR (2021)
20. Naretto, F., Monreale, A., Giannotti, F.: Evaluating the privacy exposure of interpretable global explainers. In: 2022 IEEE 4th International Conference on Cognitive Machine Intelligence (CogMI). pp. 13–19 (2022). https://doi.org/10.1109/CogMI56440.2022.00012
21. Naretto, F., Pellungrini, R., Fadda, D., Rinzivillo, S.: Exphlot: Explainable privacy assessment for human location trajectories. In: Discovery Science. DS 2023. Lecture Notes in Computer Science. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-45275-8_22
22. Naretto, F., Pellungrini, R., Monreale, A., Nardini, F.M., Musolesi, M.: Predicting and explaining privacy risk exposure in mobility data. In: Discovery Science - 23rd International Conference, DS 2020. Lecture Notes in Computer Science, Springer (2020). https://doi.org/10.1007/978-3-030-61527-7_27
23. Peters, B., Niculae, V., Martins, A.F.: Sparse sequence-to-sequence models. In: Proc. ACL (2019)
24. Ruchte, M., Grabocka, J.: Scalable pareto front approximation for deep multi-objective learning. In: ICDM. pp. 1306–1311 (2021)
25. Villani, C.: The Wasserstein distances, pp. 93–111. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
26. Yu, G., Ma, L., Wang, X., Du, W., Du, W., Jin, Y.: Towards fairness-aware multi-objective optimization. Complex & Intelligent Systems **11**(1), 50 (2025)
27. Zhang, B.H., Lemoine, B., Mitchell, M.: Mitigating unwanted biases with adversarial learning. In: AAAI/ACM Conf. on AI, Ethics, and Society (2018)
28. Zhang, Q., Liu, J., Yao, X.: Fairness-aware multiobjective evolutionary learning. IEEE Transactions on Evolutionary Computation (2024)