Fairness-aware Reweighting in Federated Learning

Paul Andrey[1], Brahim Erraji[1], and Michaël Perrot[1]

[1]Univ. Lille, Inria, CNRS, Centrale Lille, UMR 9189 - CRIStAL, F-59000 Lille, France

Abstract

We address the problem of enforcing group fairness in federated learning. To this end, we first adapt FairGrad, a recently proposed fairness method in centralized learning, to the federated setting. This comes with a large communication cost as it requires sharing the fairness level of the learned models after each gradient descent step. Unfortunately, using the usual federated learning trick consisting in taking several local steps before communicating with the server leads to models with low utility. To tackle this issue, we propose an alternative approach that combines ideas from FairGrad and cost-sensitive fairness methods. Our preliminary experiments show that this new method is either competitive or better than several state-of-the-art approaches.

Keywords: Fairness, Federated Learning.

1 Introduction

Machine Learning approaches are nowadays used to solve a wide range of problems. Based on historical data, they often reach human-like performances. Unfortunately, the datasets used to train these models sometimes reflect biases present in our society or may be under-representative of some subgroups of the population. Together with potentially inadequate algorithmic choices, this often leads to models that tend to unfairly disadvantage some individuals. For example, De-Arteaga et al. (2019) show that a model trained to predict the professional occupation of a person from its biography may perpetuate existing gender stereotypes. When human lives get impacted by automated decision-making (Chouldechova and Roth, 2020), it is of the utmost importance to ensure that such biases do not arise. Thus, fair machine learning has gained a lot of attention in the recent years (Barocas et al., 2023).

Defining fairness in Machine Learning is challenging as different problems usually call for different notions of non-discrimination. Two main kind of notions became prevalent in the recent literature. On the one hand, individual fairness advocates that similar individual should be treated similarly (Dwork et al., 2012). On the other hand, group fairness seeks to equate the performance of machine learning models across different sensitive groups, that is groups based on sensitive attributes such as gender or ethnicity (Hardt et al., 2016; Zafar et al., 2017). In this work, we consider a generic group fairness formulation encompassing several of these notions (Maheshwari and Perrot, 2023).

The goal of fair machine learning is to learn models that maximize utility while satisfying a given fairness definition. Many approaches have been proposed, that fall into several categories (Caton and Haas, 2023). Pre-processing techniques aim to debias data before training (Kamiran and Calders, 2012; Zemel et al., 2013; Calmon et al., 2017), in-processing approaches steer the training procedure towards directions that maintain fairness within the model (Maheshwari and Perrot, 2023; Celis et al., 2019; Cotter et al., 2019), and post-processing methods adjust the predictions made after training to achieve fairness (Hébert-Johnson et al., 2018; Hardt et al., 2016; Pleiss et al., 2017). Most fairness approaches have been proposed in the centralized setting where the data can be accessed at will, and only a few of them consider the federated setting (Kairouz et al., 2021), where data is partitioned across many clients and a server orchestrates them to learn a global model while only accessing the data through aggregate statistics. The goal of this paper is to bridge this gap. More precisely, we propose new algorithms to learn fair models in cross-silo federated learning, where clients have enough communication and computation resources to be responsive to every query of the server. Furthermore, we assume that the clients are honest and provide accurate information (gradients, fairness-levels, ...).

Learning fair models in federated learning is challenging. Indeed, merely applying debiasing methods locally does not ensure the composition of these local models to be fair (Dwork and Ilvento, 2019). Furthermore, even with recurrent synchronization of these models, the aggregated model is generally not fair because of the data heterogeneity across the clients (Wang et al., 2023). Nevertheless, several works have addressed fairness in federated learning from different angles. For instance, some approaches propose to balance the local datasets in a pre-processing step (Abay et al., 2020) or strive to minimize the loss of the most disadvantaged groups (Papadaki et al., 2022). Similarly, several recent approaches propose to calculate adaptive sensitive group weights (Ezzeldin et al., 2023; Zeng et al., 2021). In this work, we follow the latter trend and propose two extensions of FairGrad (Maheshwari and Perrot, 2023), a recent reweighting approach for fairness in the centralized setting that has been shown to be easy to implement and applicable to a wide range of settings.

Contributions In this work, we study new reweighting approaches to learn fair models in federated learning. First, a method called Fed-FairGrad straightforwardly extends FairGrad (Maheshwari and Perrot, 2023) by computing the weights for each sensitive group in a federated way. This method requires an additional communication cost of K floats, one for each sensitive group, between each training round. Its strict application would also require rounds to consist of a single step, which can be quite expensive. To alleviate this issue, we use a standard approach in federated learning (McMahan et al., 2017). Unfortunately, we show that this leads to models with reduced utility. Based on these observations, we propose a second approach, called Fed-FairGrad-Convex, that combines ideas from cost-sensitive fairness methods (Agarwal et al., 2018) and FairGrad (Maheshwari and Perrot, 2023). Through a series of experiments with varying levels of heterogeneity, we demonstrate that Fed-FairGrad-Convex tends to learn fair models with better utility that state-of-the-art methods.

Related Work Fairness in federated learning can be seen from two main perspectives. On the one hand, client parity ensures that the clients receive models with performances proportional to their contributions (Mohri et al., 2019; Li et al., 2019; Yue et al., 2023). On the other hand, the goal in group fairness, the problem investigated in this paper, is to ensure that the global model learned in a federated way is fair on the overall data distribution formed by a mixture of the local, potentially different, distributions. Among the different works proposed in this literature (Abay et al., 2020; Gálvez et al., 2021; Zhang et al., 2020; Padala et al., 2021; Papadaki et al., 2022; Su et al., 2024), the ones closest to ours are the ones that use some kind of reweighting scheme to balance the impact that each sensitive group has on the learned model. First, Du et al. (2021) propose to learn group-wise weights based on relaxations of the fairness constraints under consideration. Our solution also uses reweighting but with respect to the true fairness constraints, avoiding the pitfall of relaxations that may lead to unfair models (Lohaus et al., 2020). Similarly, FairFed (Ezzeldin et al., 2023) is an approach that explores the idea of assigning adaptive weights to the updates received from each client to achieve fairness. More precisely, it proposes to increase the weight of clients with updates with fairness levels closer to the global fairness level and to decrease the weight of the others. In our work, rather than altering the importance of each client update directly, we modify the importance of each sensitive group, implicitly reducing the impact of clients that only have access to examples from the advantaged groups. Finally, the closest work to ours is FedFB (Zeng et al., 2021) which proposes to alter, at each iteration of gradient descent, the sampling probabilities used to generate minibatches of sensitive groups based on the fairness level of the current model. This is an extension of the FairBatch method, designed for the centralized setting (Roh et al., 2021). Our approach is also an extension of a centralized approach to the federated setting but with a different basis, namely FairGrad (Maheshwari and Perrot, 2023).

2 Problem setting and Notations

In this work, we consider a binary classification problem in federated learning, where data is distributed across m clients. The goal is to accurately predict a label $y \in \{0, 1\}$ based on features $x \in \mathcal{X}$ using a model $h_\theta : \mathcal{X} \to [0, 1]$ while staying fair with respect to some discrete sensitive attribute $s \in \mathcal{S}$ such as gender or ethnicity. We denote $\theta \in \mathbb{R}^d$ the learnable model parameters. We assume that there exists an underlying distribution $\mathcal{T}$ over $\mathcal{X} \times \{0, 1\} \times \mathcal{S}$ and that our data can be partitioned into K different groups $\{g_1, \ldots, g_K\}$ based on their sensitive attributes and their labels (Maheshwari and Perrot, 2023). We note $\mathcal{T}_k$ the distribution of samples for a group g_k. We also note $\mathcal{T}^{(i)}$ the distribution of samples held by a client (i), and $\mathcal{T}_k^{(i)}$ that of samples that belong to both client (i) and group g_k. We implicitly assume that $\mathcal{T}$ is a convex combination of local distributions:

$$\exists (p^{(1)}, \ldots p^{(m)}) \geq 0 : \mathcal{T} = \sum_{i=1}^{m} p^{(hi)} \mathcal{T}^{(i)} \text{ such that : } \sum_{i=1}^{m} p^{(i)} = 1. \tag{1}$$

where $\forall i, p^{(i)} = \mathbb{P}(\mathcal{T}^{(i)})$, which is usual in federated learning, but not ubiquitous (Mohri et al., 2019). In practice, the true distributions are unknown and each client holds an empirical dataset $D^{(i)}$ of $n^{(i)}$ samples drawn i.i.d. from $\mathcal{T}^{(i)}$. We note $n_k^{(i)}$ the number of samples in $D^{(i)}$ belonging to group g_k such that $n = \sum_{i=1}^{m} n^{(i)}$ and $n_k = \sum_{i=1}^{m} n_k^{(i)}$. We use these empirical sample counts to approximate probabilities of belonging to an underlying distribution, that is $\mathbb{P}(\mathcal{T}^{(i)}) \approx \frac{n^{(i)}}{n}$ and $\mathbb{P}(\mathcal{T}^{(i)} | \mathcal{T}_k) \approx \frac{n_k^{(i)}}{n_k}$.

To learn a model, we consider empirical risk minimization approaches, and we denote by $L(h_\theta)$ the loss function that is to be minimized over $\mathcal{T}$. We further note $L_k(h_\theta)$ the loss over $\mathcal{T}_k$ and $L_k^{(i)}(h_\theta)$ that over $\mathcal{T}_k^{(i)}$. All of these are in practice estimated on batches of samples drawn from (subsets of) the empirical datasets.

Fairness Measure To measure fairness, we use a canonical form (Maheshwari and Perrot, 2023) that allows us to simultaneously reason about several definitions that were proposed in the literature:

$$F_k(h_\theta, \mathcal{T}) = C_k^0 + \sum_{k'=1}^{K} C_k^{k'} \mathbb{P}(h_\theta(x) \neq y | \mathcal{T}_{k'}) \tag{2}$$

where $C_k^{k'}$ are constants that depend on the fairness definition and can be estimated from empirical sample counts, and $\mathbb{P}(h_\theta(x) \neq y | \mathcal{T}_k)$ denotes the misclassification probability on samples from $\mathcal{T}_k$. A positive value of F_k means that the group g_k is advantaged, while a negative one implies that the group is disadvantaged. We refer to Maheshwari and Perrot (2023); Mangold et al. (2023) for explicit values of $C_k^{k'}$ for Accuracy Parity (Zafar et al., 2017), Demographic Parity (Calders et al., 2009), Equality of Opportunity and Equalized Odds (Hardt et al., 2016).

FairGrad for exact fairness Our algorithms extend FairGrad (Maheshwari and Perrot, 2023) to federated learning, hence we recall here its formulation in the centralized setting. It is an iterative fairness algorithm that learns adaptive sample weights based on whether they belong to an advantaged or disadvantaged group. To do so, FairGrad introduces a reweighted loss: $\mathcal{L}(h_\theta, w^t) = \sum_{k=1}^{K} w_k^t L_k(h_\theta)$ where $w^t = (w_1^t, \ldots, w_K^t)$ are group-wise weights that are defined and updated based on current fairness measures, using a group-fairness definition that follows the form previously introduced in Equation (2):

$$w_k^t = \mathbb{P}(\mathcal{T}_k) + \sum_{k'=1}^{K} C_k^{k'} \sum_{\tau=0}^{t-1} \eta F_{k'}(\theta^\tau, \mathcal{T}). \tag{3}$$

FairGrad uses stochastic optimization and an alternating approach, where at each step fairness is estimated, weights are updated, and finally model parameters are updated using the reweighted loss

$\mathcal{L}$ as $\theta^{t+1} = \theta_t - \gamma\nabla_\theta\mathcal{L}(h_\theta, w^t)$. In these formulas, γ is the usual learning rate for stochastic gradient descent (SGD), while η is a FairGrad-specific learning rate for the weights. At each step, the loss is estimated on a batch of samples. As for $\{F_k\}_{k=1}^K$, the estimates of the fairness of the current model, the authors suggest to compute them over the same training batch for computational efficiency, resulting in updating both weights and model parameters at every training step with a single forward pass.

3 FairGrad in Federated Learning

In this section, we first show how to evaluate group-fairness measures in a federated setting. Then, we propose two extensions of FairGrad (Maheshwari and Perrot, 2023). First, we introduce Fed-FairGrad a straightforward extension that we adapt to reduce its inflated communication cost. Second, we present Fed-FairGrad-Convex, a variant that aims at tackling utility issues encountered with Fed-FairGrad.

3.1 Evaluating Fairness in Federated Learning

Fairness functions that follow the canonical form in Equation (2) are straightforward to compute in a federated way. Indeed, using the law of total probabilities, we can rewrite the initial formula as:

$$F_k(\theta, \mathcal{T}) = C_k^0 + \sum_{k'=1}^{K} C_k^{k'} \mathbb{P}(h_\theta(x) \neq y | \mathcal{T}_{k'}) = C_k^0 + \sum_{k'=1}^{K} C_k^{k'} \left(\sum_{i=1}^{m} \mathbb{P}(\mathcal{T}_{k'}^{(i)} | \mathcal{T}_{k'}) \mathbb{P}(h_\theta(x) \neq y | \mathcal{T}_{k'}^{(i)}). \right) \tag{4}$$

In practice, this gives $F_k(\theta, \mathcal{T}) = C_k^0 + \sum_{k'=1}^{K} C_k^{k'} \left(\frac{1}{n_{k'}} \sum_{i=1}^{m} n_{k'}^{(i)} \mathbb{P}(h_\theta(x) \neq y | \mathcal{T}_{k'}^{(i)}) \right)$, since we assume $\mathbb{P}(\mathcal{T}_k^{(i)} | \mathcal{T}_k) \approx \frac{n_k^{(i)}}{n_k}$. It means that the clients only need to estimate and share the group-wise misclassification rate of the global model on their local data, weighted by local sample counts. This corresponds to K scalar values for each client, that are then aggregated by the server to estimate the fairness level of the global model on the overall distribution. This also requires the one-time sharing and aggregation of group-wise sample counts, so that the server can compute the $C_k^{k'}$ constants and access the n_k values.

3.2 Fed-FairGrad

In this section, we explain how to derive Fed-FairGrad from FairGrad at the expense of a large communication cost. We then consider practical issues and reduce this cost using standard techniques.

Theoretical update rules We use the total law of probabilities to decompose the reweighted loss function $\mathcal{L}$ and model parameters update rule of FairGrad into client-wise terms:

$$\mathcal{L}(h_\theta, w^t) = \sum_{k=1}^{K} w_k^t L_k(h_\theta) = \sum_{i=1}^{m} \sum_{k=1}^{K} \mathbb{P}(\mathcal{T}_k^{(i)} | \mathcal{T}_k) w_k^t L_k^{(i)}(h_\theta) = \sum_{i=1}^{m} \mathcal{L}(h_\theta, w^{(i),t}) \tag{5}$$

$$\theta^{t+1} = \theta_t - \gamma\nabla_\theta\mathcal{L}(h_\theta, w^t) = \theta_t - \gamma \sum_{i=1}^{m} \nabla_\theta\mathcal{L}(h_\theta, w^{(i),t}) \tag{6}$$

where $w^{(i),t} := (w_1^t \mathbb{P}(\mathcal{T}_1^{(i)} | \mathcal{T}_1), \dots, w_K^t \mathbb{P}(\mathcal{T}_K^{(i)} | \mathcal{T}_K))$ are client-adjusted weights. In practice, given our assumptions, $w_k^t \mathbb{P}(\mathcal{T}_k^{(i)} | \mathcal{T}_k) \approx w_k^t \frac{n_k^{(i)}}{n_k}$. Hence, having received $\frac{w_k^t}{n_k}$ for all groups (K scalars) from the server, the clients refine them based on their sample counts, run a SGD step using the reweighted loss, and finally send back the resulting gradient, similar to what is done in fairness-agnostic federated learning. The update rule for the weights remains the same as in Equation (3), but applying it requires that, before each local gradient step, the clients evaluate and share their local misclassification rates,

Algorithm 1 Fed-FairGrad

Require: m clients, an initial model h_θ, a learning rate γ, a fairness rate η and a fairness definition F.

▷ Initialize fairness functions and constraints:
    Clients share their group-wise sample counts $\{n_k^{(i)}\}_{k=1}^K$.
    Server aggregates all n_k counts, calculates $C_k^{k'}$ constants, and initializes w_k^0 values.
for each round **do**
    ▷ Estimate fairness and update weights:
        Clients compute and share group-wise misclassification rates on their training data.
        Server computes $\{F_k\}_{k=1}^K$ estimates as per Equation (4).
        Server updates loss weights w^t as per Equation (3) and shares them with clients.
    ▷ Update model parameters:
        Clients run local SGD steps as per Equation (7) and send back the resulting $\theta^{(i),t+1}$.
        Server aggregates updates into θ^{t+1} as per Equation (8) and shares it with clients.

so that the server can estimate the fairness of the model as explained in Section 3.1, introducing some extra communication and synchronization costs compared with fairness-agnostic federated learning.

This adaptation of FairGrad is straightforward mathematically, but costly. Indeed, its strict application requires two rounds of communication per gradient step: one to update fairness estimates and share the new weights, the other to aggregate the updated model parameters.

Reducing the communication cost A first possible improvement, similar to what is done in FairFed (Ezzeldin et al., 2023), would consist in computing parameter updates using weights derived from fairness measures that are one-step-behind (that is, replacing w_k^t with w_k^{t-1} in Equation (6)). This reduces the number of exchanges per round as the parameter and fairness updates can then be conducted at once, with a single synchronisation step instead of two. However, it would not affect the amount of information shared. In particular, it would not address the main communication bottleneck, which is that model parameters need to be shared back and forth after every single local gradient step.

To alleviate this issue, a standard approach is to have the clients run multiple local gradient steps between each synchronization step. This was first proposed in FedAvg (McMahan et al., 2017), and has since been shown to be grounded theoretically (Khaled et al., 2020; Mishchenko et al., 2022). In our case, this involves that, at each round, clients iteratively update local model parameters using fixed weights $w_k^t \mathbb{P}(\mathcal{T}_k^{(i)} | \mathcal{T}_k)$ based on the estimated fairness level of initial parameters θ^t. Formally, with B the number of local SGD steps, τ a local time index, and $\theta^{(i),t,0} := \theta^t$, we have:

$$\theta^{(i),t,\tau+1} = \theta^{(i),t,\tau} - \gamma \nabla_\theta \mathcal{L}\left(h_{\theta^{(i),t,\tau}}, w^{(i),t}\right) \ , \forall \tau \in \{0, \ldots, B-1\}. \tag{7}$$

Clients then send the resulting parameters $\theta^{(i),t+1} := \theta^{(i),t,B}$ to the server, which sum-aggregates them:

$$\theta^{t+1} = \theta^t - \sum_{i=1}^{m} (\theta^t - \theta^{(i),t+1}) \tag{8}$$

In this case, the fairness of the model and the weights are only updated once per training round and are then used for all the gradient steps in that round. As a consequence, we propose computing them based on the entire training datasets of clients without the one-step-behind trick to produce more robust estimates. This is a major difference with the centralized version of FairGrad, where fairness is estimated in a less robust fashion (based on a single training batch) but way more frequently (at each training step). We summarize the whole Fed-FairGrad approach in Algorithm 1.

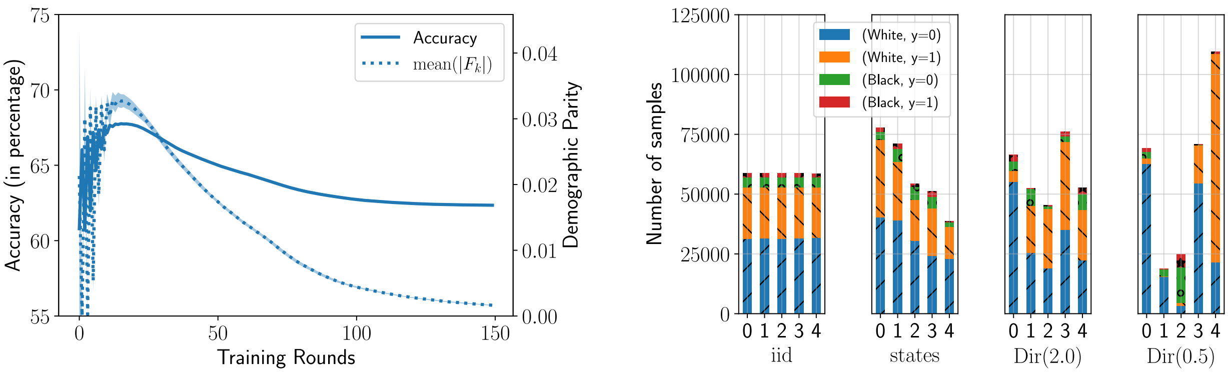

Figure 1: (Left) Example of Fed-FairGrad performance metrics. (Right) Training samples partitions. (Left) The model was trained using Fed-FairGrad ($\eta = 0.1$) for Demographic Parity on i.i.d.-partitioned data (see Section 4 for details). The metrics were computed on validation data at the end of each round and only the first 150 out of 500 rounds are included for readability. The results are averaged across 20 replicas, with the mostly negligible standard deviations rendered as surfaces around the lines. (Right) The height of the bars indicates the number of samples in a local training set. Bars are made of stacked sub-bars indicating sensitive groups, as defined by ethnicity and target labels.

Reduced utility Our experiments (detailed in Section 4) show that Fed-FairGrad tends to learn models that are fair but inaccurate. More precisely, their utility and fairness oscillates for a few iterations before dropping to levels that are close to a majority-label-predicting baseline as shown in Figure 1 (Left). We hypothesize that this is due to the fact that the weights in FairGrad can be negative, a necessary property to enforce fairness. Indeed, we believe that this creates cases where the local weighted losses become non-convex and thus may steer the local updates in vastly different directions from one training round to the other. This creates the oscillations and pushes the weights towards a solution where this does not happen: all weights are 0 except for the majority label groups. The learned model is then almost constant. To tackle this issue, we present, in the following subsection, an alternative algorithm that builds upon Fed-FairGrad by modifying its loss function to ensure that the local losses remain convex and thus that the local updates steer the updates towards similar directions from one training round to the other.

3.3 Fed-FairGrad-Convex: a convex alternative to Fed-FairGrad

In this section, we introduce Fed-FairGrad-Convex, an alternative that follows the same logic as Fed-FairGrad, updating parameters based on a reweighted loss that reflects the current group fairness measures, but is designed so that the local weighted losses are convex for a fixed set of weights. To do so, we replace the weighted loss in Equation (5) with a new loss inspired by cost-sensitive fairness approaches (Agarwal et al., 2018) that affect different cost-functions to different sensitive groups:

$$\mathcal{L}(h_\theta, w^t) = \sum_{k=1}^{K} \left(L_k(h_\theta) \frac{|w_k^t| + w_k^t}{2||w^t||_2} + L_k(-h_\theta) \frac{|w_k^t| - w_k^t}{2||w^t||_2} \right) \tag{9}$$

where L is the raw loss function, which we assume to be convex, and $w^t = (w_1^t, \ldots, w_K^t) \in \mathbb{R}^K$ are group-wise weights. Here, $-h_\theta$ denotes a model that predicts opposite labels, meaning $-h_\theta(x) = 1 - h_\theta(x)$ (as $h_\theta(x) \in [0, 1]$). Hence, if w_k is negative for an unfairly advantaged group, the loss is computed using opposite labels, so that the performance on that group is reduced. On the opposite, if w_k is positive, performance on that group improves with training. Apart from these changes in terms of loss, the decomposition across clients is similar to Fed-FairGrad, with client-adjusted weights $w_k^{(i),t} \approx w_k^t \frac{n_k^{(i)}}{n_k}$.

Regarding the weights, they are initialized as $w_k^0 = \frac{1}{\sqrt{K}}$ and updated as follows:

$$w_k^{t+1} = w_k^t - \eta F_k(\theta^t, \mathcal{T}) \quad \forall k \in \{1, \ldots, K\} \tag{10}$$

The resulting algorithm is almost identical to Algorithm 1, with the only differences being that the weights are updated using Equation (10) and the loss considered is the one in Equation (9). The computation and communication costs remain unchanged.

4 Experiments

In this section, we empirically evaluate the performance of Fed-FairGrad and Fed-FairGrad-Convex. We compare them to 3 baselines on Folktables (Ding et al., 2021), a well-known fairness dataset.

Methods We compare Fed-FairGrad (FedFG) and Fed-FairGrad-Convex (FedFG-C) with three baselines. FedAvg McMahan et al. (2017) is a standard federated learning approach that does not consider fairness constraints. FairFed (Ezzeldin et al., 2023) calculates adaptive aggregation weights based on the differences between local and global fairness levels. FedFB (Zeng et al., 2021) is a federated extension of FairBatch Roh et al. (2021) that calculates minibatch sampling probabilities based on fairness. We use linear models with sigmoid activation and cross-entropy loss. In Appendix A, we provide details on hyper-parameters, that were selected based on preliminary experiments and recommendations from the original papers.

Dataset We conduct our experiments on Folktables (Ding et al., 2021), a US census dataset where the goal is to predict the income of individuals ($\leq 50k$ or $\geq 50k$ dollars yearly). We split our data in three subsets (60 % for training, 20 % for validation and 20 % for testing). To simulate a federated context, we partition the training data among five clients using different strategies to create various degrees of heterogeneity: i.i.d. splits, states-based splits that reflect some heterogeneity due to real-life sociohistorical factors, and artificial heterogeneity based on the Dirichlet distribution (Yurochkin et al., 2019) as Dir(2.0) and Dir(0.5). The resulting sample counts are represented in Figure 1 (Right). The pre-processing and data partitioning details are provided in Appendix A.

Randomness and repetitions We consider 16 different settings resulting from the combination of our 4 dataset partitions and 4 fairness metrics. We run 20 replicas per setting and algorithm to measure the sensitivity of the results to training-time randomness factors. These affect the choice of initial model parameters, and shuffling of the local datasets of each client at the start of each round. They are controlled by seeding pseudo-random number generators, with an arbitrary base seed (20231127) that is shared across settings, and incremented when passing to the next replica of the same setting. Note that the data partition itself does not change across replicas as the study of outside-of-training randomness factors is left for future work.

Results We present representative results here and defer the others, that follow similar trends, to Appendix C, along with an expanded analysis in Appendix B. Figure 2 illustrates results for Demographic Parity with i.i.d. partition (Left) and for Equalized Odds with Dir(0.5) partition (Right).

In all our experiments, FedAvg sets a baseline for quick convergence towards relatively accurate but mostly unfair models. It becomes less accurate in more heterogeneous settings, in line with previous studies (Karimireddy et al., 2020). Second, FedFair and FedFB obtain disparate results; they either learn unfair but accurate models or learn fair but very inaccurate predictors, close to the constant model. For instance, FairFed tends to learn unfair models on the i.i.d. partition and fair models on the more heterogeneous ones. It is probably due to the fact that it operates based on fairness discrepancies between clients, which are bound to be negligible with lower levels of heterogeneity. The more erratic

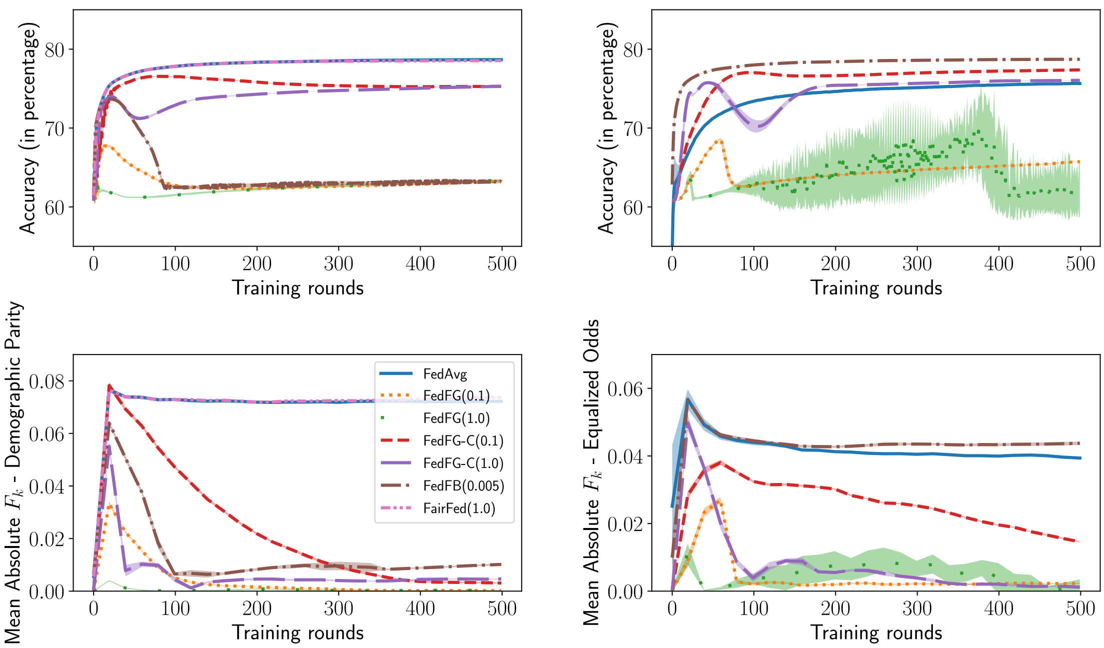

Figure 2: Settings: (Left) Demographic Parity with i.i.d. partition; (Right) Equalized Odds with Dir(0.5) partition. Each plot represents the evolution throughout training of a given metric. Each line represents the average of models trained with a given algorithm, with a surface around it denoting the standard deviation across replicas. Accuracy is computed at the end of each round, while fairness is computed at the end of rounds 1 and $\{20, 40, \ldots, 500\}$. Both are computed on the validation subsets. For reference, a model predicting the majority label would achieve 60.79 % accuracy and 0.0 mean $|F_k|$.

behaviour of FedFB remains unexplained and requires more investigation. Third, FedFG reliably learns fair models but tends to converge to very inaccurate solutions. As mentioned in Section 3.2, this is probably related to the non-convexity of the local losses. Finally, FedFG-C is the only approach that consistently manages to learn fair models with a limited decrease of utility as compared with FedAvg.

5 Conclusion And Future Work

In this work, we addressed the problem of group fairness in federated learning. We proposed two algorithms, Fed-FairGrad and Fed-FairGrad-Convex. Both methods use a reweighting scheme to achieve a better balance between the sensitive groups. Empirically, Fed-FairGrad learns fair but inaccurate models while Fed-FairGrad-Convex remains more accurate while remaining fair.

The preliminary results presented in this paper are encouraging and we envision several future works. First, we want to evaluate our methods in a wider range of settings, for example taking into account the outside-of-training randomness, considering more datasets, more complex model architectures or other heterogeneity scenarios. Second, we want to further integrate our implementation to the DecLearn open-source framework for federated learning, to provide reproducible baselines and an extendable playground for fair federated learning solutions. Third, our algorithms are compatible with Secure Aggregation (Bonawitz et al., 2016), hence we would like to implement it. Fourth, we would like to further study the trade-offs between communication and computation costs, for example by considering other communication reduction techniques, such as partial client participation or gradient compression (Wang and Ji, 2022; Haddadpour et al., 2021). Finally, longer-term perspectives include studying asynchronous training (Sprague et al., 2018), where client information is not necessarily up-to-date, and handling malicious clients that may share untrustworthy information (Blanchard et al., 2017).

References

Abay, A., Zhou, Y., Baracaldo, N., Rajamoni, S., Chuba, E., and Ludwig, H. (2020). Mitigating bias in federated learning. *arXiv preprint arXiv 2012.02447*.

Agarwal, A., Beygelzimer, A., Dudik, M., Langford, J., and Wallach, H. (2018). A reductions approach to fair classification. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR.

Barocas, S., Hardt, M., and Narayanan, A. (2023). *Fairness and machine learning: Limitations and opportunities*. MIT Press.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30.

Bonawitz, K. A., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2016). Practical secure aggregation for federated learning on user-held data. In *NIPS Workshop on Private Multi-Party Machine Learning*.

Calders, T., Kamiran, F., and Pechenizkiy, M. (2009). Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18.

Calmon, F., Wei, D., Vinzamuri, B., Natesan Ramamurthy, K., and Varshney, K. R. (2017). Optimized pre-processing for discrimination prevention. *Advances in neural information processing systems*, 30.

Caton, S. and Haas, C. (2023). Fairness in machine learning: A survey. *ACM Comput. Surv.* Just Accepted.

Celis, L. E., Huang, L., Keswani, V., and Vishnoi, N. K. (2019). Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 319–328.

Chouldechova, A. and Roth, A. (2020). A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89.

Cotter, A., Jiang, H., Gupta, M., Wang, S., Narayan, T., You, S., and Sridharan, K. (2019). Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. *Journal of Machine Learning Research*, 20(172):1–59.

De-Arteaga, M., Romanov, A., Wallach, H., Chayes, J., Borgs, C., Chouldechova, A., Geyik, S., Kenthapadi, K., and Kalai, A. T. (2019). Bias in bios: A case study of semantic representation bias in a high-stakes setting. In *proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 120–128.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. (2021). Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490.

Du, W., Xu, D., Wu, X., and Tong, H. (2021). Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 181–189. SIAM.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226.

Dwork, C. and Ilvento, C. (2019). Fairness under composition. *10th Innovations in Theoretical Computer Science*.

Ezzeldin, Y. H., Yan, S., He, C., Ferrara, E., and Avestimehr, A. S. (2023). Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7494–7502.

Gálvez, B. R., Granqvist, F., van Dalen, R., and Seigel, M. (2021). Enforcing fairness in private federated learning via the modified method of differential multipliers. In *NeurIPS 2021 Workshop Privacy in Machine Learning*.

Haddadpour, F., Kamani, M. M., Mokhtari, A., and Mahdavi, M. (2021). Federated learning with compression: Unified analysis and sharp guarantees. In *International Conference on Artificial Intelligence and Statistics*, pages 2350–2358. PMLR.

Hardt, M., Price, E., and Srebro, N. (2016). Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Hébert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. (2018). Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al. (2021). Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210.

Kamiran, F. and Calders, T. (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR.

Khaled, A., Mishchenko, K., and Richtarik, P. (2020). Tighter theory for local sgd on identical and heterogeneous data. In Chiappa, S. and Calandra, R., editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4519–4529. PMLR.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. (2019). Fair resource allocation in federated learning. In *International Conference on Learning Representations*.

Lohaus, M., Perrot, M., and Luxburg, U. V. (2020). Too relaxed to be fair. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6360–6369. PMLR.

Maheshwari, G. and Perrot, M. (2023). Fairgrad: Fairness aware gradient descent. *Transactions on Machine Learning Research*.

Mangold, P., Perrot, M., Bellet, A., and Tommasi, M. (2023). Differential privacy has bounded impact on fairness in classification. In *International Conference on Machine Learning*, pages 23681–23705. PMLR.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Mishchenko, K., Malinovsky, G., Stich, S., and Richtarik, P. (2022). ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15750–15769. PMLR.

Mohri, M., Sivek, G., and Suresh, A. T. (2019). Agnostic federated learning. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR.

Padala, M., Damle, S., and Gujar, S. (2021). Federated learning meets fairness and differential privacy. In *International Conference on Neural Information Processing*, pages 692–699. Springer.

Papadaki, A., Martinez, N., Bertran, M., Sapiro, G., and Rodrigues, M. (2022). Minimax demographic group fairness in federated learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 142–159, New York, NY, USA. Association for Computing Machinery.

Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. (2017). On fairness and calibration. *Advances in neural information processing systems*, 30.

Roh, Y., Lee, K., Whang, S. E., and Suh, C. (2021). Fairbatch: Batch selection for model fairness. In *International Conference on Learning Representations*.

Sprague, M. R., Jalalirad, A., Scavuzzo, M., Capota, C., Neun, M., Do, L., and Kopp, M. (2018). Asynchronous federated learning for geospatial applications. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 21–28. Springer.

Su, C., Yu, G., Wang, J., Li, H., Li, Q., and Yu, H. (2024). Multi-dimensional fair federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15083–15090.

Wang, G., Payani, A., Lee, M., and Kompella, R. (2023). Mitigating group bias in federated learning: Beyond local fairness.

Wang, S. and Ji, M. (2022). A unified analysis of federated learning with arbitrary client participation. *Advances in Neural Information Processing Systems*, 35:19124–19137.

Yue, X., Nouiehed, M., and Al Kontar, R. (2023). Gifair-fl: A framework for group and individual fairness in federated learning. *INFORMS Journal on Data Science*, 2(1):10–23.

Yurochkin, M., Agarwal, M., Ghosh, S., Greenewald, K., Hoang, N., and Khazaeni, Y. (2019). Bayesian nonparametric federated learning of neural networks. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR.

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. (2017). Fairness beyond disparate treatment and disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17. International World Wide Web Conferences Steering Committee.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. (2013). Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR.

Zeng, Y., Chen, H., and Lee, K. (2021). Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*.

Zhang, D. Y., Kou, Z., and Wang, D. (2020). Fairfl: A fair federated learning approach to reducing demographic bias in privacy-sensitive classification models. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 1051–1060. IEEE.

A Details on the experimental setting

In this appendix section, we provide with additional details as to the experimental setting, including dataset pre-processing, data partition schemes and hyper-parameter choices for compared algorithms.

Dataset pre-processing We conduct our experiments on Folktables (Ding et al., 2021), a US census dataset containing explanatory and dependent variables such as income or employment. We use data from the year 2018, restricted to the 5 highest-population states: California, Texas, Florida, New-York and Pennsylvania. All the experiments were conducted to predict the income of individuals (either $\leq 50k$ or $\geq 50k$ dollars yearly) using a linear model with sigmoid activation and binary cross-entropy loss. We use 9 raw features, out of which 3 are continuous, 1 is binary, and 6 are categorical. We apply min-max normalization to continuous features, and encode categorical ones into sets of dummy variables, sometimes using high-level groups from the US census technical documentation. In the end, 66 pre-processed features are used as predictors. The sensitive attribute is ethnicity. We select recordings belonging to white and black individuals and drop the rest (Zafar et al., 2017) to make the attribute binary, both for simplicity and to enable FairFed.

Dataset partitions We split our data in three subsets. Hence, we sampled uniformly from each State's data to attribute 60 % of samples for training, 20 % for validation and 20 % for testing. To simulate a federated context, we further partition the training data among five clients using different strategies to create heterogeneity. First, the data is shuffled and split evenly, resulting in a *i.i.d.* setting. Second, the five *states*' datasets are attributed to clients, resulting in some heterogeneity due to real-life sociohistorical factors. Finally, some artificial heterogeneity is introduced using the Dirichlet distribution Yurochkin et al. (2019): for each sensitive group, the proportion of its samples attributed to each client is sampled from Dir(α). We use either Dir(2.0) or Dir(0.5), the latter being more heterogeneous than the former. The resulting subset-wise and client-wise sample counts are represented in Figure 3.

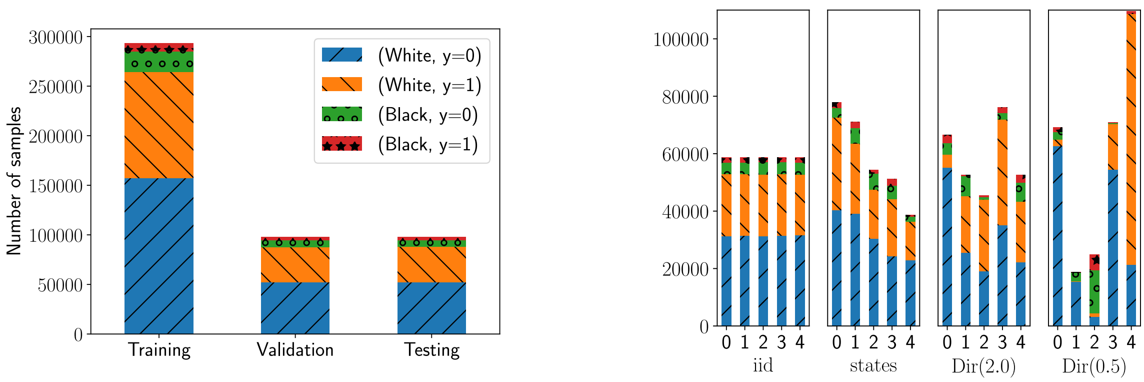

Figure 3: Distribution of samples across subsets (Left) and clients (Right). The height of the bars indicates the total number of samples in a global subset or local training sets. Bars are made of stacked sub-bars indicating sensitive groups, as defined by ethnicity and target labels.

Hyper-parameters We set a number of common hyper-parameters for all our experiments based on preliminary results. We run 500 rounds of federated training. During each round, clients first randomly shuffle their training data, then run a full epoch of training with 512-samples batches. They use 0.05 as learning rate, and clip gradients so that their global L2-norm cannot exceed 0.05.

Apart from FedAvg, each algorithm has a differently-named hyper-parameter that impacts the update rule associated with fairness constraints, acting as a sort of learning rate. For FairFed, the β parameter impacts the update rule for client-updates-averaging weights; we set it to $\beta = 1.0$ as advised in the original paper. For FedFB, the α parameter impacts the update rule for group-wise sampling weights; we set it to $\alpha = 0.005$ based on preliminary results. For Fed-FG and FedFG-C, the η parameter is the learning rate for updating the fairness weights; we set it either to $\eta = 0.1$ or $\eta = 1.0$. The former would be our choice in the centralized setting, whereas we hypothesized the latter to be more suitable for the federated setting due to weights being updated only once per epoch.

A precision on settings Some baselines are only applicable to a subset of fairness definitions. FairFed only applies to Demographic Parity and Equality of Opportunity, while FedFB applies to all of our definitions but Accuracy Parity. As for FedAvg, since it is agnostic to fairness, we only run it once per data partition (with 20 replicas) and evaluate the resulting models' fairness by all definitions.

Implementation Our experiments were implemented using the DecLearn open-source Python package for federated learning. Both our contributed algorithms and baselines from the literature were implemented on a dedicated software branch. Federated learning was simulated, running concurrent routines for the clients and server with network communications on the localhost.

B Expanded commentary of results

In this appendix section, we provide a more detailed overview of our experimental results than that exposed in Section 4.

FedAvg sets a baseline for quick convergence towards a relativelyaccurate unfair model. In most settings, it achieves more than 78 % accuracy and 71 % precision and recall, with negligible variance. Metrics are slightly lower on the Dir(0.5) partition, which is expected given client heterogeneity (Karimireddy et al., 2020). The resulting models are unfair by all definitions, save for Accuracy Parity.

FedFG-C is the only algorithm that achieves fairness in all settings, with a limited decrease of utility as compared with FedAvg. It does so using $\eta = 1.0$, usually converging in less than 150 rounds. In 14 out of 16 settings, it achieves more than 75 % accuracy and more than 67 % precision and recall, with negligible variance. With $\eta = 0.1$, models are much slower to converge, sometimes not being done after 500 rounds, resulting in a less fair (albeit sometimes more accurate) model than when using $\eta = 1.0$. We note that accuracy usually stabilizes early in the training process, and appears to be sparingly degraded as fairness is being optimized. We observe a similar pattern for Fed-FairGrad, that starts with optimizing utility (although to a lower point) prior to enforcing fairness (albeit in an utility-degrading manner).

Most other algorithms converge to solutions that are fair but close to a constant-prediction model. As exposed at the end of Section 3.2, this is the case for FedFG, which save for accuracy parity settings achieves at most 66 % accuracy, with around 90 % precision and below 15 % recall. This is better than a constant-prediction baseline, but still far from the utility of FedAvg. Indeed, a model that always predicts the majority label $y = 0$ would achieve perfect fairness ($\forall k,\ F_k = 0$) as per all definitions but accuracy parity, 60.79 % accuracy and 0 % precision and recall. For Demographic

Parity, FedFB also quickly converges to a model with less that 64 % accuracy and more than 93 % precision, which is fairer than FedAvg baselines but not as fair as models obtained using Fed-FG and Fed-FG-C. Additional experiments on the i.i.d. partition with distinct values of the α hyper-parameter (0.001, 0.0025, 0.01, 0.05, 0.1) all converged to similar results, save for some being fairer than others. On non-i.i.d. partitions, FairFed, which is only defined for Demographic Parity and Equality of Opportunity, mostly converges to constant-prediction models, outputting either the majority or minority label depending on settings.

Some baseline algorithms fail to enforce fairness altogether. FairFed does so on the i.i.d. partition, producing nearly the same results as FedAvg. This is explained by the fact that it operates based on fairness discrepancies between clients, which are bound to be negligible in the i.i.d. setting. This may also explain why FairFed exposes higher variance than other algorithms and sometimes fails to enforce fairness in heterogeneous settings without a clear pattern to it. FedFB fails to enforce Equality of Opportunity or Equalized Odds fairness, converging to a model that is similar to FedAvg in accuracy, but as bad or even worse as to fairness. This remains to be explained. On the Dir(0.5) partition, FedFB notably achieves better utility than any other method, but worse fairness than FedAvg. This may be be explained by the reweighting of the loss by sensitive group making the problem homogeneous across clients, since they are only made heterogeneous as to these sensitive groups' distribution.

C Additional Results

In this appendix section, we provide exhaustive results of our experiments. For each setting, defined by a data partition and a fairness definition, we present a table with metrics from the models resulting from the compared algorithms after the full 500 rounds of training. All metrics are computed on the validation dataset. Accuracy, precision and recall are presented as percentages with 10^{-2} precision; the higher the more useful the model, provided precision and recall are somewhat balanced. The average, minimum and maximum values of F_k fairness measures are provided with 10^{-4} precision; the lower the better. Values are averaged across 20 replicas. We omit standard deviations for readability, noting that they are almost always negligible. In each table, the first line indicates results for a model that always predicts the majority label, which are the same across settings.

C.1 Results for Accuracy Parity

Table 1: Results for Accuracy Parity fairness on i.i.d. partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k \rvert$-avg ↓	$\lvert F_k \rvert$-min ↓	$\lvert F_k \rvert$-max ↓
Constant	60.79	0.00	0.00	0.0676	0.0135	0.1218
FedAvg	78.68	73.52	71.30	0.0075	0.0015	0.0135
FedFG($\eta = 0.1$)	71.97	70.99	64.56	0.0196	0.0039	0.0353
FedFG($\eta = 1.0$)	72.71	60.10	90.44	0.0154	0.0031	0.0278
FedFG-C($\eta = 0.1$)	78.57	73.18	71.56	0.0076	0.0015	0.0137
FedFG-C($\eta = 1.0$)	76.82	71.85	70.90	0.0079	0.0016	0.0142

Table 2: Results for Accuracy Parity fairness on states partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0676	0.0135	0.1218
FedAvg	78.63	73.34	71.50	0.0069	0.0014	0.0124
FedFG($\eta = 0.1$)	72.73	70.43	67.06	0.0181	0.0036	0.0327
FedFG($\eta = 1.0$)	72.44	59.78	90.75	0.0163	0.0032	0.0294
FedFG-C($\eta = 0.1$)	78.58	72.88	72.25	0.0074	0.0015	0.0133
FedFG-C($\eta = 1.0$)	76.02	69.72	74.48	0.0095	0.0019	0.0171

Table 3: Results for Accuracy Parity fairness on Dir(2.0) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0676	0.0135	0.1218
FedAvg	78.52	72.11	73.73	0.0065	0.0013	0.0118
FedFG($\eta = 0.1$)	78.39	72.86	71.51	0.0039	0.0008	0.0070
FedFG($\eta = 1.0$)	78.36	73.61	69.82	0.0032	0.0006	0.0058
FedFG-C($\eta = 0.1$)	78.31	70.44	76.96	0.0043	0.0009	0.0078
FedFG-C($\eta = 1.0$)	78.20	69.84	78.16	0.0027	0.0005	0.0049

Table 4: Results for Accuracy Parity fairness on Dir(0.5) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0676	0.0135	0.1218
FedAvg	75.65	64.10	86.09	0.0038	0.0008	0.0068
FedFG($\eta = 0.1$)	65.95	86.70	15.62	0.0392	0.0078	0.0706
FedFG($\eta = 1.0$)	65.50	92.76	13.03	0.0493	0.0098	0.0888
FedFG-C($\eta = 0.1$)	77.83	75.11	64.98	0.0057	0.0011	0.0103
FedFG-C($\eta = 1.0$)	77.80	74.57	65.83	0.0046	0.0009	0.0083

C.2 Results for Demographic Parity

Table 5: Results for Demographic Parity fairness on i.i.d. partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.68	73.52	71.30	0.0722	0.0144	0.1300
FairFed($\beta = 1.0$)	78.55	73.61	70.61	0.0737	0.0147	0.1328
FedFB($\alpha = 0.005$)	63.17	93.69	6.51	0.0102	0.0020	0.0183
FedFG($\eta = 0.1$)	63.27	90.35	7.07	0.0002	0.0000	0.0003
FedFG($\eta = 1.0$)	63.37	90.19	7.39	0.0003	0.0001	0.0005
FedFG-C($\eta = 0.1$)	75.28	69.03	67.03	0.0031	0.0006	0.0056
FedFG-C($\eta = 1.0$)	75.28	69.22	66.54	0.0047	0.0009	0.0084

Table 6: Results for Demographic Parity fairness on states partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.63	73.34	71.50	0.0727	0.0145	0.1309
FairFed($\beta = 1.0$)	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedFB($\alpha = 0.005$)	63.36	93.55	7.02	0.0112	0.0022	0.0201
FedFG($\eta = 0.1$)	63.59	90.19	8.01	0.0003	0.0001	0.0006
FedFG($\eta = 1.0$)	63.74	90.04	8.46	0.0003	0.0001	0.0006
FedFG-C($\eta = 0.1$)	75.31	67.85	70.34	0.0038	0.0008	0.0069
FedFG-C($\eta = 1.0$)	75.24	67.97	69.70	0.0044	0.0009	0.0079

Table 7: Results for Demographic Parity fairness on Dir(2.0) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.52	72.11	73.73	0.0748	0.0149	0.1348
FairFed($\beta = 1.0$)	39.21	39.21	100.00	0.0000	0.0000	0.0000
FedFB($\alpha = 0.005$)	63.21	93.51	6.62	0.0104	0.0021	0.0187
FedFG($\eta = 0.1$)	64.34	89.06	10.31	0.0002	0.0000	0.0004
FedFG($\eta = 1.0$)	64.53	88.88	10.89	0.0003	0.0001	0.0005
FedFG-C($\eta = 0.1$)	75.09	66.49	73.53	0.0047	0.0009	0.0085
FedFG-C($\eta = 1.0$)	75.10	66.62	73.15	0.0064	0.0013	0.0116

Table 8: Results for Demographic Parity fairness on Dir(0.5) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\|F_k\|$-avg ↓	$\|F_k\|$-min ↓	$\|F_k\|$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	75.65	64.10	86.09	0.0723	0.0144	0.1302
FairFed($\beta = 1.0$)	39.26	39.22	99.86	0.0056	0.0011	0.0101
FedFB($\alpha = 0.005$)	64.05	93.29	8.95	0.0129	0.0026	0.0233
FedFG($\eta = 0.1$)	65.52	87.92	13.98	0.0003	0.0001	0.0005
FedFG($\eta = 1.0$)	65.93	87.70	15.26	0.0010	0.0002	0.0017
FedFG-C($\eta = 0.1$)	74.81	78.01	49.78	0.0146	0.0029	0.0264
FedFG-C($\eta = 1.0$)	72.24	79.19	39.61	0.0030	0.0006	0.0054

C.3 Results for Equality of Opportunity

Table 9: Results for Equality of Opportunity fairness on i.i.d. partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k \rvert$-avg ↓	$\lvert F_k \rvert$-min ↓	$\lvert F_k \rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.68	73.52	71.30	0.0321	0.0000	0.1196
FairFed($\beta = 1.0$)	78.09	73.19	69.63	0.0385	0.0000	0.1436
FedFB($\alpha = 0.005$)	78.59	73.56	70.85	0.0414	0.0000	0.1541
FedFG($\eta = 0.1$)	64.01	90.64	9.16	0.0004	0.0000	0.0014
FedFG($\eta = 1.0$)	64.01	90.61	9.17	0.0003	0.0000	0.0009
FedFG-C($\eta = 0.1$)	77.28	72.17	68.44	0.0081	0.0000	0.0303
FedFG-C($\eta = 1.0$)	75.83	71.14	64.51	0.0003	0.0000	0.0011

Table 10: Results for Equality of Opportunity fairness on states partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k \rvert$-avg ↓	$\lvert F_k \rvert$-min ↓	$\lvert F_k \rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.63	73.34	71.50	0.0328	0.0000	0.1223
FairFed($\beta = 1.0$)	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedFB($\alpha = 0.005$)	78.57	73.48	70.96	0.0426	0.0000	0.1588
FedFG($\eta = 0.1$)	64.33	90.37	10.11	0.0007	0.0000	0.0026
FedFG($\eta = 1.0$)	64.34	90.28	10.14	0.0005	0.0000	0.0018
FedFG-C($\eta = 0.1$)	77.28	71.22	70.56	0.0099	0.0000	0.0368
FedFG-C($\eta = 1.0$)	75.90	70.14	67.11	0.0011	0.0000	0.0040

Table 11: Results for Equality of Opportunity fairness on Dir(2.0) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k \rvert$-avg ↓	$\lvert F_k \rvert$-min ↓	$\lvert F_k \rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.52	72.11	73.73	0.0324	0.0000	0.1205
FairFed($\beta = 1.0$)	69.00	88.03	24.23	0.0089	0.0000	0.0333
FedFB($\alpha = 0.005$)	78.61	73.56	70.94	0.0423	0.0000	0.1575
FedFG($\eta = 0.1$)	64.81	90.24	11.48	0.0006	0.0000	0.0024
FedFG($\eta = 1.0$)	64.81	90.16	11.49	0.0006	0.0000	0.0021
FedFG-C($\eta = 0.1$)	77.23	70.44	72.23	0.0092	0.0000	0.0342
FedFG-C($\eta = 1.0$)	75.94	69.28	69.43	0.0015	0.0000	0.0057

Table 12: Results for Equality of Opportunity fairness on Dir(0.5) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k\rvert$-avg ↓	$\lvert F_k\rvert$-min ↓	$\lvert F_k\rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	75.65	64.10	86.09	0.0243	0.0000	0.0907
FairFed($\beta = 1.0$)	70.68	61.74	85.01	0.0163	0.0000	0.0607
FedFB($\alpha = 0.005$)	78.70	73.32	71.80	0.0367	0.0000	0.1368
FedFG($\eta = 0.1$)	65.25	90.31	12.72	0.0001	0.0000	0.0003
FedFG($\eta = 1.0$)	65.74	89.06	14.45	0.0061	0.0000	0.0228
FedFG-C($\eta = 0.1$)	76.54	66.10	82.48	0.0160	0.0000	0.0597
FedFG-C($\eta = 1.0$)	76.12	69.84	68.79	0.0016	0.0000	0.0060

C.4 Results for Equalized Odds

Table 13: Results for Equalized Odds fairness on i.i.d. partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k\rvert$-avg ↓	$\lvert F_k\rvert$-min ↓	$\lvert F_k\rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.68	73.52	71.30	0.0442	0.0058	0.1196
FedFB($\alpha = 0.005$)	78.65	73.85	70.53	0.0423	0.0052	0.1171
FedFG($\eta = 0.1$)	64.14	90.56	9.55	0.0022	0.0003	0.0043
FedFG($\eta = 1.0$)	56.96	13.05	19.98	0.0003	0.0000	0.0007
FedFG-C($\eta = 0.1$)	77.25	72.33	67.99	0.0103	0.0015	0.0268
FedFG-C($\eta = 1.0$)	75.67	71.48	63.16	0.0017	0.0002	0.0038

Table 14: Results for Equalized Odds fairness on states partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k\rvert$-avg ↓	$\lvert F_k\rvert$-min ↓	$\lvert F_k\rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.63	73.34	71.50	0.0450	0.0058	0.1223
FedFB($\alpha = 0.005$)	78.62	73.68	70.72	0.0432	0.0053	0.1198
FedFG($\eta = 0.1$)	64.48	90.38	10.54	0.0020	0.0002	0.0048
FedFG($\eta = 1.0$)	53.24	13.72	35.00	0.0000	0.0000	0.0000
FedFG-C($\eta = 0.1$)	77.29	71.37	70.27	0.0107	0.0013	0.0296
FedFG-C($\eta = 1.0$)	75.83	70.83	65.23	0.0014	0.0001	0.0033

Table 15: Results for Equalized Odds fairness on Dir(2.0) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k\rvert$-avg ↓	$\lvert F_k\rvert$-min ↓	$\lvert F_k\rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	78.52	72.11	73.73	0.0459	0.0065	0.1205
FedFB($\alpha = 0.005$)	78.69	73.85	70.67	0.0431	0.0053	0.1193
FedFG($\eta = 0.1$)	64.96	89.92	11.97	0.0022	0.0002	0.0055
FedFG($\eta = 1.0$)	57.37	16.01	22.95	0.0002	0.0000	0.0006
FedFG-C($\eta = 0.1$)	77.27	70.54	72.17	0.0107	0.0015	0.0282
FedFG-C($\eta = 1.0$)	75.99	70.20	67.38	0.0025	0.0002	0.0075

Table 16: Results for Equalized Odds fairness on Dir(0.5) partition

algorithm	Accuracy ↑	Precision ↑	Recall ↑	$\lvert F_k\rvert$-avg ↓	$\lvert F_k\rvert$-min ↓	$\lvert F_k\rvert$-max ↓
Constant	60.79	0.00	0.00	0.0000	0.0000	0.0000
FedAvg	75.65	64.10	86.09	0.0394	0.0067	0.0907
FedFB($\alpha = 0.005$)	78.72	74.04	70.41	0.0437	0.0052	0.1223
FedFG($\eta = 0.1$)	65.74	89.74	14.24	0.0023	0.0002	0.0056
FedFG($\eta = 1.0$)	62.11	12.48	4.21	0.0009	0.0000	0.0029
FedFG-C($\eta = 0.1$)	77.37	68.65	77.84	0.0146	0.0006	0.0501
FedFG-C($\eta = 1.0$)	76.06	73.09	61.63	0.0012	0.0001	0.0035