
Global Group Fairness in Federated Learning via Function Tracking

Yves Rychener¹

¹EPFL, Switzerland

Daniel Kuhn¹

Yifan Hu^{1,2}

²ETH Zurich, Switzerland

Abstract

We investigate group fairness regularizers in federated learning, aiming to train a globally fair model in a distributed setting. Ensuring global fairness in distributed training presents unique challenges, as fairness regularizers typically involve probability metrics between distributions across all clients and are not naturally separable by client. To address this, we introduce a function-tracking scheme for the global fairness regularizer based on a Maximum Mean Discrepancy (MMD), which incurs a small communication overhead. This scheme seamlessly integrates into most federated learning algorithms while preserving rigorous convergence guarantees, as demonstrated in the context of FedAvg. Additionally, when enforcing differential privacy, the kernel-based MMD regularization enables straightforward analysis through a change of kernel, leveraging an intuitive interpretation of kernel convolution. Numerical experiments confirm our theoretical insights.

1 INTRODUCTION

Many modern datasets are stored decentrally over multiple devices, organizations or jurisdictions and contain sensitive information. Federated learning (McMahan et al., 2017) is proposed to train machine learning models on the aggregated (*global*) dataset without exchanging data between local devices to protect privacy. On the other hand, these machine learning models are used in various aspects of daily life and can significantly impact people’s lives. Unfortunately, cutting-edge AI models can learn or even amplify biases in datasets, which results in discriminatory predictions (Choulde-

chova, 2017; Dastin, 2018; Angwin et al., 2022). Ensuring that machine learning models treat different demographic groups fairly has become increasingly important. The desire to eliminate algorithmic bias has sparked the development of new algorithms in the field of fair machine learning. As both privacy and fairness become more critical, there is a need to train fair models on distributed datasets using federated learning. In this work, we study fair federated learning to protect privacy while eliminating algorithmic bias.

In the following, we assume that each datapoint accommodates a binary attribute A (called the sensitive attribute), which encodes membership in a demographic group we do not wish to discriminate against (e.g., race, gender, religion). We aim to train a model that satisfies a notion of group fairness on the global scale, that is, on the aggregated dataset (*global group fairness*). We use Figure 1 to explain how this fairness notion differs from others. To this end, let \mathbb{P}_k be the distribution of the dataset of the k -th client, and consider the global mixture distribution $\mathbb{P} = \sum_{k=1}^K \nu_k \mathbb{P}_k$, where the mixture weight ν_k encodes the size of the k -th dataset. In the following we say that a model h is fair in the sense of demographic parity if the prediction $h(X)$ has the same distribution across several demographic groups.

Global group fairness holds if $\mathbb{P}^{h(X)|A=0} = \mathbb{P}^{h(X)|A=1}$, i.e., if the prediction distributions match across both demographic groups encircled in green in Figure 1.

Local group fairness holds if $\mathbb{P}_k^{h(X)|A=0} = \mathbb{P}_k^{h(X)|A=1}$ for all k , i.e., if the prediction distributions match across all demographic groups and clients encircled in blue.

Client fairness holds if $\mathbb{P}_k^{h(X)} = \mathbb{P}_{k'}^{h(X)}$ for all k, k' , i.e., if the prediction distributions match across all clients encircled in red.

By Simpson’s paradox (Simpson, 1951), none of the above fairness notions implies the other. To see this, assume that $K = 2$ and that $h(X)$ conditional on $A = a$ follows a shifted standard normal distribution with mean 0 if $k + a$ is even and with mean 1 if $k + a$ is odd. Thus, h is globally fair because $\mathbb{P}^{h(X)|A=a} = \frac{1}{2}\mathcal{N}(0, 1) + \frac{1}{2}\mathcal{N}(1, 1)$ for all $a \in \{0, 1\}$. However, h does

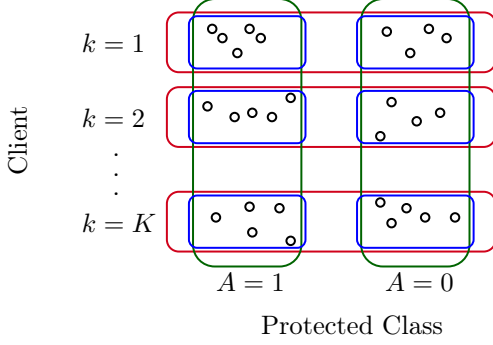


Figure 1: Illustration of client fairness (red), global group fairness (green) and local group fairness (blue)

not satisfy local statistical parity because $\mathbb{P}_1^{h(X)|A=0} = \mathcal{N}(1, 1) \neq \mathcal{N}(0, 1) = \mathbb{P}_1^{h(X)|A=1}$. A similar situation will be explored in Section 6 below.

We emphasize that different applications necessitate different fairness criteria. For example, an algorithm that assesses the quality of images across multiple mobile phones may have to rate a similar percentage of images highly on each device in order to ensure user satisfaction (*client fairness*). Similarly, multiple companies that train an algorithm for ranking job applicants may want to avoid racial discrimination at each individual company (*local group fairness*). Finally, multiple banks that train a model to predict their clients’ credit worthiness may want to avoid gender discrimination across the entire population without exchanging proprietary data (*global group fairness*). In practice, fairness criteria should be selected case-by-case. Global group fairness is not always the right choice.

Focusing on a general notion of group fairness, we propose a metric-regularized fair federated learning objective that incorporates a fairness-promoting regularization term into the classical federated learning framework. We show that a metric-based fairness regularizer cannot be decomposed as a weighted sum over local client objectives, making classical federated learning algorithms inapplicable. To address this, we introduce an approximation of the Maximum Mean Discrepancy (MMD) metric regularizer, updated at each communication round. This approach offers several advantages. First, the fairness-accuracy tradeoff can be easily adjusted by varying the regularization strength. Second, the metric-based regularization applies to a broad range of group fairness definitions. Third, it requires only a simple modification of existing federated learning methods. Finally, it is compatible with both cross-device and cross-silo settings and integrates seamlessly into most federated learning algorithms. We also demonstrate that differential privacy can be preserved while constructing the approximation.

Our main contributions are (1) a client-decomposable approximation of the metric-based fairness regularizer, (2) the derivation of the convergence rate for fairness-regularized FedAvg, and (3) an interpretation of differential privacy as a kernel change for MMD.

Notation We denote the space of probability distributions on $\mathcal{Z} \subseteq \mathbb{R}^d$ by $\mathcal{P}(\mathcal{Z})$. For any $\mathbb{P} \in \mathcal{P}(\mathcal{Z})$ and $E \subseteq \mathcal{Z}$, we denote the probability distribution of Z conditioned on the event $Z \in E$ by $\mathbb{P}[\cdot|Z \in E]$. We sometimes use $\mathbb{P}^{Z|Z \in E}$ as shorthand for this distribution. We use subscripts to index different distributions (e.g., \mathbb{P}_k) and the respective expectation operators (e.g., $\mathbb{E}_k[\cdot]$). We further use $\mathcal{L}(\mathcal{Z}, \mathcal{Z}')$ to denote the space of Borel measurable functions from \mathcal{Z} to \mathcal{Z}' and, for a given $w \in \mathcal{L}(\mathcal{Z}, [0, \infty))$, we use $\mathcal{P}_w(\mathcal{Z})$ to denote the set of all $\mathbb{P} \in \mathcal{P}(\mathcal{Z})$ that satisfy $\int_{\mathcal{Z}} w(z) \mathbb{P}(dz) < \infty$.

2 RELATED WORK

Group Fairness The seminal work of Zafar et al. (2017a) introduced a covariance-based fairness constraint for logistic regression. Several studies extend this by proposing new fairness constraints (Donini et al., 2018; Wu et al., 2019; Zafar et al., 2019, 2017b). Rather than enforcing strict constraints, researchers often use fairness regularization for penalizing unfair models in the objective (Donini et al., 2018; Oneto et al., 2020; Taskesen et al., 2020; Wu et al., 2019; Zafar et al., 2017b), which enables the use of gradient-based methods for training deep neural networks. Lohaus et al. (2020) examine how fairness-regularized problems align with the original fairness constraint. A flexible and effective way to construct fairness regularizers is through metrics or divergences of probability distributions (Oneto et al., 2020; Prost et al., 2019; Rychener et al., 2022; Risser et al., 2022). For example, statistical parity can be enforced by penalizing differences between $\mathbb{P}^{h(X)|A=a}$ for $a \in \{0, 1\}$. We refer to (Rychener et al., 2022) for discussions on fairness notions that can be addressed with this approach. In this paper, we use the MMD metric for fairness regularization due to its favorable computational properties. MMD metrics have already been used in centralized learning (Oneto et al., 2020; Prost et al., 2019; Rychener et al., 2022).

Federated Learning McMahan et al. (2017) introduced the FedAVG algorithm, which trains models on distributed data while periodically communicating updates to a central node, enhancing data privacy. Since then, federated learning has garnered significant attention, leading to novel algorithms (Karimireddy et al., 2020; Li et al., 2020; Wang et al., 2019), convergence rate analyses (Karimireddy et al., 2020; Khaled et al., 2020; Li et al., 2020; Yu et al., 2019), and methods for

enforcing differential privacy (DP) (McMahan et al., 2018; Nikolaenko et al., 2013; Dwork et al., 2006) or leveraging homomorphic encryption (Nikolaenko et al., 2013). For a comprehensive overview of recent advances and open challenges, see (Kairouz et al., 2021).

Fair Federated Learning The federated learning literature considers three main notions of fairness. Client fairness arises in cross-device federated learning and ensures that individual clients are not disadvantaged (Mohri et al., 2019; Deng et al., 2020; Yue et al., 2021). Global group fairness more closely aligns with classical centralized group fairness, aiming to prevent systemic discrimination of individuals, ensuring similar treatment across demographic groups spanning multiple clients. One approach to achieving global group fairness in federated learning is adjusting FedAvg’s aggregation weights to promote a fairer aggregated model (Du et al., 2021; Ezzeldin et al., 2023; Salazar et al., 2023). Another direction extends centralized fairness algorithms to the federated setting, such as FairBatch (Roh et al., 2021; Zeng et al., 2021), which adaptively reweights samples during local batch construction, min-max fairness, which minimizes loss for the worst-off protected group (Papadaki et al., 2022), and pre- and post-processing techniques (Pentya et al., 2022).

Some group fairness notions, particularly those defined in terms of expectations—where fairness is enforced by matching expectations rather than distributions across protected groups—are inherently client-decomposable. This allows existing federated learning algorithms to be applied directly (Chu et al., 2021; Hu et al., 2024a). However, while useful in certain contexts, these fairness notions may be too simplistic in others. For instance, two prediction distributions with the same expectation but different variances would be considered fair under expectation-based fairness yet may still be unfair from a distributional perspective.

Lastly, local group fairness is studied by Abay et al. (2020); Cui et al. (2021). While beneficial in some cases, our Proposition 3.1 and recent findings by Hamman and Dutta (2023) indicate that it does not necessarily ensure group fairness in the aggregated distribution. To our best knowledge, global group fairness regularizers have not yet been explored in federated learning.

3 FAIR FEDERATED LEARNING WITH MMD METRICS

We consider learning problems of the form

$$\min_{h \in \mathcal{H}} \mathbb{E}[L(h(X), Y)], \quad (1)$$

where $X \in \mathcal{X} \subseteq \mathbb{R}^d$ and $Y \in \mathcal{Y} \subseteq \mathbb{R}$ denote the feature vector and target, respectively, $L : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ stands

for the loss function, and $h : \mathcal{X} \rightarrow \mathcal{Y}$ denotes a model chosen from a space \mathcal{H} of Borel measurable functions. We further assume that there is a protected attribute $A \in \mathcal{A} = \{0, 1\}$, which encodes a sensitive property of a person such as gender, race, or religion. Extending this framework to multiple or categorical protected attributes is feasible. The method to be developed below can accommodate such cases by incorporating additional fairness penalties into the objective.

We consider a federated learning setting with K clients and represent datasets using probability distributions. While various data partitioning schemes have been explored in the federated learning literature (see (Kairouz et al., 2021) for an overview), we focus on horizontal federated learning, where clients own different samples, all of which share the same feature space. We assume that the global distribution $\mathbb{P} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$ of the features, targets and protected attributes can be represented as a mixture $\mathbb{P} = \sum_{k=1}^K \nu_k \mathbb{P}_k$ of local (client-wise) distributions $\mathbb{P}_k \in \mathcal{P}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$, with client weights $\nu_k \geq 0$ satisfying $\sum_{k=1}^K \nu_k = 1$.

To avoid discriminatory outcomes when solving (1), we enforce a form of group fairness, which ensures that the demographic groups corresponding to $A = 0$ and $A = 1$ are treated similarly. We say that a hypothesis h is ε -group fair (Rychener et al., 2022) for some $\varepsilon \geq 0$ if

$$\begin{aligned} \mathcal{D} \left(\mathbb{P}^{S(h(X), Y) | A=0, (X, Y) \in \mathcal{C}_j}, \mathbb{P}^{S(h(X), Y) | A=1, (X, Y) \in \mathcal{C}_j} \right) \\ \leq \varepsilon \quad \forall j \in \mathcal{J}, \end{aligned}$$

where \mathcal{D} is a dissimilarity measure for probability distributions, $S : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ is a scoring function that assigns each prediction $h(X)$ and output Y a score such as the task loss or some prediction score, and $\mathcal{C} = \{\mathcal{C}_j\}_{j \in \mathcal{J}}$ is a family of conditioning sets with $\mathcal{C}_j \subseteq \mathcal{X} \times \mathcal{Y}$. This generic fairness notion encompasses a variety of popular fairness criteria as shown in Table 1. For example, risk parity (Donini et al., 2018; Maity et al., 2021) aims to match the distributions of the prediction loss $L(h(X), Y)$ across the two demographic groups, while equal opportunity (Hardt et al., 2016; Pleiss et al., 2017) aims to match the distributions of the prediction $h(X)$ across the subclasses of positively labeled datapoints within the two groups. In general, the unfairness of a hypothesis h is small if, for every j , the distributions of the scores $S(h(X), Y)$ conditional on $A = a$ and $(X, Y) \in \mathcal{C}_j$ are similar across all $a \in \mathcal{A}$.

For ease of exposition, from now on we focus on statistical parity, that is, we set $S(\hat{y}, y) = \hat{y}$ and $\mathcal{C} = \{\mathcal{X} \times \mathcal{Y}\}$. However, all proposed methods readily extend to other group-fairness notions, including equal opportunity, equalized odds, and risk parity, see Appendix A.2.

Following Donini et al. (2018); Oneto et al. (2020); Rychener et al. (2022); Taskesen et al. (2020); Wu et al.

Table 1: Common group fairness criteria induced by different scoring functions and families of conditioning sets (here, \mathcal{X}_s is any conditioning set contained in \mathcal{X})

Fairness Criterion	$S(\hat{y}, y)$	\mathcal{C}
Statistical parity (Agarwal et al., 2019)	\hat{y}	$\{\mathcal{X} \times \mathcal{Y}\}$
Equal opportunity (Hardt et al., 2016; Pleiss et al., 2017)	\hat{y}	$\{\mathcal{X} \times \{1\}\}$
Equalized odds (Hardt et al., 2016; Pleiss et al., 2017)	\hat{y}	$\{\mathcal{X} \times \{y\}\}_{y \in \mathcal{Y}}$
Risk Parity (Donini et al., 2018; Maity et al., 2021)	$L(\hat{y}, y)$	$\{\mathcal{X} \times \mathcal{Y}\}$
Conditional statistical parity (Corbett-Davies et al., 2017)	\hat{y}	$\{\mathcal{X}_s \times \mathcal{Y}\}$
Prob. predictive equality (Corbett-Davies et al., 2017; Pleiss et al., 2017)	\hat{y}	$\{\mathcal{X} \times \{0\}\}$

(2019); Zafar et al. (2017b), we enforce group fairness in (1) by solving the fairness-regularized problem

$$\min_{h \in \mathcal{H}} \mathbb{E}[L(h(X), Y)] + \lambda \rho \left(\mathcal{D}(\mathbb{P}^{h(X)|A=0}, \mathbb{P}^{h(X)|A=1}) \right), \quad (2)$$

where $\rho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a non-decreasing regularization function. Since problem (1) is client-decomposable, it can be directly solved using a federated learning approach. In contrast, problem (2) lacks client-wise decomposition because the fairness regularizer couples clients in a nontrivial way. This observation is formally stated in the following proposition.

Proposition 3.1 (Impossibility of Client-Decomposition). *Let $\rho : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be a regularization function with $\rho(v) = 0$ if and only if $v = 0$. Then, there exists no dissimilarity measure $\mathcal{D} : \mathcal{P}(\mathcal{Y}) \times \mathcal{P}(\mathcal{Y}) \rightarrow \mathbb{R}_+$ satisfying the identity of indiscernibles such that*

$$\begin{aligned} & \rho \left(\mathcal{D} \left(\sum_{k=1}^K \nu_k \mathbb{P}_k^{h(X)|A=0}, \sum_{k=1}^K \nu_k \mathbb{P}_k^{h(X)|A=1} \right) \right) \\ &= \sum_{k=1}^K \nu_k \rho \left(\mathcal{D}(\mathbb{P}_k^{h(X)|A=0}, \mathbb{P}_k^{h(X)|A=1}) \right) \end{aligned} \quad (3)$$

for all $h \in \mathcal{H}$, $\nu_k \geq 0$ with $\sum_{k=1}^K \nu_k = 1$ and probability measures $\mathbb{P}_k \in \mathcal{P}(\mathcal{X} \times \mathcal{Y} \times \mathcal{A})$, $k = 1, \dots, K$.

If a decomposition of the unfairness penalty, as described in Proposition 3.1, existed, standard federated learning algorithms could solve the fairness-regularized problem by replacing the original local gradient estimator for (1) with a fairness-regularized local gradient estimator for (2), leveraging the right-hand side of (3). However, since such a decomposition is impossible, fairness-regularized federated learning presents a unique challenge, as the non-decomposability of group fairness regularizers appears to be intrinsic. This aligns with recent findings that local fairness does not generally imply global fairness, and vice versa, from an

information-theoretic perspective (Hamman and Dutta, 2023). The key question, then, is how to efficiently solve problem (2) within a federated learning framework.

From now on, we will set the dissimilarity measure \mathcal{D} to a maximum mean discrepancy (MMD).

Definition 3.2 (Maximum Mean Discrepancy). *If $\kappa \in \mathcal{L}(\mathbb{R}^n \times \mathbb{R}^n, \mathbb{R})$ is a positive definite kernel and $w \in \mathcal{L}(\mathbb{R}^n, [1, \infty))$ satisfies $\sup_{z \in \mathbb{R}^n} \kappa(z, z')/w(z) < \infty$ for all $z' \in \mathbb{R}^n$, then the maximum mean discrepancy between $\mathbb{P}_1, \mathbb{P}_2 \in \mathcal{P}_w(\mathbb{R}^n)$ relative to κ is given by*

$$\begin{aligned} \mathcal{D}_{\text{MMD}}(\mathbb{P}_1, \mathbb{P}_2) &= \left(\int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa(z, z') \mathbb{P}_1(dz) \mathbb{P}_1(dz') \right. \\ &\quad + \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa(z, z') \mathbb{P}_2(dz) \mathbb{P}_2(dz') \\ &\quad \left. - 2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa(z, z') \mathbb{P}_1(dz) \mathbb{P}_2(dz') \right)^{\frac{1}{2}}. \end{aligned}$$

Definition 3.2 characterizes the MMD distance between multivariate distributions on \mathbb{R}^n . However, we defined the score $S(\hat{y}, y)$ as a scalar, and thus our fairness regularizer only requires an MMD distance for univariate distributions. Extensions to vector-valued scores are possible using a similar construction. Among all possible fairness regularizers in (2) obtained by combining some regularization function ρ with a dissimilarity measure \mathcal{D} , the squared MMD distance stands out for multiple reasons. First, for common kernels κ and finite-dimensional parametrizations h_θ of the hypotheses in \mathcal{H} , the fairness regularizer $\mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2$ constitutes a smooth function of the parameter θ . Second, in Section 4, we introduce a biased local gradient estimator for this fairness regularizer based on a simple function tracking scheme. This estimator remains unbiased at the communication round, when client updates are aggregated by the server, but becomes biased during local updates performed between communication rounds. Despite this bias, existing convergence results for stochastic gradient descent (SGD) in federated learning can be naturally extended to account for it. Furthermore, the effect of differential privacy noise can be interpreted as a kernel convolution (see Section 5), which provides a clearer understanding of its impact on learning. This interpretation also allows generalization bounds to be directly applied (Rychener et al., 2022). For a rigorous restatement of this result, see Proposition A.1 in the appendix. From now on we refer to the following instance of (2) with $\mathcal{D} = \mathcal{D}_{\text{MMD}}$ and $\rho(z) = \lambda z^2$ for some $\lambda \geq 0$ as the MMD-fair federated learning problem.

$$\min_{h \in \mathcal{H}} \mathbb{E}[L(h(X), Y)] + \lambda \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h(X)|A=0}, \mathbb{P}^{h(X)|A=1})^2. \quad (4)$$

4 ALGORITHM DESIGN

We now propose an efficient function tracking scheme for the fairness-regularized federated learning problem (4). To this end, we focus on a parameterized hypothesis class $\mathcal{H} = \{h_\theta\}_{\theta \in \Theta}$, which includes models such as linear predictors and neural networks, and design a federated gradient-based algorithm. The objective function of problem (4) is henceforth denoted as

$$F(\theta) = \mathbb{E}[L(h_\theta(X), Y)] + \lambda \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2.$$

Note that $F(\theta)$ consists of the task loss $\mathbb{E}[L(h_\theta(X), Y)]$, which can be naturally decomposed into a weighted average of local losses $\mathbb{E}_k[L(h_\theta(X), Y)]$, and the fairness loss $\mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2$ weighted by λ .

In the absence of a fairness regularizer, the classical federated learning algorithm FedAvg (McMahan et al., 2017) operates as follows. At each communication round t , the central server sends the current global model h_{θ^t} to a selected subset of clients. Each participating client then updates h_{θ^t} locally by performing E steps of stochastic gradient descent (SGD) on its own loss function $\mathbb{E}_k[L(h_\theta(X), Y)]$, where the expectation is taken with respect to the local data distribution \mathbb{P}_k of client k . The client transmits the updated local model $h_{\theta_{E+1,k}^t}$ back to the central server. The server then aggregates the received local models by computing their weighted average, where the contribution of each client is proportional to the number ν_k of data samples it holds. Specifically, the new global model $h_{\theta^{t+1}}$ is obtained by weighting each client's update according to its dataset size. This process is repeated for multiple communication rounds until convergence.

Unfortunately, Proposition 3.1 shows that the fairness loss does not admit a client-wise decomposition. This poses a challenge in federated learning, where transferring client data to a central server for computing gradients of the fairness loss and updating the global model is infeasible due to privacy constraints. To overcome this, we seek to incorporate fairness-aware updates at the client level. Specifically, we construct a client-decomposable approximation of the gradient of the fairness loss that each client can integrate into its local updates during the second step of FedAvg.

For a fixed parameter θ , we sample a finite number of predictions from $\mathbb{P}^{h_\theta(X)|A=a}$ across all clients' datasets and collect them in the set \mathcal{Y}_a for any $a \in \mathcal{A}$. Using these datasets, we then construct the auxiliary function

$$f(\theta; \mathcal{Y}_0, \mathcal{Y}_1) = 2 \left(\int_{\mathbb{R}^n} C(h_\theta(x); \mathcal{Y}_0, \mathcal{Y}_1) \mathbb{P}^{X|A=0}(dx) - \int_{\mathbb{R}^n} C(h_\theta(x); \mathcal{Y}_0, \mathcal{Y}_1) \mathbb{P}^{X|A=1}(dx) \right),$$

where

$$C(\hat{y}; \mathcal{Y}_0, \mathcal{Y}_1) = \frac{1}{|\mathcal{Y}_0|} \sum_{y \in \mathcal{Y}_0} \kappa(\hat{y}, y) - \frac{1}{|\mathcal{Y}_1|} \sum_{y \in \mathcal{Y}_1} \kappa(\hat{y}, y).$$

Then, the gradient of f with respect to θ is an unbiased estimator for the gradient of the fairness loss.

Lemma 4.1 (Unbiased Gradient Estimator). *If the kernel κ is continuous and the model $h_\theta(X)$ is \mathbb{P} -almost surely continuously differentiable in θ , then we have*

$$\begin{aligned} \nabla_\theta \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2 \\ = \mathbb{E}[\nabla_\theta f(\theta; \mathcal{Y}_0, \mathcal{Y}_1)]. \end{aligned}$$

We emphasize that $\nabla_\theta f(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ stands for the partial gradient of $f(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ with respect to its first argument. Thus, the gradient ignores the dependence of \mathcal{Y}_0 and \mathcal{Y}_1 on θ . We also emphasize that $f(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ is unavailable in practice because the distributions $\mathbb{P}^{X|A=a}$ for $a \in \mathcal{A}$ are unknown. However, if all clients were to share their data, then $f(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ could be estimated by sampling from these distributions. By construction, the function f admits the client decomposition

$$f(\theta; \mathcal{Y}_0, \mathcal{Y}_1) = \sum_{k=1}^K \nu_k f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1),$$

where the individual client functions f_k are given by

$$\begin{aligned} f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1) = 2 \left(\alpha_k^0 \int_{\mathbb{R}^n} C(h_\theta(x); \mathcal{Y}_0, \mathcal{Y}_1) \mathbb{P}_k^{X|A=0}(dx) \right. \\ \left. - \alpha_k^1 \int_{\mathbb{R}^n} C(h_\theta(x); \mathcal{Y}_0, \mathcal{Y}_1) \mathbb{P}_k^{X|A=1}(dx) \right), \end{aligned}$$

and where $\alpha_k^a = \mathbb{P}_k(A = a)/\mathbb{P}(A = a)$ corrects the mixture weight ν_k after conditioning on $A = a$. Note again that $f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ is unavailable because the distributions $\mathbb{P}_k^{X|A=a}$ for $a \in \mathcal{A}$ are unknown. However, the k -th client can estimate $f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ by using only privately owned samples from these distributions. The above client decomposition of f readily leads to an estimator for the overall loss that is both unbiased and admits a client decomposition, that is, we have

$$F(\theta) = \mathbb{E} \left[\sum_{k=1}^K \nu_k \left(\mathbb{E}_k[L(h_\theta(X), Y)] + \lambda f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1) \right) \right].$$

Note that Lemma 4.1 does not contradict the impossibility result in Proposition 3.1 because the prediction datasets \mathcal{Y}_0 and \mathcal{Y}_1 appearing in $f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$ are constructed from information that originates from *all* clients. Thus, in order to construct $f_k(\theta; \mathcal{Y}_0, \mathcal{Y}_1)$, the k -th client requires information from other clients. Nevertheless, sharing \mathcal{Y}_0 and \mathcal{Y}_1 does not require transmitting each client's original dataset but rather a few scalar predictions, thereby preserving data privacy.

We now detail our proposed algorithm for fair federated learning; see also Algorithm 1. It is inspired by the classical FedAvg algorithm and works as follows. At each communication round t and for each $a \in \mathcal{A}$, the current model h_{θ^t} is used to sample a finite number of predictions from $\mathbb{P}^{h_{\theta^t}(X)|A=a}$, which are collected in \mathcal{Y}_a^t . The prediction sets \mathcal{Y}_0^t and \mathcal{Y}_1^t , along with the model h_{θ^t} , are transmitted to a randomly selected batch of clients. Each client then constructs unbiased gradient estimators for $\mathbb{E}_k[L(h_{\theta}(X), Y)] + \lambda f_k(\theta; \mathcal{Y}_0^t, \mathcal{Y}_1^t)$, performs multiple local SGD updates, and communicates the final iterate to the central server. The server subsequently aggregates the locally updated models to obtain θ^{t+1} . The communication overhead of this algorithm exceeds that of standard FedAvg. Specifically, in each communication round t , the server must exchange information about the sets \mathcal{Y}_0^t and \mathcal{Y}_1^t with the clients. However, the additional cost is moderate because these sets contain only one-dimensional samples. Indeed, the total bit size of \mathcal{Y}_0^t and \mathcal{Y}_1^t is often negligible compared to that of the high-dimensional iterate θ^t .

Lemma 4.1 implies that, in each communication round t , $\nabla_{\theta} f(\theta, \mathcal{Y}_0^t, \mathcal{Y}_1^t)$ serves as an unbiased estimator for $\nabla_{\theta} \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_{\theta}(X)|A=0}, \mathbb{P}^{h_{\theta}(X)|A=1})^2$ at $\theta = \theta^t$. However, we emphasize that $\nabla_{\theta} f(\theta, \mathcal{Y}_0^t, \mathcal{Y}_1^t)$ is a biased gradient estimator for the fairness loss when evaluated at $\theta \neq \theta^t$, since \mathcal{Y}_0^t and \mathcal{Y}_1^t are generated under θ^t . As these sets remain fixed throughout each communication round, the clients' SGD updates are biased. This bias is mitigated, however, as \mathcal{Y}_0 and \mathcal{Y}_1 are refreshed at the start of each communication round. Indeed, resampling prevents bias accumulation during local updates and ensures tracking of the population-level counterparts of the C functions. The sizes of \mathcal{Y}_0 and \mathcal{Y}_1 affect the variances of the underlying estimators. However, as each client performs local SGD updates, the gradient estimator for the task loss also introduces variance. In theory, $O(1)$ samples suffice to construct \mathcal{Y}_0 and \mathcal{Y}_1 .

The steps where Algorithm 1 differs from FedAvg are highlighted by gray boxes. On lines 2–3, an additional one-time communication between the clients and the server is required to compute the weights $\alpha_k^a = \mathbb{P}_k(A = a)/\mathbb{P}(A = a)$. In practice, these weights can be determined if the server knows the number of samples each client has from each demographic group $a \in \mathcal{A}$. Exchanging sample size information is generally uncontroversial. On lines 5–6, \mathcal{Y}_0^t and \mathcal{Y}_1^t must be generated and communicated to clients, incurring a moderate additional communication cost. As shown in Appendix C.3, using 50 samples for \mathcal{Y}_0^t and \mathcal{Y}_1^t suffices in our experiments. Furthermore, the potential information leakage from sharing these samples is minimal, especially since, as demonstrated in Section 5.2, they can be effectively protected with differential pri-

Algorithm 1 MMD-Fair FedAvg

```

1: Initialize  $\theta^1$ 
2: Collect information to estimate  $\alpha_k^a, a \in \mathcal{A}$ 
3: Communicate  $\alpha_k^a, a \in \mathcal{A}$ , to each client  $k$ 
4: for  $t = 1, \dots, T$  do
5:   Construct  $\mathcal{Y}_a^t, a \in \mathcal{A}$ , using  $\theta^t$ 
6:   Communicate  $\theta^t$  and  $\mathcal{Y}_a^t, a \in \mathcal{A}$ , to all clients
7:   Sample a subset  $\mathcal{S} \subseteq \{1, \dots, K\}$  of  $S$  clients
8:   for  $k \in \mathcal{S}$  in parallel do
9:     Initialize  $\theta_{1,k}^t = \theta^t$ 
10:    for local iteration  $i = 1, \dots, E$  do
11:      Compute a stochastic gradient  $g_{i,k}^t$  with
        
$$\mathbb{E}[g_{i,k}^t | \mathcal{Y}_0^t, \mathcal{Y}_1^t] = \nabla_{\theta} (\mathbb{E}_k[L(h_{\theta_{i,k}^t}(X), Y)] + \lambda f_k(\theta_{i,k}^t; \mathcal{Y}_0^t, \mathcal{Y}_1^t))$$

12:      Update  $\theta_{i+1,k}^t = \theta_{i,k}^t - \eta_l g_{i,k}^t$ 
13:    end for
14:    Communicate  $\theta_{E+1,k}^t$  to server
15:  end for
16:  Update  $\theta^{t+1} = \theta^t + \eta_g \left( \frac{1}{S} \sum_{k \in \mathcal{S}} (\theta_{E+1,k}^t - \theta^t) \right)$ 
17: end for
    
```

vacy. Finally, on line 11, the gradient of the fairness loss must be approximated in the client's objective function. All these modifications are simple, compatible with most federated learning algorithms beyond FedAvg, and easy to implement.

5 ANALYSIS OF ALGORITHM 1

We now analyze the convergence rate of Algorithm 1 and show how it can be made differentially private.

5.1 Convergence Analysis

Theorem 5.1 (Convergence of Algorithm 1). *Assume that the kernel function κ and the loss function L are Lipschitz continuous and Lipschitz smooth. Assume further that the model $h_{\theta}(X)$ is \mathbb{P} -almost surely Lipschitz continuous and Lipschitz smooth in θ and that the gradient estimator $g_{i,k}^t$ has bounded variance. Then, there exist a maximum stepsize $\bar{\eta} > 0$ and probabilities $\{w_t\}_{t=1}^T$ such that for any training hyperparameters $E \in \mathbb{N}$, $\eta_l > 0$ and $\eta_g \geq 1$ with $E\eta_l\eta_g \leq \bar{\eta}$, we have*

$$\mathbb{E} \|\nabla_{\theta} F(\bar{\theta})\|^2 \leq \mathcal{O} \left(\frac{1}{T+1} + \frac{1}{\sqrt{ES(T+1)}} + \frac{1}{E^{\frac{1}{3}}(T+1)^{\frac{2}{3}}} \right),$$

where $\bar{\theta} = \theta^t$ with probability w_t and θ^t is the t -th global iterate of Algorithm 1 for every $t = 1, \dots, T$.

Our convergence analysis builds on that for FedAvg when the objective is nonconvex and only a random subset of clients participates in any round (Karimireddy et al., 2020). It shows that Algorithm 1 requires the same number of communication rounds, local update steps, and clients per round as FedAvg. When clients are sampled randomly, applying momentum acceleration or variance reduction (Cheng et al., 2024) can result in state-of-the-art convergence rates. However, as our focus is on integrating global fairness regularization into federated learning, we leave further convergence improvements using these ideas for future work. Our proof techniques are significantly different from those that are typically used in the context of biased gradient methods (Hu, 2024; Hu et al., 2024b).

We briefly outline the proof idea for Theorem 5.1. The analysis by Karimireddy et al. (2020) relates local update gradients to the gradient at the previous communication round. Even though the local gradient estimators are generally biased due to the fairness regularization, Lemma 4.1 shows that they remain unbiased at the communication round when \mathcal{Y}_0^t and \mathcal{Y}_1^t are sampled. As a result, the bias from fairness regularization is reset at each communication round. We use this property to analyze the bias similarly to how Karimireddy et al. (2020) analyze local gradients. The full proof is provided in Appendix B.3.1.

5.2 Differential Privacy

To address privacy concerns arising from sharing predictions $h_\theta(X)$ and parameters $\theta_{E+1,k}^t$, federated learning commonly employs differential privacy (DP) to prevent sensitive information from being leaked through network communications. Specifically, additive DP mechanisms introduce random noise to shared parameters, ensuring that clients cannot reconstruct data samples from other participants. The protection of model updates $\theta_{E+1,k}^t$ has already been studied extensively in the federated learning literature (see (El Ouadrhiri and Abdelhadi, 2022) for a survey). We therefore show how DP can be used to safeguard shared predictions $h_\theta(X)$. This can be achieved by adding Gaussian noise (the *Gaussian mechanism* (Balle and Wang, 2018)) or Laplacian noise (the *Laplacian mechanism* (Koufogiannis et al., 2015)) to the predictions shared on line 5 of Algorithm 1. For each prediction $\widehat{Y} \in \mathcal{Y}_a$, $a \in \mathcal{A}$, we construct a privacy-protected version $\tilde{Y} = \widehat{Y} + \xi$ with $\xi \sim \mu_{\text{DP}}$, and we denote by $\tilde{\mathcal{Y}}_a$ the resulting privacy-protected set. If the predictions \widehat{Y} are bounded, as is the case for scores after a sigmoid activation in neural networks, the variance (for the Gaussian mechanism) or scale (for the Laplacian mechanism) can be calibrated and remains finite. Even in regression tasks, it is often reasonable to assume that \widehat{Y} is bounded or can be

clipped, as most real-world quantities to be estimated are naturally constrained. While DP noise is typically seen as a nuisance, we now show that noise added to shared predictions \widehat{Y} can instead be interpreted as a kernel modification in the MMD metric.

Proposition 5.2 (Effect of Differential Privacy). *Assume that $\xi \sim \mu_{\text{DP}}$, $a, b \stackrel{i.i.d.}{\sim} \mu'$ and $\xi \stackrel{d}{=} a - b$. Construct the differentially private sets $\tilde{\mathcal{Y}}_0$ and $\tilde{\mathcal{Y}}_1$ by adding i.i.d. samples from μ_{DP} to each element of \mathcal{Y}_0 and \mathcal{Y}_1 , respectively. If the kernel κ satisfies $\kappa(x, y) = v(x - y)$ for some real-valued function v , then we have*

$$\begin{aligned} \tilde{C}(z; \mathcal{Y}_0, \mathcal{Y}_1) &:= \mathbb{E} \left[C(z; \tilde{\mathcal{Y}}_0, \tilde{\mathcal{Y}}_1) \middle| \mathcal{Y}_0, \mathcal{Y}_1 \right] \\ &= \frac{1}{|\mathcal{Y}_0|} \sum_{y \in \mathcal{Y}_0} \tilde{\kappa}(z, y) - \frac{1}{|\mathcal{Y}_1|} \sum_{y \in \mathcal{Y}_1} \tilde{\kappa}(z, y), \end{aligned}$$

that is, $\tilde{C}(z; \mathcal{Y}_0, \mathcal{Y}_1)$ is a variant of $C(z; \mathcal{Y}_0, \mathcal{Y}_1)$, where the original kernel $\kappa(x, y)$ is replaced with the DP kernel $\tilde{\kappa}(x, y) = \int_{\mathbb{R}} \kappa(x, y + \xi) \mu_{\text{DP}}(d\xi)$. Note that $\tilde{\kappa}(x, y)$ can be viewed as the convolution of μ_{DP} and κ . Furthermore, if κ is positive semidefinite, then so is $\tilde{\kappa}$.

The assumptions of Proposition 5.2 hold for many DP mechanisms and distributions of interest. The conditions on μ_{DP} are satisfied by both the Gaussian mechanism (where μ' is a Gaussian distribution with lower variance than μ_{DP}) and the Laplacian mechanism (where μ' is an exponential distribution). The condition on κ holds for many commonly used kernels, such as the Gaussian and Laplacian/Abelian kernels. However, one can show that the condition $\kappa(x, y) = v(x - y)$ implies $|\kappa(x, y)| \leq |v(0)|$, which can only be satisfied by bounded kernels. By Proposition 5.2, adding differential privacy ensures that f remains an unbiased gradient of the MMD distance between prediction distributions at the communication round, preserving the validity of all preceding arguments. However, the expected gradient corresponds to a different objective—namely, the MMD-fair learning problem (4) with kernel $\tilde{\kappa}$ instead of κ . An interesting corollary of Proposition 5.2 arises when differential privacy is enforced using a Gaussian mechanism.

Corollary 5.3 (Differentiable Privacy with Gaussian Kernel). *Under the assumptions of Proposition 5.2, if κ is the Gaussian Kernel and μ_{DP} is a Gaussian distribution (i.e., a Gaussian mechanism is used), then $\tilde{\kappa}$ is a Gaussian kernel with increased variance.*

Corollary 5.3 immediately follows from Proposition 5.2 because the family of Gaussian distributions is closed under convolution.

6 EXPERIMENTS

We conduct classification experiments on both synthetic and real data to demonstrate that MMD-regularized federated learning performs comparably to centralized training on the global dataset while surpassing other fair federated learning methods in terms of global statistical parity. In all experiments, we report the degree of SP unfairness of the tested classifiers, which is defined as the absolute difference between the positive prediction frequencies in the two demographic groups corresponding to $A = 0$ and $A = 1$, respectively. All results are averaged over 10 different random seeds, and we use a distance-based kernel for training.

Synthetic Data First, to highlight the advantage of a global fairness regularizer, we illustrate the implications of Proposition 3.1 on learning performance. Consider a setting with $K = 10$ clients, each with equal weight ($\nu_k = \frac{1}{10}$ for all $k = 1, \dots, K$). For all $k = 1, \dots, K$, we assume that $\mathbb{P}_k[A = 0] = \mathbb{P}_k[A = 1] = \frac{1}{2}$ and that $X \sim \mathcal{N}(\mu(k, A), \mathbb{I})$, where \mathbb{I} is the 10-dimensional identity matrix, and $\mu(k, a)$ is a vector with all entries equal to 1 (−1) if $k + a$ is even (odd). The label is set to $Y = 1$ if $\mathbb{1}^\top X > 0$ and to $Y = 0$ otherwise. We sample 200 data points per client and compare our approach to two baselines. In the *centralized* method, all clients send their data to a server, which solves (4) on the pooled dataset. In the *local group fairness* method, federated learning is performed, but each client k uses the local fairness regularizer

$$\mathcal{D}_{\text{MMD}} \left(\mathbb{P}_k^{h_\theta(X)|A=0}, \mathbb{P}_k^{h_\theta(X)|A=1} \right)^2.$$

We also compare against FairFed (Ezzeldin et al., 2023), which employs an adaptive, fairness-aware client reweighting and can be combined with local fairness methods. For a fair comparison, we integrate FairFed’s global fair aggregation with MMD-based local fair training. All methods train a fairness-regularized logistic regression model. Figure 2 presents accuracy and unfairness profiles for different regularization weights λ . Our approach, like the centralized method, maintains accuracy even for higher values of λ , but it achieves global fairness already for low λ . In contrast, the local group fairness method significantly biases the model when fairness regularization is imposed, and reducing client-level unfairness does not necessarily lower global unfairness. FairFed suffers from similar problems, likely due to its combination of fair aggregation with a local fairness penalty. The Bayes classifier on the global distribution \mathbb{P} , given by $\mathbb{1}[\mathbb{1}^\top X > 0]$, achieves perfect accuracy while satisfying statistical parity, as $\mathbb{P}[\mathbb{1}^\top X > 0|A = 0] = \mathbb{P}[\mathbb{1}^\top X > 0|A = 1] = \frac{1}{2}$. However, when restricted to a single client, the Bayes classifier no longer satisfies statistical parity.

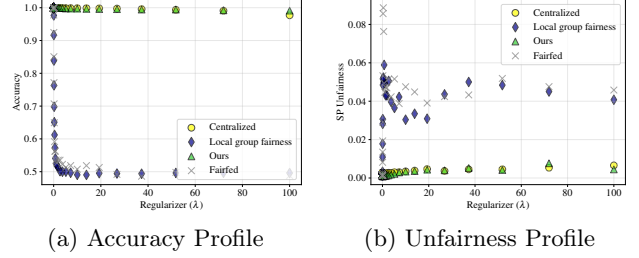


Figure 2: Performance comparison on synthetic data

Communities & Crime Dataset We evaluate our algorithm on the Communities & Crime dataset.¹ This dataset includes law enforcement and socio-economic data for 1,994 US communities, described by 99 features. The task is to classify whether the incidence of violent crimes exceeds the national average. Following Calders et al. (2013), we define a binary protected attribute by thresholding the percentage of Black or African American residents at the median. We assume data is stored by state, with states unwilling to share their communities’ data. States with fewer than four communities are removed. All methods train a two-layer feedforward neural network with 16 hidden units and ReLU activation. We compare our approach, which solves (4) using Algorithm 1 with $|\mathcal{Y}_0| = |\mathcal{Y}_1| = 100$, against several baselines: a *centralized* model trained on aggregated data using SGD on (4); FedAvg with a local MMD-fairness regularizer (*local group fairness*); *agnostic* federated learning (Mohri et al., 2019), which enforces client fairness without additional hyperparameters; *FedFB*, a federated version of FairBatch (Zeng et al., 2021); and FairFed (Ezzeldin et al., 2023), which combines fair aggregation with local fairness penalties. Hyperparameter details are provided in Appendix C. Figure 3a presents the Pareto frontier of the accuracy-unfairness tradeoff as we vary the regularization parameter λ . We observe that (Mohri et al., 2019) fails to induce group fairness, as client fairness does not necessarily imply global group fairness. Furthermore, our method outperforms the local group fairness approach, indicating bias heterogeneity across clients, consistent with our synthetic experiment. Finally, FairFed performs comparably to our method, and both achieve a fairness-accuracy tradeoff similar to training on aggregated data.

Compas Dataset For our next experiment, we consider the COMPAS dataset.² It contains records of defendants in Broward County, Florida, described by eight features, including age, gender, race, and prior offenses—features used by the COMPAS (Correctional

¹<http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime>

²<https://www.kaggle.com/danofer/compass>

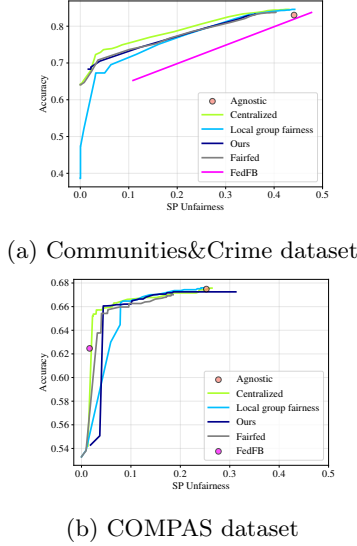


Figure 3: Performance comparison on real data

Offender Management Profiling for Alternative Sanctions) algorithm. We use race as the protected attribute and predict recidivism within two years of the original offense. After preprocessing, the dataset contains 5,278 datapoints, which we assume are distributed across $K = 3$ clients, split by age group (25 or younger, 25–45, and 45 or older). All hyperparameters and baselines remain the same as in the Communities & Crime experiment. Figure 3b presents the Pareto frontier of the accuracy-unfairness tradeoff. At high accuracy levels, both our method and the local group fairness approach perform comparably to the centralized baseline. However, the local group fairness method sacrifices significant accuracy to achieve fairness below 0.1, whereas our approach maintains competitiveness with the centralized method even at fairness values of 0.05. For stronger regularization, our method exhibits lower accuracy and higher unfairness, which we attribute to noise from sampling the sets \mathcal{Y}_0 and \mathcal{Y}_1 . Addressing this would require tuning the step size and the sizes of \mathcal{Y}_0 and \mathcal{Y}_1 , but we opted against this to ensure a fair comparison. FedFB achieves strong fairness but at the cost of accuracy, and its accuracy-fairness tradeoff is not adjustable. FairFed performs well but does not match the performance of our method.

In summary, the proposed fairness-regularized algorithm produces globally fair predictors without unnecessary accuracy loss. Moreover, it performs comparably to centralized training while sharing only $|\mathcal{Y}_0| = |\mathcal{Y}_1| = 100$ samples. As shown in Appendix C.3, the algorithm is robust to variations in the sample sizes.

7 Conclusions

We introduce a global fairness regularizer for federated learning and propose a distributed training method to enhance global group fairness in federated algorithms. When applying such regularizers, it is essential to carefully assess the appropriate fairness notion (if any) and recognize that fairness regularization can have unintended effects, particularly on heavily biased distributions. Our approach is compatible with most group fairness notions, as shown in Table 1. Additionally, federated learning algorithms like FedAvg can lead to data leakage even when only gradients are shared. Thus, to strengthen security and privacy in practical applications, our methods should be combined with differential privacy (Dwork et al., 2006; McMahan et al., 2018) or homomorphic encryption (Nikolaenko et al., 2013).

Acknowledgments This work was supported as a part of the NCCR Automation, a National Centre of Competence in Research, funded by the Swiss National Science Foundation (grant number 51NF40_225155).

References

- Annie Abay, Yi Zhou, Nathalie Baracaldo, Shashank Rajamoni, Ebube Chuba, and Heiko Ludwig. Mitigating bias in federated learning. *arXiv preprint arXiv:2012.02447*, 2020. (Cited on page 3.)
- Alekh Agarwal, Miroslav Dudik, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pages 120–129, 2019. (Cited on page 4.)
- Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pages 254–264. Auerbach Publications, 2022. (Cited on page 1.)
- Borja Balle and Yu-Xiang Wang. Improving the Gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403, 2018. (Cited on page 7.)
- Toon Calders, Asim Karim, Faisal Kamiran, Wasif Ali, and Xiangliang Zhang. Controlling attribute effect in linear regression. In *IEEE International Conference on Data Mining*, pages 71–80, 2013. (Cited on page 8.)
- Ziheng Cheng, Xinmeng Huang, Pengfei Wu, and Kun Yuan. Momentum benefits non-iid federated learning simply and provably. In *International Conference on Learning Representations*, 2024. (Cited on page 7.)

- Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. (Cited on page 1.)
- Lingyang Chu, Lanjun Wang, Yanjie Dong, Jian Pei, Zirui Zhou, and Yong Zhang. Fedfair: Training fair models in cross-silo federated learning. *arXiv preprint arXiv:2109.05662*, 2021. (Cited on page 3.)
- Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806, 2017. (Cited on page 4.)
- Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. In *Advances in Neural Information Processing Systems*, pages 26091–26102, 2021. (Cited on page 3.)
- Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women. <https://reuters/20d9fPr>, 2018. (Cited on page 1.)
- Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Distributionally robust federated averaging. In *Advances in neural information processing systems*, pages 15111–15122, 2020. (Cited on pages 3 and 21.)
- Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, pages 325–333, 2018. (Cited on pages 2, 3, and 4.)
- Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *International Conference on Data Mining*, pages 181–189, 2021. (Cited on page 3.)
- Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography*, pages 265–284, 2006. (Cited on pages 3 and 9.)
- Ahmed El Ouadrhiri and Ahmed Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380, 2022. (Cited on page 7.)
- Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *AAAI Conference on Artificial Intelligence*, pages 7494–7502, 2023. (Cited on pages 3, 8, and 22.)
- Faisal Hamman and Sanghamitra Dutta. Demystifying local & global fairness trade-offs in federated learning using partial information decomposition. In *International Conference on Learning Representations*, 2023. (Cited on pages 3 and 4.)
- Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, pages 3315–3323, 2016. (Cited on pages 3 and 4.)
- Shengyuan Hu, Zhiwei Steven Wu, and Virginia Smith. Fair federated learning via bounded group loss. In *Conference on Secure and Trustworthy Machine Learning*, pages 140–160, 2024a. (Cited on page 3.)
- Yifan Hu. Stochastic biased gradient methods. *Encyclopedia of Optimization (in press)*, pages 1–9, 2024. (Cited on page 7.)
- Yifan Hu, Jie Wang, Xin Chen, and Niao He. Multi-level Monte-Carlo gradient methods for stochastic optimization with biased oracles. *arXiv preprint arXiv:2408.11084*, 2024b. (Cited on page 7.)
- Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, and Rachel Cummings. Advances and open problems in federated learning et al. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. (Cited on page 3.)
- Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143, 2020. (Cited on pages 2, 7, 17, 18, 19, and 20.)
- Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local SGD on identical and heterogeneous data. In *Artificial Intelligence and Statistics*, pages 4519–4529, 2020. (Cited on page 2.)
- Fragkiskos Koufogiannis, Shuo Han, and George J Pappas. Optimality of the Laplace mechanism in differential privacy. *arXiv preprint arXiv:1504.00065*, 10, 2015. (Cited on page 7.)
- Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020. (Cited on page 2.)
- Michael Lohaus, Michaël Perrot, and Ulrike von Luxburg. Too relaxed to be fair. In *International Conference on Machine Learning*, pages 6360–6369, 2020. (Cited on page 2.)

- Subha Maity, Debarghya Mukherjee, Mikhail Yurochkin, and Yuekai Sun. Does enforcing fairness mitigate biases caused by subpopulation shift? In *Advances in Neural Information Processing Systems*, pages 25773–25784, 2021. (Cited on pages 3 and 4.)
- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017. (Cited on pages 1, 2, and 5.)
- H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. (Cited on pages 3 and 9.)
- Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pages 4615–4625, 2019. (Cited on pages 3, 8, and 21.)
- Valeria Nikolaenko, Udi Weinsberg, Stratis Ioannidis, Marc Joye, Dan Boneh, and Nina Taft. Privacy-preserving ridge regression on hundreds of millions of records. In *Symposium on Security and Privacy*, pages 334–348, 2013. (Cited on pages 3 and 9.)
- Luca Oneto, Michele Donini, Giulia Luise, Carlo Ciliberto, Andreas Maurer, and Massimiliano Pontil. Exploiting MMD and Sinkhorn divergences for fair and transferable representation learning. In *Advances in Neural Information Processing Systems*, pages 15360–15370, 2020. (Cited on pages 2 and 3.)
- Afroditi Papadaki, Natalia Martinez, Martin Bertran, Guillermo Sapiro, and Miguel Rodrigues. Minimax demographic group fairness in federated learning. In *Conference on Fairness, Accountability, and Transparency*, 2022. (Cited on page 3.)
- Sikha Pentyala, Nicola Neophytou, Anderson Nascimento, Martine De Cock, and Golnoosh Farnadi. Privfairfl: Privacy-preserving group fairness in federated learning. *arXiv preprint arXiv:2205.11584*, 2022. (Cited on page 3.)
- Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems*, pages 5680–5689, 2017. (Cited on pages 3 and 4.)
- Flavien Prost, Hai Qian, Qiuwen Chen, Ed H Chi, Jilin Chen, and Alex Beutel. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *arXiv preprint arXiv:1910.11779*, 2019. (Cited on page 2.)
- Laurent Risser, Alberto Gonzalez Sanz, Quentin Vincenot, and Jean-Michel Loubes. Tackling algorithmic bias in neural-network classifiers using Wasserstein-2 regularization. *Journal of Mathematical Imaging and Vision*, 64(6):672–689, 2022. (Cited on page 2.)
- Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. In *International Conference on Learning Representations*, 2021. (Cited on page 3.)
- Yves Rychener, Bahar Taskesen, and Daniel Kuhn. Metrizing fairness. *arXiv preprint arXiv:2205.15049*, 2022. (Cited on pages 2, 3, 4, and 13.)
- Teresa Salazar, Miguel Fernandes, Helder Araújo, and Pedro Henriques Abreu. Fair-fate: Fair federated learning with momentum. In *International Conference on Computational Science*, pages 524–538, 2023. (Cited on page 3.)
- Edward H Simpson. The interpretation of interaction in contingency tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 13(2):238–241, 1951. (Cited on page 1.)
- Bahar Taskesen, Viet Anh Nguyen, Daniel Kuhn, and Jose Blanchet. A distributionally robust approach to fair classification. *arXiv:2007.09530*, 2020. (Cited on pages 2 and 3.)
- Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soumya Kar. Matcha: Speeding up decentralized SGD via matching decomposition sampling. In *Indian Control Conference*, pages 299–300, 2019. (Cited on page 2.)
- Yongkai Wu, Lu Zhang, and Xintao Wu. On convexity and bounds of fairness-aware classification. In *The World Wide Web Conference*, pages 3356–3362, 2019. (Cited on pages 2 and 3.)
- Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *AAAI Conference on Artificial Intelligence*, pages 5693–5700, 2019. (Cited on page 2.)
- Xubo Yue, Maher Nouiehed, and Raed Al Kontar. Gifair-fl: A framework for group and individual fairness in federated learning. *arXiv preprint arXiv:2108.02741*, 2021. (Cited on page 3.)
- Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *International Conference on World Wide Web*, pages 1171–1180, 2017a. (Cited on page 2.)

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial Intelligence and Statistics*, pages 962–970, 2017b. (Cited on pages 2 and 4.)

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019. (Cited on page 2.)

Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021. (Cited on pages 3, 8, and 22.)

Checklist

1. For all models and algorithms presented, check if you include:
 - (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes:** All assumptions, algorithms and model settings can be either found in the main paper or in the appendix.
 - (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **Yes:** Section 5 gives an analysis of the central properties, generalization bounds are given in the appendix.
 - (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **Yes:** Anonymized source code is provided in the attached zip folder. All code will be released on publication.
2. For any theoretical claim, check if you include:
 - (a) Statements of the full set of assumptions of all theoretical results. **Yes:** All assumptions are given.
 - (b) Complete proofs of all theoretical results. **Yes:** Proofs are given in the appendix.
 - (c) Clear explanations of any assumptions. **Yes:** Assumptions are formulated in a verifiable manner. Where applicable, we discuss the validity of these assumptions in common settings.
3. For all figures and tables that present empirical results, check if you include:
 - (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes:** All code is provided in the attached zip file.
 - (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes:** Hyperparameters are provided in the appendix.
 - (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes:** All metrics are described.
 - (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **No:** We do not give timing results. We therefore feel that these details do not add to the understanding of the results.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
 - (a) Citations of the creator If your work uses existing assets. **Yes:** The datasets we use are cited and a link to obtain the data is provided.
 - (b) The license information of the assets, if applicable. **Not applicable:** Existing assets are only used as data for testing our code. No new assets are created.
 - (c) New assets either in the supplemental material or as a URL, if applicable. **No new assets are created.:** Information about consent from data providers/curators. **Not applicable:** We only use public datasets.
 - (d) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not applicable:** No personally identifiable information or offensive content is used.
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
 - (a) The full text of instructions given to participants and screenshots. **Not Applicable:** We only use public datasets, no crowdsourcing or human subjects were used.
 - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable:** We only use public datasets, no crowdsourcing or human subjects were used.
 - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable:** We only use public datasets, no crowdsourcing or human subjects were used.

Appendix

A Auxiliary Results

A.1 Generalization Bound

We formally restate that generalization bound by [Rychener et al. \(2022\)](#). To this end, we denote the empirical distribution constructed from N training samples $\{(X_i, Y_i, A_i)\}_{i=1}^N$ by $\hat{\mathbb{P}}$.

Proposition A.1 (Generalization Bounds ([Rychener et al., 2022](#), Theorem A.2)). *Suppose $\mathcal{H} \subseteq \mathcal{L}(\mathcal{X}, \{0, 1\})$ is a family of binary functions with VC dimension \mathcal{V} , and let \mathcal{D}_{MMD} be the MMD distance induced by a kernel κ . Select any $N \in \mathbb{N}$, and define $N_1 = \sum_{i=1}^N A_i$. Then, for any $h \in \mathcal{H}$ and $\delta \in (0, 1)$, conditioned on the event $\mathcal{V} < N_1 < N - \mathcal{V}$, the accuracy and MMD unfairness of h jointly satisfy*

$$\mathbb{P}[h(X) = Y] \geq \hat{\mathbb{P}}[h(X) = Y] - \frac{1}{2} \sqrt{\frac{4}{N} \left(\mathcal{V} \left(\log \left(\frac{2N}{\mathcal{V}} \right) + 1 \right) - \log(\delta/12) \right)}$$

and

$$\begin{aligned} \mathcal{D}_{\text{MMD}} \left(\mathbb{P}^{h(X)|A=0}, \mathbb{P}^{h(X)|A=1} \right) &\leq \mathcal{D}_{\text{MMD}} \left(\hat{\mathbb{P}}^{h(X)|A=0}, \hat{\mathbb{P}}^{h(X)|A=1} \right) + \frac{\alpha}{2} \sqrt{\frac{4}{N_1} \left(\mathcal{V} \left(\log \left(\frac{2N_1}{\mathcal{V}} \right) + 1 \right) - \log(\delta/12) \right)} \\ &\quad + \frac{\alpha}{2} \sqrt{\frac{4}{N - N_1} \left(\mathcal{V} \left(\log \left(\frac{2(N - N_1)}{\mathcal{V}} \right) + 1 \right) - \log(\delta/12) \right)} \end{aligned}$$

with probability at least $1 - \delta$, where $\alpha = \sup_{\psi \in \Psi} |\psi(0) - \psi(1)|$ and $\Psi = \{\psi \in \mathbb{H}_\kappa : \|\psi\|_{\mathbb{H}_\kappa} \leq 1\}$ denotes the unit ball in the reproducing kernel Hilbert space generated by the kernel κ .

We note that Proposition A.1 readily extends to other fairness notions beyond statistical parity.

A.2 Application to Other Fairness Notions

We now briefly outline how Algorithm 1 can be extended to other group fairness notions, such as those in Table 1. First, we require a fairness regularizer for each conditioning set $\mathcal{C}_j \in \mathcal{C}$. Indexing the regularizers by $j \in \mathcal{J}$, the following changes must be made compared to Algorithm 1 described in Section 4.

1. Instead of sampling predictions from $\mathbb{P}^{h_\theta(X)|A=a}$ and collecting them in \mathcal{Y}_a for any $a \in \mathcal{A}$, we must sample predictions from $\mathbb{P}^{S(h_\theta(X), Y)|A=a, (X, Y) \in \mathcal{C}_j}$ and collect them in $\mathcal{Y}_{a,j}$ for every $a \in \mathcal{A}$ and $j \in \mathcal{J}$.
2. Instead of decomposing f into client-wise functions f_k , we need to decompose an analogue f^j of f , one for every $j \in \mathcal{J}$, into client-wise functions f_k^j . These functions are defined as

$$\begin{aligned} f_k^j(\theta; \mathcal{Y}_{0,j}, \mathcal{Y}_{1,j}) &= 2 \left(\alpha_{k,j}^0 \int_{\mathbb{R}^n} C \left(S(h_{\theta^t}(X), Y); \mathcal{Y}_{0,j}, \mathcal{Y}_{1,j} \right) \mathbb{P}_k^{X|A=0, (X, Y) \in \mathcal{C}_j}(\mathrm{d}x) \right. \\ &\quad \left. - \alpha_{k,j}^1 \int_{\mathbb{R}^n} C \left(S(h_{\theta^t}(X), Y); \mathcal{Y}_{0,j}, \mathcal{Y}_{1,j} \right) \mathbb{P}_k^{X|A=1, (X, Y) \in \mathcal{C}_j}(\mathrm{d}x) \right). \end{aligned}$$

The main differences to the definition of f_k are that we evaluate C on $S(h_{\theta^t}(X), Y)$ instead of $h_{\theta^t}(X)$ and that we not only condition on $A = a$ but also on $(X, Y) \in \mathcal{C}_j$.

3. Instead of computing the weights α_k^a , we need to construct more structured weights $\alpha_{k,j}^a$ by not only conditioning on $A = a$ but also on $(X, Y) \in \mathcal{C}_j$. Specifically, we define

$$\alpha_{k,j}^a = \frac{\mathbb{P}_k[A = a, (X, Y) \in \mathcal{C}_j]}{\mathbb{P}[A = a, (X, Y) \in \mathcal{C}_j]}.$$

These generalizations lead to Algorithm 2 below.

Algorithm 2 General MMD-Fair FedAvg

```

1: Initialize  $\theta^1$ 
2: Collect information to estimate  $\alpha_{k,j}^a, a \in \mathcal{A}, j \in \mathcal{J}$ 
3: Communicate  $\alpha_{k,j}^a, a \in \mathcal{A}, j \in \mathcal{J}$ , to each client  $k$ 
4: for  $t = 1, \dots, T$  do
5:   Construct  $\mathcal{Y}_{a,j}^t, a \in \mathcal{A}$ , using  $\theta^t$ 
6:   Communicate  $\theta^t$  and  $\mathcal{Y}_{a,j}^t, a \in \mathcal{A}, j \in \mathcal{J}$ , to all clients
7:   Sample subset  $\mathcal{S} \subseteq \{1, \dots, K\}$  of  $S$  clients
8:   for client  $k \in \mathcal{S}$  in parallel do
9:     Initialize  $\theta_{1,k}^t = \theta^t$ 
10:    for local iteration  $i = 1, \dots, E$  do
11:      Compute a stochastic gradient estimator  $g_{i,k}^t$  such that:
          
$$\mathbb{E} [g_{i,k}^t | \mathcal{Y}_{a,j}^t, \forall a \in \mathcal{A}, \forall j \in \mathcal{J}] = \nabla_{\theta} (\mathbb{E}_k [L(h_{\theta_{i,k}^t}(X), Y)] + \lambda \sum_{j \in \mathcal{J}} f_k^j(\theta_{i,k}^t; \mathcal{Y}_{0,j}^t, \mathcal{Y}_{1,j}^t))$$

12:      Update  $\theta_{i+1,k}^t = \theta_{i,k}^t - \eta g_{i,k}^t$ 
13:    end for
14:    Communicate  $\theta_{E+1,k}^t$  to server
15:  end for
16:  Update  $\theta^{t+1} = \theta^t + \eta_g \left( \frac{1}{S} \sum_{k \in \mathcal{S}} (\theta_{E+1,k}^t - \theta^t) \right)$ 
17: end for
    
```

B Proofs

We now present all proofs, organized by section.

B.1 Proofs of Section 3

Proof of Proposition 3.1. We prove the proposition by constructing a distribution \mathbb{P} for which the equality cannot hold for any dissimilarity measure \mathcal{D} satisfying the identity of indiscernibles and any regularization function ρ with the given properties. For simplicity, assume $K = 2$; the extension to higher K is straightforward.

Assume, for contradiction, that \mathcal{D} satisfies the identity of indiscernibles. Consider the distributions

$$\mathbb{P}_1^{h(X)|A=0} = \mathbb{P}_2^{h(X)|A=1} \neq \mathbb{P}_1^{h(X)|A=1} = \mathbb{P}_2^{h(X)|A=0}$$

with mixing weights $\nu_1 = \nu_2 = 0.5$. It is easily verified that

$$\frac{1}{2} \mathbb{P}_1^{h(X)|A=1} + \frac{1}{2} \mathbb{P}_2^{h(X)|A=1} = \frac{1}{2} \mathbb{P}_1^{h(X)|A=0} + \frac{1}{2} \mathbb{P}_2^{h(X)|A=0}.$$

Thus,

$$\mathcal{D} \left(\frac{1}{2} \mathbb{P}_1^{h(X)|A=1} + \frac{1}{2} \mathbb{P}_2^{h(X)|A=1}, \frac{1}{2} \mathbb{P}_1^{h(X)|A=0} + \frac{1}{2} \mathbb{P}_2^{h(X)|A=0} \right) = 0.$$

However, by the identity of indiscernibles,

$$\mathcal{D}(\mathbb{P}_1^{h(X)|A=1}, \mathbb{P}_1^{h(X)|A=0}) > 0 \quad \text{and} \quad \mathcal{D}(\mathbb{P}_2^{h(X)|A=1}, \mathbb{P}_2^{h(X)|A=0}) > 0.$$

Since $\rho(v) = 0$ if and only if $v = 0$, the equality in the proposition statement cannot hold, leading to a contradiction. Thus, the claim follows. \square

B.2 Proofs of Section 4

Proof of Lemma 4.1. An elementary calculation shows that

$$\begin{aligned} & \nabla_{\theta} f(\theta; \mathcal{Y}_0, \mathcal{Y}_1) \\ &= 2 \nabla_{\theta} \left(\int_{\mathbb{R}^n} C(h_{\theta}(x); \mathcal{Y}_0, \mathcal{Y}_1) \mathbb{P}^{X|A=0}(\mathrm{d}x) - \int_{\mathbb{R}^n} C(h_{\theta}(x); \mathcal{Y}_0, \mathcal{Y}_1) \mathbb{P}^{X|A=1}(\mathrm{d}x) \right) \end{aligned}$$

$$\begin{aligned}
 &= 2 \left(\int_{\mathbb{R}^n} C'(h_\theta(x); \mathcal{Y}_0, \mathcal{Y}_1) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=0}(\mathrm{d}x) - \int_{\mathbb{R}^n} C'(h_\theta(x); \mathcal{Y}_0, \mathcal{Y}_1) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=1}(\mathrm{d}x) \right) \\
 &= 2 \left(\int_{\mathbb{R}^n} \left(\frac{1}{|\mathcal{Y}_0|} \sum_{y \in \mathcal{Y}_0} \kappa^{(1)}(h_\theta(x), y) - \frac{1}{|\mathcal{Y}_1|} \sum_{y \in \mathcal{Y}_1} \kappa^{(1)}(h_\theta(x), y) \right) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=0}(\mathrm{d}x) \right. \\
 &\quad \left. - \int_{\mathbb{R}^n} \left(\frac{1}{|\mathcal{Y}_0|} \sum_{y \in \mathcal{Y}_0} \kappa^{(1)}(h_\theta(x), y) - \frac{1}{|\mathcal{Y}_1|} \sum_{y \in \mathcal{Y}_1} \kappa^{(1)}(h_\theta(x), y) \right) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=1}(\mathrm{d}x) \right),
 \end{aligned}$$

where we use $\kappa^{(1)}$ to denote derivative of the kernel with respect to its first argument. Therefore, we find

$$\begin{aligned}
 &\mathbb{E} [\nabla_\theta f(\theta; \mathcal{Y}_0, \mathcal{Y}_1)] \\
 &= 2 \left(\int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=0}(\mathrm{d}x') \right. \right. \\
 &\quad \left. \left. - \int_{\mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=1}(\mathrm{d}x') \right) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=0}(\mathrm{d}x) \right. \\
 &\quad \left. - \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=0}(\mathrm{d}x') \right. \right. \\
 &\quad \left. \left. - \int_{\mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=1}(\mathrm{d}x') \right) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=1}(\mathrm{d}x) \right),
 \end{aligned} \tag{5}$$

where we used the identity

$$\mathbb{E} \left[\frac{1}{|\mathcal{Y}_a|} \sum_{y \in \mathcal{Y}_a} \kappa^{(1)}(y', y) \right] = \int_{\mathbb{R}^n} \kappa^{(1)}(y', h_\theta(x)) \mathbb{P}^{X|A=a}(\mathrm{d}x) \quad \forall a \in \mathcal{A}, y' \in \mathbb{R}^n.$$

Next, we address the gradient of the fairness loss. By the definition of the MMD metric, we have

$$\begin{aligned}
 \nabla_\theta \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2 &= \nabla_\theta \left(\int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=0}(\mathrm{d}x) \mathbb{P}^{X|A=0}(\mathrm{d}x') \right. \\
 &\quad \left. + \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=1}(\mathrm{d}x) \mathbb{P}^{X|A=1}(\mathrm{d}x') \right. \\
 &\quad \left. - 2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=0}(\mathrm{d}x) \mathbb{P}^{X|A=1}(\mathrm{d}x') \right).
 \end{aligned}$$

Interchanging the gradient with the integrals using Leibniz's integral rule, which applies due to the theorem's assumptions, the last expression is equivalent to

$$\begin{aligned}
 \nabla_\theta \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2 &= \left(\int_{\mathbb{R}^n \times \mathbb{R}^n} \nabla_\theta \kappa(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=0}(\mathrm{d}x) \mathbb{P}^{X|A=0}(\mathrm{d}x') \right. \\
 &\quad \left. + \int_{\mathbb{R}^n \times \mathbb{R}^n} \nabla_\theta \kappa(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=1}(\mathrm{d}x) \mathbb{P}^{X|A=1}(\mathrm{d}x') \right. \\
 &\quad \left. - 2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \nabla_\theta \kappa(h_\theta(x), h_\theta(x')) \mathbb{P}^{X|A=0}(\mathrm{d}x) \mathbb{P}^{X|A=1}(\mathrm{d}x') \right)
 \end{aligned}$$

By leveraging the chain rule for multivariable functions, we then obtain

$$\nabla_\theta \kappa(h_\theta(x), h_\theta(x')) = \kappa^{(1)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x) + \kappa^{(2)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x'),$$

where $\kappa^{(1)}(x, x') = \partial \kappa(x, x') / \partial x$ and $\kappa^{(2)}(x, x') = \partial \kappa(x, x') / \partial x'$. By symmetry of the kernel κ , we thus find

$$\nabla_\theta \kappa(h_\theta(x), h_\theta(x')) = \kappa^{(1)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x) + \kappa^{(1)}(h_\theta(x'), h_\theta(x)) \nabla_\theta h_\theta(x').$$

In summary, we therefore have

$$\nabla_\theta \mathcal{D}_{\text{MMD}}(\mathbb{P}^{h_\theta(X)|A=0}, \mathbb{P}^{h_\theta(X)|A=1})^2$$

$$\begin{aligned}
 &= \left(2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=0}(\mathrm{d}x) \mathbb{P}^{X|A=0}(\mathrm{d}x') \right. \\
 &\quad + 2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=1}(\mathrm{d}x) \mathbb{P}^{X|A=1}(\mathrm{d}x') \\
 &\quad - 2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=0}(\mathrm{d}x) \mathbb{P}^{X|A=1}(\mathrm{d}x') \\
 &\quad \left. - 2 \int_{\mathbb{R}^n \times \mathbb{R}^n} \kappa^{(1)}(h_\theta(x), h_\theta(x')) \nabla_\theta h_\theta(x) \mathbb{P}^{X|A=1}(\mathrm{d}x) \mathbb{P}^{X|A=0}(\mathrm{d}x') \right).
 \end{aligned}$$

The claim then follows by comparing this expression with (5). \square

B.3 Proofs of Section 5

B.3.1 Proofs of Section 5.1

The proof of Theorem 5.1 follows from the following, more general theorem for biased local gradients. It shows the convergence rate of Algorithm 3, which is an abstraction of Algorithm 1.

Algorithm 3 FedAvg with Function-Tracking

```

1: Initialize  $\theta^1$ 
2: for  $t = 1, \dots, T$  do
3:   Build function estimates  $f_k^t$ 
4:   Communicate  $\theta^t$  to clients
5:   Sample subset  $\mathcal{S} \subseteq \{1, \dots, K\}$  of  $S$  clients
6:   for  $k \in \mathcal{S}$  in parallel do
7:     Initialize  $\theta_{1,k}^t = \theta^t$ 
8:     for epoch  $i = 1, \dots, E$  do
9:       Compute stochastic gradient  $g_k^t(\theta_{i,k}^t)$  of  $f_k^t(\theta_{i,k}^t)$ 
10:      Update  $\theta_{i+1,k}^t = \theta_{i,k}^t - \eta_l g_k^t(\theta_{i,k}^t)$ 
11:    end for
12:    Communicate  $\theta_{E+1,k}^t$  to server
13:  end for
14:  Update  $\theta^{t+1} = \theta^t + \eta_g \left( \frac{1}{S} \sum_{k \in \mathcal{S}} (\theta_{E+1,k}^t - \theta^t) \right)$ 
15: end for
    
```

Assumption B.1 (Unbiased Function Estimates). *Let f_k^t be an unbiased estimator for the objective function F in the sense that $\mathbb{E}[\nabla_\theta \sum_{k=1}^K f_k^t(\theta^t)] = \nabla_\theta F(\theta^t)$, where θ^t is the t -th global iterate of Algorithm 3. Assume further that the variance of $\nabla_\theta \frac{1}{K} \sum_{k=1}^K f_k^t(\theta^t)$ is bounded by ω^2 .*

Assumption B.2 (G-B Gradient Dissimilarity). *There exist constants $G \geq 0$ and $B \geq 1$ such that*

$$\frac{1}{K} \sum_{k=1}^K \|\nabla_\theta f_k^t(\theta)\|^2 \leq G^2 + B^2 \left\| \nabla_\theta \frac{1}{K} \sum_{k=1}^K f_k^t(\theta) \right\|^2.$$

Theorem B.3 (Convergence Rate of Algorithm 3). *Assume that the estimators f_k^t and the objective function F are β -smooth and that g_k^t are unbiased gradient estimators of f_k^t with variance at most σ^2 . If $E \in \mathbb{N}$, $\eta_l > 0$, $\eta_g \geq 1$, and the pseudo-stepsize $\tilde{\eta} = E\eta_l\eta_g$ satisfies $\tilde{\eta} \leq (60\beta(B^2 + 1))^{-1}$, and if Assumptions B.1 and B.2 hold, then there exist probabilities $\{w_t\}_{t=1}^T$, such that*

$$\mathbb{E} \|\nabla F(\bar{\theta})\|^2 \leq \mathcal{O} \left(\frac{\beta B^2 \bar{F}}{T+1} + \frac{\beta M_1 \bar{F}^{\frac{1}{2}}}{\sqrt{T+1}} + \frac{\beta^{\frac{2}{3}} M_2 \bar{F}^{\frac{2}{3}}}{(T+1)^{\frac{2}{3}}} \right),$$

where $\bar{\theta} = \theta^t$ with probability w_t , θ^t is the t -th global iterate of Algorithm 3 for every $t = 1, \dots, T$, while

$$\bar{F} = \|\nabla F(\theta^0)\|^2,$$

$$M_1^2 = \frac{\sigma^2}{2EK} + \left(1 - \frac{S}{K}\right) \frac{2}{S} G^2 + 2(B^2 + 1)\omega^2,$$

$$M_2 = \left(\frac{3\sigma^2}{E\eta_g^2} + 6(G^2 + \omega^2 B^2) \right)^{\frac{1}{3}}.$$

The proof of Theorem B.3 is inspired by (Karimireddy et al., 2020). We give it below for completeness.

Proof of Theorem B.3. Before proceeding with the proof, we define some notation. Let $\{\mathcal{F}_t\}_{t \geq 0}$ denote the filtration generated by the algorithm. We use the shorthand $\mathbb{E}^t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_t]$ and omit the superscript when clear from context. The objective function is denoted by $F(\theta)$, and at each communication round t , it is approximated by $f^t(\theta) = \frac{1}{K} \sum_{k=1}^K f_k^t(\theta)$. The proof follows three steps and parallels the convergence rate proof in (Karimireddy et al., 2020). To maintain alignment with their approach, we adopt the bound $\tilde{\eta} \leq (8\beta(B^2 + 1))^{-1}$ as in (Karimireddy et al., 2020), which is implied by our theorem assumptions. The bound $\tilde{\eta} \leq (60\beta(B^2 + 1))^{-1}$ is used at the final step together with (Karimireddy et al., 2020, Lemma 2).

Part 1: Bounding Client Drift In order to quantify how the local models of the clients drift away from the model in the last communication round, we bound

$$\mathcal{E}^t = \mathbb{E} \left[\frac{1}{EK} \sum_{i,k} \|\theta_{i,k}^t - \theta^t\|^2 \right].$$

Following Karimireddy et al. (2020), we can derive the following estimate.

$$\begin{aligned} \mathbb{E} \|\theta_{i,k}^t - \theta^t\|^2 &= \mathbb{E} \|\theta_{i-1,k}^t - \theta^t - \eta_l g_k^t(\theta_{i-1,k}^t)\|^2 \\ &\leq \mathbb{E} \|\theta_{i-1,k}^t - \theta^t - \eta_l \nabla f_k^t(\theta_{i-1,k}^t)\|^2 + \eta_l^2 \sigma^2 \\ &\leq \left(1 + \frac{1}{E-1}\right) \mathbb{E} \|\theta_{i-1,k}^t - \theta^t\|^2 + E\eta_l^2 \mathbb{E} \|\nabla f_k^t(\theta_{i-1,k}^t)\|^2 + \eta_l^2 \sigma^2 \\ &= \left(1 + \frac{1}{E-1}\right) \mathbb{E} \|\theta_{i-1,k}^t - \theta^t\|^2 + \frac{\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta_{i-1,k}^t)\|^2 + \frac{\tilde{\eta}^2 \sigma^2}{E^2 \eta_g^2} \\ &\leq \left(1 + \frac{1}{E-1}\right) \mathbb{E} \|\theta_{i-1,k}^t - \theta^t\|^2 + \frac{2\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta_{i-1,k}^t) - \nabla f_k^t(\theta^t)\|^2 + \frac{2\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 + \frac{\tilde{\eta}^2 \sigma^2}{E^2 \eta_g^2} \\ &\leq \left(1 + \frac{1}{E-1} + \frac{2\tilde{\eta}^2 \beta^2}{E\eta_g}\right) \mathbb{E} \|\theta_{i-1,k}^t - \theta^t\|^2 + \frac{2\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 + \frac{\tilde{\eta}^2 \sigma^2}{E^2 \eta_g^2} \\ &\leq \left(1 + \frac{2}{E-1}\right) \mathbb{E} \|\theta_{i-1,k}^t - \theta^t\|^2 + \frac{2\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 + \frac{\tilde{\eta}^2 \sigma^2}{E^2 \eta_g^2} \end{aligned}$$

We first use the definition of the local update, followed by the mean and variance of the randomness of $g_k^t(\theta_{i-1,k}^t)$. The remaining steps follow from a repeated application of the relaxed triangle inequality (Karimireddy et al., 2020, Lemma 3), the definition of $\tilde{\eta} = E\eta_g\eta_l$, bounds on $\tilde{\eta}$ and η_g and the β -smoothness of f_k^t . Unrolling computations, we find that

$$\begin{aligned} \mathbb{E} \|\theta_{i,k}^t - \theta^t\|^2 &\leq \sum_{\tau=1}^{i-1} \left(\frac{2\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 + \frac{\tilde{\eta}^2 \sigma^2}{E^2 \eta_g^2} \right) \left(1 + \frac{2}{E-1}\right)^\tau \\ &\leq \left(\frac{2\tilde{\eta}^2}{E\eta_g} \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 + \frac{\tilde{\eta}^2 \sigma^2}{E^2 \eta_g^2} \right) 3E. \end{aligned}$$

Averaging over i and k (recalling again that $\eta_g \geq 1$, we get

$$\begin{aligned} \mathcal{E}^t &\leq \frac{1}{K} \sum_{k \in [K]} 6\tilde{\eta}^2 \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 + \frac{3\tilde{\eta}^2 \sigma^2}{E\eta_g^2} \\ &\leq \frac{3\tilde{\eta}^2 \sigma^2}{E\eta_g^2} + 6\tilde{\eta}^2 G^2 + 6\tilde{\eta}^2 B^2 \mathbb{E} \|\nabla f^t(\theta^t)\|^2 \end{aligned}$$

$$\leq \frac{3\tilde{\eta}^2\sigma^2}{E\eta_g^2} + 6\tilde{\eta}^2(G^2 + \omega^2 B^2) + 6\tilde{\eta}^2 B^2 \|\nabla F(\theta^t)\|^2, \quad (6)$$

where the last step uses the bounded mean-variance decomposition together with the bound on variance of f^t and Assumption B.1.

Part 2: Linearize Progress We next estimate the progress made in each communication round using a linearization of the objective. It satisfies

$$\begin{aligned} \langle \nabla F(\theta^t), \mathbb{E}[\theta^{t+1} - \theta^t] \rangle &= -\frac{\tilde{\eta}}{EK} \sum_{i,k} \langle \nabla F(\theta^t), \mathbb{E}[\nabla f_k^t(\theta_{i,k}^t)] \rangle \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \frac{\tilde{\eta}}{2} \left\| \frac{1}{EK} \sum_{i,k} \mathbb{E}[\nabla f_k^t(\theta_{i,k}^t)] - \nabla F(\theta^t) \right\|^2 \\ &= -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \frac{\tilde{\eta}}{2} \left\| \frac{1}{EK} \sum_{i,k} \mathbb{E}[\nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t) + \nabla f_k^t(\theta^t)] - \nabla F(\theta^t) \right\|^2 \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \tilde{\eta} \left\| \frac{1}{EK} \sum_{i,k} \mathbb{E}[\nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t)] \right\|^2 + \tilde{\eta} \|\mathbb{E}[\nabla f^t(\theta^t)] - \nabla F(\theta^t)\|^2 \\ &= -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \tilde{\eta} \left\| \frac{1}{EK} \sum_{i,k} \mathbb{E}[\nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t)] \right\|^2 \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \frac{\tilde{\eta}}{EK} \sum_{i,k} \|\mathbb{E}[\nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t)]\|^2 \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \frac{\tilde{\eta}}{EK} \sum_{i,k} \mathbb{E} \|\nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t)\|^2 \\ &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \tilde{\eta} \beta^2 \mathcal{E}^t. \end{aligned} \quad (7)$$

The first equality follows from the definition of the communication round update, we then use the inequality $-ab \leq \frac{1}{2}((b-a)^2 - a^2)$. The third equation follows from Assumption B.1. The inequalities follow from the relaxed triangle inequality (Karimireddy et al., 2020, Lemma 3) and Jensen's inequality.

Part 3: Second-Order Terms Following again Karimireddy et al. (2020)

$$\begin{aligned} \mathbb{E}[\|\theta^{t+1} - \theta^t\|^2] &= \mathbb{E}[\| -\frac{\tilde{\eta}}{ES} \sum_{i=1}^E \sum_{k \in S} g_k^t(\theta_{i,k}^t) \|^2] \\ &\leq \tilde{\eta}^2 \mathbb{E}[\| \frac{1}{ES} \sum_{i=1}^E \sum_{k \in S} \nabla f_k^t(\theta_{i,k}^t) \|^2] + \frac{\tilde{\eta}^2 \sigma^2}{EK} \\ &= \frac{\tilde{\eta}^2 \sigma^2}{EK} + \tilde{\eta}^2 \mathbb{E}[\| \frac{1}{ES} \sum_{i=1}^E \sum_{k \in S} \nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t) + \nabla f_k^t(\theta^t) \|^2] \\ &\leq \frac{\tilde{\eta}^2 \sigma^2}{EK} + 2\tilde{\eta}^2 \mathbb{E}[\| \frac{1}{ES} \sum_{i=1}^E \sum_{k \in S} \nabla f_k^t(\theta_{i,k}^t) - \nabla f_k^t(\theta^t) \|^2] + 2\tilde{\eta}^2 \mathbb{E}[\| \frac{1}{S} \sum_{k \in S} \nabla f_k^t(\theta^t) \|^2] \\ &\leq \frac{\tilde{\eta}^2 \sigma^2}{EK} + 2\beta^2 \tilde{\eta}^2 \mathcal{E}^t + 2\tilde{\eta}^2 \mathbb{E}[\| \frac{1}{S} \sum_{k \in S} \nabla f_k^t(\theta^t) \|^2] \\ &\leq \frac{\tilde{\eta}^2 \sigma^2}{EK} + 2\beta^2 \tilde{\eta}^2 \mathcal{E}^t + 4\tilde{\eta}^2 \mathbb{E} \|\nabla f^t(\theta^t)\|^2 + (1 - \frac{S}{K}) \frac{4\tilde{\eta}^2}{SK} \sum_{k=1}^K \mathbb{E} \|\nabla f_k^t(\theta^t)\|^2 \end{aligned}$$

$$\begin{aligned}
 &\leq \frac{\tilde{\eta}^2 \sigma^2}{EK} + 2\tilde{\eta}^2 \beta^2 \mathcal{E}^t + 4(B^2 + 1)\tilde{\eta}^2 \mathbb{E} \|\nabla f^t(\theta^t)\|^2 + (1 - \frac{S}{K}) \frac{4\tilde{\eta}^2}{S} G^2 \\
 &\leq \frac{\tilde{\eta}^2 \sigma^2}{EK} + 2\tilde{\eta}^2 \beta^2 \mathcal{E}^t + 4(B^2 + 1)\tilde{\eta}^2 \|\nabla F(\theta^t)\|^2 + (1 - \frac{S}{K}) \frac{4\tilde{\eta}^2}{S} G^2 + 4(B^2 + 1)\tilde{\eta}^2 \omega^2, \quad (8)
 \end{aligned}$$

where we first apply the definition of the local updates and the mean and variance of the randomness of $g_k^t(\theta_{i-1,k}^t)$. The second inequality then follows from the relaxed triangle inequality (Karimireddy et al., 2020, Lemma 3). We then apply the β -smoothness of f_k^t . The fourth and fifth inequality leverage the mean-variance decomposition and Assumption B.2 respectively. Finally, we apply the bounded mean-variance decomposition together with Assumption B.1.

Combining all Parts We will now combine all parts using the local expansion

$$\begin{aligned}
 \mathbb{E}[F(\theta^{t+1})] - F(\theta^t) &\leq \langle \nabla F(\theta), \mathbb{E}[\theta^{t+1} - \theta^t] \rangle + \frac{\beta}{2} \mathbb{E} [\|\theta^{t+1} - \theta^t\|^2] \\
 &\leq -\frac{\tilde{\eta}}{2} \|\nabla F(\theta^t)\|^2 + \tilde{\eta} \beta^2 \mathcal{E}^t \\
 &\quad + \frac{\beta}{2} \left(\frac{\tilde{\eta}^2 \sigma^2}{EK} + 2\tilde{\eta}^2 \beta^2 \mathcal{E}^t + 4(B^2 + 1)\tilde{\eta}^2 \|\nabla F(\theta^t)\|^2 + (1 - \frac{S}{K}) \frac{4\tilde{\eta}^2}{S} G^2 + 4(B^2 + 1)\tilde{\eta}^2 \omega^2 \right) \\
 &= \left(-\frac{\tilde{\eta}}{2} + 2(B^2 + 1)\beta\tilde{\eta}^2 \right) \|\nabla F(\theta^t)\|^2 + (\tilde{\eta}\beta^2 + \tilde{\eta}^2\beta^3) \mathcal{E}^t \\
 &\quad + \frac{\tilde{\eta}^2 \beta \sigma^2}{2EK} + (1 - \frac{S}{K}) \frac{2\tilde{\eta}^2 \beta}{S} G^2 + 2(B^2 + 1)\tilde{\eta}^2 \beta \omega^2 \\
 &\leq \left(-\frac{\tilde{\eta}}{2} + 2(B^2 + 1)\beta\tilde{\eta}^2 \right) \|\nabla F(\theta^t)\|^2 \\
 &\quad + (\tilde{\eta}\beta^2 + \tilde{\eta}^2\beta^3) \left(\frac{3\tilde{\eta}^2 \sigma^2}{E\eta_g^2} + 6\tilde{\eta}^2 (G^2 + \omega^2 B^2) + 6\tilde{\eta}^2 B^2 \|\nabla F(\theta^t)\|^2 \right) \\
 &\quad + \frac{\tilde{\eta}^2 \beta \sigma^2}{2EK} + (1 - \frac{S}{K}) \frac{2\tilde{\eta}^2 \beta}{S} G^2 + 2(B^2 + 1)\tilde{\eta}^2 \beta \omega^2 \\
 &= \left(-\frac{\tilde{\eta}}{2} + 2(B^2 + 1)\beta\tilde{\eta}^2 + 6\tilde{\eta}^3 \beta^2 B^2 + 6\tilde{\eta}^4 \beta^3 B^2 \right) \|\nabla F(\theta^t)\|^2 \\
 &\quad + (\tilde{\eta}\beta^2 + \tilde{\eta}^2\beta^3) \left(\frac{3\tilde{\eta}^2 \sigma^2}{E\eta_g^2} + 6\tilde{\eta}^2 (G^2 + \omega^2 B^2) \right) \\
 &\quad + \frac{\tilde{\eta}^2 \beta \sigma^2}{2EK} + (1 - \frac{S}{K}) \frac{2\tilde{\eta}^2 \beta}{S} G^2 + 2(B^2 + 1)\tilde{\eta}^2 \beta \omega^2 \\
 &\leq \tilde{\eta} \left(-\frac{1}{2} + \frac{2}{8} + \frac{6}{64} + \frac{6}{512} \right) \|\nabla F(\theta^t)\|^2 \\
 &\quad + \tilde{\eta} \beta^2 \left(1 + \frac{1}{8} \right) \left(\frac{3\tilde{\eta}^2 \sigma^2}{E\eta_g^2} + 6\tilde{\eta}^2 (G^2 + \omega^2 B^2) \right) \\
 &\quad + \frac{\tilde{\eta}^2 \beta \sigma^2}{2EK} + (1 - \frac{S}{K}) \frac{2\tilde{\eta}^2 \beta}{S} G^2 + 2(B^2 + 1)\tilde{\eta}^2 \beta \omega^2 \\
 &= -\frac{37\tilde{\eta}}{256} \|\nabla F(\theta^t)\|^2 \\
 &\quad + \frac{9\tilde{\eta}^3 \beta^2}{8} \left(\frac{3\sigma^2}{E\eta_g^2} + 6(G^2 + \omega^2 B^2) \right) \\
 &\quad + \tilde{\eta}^2 \beta \left(\frac{\sigma^2}{2EK} + (1 - \frac{S}{K}) \frac{2}{S} G^2 + 2(B^2 + 1)\omega^2 \right). \quad (9)
 \end{aligned}$$

The first inequality follows from the β -smoothness of F . We then apply equations (7) and (8). The third and fourth inequalities leverage equation (6) and the bound on the pseudo stepsize $\tilde{\eta} \leq (8\beta(B^2 + 1))^{-1}$, which implies $\tilde{\eta}\beta \leq 1/8$ and $\tilde{\eta}\beta B^2 \leq 1/8$. All equalities are rearranging terms. Take the average of (9) over all $t = 1, \dots, T$ and

apply (Karimireddy et al., 2020, Lemma 2) with

$$\begin{aligned}\eta &= \frac{37\tilde{\eta}}{256} \\ \eta_{\max} &= \frac{37}{2048\beta(B^2 + 1)} \\ c_1 &\propto \beta \left(\frac{\sigma^2}{2EK} + \left(1 - \frac{S}{K}\right) \frac{2}{S} G^2 + 2(B^2 + 1)\omega^2 \right) \\ c_2 &\propto \beta^2 \left(\frac{3\sigma^2}{E\eta_g^2} + 6(G^2 + \omega^2 B^2) \right).\end{aligned}$$

We have

$$\mathbb{E}\|\nabla F(\bar{\theta}^T)\|^2 \leq \mathcal{O} \left(\frac{\beta B^2 \bar{F}}{T+1} + \frac{\beta M_1 \bar{F}^{\frac{1}{2}}}{\sqrt{T+1}} + \frac{\beta^{\frac{2}{3}} M_2 \bar{F}^{\frac{2}{3}}}{(T+1)^{\frac{2}{3}}} \right),$$

where

$$\begin{aligned}\bar{F} &= \|\nabla F(\theta^0)\|^2 \\ M_1^2 &= \frac{\sigma^2}{2EK} + \left(1 - \frac{S}{K}\right) \frac{2}{S} G^2 + 2(B^2 + 1)\omega^2 \\ M_2 &= \left(\frac{3\sigma^2}{E\eta_g^2} + 6(G^2 + \omega^2 B^2) \right)^{\frac{1}{3}}.\end{aligned}$$

Thus, the claim follows. \square

We are now ready to prove Theorem 5.1.

Proof of Theorem 5.1. Let L_L, L_K, L_h and $\beta_L, \beta_K, \beta_h$ denote the Lipschitzness and smoothness parameters of L, κ and h_θ respectively. We apply Theorem B.3. The unbiasedness in Assumption B.1 is implied by Lemma 4.1. Since L, κ and h_θ are all Lipschitz continuous, so is f^t with Lipschitz constant $L_L L_h + 8\lambda L_K L_h$. This implies that $\|\nabla f^t(\theta)\| \leq L_L L_h + 8\lambda L_K L_h$, which by Popoviciu's inequality implies that $\omega^2 = \frac{1}{4}(L_L L_h + 8\lambda L_K L_h)^2$. The bound on $\|\nabla f^t(\theta)\|$ further implies that Assumption B.2 holds with $G = (L_L L_h + 8\lambda L_K L_h)^2$ and $B = 1$. The continuity and smoothness also imply that f_k^t and F are β -smooth for some β . Finally, we note that $g_{i,k}^t$ in Line 10 of Algorithm 1 is unbiased and has bounded variance by assumption. In summary, the assumptions of Theorem B.3 hold with the specific values

$$\begin{aligned}\omega^2 &= \frac{1}{4}(L_L L_h + 8\lambda L_K L_h)^2 \\ \sigma^2 &< \infty \\ G &= (L_L L_h + 8\lambda L_K L_h)^2 \\ L &= 1 \\ \beta &= \beta_L L_h + 8\beta_K L_h.\end{aligned}$$

\square

B.3.2 Proofs of Section 5.2

Proof of Proposition 5.2. The independence of the DP noise and linearity of expectation imply

$$\begin{aligned}\mathbb{E}_{\text{DP}} [C(z; \tilde{\mathcal{Y}}_0, \tilde{\mathcal{Y}}_1)] &= \mathbb{E}_{\text{DP}} \left[\frac{1}{|\mathcal{Y}_0|} \sum_{y \in \mathcal{Y}_0} \kappa(z, y + \xi_y) - \frac{1}{|\mathcal{Y}_1|} \sum_{y \in \mathcal{Y}_1} \kappa(z, y + \xi_y) \right] \\ &= \frac{1}{|\mathcal{Y}_0|} \sum_{y \in \mathcal{Y}_0} \mathbb{E}_{\text{DP}} [\kappa(z, y + \xi_y)] - \frac{1}{|\mathcal{Y}_1|} \sum_{y \in \mathcal{Y}_1} \mathbb{E}_{\text{DP}} [\kappa(z, y + \xi_y)].\end{aligned}$$

Taking $\tilde{\kappa}(x, y) = \mathbb{E}_{\text{DP}}[\kappa(z, y + \xi_y)]$, we now verify that $\tilde{\kappa}$ satisfies the conditions of Definition 3.2. We first show that $\tilde{\kappa}$ is symmetric PSD. For symmetry, note that

$$\tilde{\kappa}(x, y) = \mathbb{E}_{\text{DP}}[\kappa(x, y + \xi)] = \mathbb{E}_{\text{DP}}[v(x - y - \xi)] = \mathbb{E}_{\text{DP}}[v(x - y + \xi)] = \mathbb{E}_{\text{DP}}[\kappa(y, x + \xi)] = \tilde{\kappa}(y, x),$$

where the middle equality holds because μ_{DP} is symmetric. To show that $\tilde{\kappa}$ is PSD, define the distribution μ' such that if $a, b \stackrel{iid.}{\sim} \mu'$, $\xi = a - b \sim \mu_{\text{DP}}$. Observe now that

$$\begin{aligned} \tilde{\kappa}(x, y) &= \mathbb{E}^{\xi \sim \mu_{\text{DP}}}[\kappa(x, y + \xi)] = \mathbb{E}^{\xi \sim \mu_{\text{DP}}}[v(x - y - \xi)] \\ &= \mathbb{E}^{a, b \sim \mu'}[v((x + a) - (y + b))] = \mathbb{E}^{a, b \sim \mu'}[\kappa(x + a, y + b)] \\ &= \mathbb{E}^{a, b \sim \mu'}[\langle \psi(x + a), \psi(y + b) \rangle_{\mathcal{H}}] \\ &= \langle \mathbb{E}^{a \sim \mu'}[\psi(x + a)], \mathbb{E}^{b \sim \mu'}[\psi(y + b)] \rangle_{\mathcal{H}}, \end{aligned}$$

which implies that $\tilde{\kappa}$ is PSD. It remains to show that there exists a $w \in \mathcal{L}(\mathbb{R}^n, [1, \infty))$ such that $\tilde{\kappa}$ satisfies $\sup_{z \in \mathbb{R}^n} \kappa(z, z')/w(z) < \infty$ for all $z' \in \mathbb{R}^n$. Since κ is a PSD kernel, we know that

$$M = \begin{pmatrix} \kappa(z, z) & \kappa(z, z') \\ \kappa(z, z') & \kappa(z', z') \end{pmatrix} \succeq 0 \quad \forall z, z' \in \mathbb{R}$$

This implies that $\det(M) \geq 0$, which is equivalent to $\kappa(z, z)\kappa(z', z') \geq \kappa(z, z')^2$. Under the assumptions of the Proposition, $\kappa(x, y) = v(x - y)$ for some function v , which implies that $v(0)^2 \geq \kappa(z, z')^2$, which implies $|\kappa(z, z')| \leq |v(0)| = C < \infty$ for all $z, z' \in \mathbb{R}$. Thus

$$\sup_{z \in \mathbb{R}^n} \tilde{\kappa}(z, z') = \sup_{z \in \mathbb{R}^n} \int_{\mathbb{R}} \kappa(z, z' + \xi) \mu_{\text{DP}}(\xi) d\xi \leq \int_{\mathbb{R}} C \mu_{\text{DP}}(\xi) d\xi \leq C < \infty,$$

and the condition $\sup_{w \in \mathbb{R}^n} \tilde{\kappa}(z, z')/w(z) < \infty$ is satisfied with $w(z) = 1$. This completes the proof. \square

C Experimentation Details and Further Results

The code for all experiments is available on GitHub³. In all experiments, we perform a train-test split on each client, with 25% of the samples used for the test set. All other major details are discussed below.

C.1 Details for Experiments in Section 6

Synthetic Data For all methods, we vary the fairness regularizer from 10^{-5} to 100 in 50 logarithmically spaced steps. The centralized method uses the SGD optimizer with a stepsize of 0.05 and trains for 1000 epochs. Our method and the *local group fairness* method use the FedAvg Algorithm, with local stepsize $\eta_l = 0.05$, global stepsize $\eta_g = 1$ and trains for $T = 100$ communication rounds with $E = 50$ local epochs each. We do not use client subsampling for our experiments, that is we set $S = K$.

Real Data We give the hyperparameters below. Whenever using MMD, we apply a distance-induced kernel, in which case the MMD corresponds to the energy distance.

- **Agnostic:** We use the idea presented by Mohri et al. (2019) with the algorithm implemented by Deng et al. (2020), where we apply local stepsize $\eta_l = 0.05$ and train for $T = 100$ communication rounds with $E = 10$ local epochs each. At each communication round, the client stepsize η_l is multiplied by a factor 0.99. We do not use client subsampling for our experiments ($S = K$). The local batchsize is set to 100.
- **Centralized:** We vary the fairness regularizer from 10^{-5} to 10 in 50 logarithmically spaced steps. The centralized method uses the SGD optimizer with a stepsize of 0.05 and trains for 1000 epochs. We use the full gradient to get the best idea of “what is possible” in a centralized setting.

³<https://github.com/yvesrychener/Fair-FL>

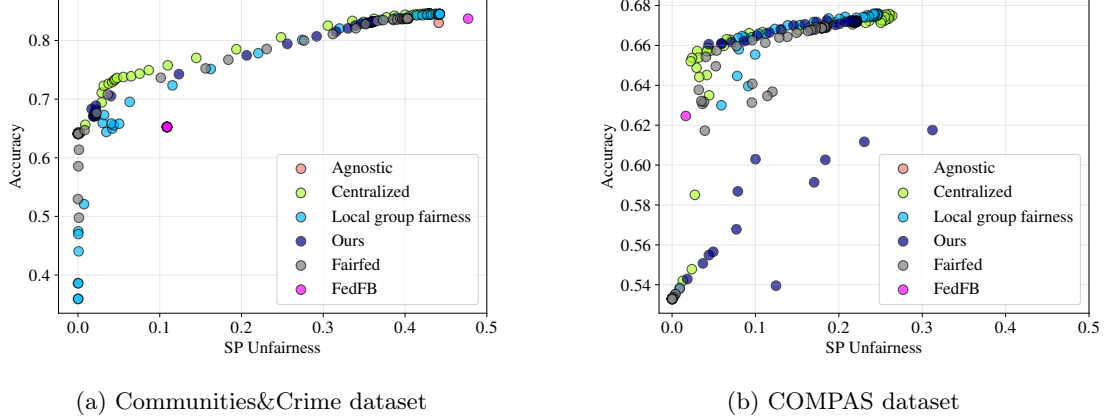


Figure 4: Performance Comparison on real data

- **Local group fairness:** We vary the fairness regularizer from 10^{-5} to 100 in 50 logarithmically spaced steps. The *local group fairness* method uses the FedAvg Algorithm, with local stepsize $\eta_l = 0.05$, global stepsize $\eta_g = 1$ and trains for $T = 100$ communication rounds with $E = 50$ local epochs each. At each communication round, the client stepsize η_l is multiplied by a factor 0.99. We do not use client subsampling for our experiments ($S = K$). The local batchsize is set to 100.
- **Ours:** We vary the fairness regularizer from 10^{-5} to 10 in 50 logarithmically spaced steps. We use the FedAvg Algorithm, with local stepsize $\eta_l = 0.01$, global stepsize $\eta_g = 1$ and trains for $T = 100$ communication rounds with $E = 50$ local epochs each. At each communication round, the client stepsize η_l is multiplied by a factor 0.99. We do not use client subsampling for our experiments ($S = K$). The local batchsize is set to 100.
- **FedFB:** We use the code by Zeng et al. (2021). We found a learning rate of 0.001 to perform the best and show results for $\alpha \in \{0.001, 0.05, 0.08, 0.1, 0.2, 0.5, 1, 2\}$. All other hyperparameters take the default values provided by Zeng et al. (2021).⁴
- **FairFed** We implement the algorithm Proposed in Ezzeldin et al. (2023) as closely as possible. In order to have a fair comparison, we choose to use a local MMD regularizer for local fair training. We compare using $\beta \in \{0.5, 1, 2\}$ but do not observe a significant effect of varying β . The local regularization strength λ is varied from 10^{-5} to 10 in 50 logarithmically spaced steps.

C.2 Scatter Plots of Experiments in Section 6

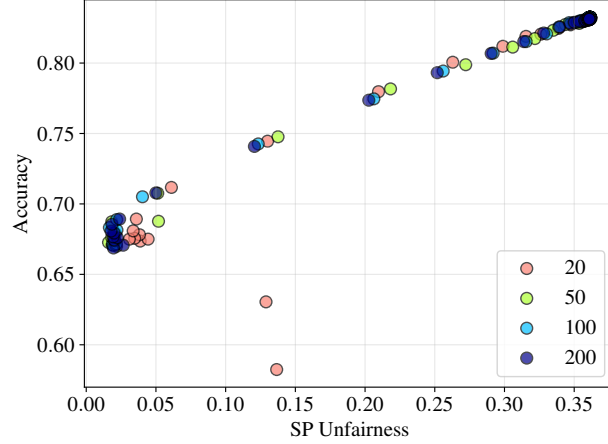
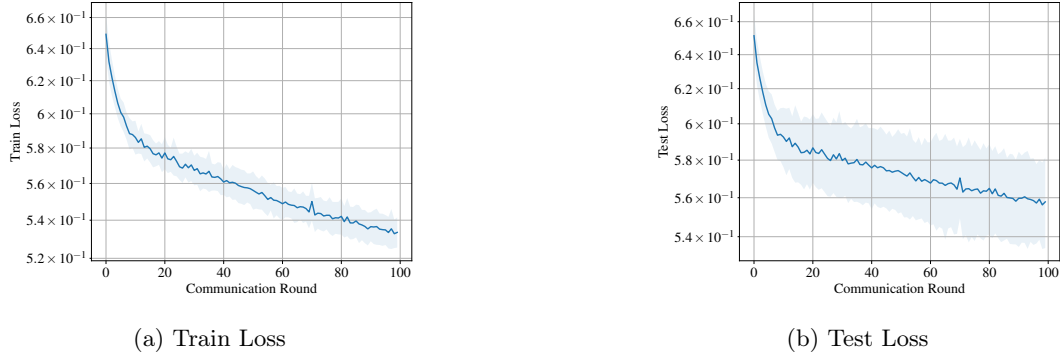
While the Pareto frontier is easy to interpret, it hides some valuable information, such as suboptimal performance away from the Pareto frontier. In Figure 4 we provide the scatter plot version of the plots in Figure 3.

C.3 Ablation Study

Effect of $|\mathcal{Y}_0|$ and $|\mathcal{Y}_1|$ In Figure 5, we plot the unfairness-accuracy trade off curve for different sizes of the sets $|\mathcal{Y}_0|$ and $|\mathcal{Y}_1|$. We use the same value for the other hyperparameters as in the experiment on real data. We see that for 20 samples, the high variance of the gradients results in subpar performance, especially for large values of λ , which results in low accuracy and relatively high unfairness. However, using 50 samples is already enough to well estimate the distributions $\mathbb{P}^{h(X)|A=0}$ and $\mathbb{P}^{h(X)|A=1}$. This suggests that good performance can be achieved even with sharing few samples, thus reducing communication cost. This is because $h_\theta(X)$ is a scalar, whose distribution can be well estimated even with few samples available.

Convergence Rate To verify the results of Section 4 and illustrate that Algorithm 1 minimizes (4), we plot the training and testing loss over communication rounds. For this experiment, we again use the Communities&Crime dataset and set $\lambda = 0.5$. Figure 6 plots the convergence on the training and testing dataset. We see that

⁴<https://github.com/yzeng58/Improving-Fairness-via-Federated-Learning>

Figure 5: Performance for varying $|\mathcal{Y}_0|$ and $|\mathcal{Y}_1|$ Figure 6: Train and Test Loss (mean \pm standard deviation over 10 runs)

Algorithm (1) indeed minimizes the objective of (4). Figure 7 shows more detailed results, where we individually show the crossentropy and MMD part of the loss.

Heterogeneity Ablation To investigate how the proposed algorithm behaves with increased client heterogeneity, we use two experiments. In the first experiment, we consider a setting with $K = 10$ clients with equal weights (that is $\nu_k = \frac{1}{10}$ for all $k = 1, \dots, K$). Further, for all $k = 1, \dots, K$, we assume $\mathbb{P}_k[A = 0] = \mathbb{P}_k[A = 1] = \frac{1}{2}$ and $X \sim \alpha \mathcal{N}(\mu(k, A), \mathbb{I}) + (1 - \alpha) \mathcal{N}(\mu(k, A^c), \mathbb{I})$, where \mathbb{I} is the 10-dimensional identity matrix, $\mu(k, A)$ is a vector of 1s (−1s) if $k + A$ is even (odd), A^c denotes the complement of A (i.e. $A^c = 1$ if $A = 0$ and $A^c = 0$ otherwise) and α is a heterogeneity parameter: If $\alpha = 0.5$, then all clients hold data sampled from the same distribution, if $\alpha = 1$ we are in the setting of the synthetic experiment in Figure 2 and all clients hold different distributions. The label is given by $Y = 1$ if $\mathbf{1}^\top X > 0$ and $Y = 0$ otherwise. We sample 200 datapoints for each client and compare the same methods as in the synthetic experiment. We use $\lambda = 50$ and otherwise use the same hyperparameters as in the main text, except that we use $\beta = 0.5$ for Fairfed as it shows slightly better performance for low heterogeneity values. The plots are given in Figure 8, showing the 95% confidence interval of mean performance ($mean \pm 1.96 \text{ standard error}$) over 10 runs. We see that our method continues to perform well as client heterogeneity increases, while other baseline methods suffer in performance.

In a second experiment, we use the same data generating process and hyperparameters as for the previous experiment. However, this time, we restrict ourselves to 6 clients with varying dataset size of, in order, 50, 150, 250, 100, 50 and 100 samples. This choice provides a heterogenous dataset size and heterogenous client weights ranging from 0.071 for clients 0 and 4 to 0.357 for client 2, while ensuring that the Bayes classifier is fair. The results are shown in Figure 9, showing the 95% confidence interval of mean performance ($mean \pm 1.96 \text{ standard error}$) over 10 runs. They show that our method continues to perform well if clients contain datasets of different sizes.

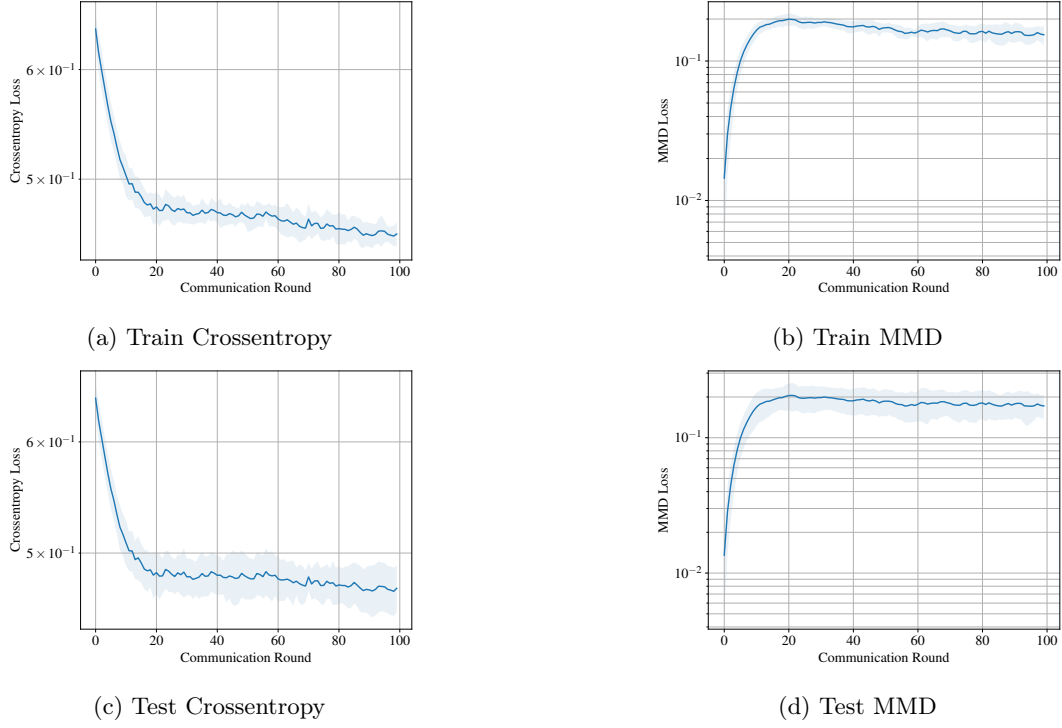


Figure 7: Decomposed Train and Test Loss (mean±standard deviation over 10 runs)

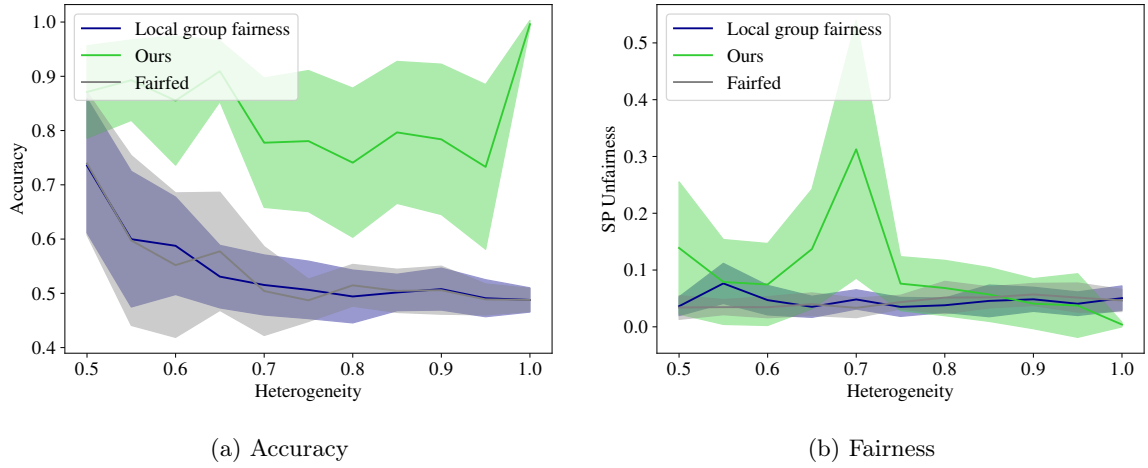


Figure 8: Heterogeneity Ablation for varying Heterogeneity α

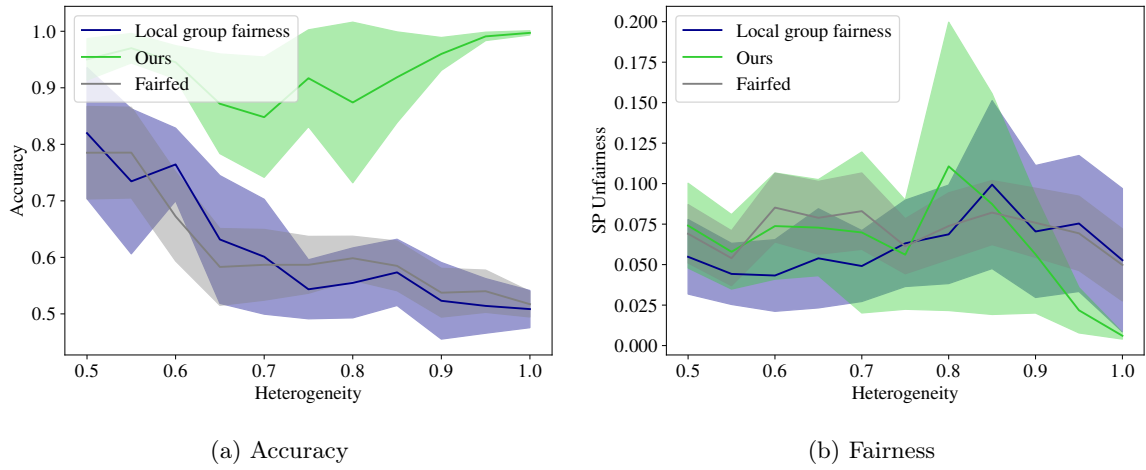


Figure 9: Heterogeneity Ablation for varying Heterogeneity α with differently weighted clients