Group Fair Federated Learning via Stochastic Kernel Regularization

Anonymous authors
Paper under double-blind review

Abstract

Ensuring **group fairness** in federated learning (FL) presents unique challenges due to data heterogeneity and communication constraints. We propose Kernel Fair Federated Learning (`KFFL`), a novel framework that incorporates group fairness into FL models using the Kernel Hilbert-Schmidt Independence Criterion (KHSIC) as a fairness regularizer. To address scalability, `KFFL` approximates KHSIC with Random Feature Maps (RFMs), significantly reducing computational and communication overhead while achieving *group fairness*.

To address the resulting non-convex optimization problem, we propose `FedProxGrad`, a federated proximal gradient algorithm that guarantees convergence. Through experiments on standard benchmark datasets across both IID and Non-IID settings for regression and classification tasks, `KFFL` demonstrates its ability to balance accuracy and fairness effectively, outperforming existing methods by comprehensively exploring the Pareto Frontier. Furthermore, we introduce `KFFL-TD`, a time-delayed variant that further reduces communication rounds, enhancing efficiency in decentralized environments.

Our work advances group fair federated learning by providing a scalable and efficient solution to promote equitable machine learning models in decentralized environments

1 Introduction

Unintended unfairness in machine learning models poses significant challenges, particularly in decision-making processes that impact specific population groups Dwork et al. (2012a); Agarwal et al. (2019b); Jalal et al. (2021). For instance, the COMPAS software, used in judicial decision-making for criminal offenses, has been shown to yield unjust outcomes disproportionately affecting the African American community Dressel & Farid (2018); Barenstein (2019). Such findings underscore the need for model outputs to be fair with respect to protected demographic attributes like gender and race. Ensuring demographic fairness has therefore emerged as a critical challenge in machine learning, driving efforts to develop robust solutions for mitigating bias and ensuring equitable model deployment.

Recent advances in bias mitigation algorithms Jalal et al. (2021); Correa et al. (2021); Agarwal et al. (2019b); Memarrast et al. (2023) largely depend on fairness regularizers incorporated into the training objective, which typically requires centralized access to data. However, in federated learning (FL) Li et al. (2020), where data is distributed across clients, privacy regulations and bandwidth constraints often prohibit raw data sharing, making centralized approaches to group fairness impractical. Furthermore, implementing *fairness regularizers* in the distributed FL setting presents additional challenges, including communication overhead, computational costs, and data heterogeneity, all of which complicate training a globally fair model.

Consequently, prior efforts to achieve *group fairness* in federated settings have primarily focused on aligning local and global fairness metrics Ezzeldin et al. (2023); Papadaki et al. (2022), often avoiding the direct incorporation of regularizer terms to ensure statistical group fairness. However, this approach faces theoretical challenges, as no universally consistent evaluation metric exists to enforce group fairness across all clients. Additionally, applying local debiasing mechanisms at individual clients alone is inadequate to ensure group fairness in the globally trained model.

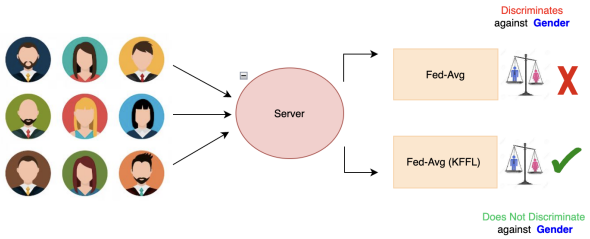

Figure 1: `FedAvg` can result in models whose predictions are biased with respect to sensitive attributes such as race or gender. `KFFL` is a principled approach designed specifically to mitigate demographic bias when training a model in a distributed setting.

In the literature, *fair federated learning* often uses the terms **client fairness** and **group fairness** interchangeably. However, these concepts address distinct objectives. Works such as Chaudhury et al. (2022); Li et al. (2019); Donahue & Kleinberg (2021); Cui et al. (2021); Du et al. (2021) focus on client fairness in federated learning, aiming to ensure that the model performs equitably across clients' data Mohri et al. (2019), thereby mitigating disparities arising from data heterogeneity among clients. In contrast, **group fairness** Ezzeldin et al. (2023); Papadaki et al. (2022) seeks to achieve fairness across different demographic groups. This involves establishing performance guarantees to ensure the global model is fair with respect to sensitive attributes, such as race or gender. More discussion on the related works is available in the Appendix A.1. This work aims to address the latter objective ie: **group fairness** in the trained global model.

Particularly, we propose a novel approach that integrates the Kernel Hilbert-Schmidt Independence Criterion (KHSIC) Gretton et al. (2005b) as a fairness regularizer into FL. KHSIC is a powerful measure of statistical dependence capable of capturing complex, non-linear relationships between variables, making it well-suited for enforcing group fairness

Why Choose KHSIC? The Hilbert-Schmidt Independence Criterion (HSIC) has been effectively used as a fairness regularizer in centralized regression models, as demonstrated in Pérez-Suay et al. (2017). KHSIC, a kernelized version of HSIC, stands out due to its ability to capture complex, nonlinear dependencies among random variables through the use of kernel functions. Additionally, KHSIC provides a theoretical guarantee: a low KHSIC value between model outputs and sensitive attributes ensures approximate statistical parity Kim & Gittens (2021).Compared to the closely related Rényi correlation Baharlouei et al. (2019b), KHSIC is more practical for measuring dependence because its empirical computation reduces to straightforward linear algebra operations. This makes KHSIC not only theoretically robust but also computationally efficient in practical applications.

However, directly applying KHSIC in a federated learning (FL) setup presents challenges, as it is computationally expensive and communication-intensive due to the necessity of computing and exchanging large kernel matrices. To address these limitations, we leverage Random Feature Maps (RFMs) Rahimi & Recht (2007) to approximate kernel functions. This approximation significantly reduces both computational and communication costs, making the integration of KHSIC as a fairness regularizer more feasible and efficient in federated learning setting.

Furthermore, to efficiently solve the distributed optimization problem incorporating the fairness regularizer, we propose `FedProxGrad`, a federated proximal gradient algorithm. This method ensures convergence for non-convex composite optimization problems that arise from the integration of the fairness term.

Our main contributions are as follows:

- We propose `KFFL`, a novel federated learning algorithm that incorporates group fairness using KHSIC as a fairness regularizer. To the best of our knowledge, this is the first work to adapt KHSIC for use in federated learning, addressing the unique challenges of the federated setting.

- We develop a communication-efficient approximation using Random Feature Maps to reduce the computational and communication overhead associated with KHSIC. This allows us to avoid transmitting large kernel matrices, reducing communication costs by orders of magnitude.
- We introduce `FedProxGrad`, a federated proximal gradient algorithm that provides convergence guarantees for non-convex composite optimization problems. `FedProxGrad` allows both terms of the composite objective to be non-convex, as opposed to prior works on federated composite optimization Wang & Li (2023); Bao et al. (2022); Yuan et al. (2021); Tran Dinh et al. (2021).
- We conduct extensive experiments on standard benchmark datasets under both IID and Non-IID data distributions on both classification and regression tasks. Our results demonstrate that `KFFL` effectively balances the trade-off between accuracy and fairness, outperforming existing baselines and exploring the Pareto frontier more comprehensively.
- We analyze the communication overhead of `KFFL` and introduce a time-delayed variant, `KFFL-TD`, which further reduces communication rounds while maintaining performance. This makes our method more practical for real-world FL applications where communication resources are limited.

2 Preliminaries

The goal of fair learning is to ensure that the model's output exhibits no dependencies on sensitive attributes.

We assume the observations are sampled i.i.d. from a joint distribution $\mathbb{P}(X, S, Y)$ to obtain training data $\{\mathbf{x}_i, \mathbf{s}_i, y_i\}_{i=1}^n$. Here, $\mathbf{x}_i$ contains the non-sensitive covariates, $\mathbf{s}_i$ contains sensitive covariates (which may be a binary scalar s_i or multi-dimensional vector $\mathbf{s}_i$), and y_i is the ground truth label for the i-th sample. This dataset is employed to train a classifier $f(\mathbf{x}; \boldsymbol{\omega})$, where $\boldsymbol{\omega}$ denotes the model parameters.[1]

As an example, consider the task of training a binary classifier for making hiring decisions. Here $\mathbf{x}$ consists of features that are ethically and legally allowable for use in making hiring decisions, s represents *binary* sensitive features such as the individual's sex or marital status, and the ground truth decisions are $y_i \in \{0, 1\}$. The classifier makes predictions $\hat{y}_i = f(\mathbf{x}_i; \boldsymbol{\omega}) \in \{0, 1\}$, where 1 signifies a decision to hire.

2.1 Metrics for Group Fairness

The two most widely used definitions of group fairness for evaluating classifier fairness Ezzeldin et al. (2023) are statistical parity and equalized odds Dwork et al. (2012a). These measures aim to assess equitable treatment across protected groups in trained models.

- **Statistical Parity Difference (SPD)**: A model achieves statistical parity when the probability of a positive outcome is independent of the sensitive variable Dwork et al. (2012b), i.e., $\hat{y} \perp\!\!\!\perp s$. In the context of binary sensitive variables, a measure of statistical parity is given by

$$\text{SPD} = \Pr(\hat{y} = 1 \mid s = 1) - \Pr(\hat{y} = 1 \mid s = 0). \tag{1}$$

- **Equalized Odds (EOD)**: A model satisfies equalized odds when the true positive rates do not depend on the sensitive variable Romano et al. (2020). In the context of binary sensitive variables, a measure of equalized odds is given by

$$\text{EOD} = \Pr(\hat{y} = 1 \mid y = 1, s = 1) - \Pr(\hat{y} = 1 \mid y = 1, s = 0). \tag{2}$$

Equalized odds and statistical parity values closer to zero indicate that the model is fair (with respect to those particular definitions). We develop our framework in the context of SPD but, as is common in the literature,

[1] We follow the notation convention where boldface lowercase letters (e.g., $\mathbf{v}$) denote vectors, non-bold lowercase letters (e.g., v) represent scalars, and boldface uppercase letters (e.g., $\mathbf{M}$) signify matrices. For example, in the equation $\mathbf{v} = \mathbf{M}\mathbf{u} + b$, $\mathbf{v}$ and $\mathbf{u}$ are vectors, $\mathbf{M}$ is a matrix, and b is a scalar.

utilize both these notions in measuring its performance empirically in Section 5 when the underlying model is a classifier.

Fair regression has been extensively studied in centralized settings, with notable contributions such as Chzhen et al. (2020a); Agarwal et al. (2019a); Chzhen et al. (2020b). However, training regression models in distributed settings, where the target variable $\hat{y}$ is continuous, remains an underexplored area. Traditional fairness metrics like **Equalized Odds (EOD)** and **Statistical Parity Difference (SPD)** are not directly applicable in this context. Instead, the **Kolmogorov-Smirnov (KS)** distance is commonly employed to evaluate fairness. This metric captures the maximum disparity between the distributions of model predictions for different sensitive groups, providing a robust measure for regression tasks (see Appendix F for more details).

- **Kolmogorov-Smirnov (KS) Distance** Chzhen et al. (2020a):

  For each sensitive group $s \in \mathcal{S}$, let the set of indices of samples belonging to that group be represented as:

  $$\mathcal{I}^s = \{i \in \{1, 2, \ldots, n\} : s_i = s\}.$$

  The empirical cumulative distribution function (CDF) of the model predictions for group s is defined as:

  $$F^s(t; \boldsymbol{\omega}) = \frac{1}{|\mathcal{I}^s|} \sum_{i \in \mathcal{I}^s} \mathbb{1}\{f(\mathbf{x}_i; \boldsymbol{\omega}) \leq t\}, \tag{3}$$

  where $f(\mathbf{x}_i; \boldsymbol{\omega})$ is the model's prediction for input $\mathbf{x}_i$ parameterized by $\boldsymbol{\omega}$.

  The KS distance between the predictions for any two sensitive groups s and s' is given by:

  $$\mathrm{KS}(\boldsymbol{\omega}) = \max_{s, s' \in \mathcal{S}} \sup_{t \in \mathbb{R}} \left| F^s(t; \boldsymbol{\omega}) - F^{s'}(t; \boldsymbol{\omega}) \right|. \tag{4}$$

  The KS distance measures the largest difference between the CDFs of predictions for any two sensitive groups s and s'. A smaller KS value indicates lower disparity between groups, supporting fairness objectives in regression tasks.

2.2 Kernel Hilbert-Schmidt Independence Criterion (KHSIC)

The Kernel Hilbert-Schmidt Independence Criterion (**KHSIC**) of Gretton et al. (2005a) is key to our approach to ensuring statistical parity. The KHSIC is predicated on the observation that $\hat{y}$ and s are independent if and only if *every* function of $\hat{y}$ is uncorrelated with *every* function of s.

Given two Reproducing Kernel Hilbert Spaces $\mathcal{F}$ and $\mathcal{G}$, the KHSIC quantifies the dependence of these random variables by measuring the correlation of every function of $\hat{y}$ in $\mathcal{F}$ with every function of s in $\mathcal{G}$:

$$\psi_{\mathrm{pop}}(\hat{y}, s) = \sup_{\substack{h \in \mathcal{F},\, g \in \mathcal{G} \\ \|h\|_{\mathcal{F}}^2 \leq 1,\, \|g\|_{\mathcal{G}}^2 \leq 1}} \mathrm{Cov}(h(\hat{y}), g(s)). \tag{5}$$

Under some regularity conditions on $\mathcal{F}$ and $\mathcal{G}$ (universality), this population KHSIC is zero if and only if $\hat{y}$ and s are independent. More generally, the population KHSIC gives an upper bound on the total variation distance between the joint distribution $\mathbb{P}_{\hat{y},s}$ and the product of the marginals $\mathbb{P}_{\hat{y}} \otimes \mathbb{P}_s$, and thus quantifies the dependence of $\hat{y}$ and s Kim & Gittens (2021).

The KHSIC is more practically useful as a measure of dependence than the closely related Rényi correlation: the latter is defined in terms of the probability density functions (PDFs) of $\hat{y}$ and s. This approach does not scale to high-dimensional inputs because accurately estimating the PDFs of $\hat{y}$ and s requires an exponential number of observations relative to their dimensionality. By comparison, empirical estimation of the KHSIC reduces to simple and scalable linear algebraic computations. The empirical KHSIC is given by Gretton et al. (2005a)

$$\psi_{\mathrm{emp}}(\hat{\mathbf{y}}, \mathbf{s}) = \frac{1}{(n-1)^2} \mathrm{Tr}\left(\mathbf{H}\mathbf{K}_{\mathbf{s}}\mathbf{H}^2\mathbf{K}_{\hat{\mathbf{y}}}\mathbf{H}\right), \tag{6}$$

where

$$\mathbf{K}_{\hat{\mathbf{y}}} = [\kappa_{\hat{y}}(\hat{y}_i, \hat{y}_j)]_{i,j=1}^n, \quad \mathbf{K}_{\mathbf{s}} = [\kappa_s(\mathbf{s}_i, \mathbf{s}_j)]_{i,j=1}^n, \quad \mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top.$$

The matrices $\mathbf{K}_{\hat{\mathbf{y}}}$ and $\mathbf{K}_{\mathbf{s}}$ are $n \times n$ kernel matrices evaluated on the training data corresponding to the kernels. [2] Gretton et al. (2005a) show that $\psi_{\text{emp}}(\hat{\mathbf{y}}, \mathbf{s})$ is a $\frac{1}{\sqrt{n}}$-consistent estimator of $\psi_{\text{pop}}(\hat{y}, s)$.

We propose to measure the fairness (in the sense of statistical parity) of models by using the KHSIC:

$$\psi(\boldsymbol{\omega}; \mathbf{X}, \mathbf{S}) = \frac{1}{(n-1)^2} \operatorname{Tr}\left(\mathbf{H}\mathbf{K}_{\mathbf{s}}\mathbf{H}\mathbf{K}_{\hat{\mathbf{y}}}\mathbf{H}\right). \tag{7}$$

The notation $\psi(\boldsymbol{\omega}; \mathbf{X}, \mathbf{S})$ emphasizes that the fairness measure depends on $\boldsymbol{\omega}$ through the predictions $\hat{\mathbf{y}} = f(\mathbf{X}; \boldsymbol{\omega})$.

2.3 Fair Learning Objective

Consequently, we propose to learn fair models by using $\psi(\boldsymbol{\omega}; \mathbf{X}, \mathbf{S})$ as a regularizer:

$$\boldsymbol{\omega}_\star = \operatorname{argmin}_{\boldsymbol{\omega}} \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(\mathbf{x}_i; \boldsymbol{\omega})) + \lambda \psi(\boldsymbol{\omega}; \mathbf{X}, \mathbf{S}). \tag{8}$$

The first term in the objective function represents the loss $\ell(y_i, f(\mathbf{x}_i; \boldsymbol{\omega}))$, which quantifies the discrepancy between the model's predictions and the ground truth labels. This loss function can be instantiated as the cross-entropy loss for classification tasks or the mean squared error (MSE) for regression tasks. The second term acts as a fairness regularizer by measuring the dependence between the model's predictions and the sensitive attributes using the KHSIC criterion. The regularization parameter λ controls the trade-off between optimizing predictive performance and enforcing fairness, thereby balancing the two objectives.

Solving Equation (8) in a centralized setting is conceptually straightforward but computationally challenging because, at each iteration, it involves computing matrices $\mathbf{K}_{\hat{\mathbf{y}}}(\boldsymbol{\omega})$ of size $n \times n$ and computing gradients through them. The computational burden is compounded in the federated learning setting by the additional communication burden: clients need to communicate these $n \times n$ matrices to compute their local contributions to the gradients. Hence, a straightforward adaptation of this centralized approach to a distributed setting is prohibitive from the perspectives of both computation and communication. In the next section (Section 3), we introduce a novel method `FedProxGrad` to overcome these challenges in the federated setting.

3 Composite Optimization in the Federated Setting

Employing the fair ML formulation of equation (8) in a federated setting requires solving a federated composite optimization (FCO) problem that can be written in the standard form

$$\operatorname{argmin}_{\boldsymbol{\omega}} \sum_{i=1}^m \ell^i(\boldsymbol{\omega}) + \psi(\boldsymbol{\omega}),$$

where the ℓ^i are data-fitting terms local to each client, and ψ is the global fairness regularizer. **For brevity, we have absorbed the regularization constant λ into ψ.** We denote the sum of local-data fitting terms with $\ell(\boldsymbol{\omega}) = \sum_{i=1}^m \ell^i(\boldsymbol{\omega})$ and the composite objective with $F(\boldsymbol{\omega}) = \ell(\boldsymbol{\omega}) + \psi(\boldsymbol{\omega})$. In Equation (8) the fairness term ψ is non-convex and the data fitting term, ℓ, may also be non-convex. The next section (Section 4) shows how to reduce the communication complexity involved in using a kernel-based fairness regularizer for ψ. This section provides an algorithm for solving the resulting non-convex FCO problem.

Several existing works provide algorithms for the FCO problem–Wang & Li (2023); Bao et al. (2022); Yuan et al. (2021) develop algorithms that require the ℓ^i and ψ to be convex, while the algorithm of Tran Dinh

[2] For notational brevity, we use the notation κ to refer to two potentially different kernel functions on the features and the sensitive variables.

Algorithm 1 Federated Proximal Gradient Descent (`FedProxGrad`)

1: **Input:** $\boldsymbol{\omega}_0$, T, α
2: **for** $t = 0, \cdots, T-1$ **do**
3: Server computes $\mathbf{g}_t$, a stochastic gradient estimator for $\nabla\psi(\boldsymbol{\omega}_t)$, and computes $\boldsymbol{\omega}_{t+1/2} = \boldsymbol{\omega}_t - \alpha\mathbf{g}_t$
4: Server sends $\boldsymbol{\omega}_{t+1/2}$ to clients
5: Each client for $i \in \{1, \ldots, m\}$ computes

$$\boldsymbol{\omega}_{t+1}^i = \operatorname{argmin}_{\boldsymbol{\omega}} \ell^i(\boldsymbol{\omega}) + \frac{1}{2\alpha}\|\boldsymbol{\omega} - \boldsymbol{\omega}_{t+1/2}\|_2^2$$

6: Each client returns $\boldsymbol{\omega}_{t+1}^i$ back to the server
7: Server aggregates the device models to form

$$\boldsymbol{\omega}_{t+1} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{\omega}_{t+1}^i.$$

8: **end for**

et al. (2021) requires only ψ to be convex—but, to our knowledge, no extant optimization algorithms for FCO guarantees convergence for problems where the ℓ^i and ψ are both non-convex. To fill this gap, we introduce `FedProxGrad`, a federated proximal gradient descent algorithm.

The `FedProxGrad` algorithm, described in Algorithm 1, extends the stochastic proximal gradient algorithm in a straightforward manner from the centralized setting to the federated setting. The analysis of its convergence follows closely that of `FedProx`, so we introduce the same notions used in its convergence Li et al. (2020).

Definition 1 (γ-suboptimality)**.** Let $\ell_t^i(\boldsymbol{\omega}) = \ell^i(\boldsymbol{\omega}) + \frac{1}{2\alpha}\|\boldsymbol{\omega} - \boldsymbol{\omega}_{t+1/2}\|^2$ (see Algorithm 1 for the definition of $\boldsymbol{\omega}_{t+1/2}$). Given $\gamma \in [0, 1]$, a point $\widehat{\boldsymbol{\omega}}$ is a γ-suboptimal solution of $\operatorname{argmin}_{\boldsymbol{\omega}} \ell_t^i(\boldsymbol{\omega})$ if $\|\nabla\ell_t^i(\widehat{\boldsymbol{\omega}})\| \leq \gamma\|\nabla\ell_t^i(\boldsymbol{\omega}_t)\|$. Smaller γ correspond to higher accuracy.

This condition on the local solvers ensures that the local solution for the $(t+1)$th iterate is a factor of γ less suboptimal than the tth iterate. This condition is agnostic to the particular solvers employed, and can be achieved using an deterministic or randomized solvers that use full gradients or stochastic gradients.

Definition 2 ((G, B)-Bounded Dissimilarity)**.** The local functions ℓ^i are (G, B)-boundedly dissimilar at $\boldsymbol{\omega}$ if $\mathbb{E}_i\|\nabla\ell^i(\boldsymbol{\omega}) + \nabla\psi(\boldsymbol{\omega})\|^2 \leq B^2\|\nabla F(w)\|^2 + G^2$.

This condition is standard Li et al. (2020); Karimireddy et al. (2020), and ensures that local progress on the individual clients can be translated to global progress.

The definition of γ-suboptimality and the bounded dissimilarity condition have been modified from those used in Li et al. (2020) to facilitate the analysis of a composite objective

Theorem 1. *Assume that the functions ℓ^i, ψ, and F are L-smooth; the functions ℓ^i are L_--weakly convex; F is bounded below by a constant $F^\star$; the bounded dissimilarity condition (Definition 2) holds with $G = 0$; and that the stochastic gradient estimate for the fairness regularizer satisfies $\mathbb{E}[\mathbf{g}_t \,|\, \boldsymbol{\omega}_t] = \nabla\psi(\boldsymbol{\omega}_t)$ and*

$$\mathbb{E}[\|\mathbf{g}_t - \nabla\psi(\boldsymbol{\omega})\|_2^2 \mid \boldsymbol{\omega}] \leq \sigma^2$$

for all $\boldsymbol{\omega}$. If the local solvers on each client ensure γ suboptimal solutions (Definition 1) with parameter $\gamma \leq \frac{1}{8(B+1)}$ and the global stepsize is chosen to satisfy

$$\alpha < \min\left\{\frac{1}{20}, \frac{1}{2L_-}, \frac{1}{120L(B+1)}, \frac{1}{5LB^2}\right\},$$

then the sequence of iterates generated by FedProxGrad satisfies

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\left[\|\nabla F(\boldsymbol{\omega}_t)\|_2^2\right] \leq \frac{F(\boldsymbol{\omega}_0) - F^\star}{\alpha T} + 4\sigma^2.$$

This result shows that, for an appropriate choice of hyperparameters, the `FedProxGrad` algorithm converges at a rate of $\frac{1}{T}$ up to the noise level of the stochastic gradient. A proof is provided in the Appendix A.2

4 Communication-Efficient Kernel Regularized Fair Learning

Random Feature Maps for Kernel Approximation:

In the centralized setting, kernel methods pose a computational challenge due to their inherent complexity, requiring optimization with a kernel matrix incurring a computational cost of up to $\mathcal{O}(n^3)$. One line of research for reducing this burden, starting with the seminal work of Rahimi & Recht (2007), uses random feature maps (RFMs). A random feature map for a shift-invariant kernel function κ is a random function $\phi : \mathbb{R}^p \to \mathbb{R}^D$ that is constructed to satisfy $\kappa(\mathbf{x}, \mathbf{y}) = \mathbb{E}\langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle$ for any two vectors $\mathbf{x}$ and $\mathbf{y}$, where the expectation is taken over the randomness in ϕ. RFMs enable the efficient formation of randomized low-rank approximations to kernel matrices. In particular, if the rows of $\mathbf{Z}_S \in \mathbb{R}^{n\times D}$ consist of the application of an RFM ϕ to the $\mathbf{s}_i$, and the rows of $\mathbf{Z}_{f(\omega)} \in \mathbb{R}^{n\times D}$ likewise consist of the application an RFM to the observed $\mathbf{x}_i$, then

$$\mathbf{K}_S = \mathbb{E}[\mathbf{Z}_S\mathbf{Z}_S^T] \quad \text{and} \quad \mathbf{K}_{f(\boldsymbol{\omega})} = \mathbb{E}[\mathbf{Z}_{f(\boldsymbol{\omega})}\mathbf{Z}_{f(\boldsymbol{\omega})}^T], \tag{9}$$

and the variance goes down as the number of random features D increases, so $\mathbf{Z}_S\mathbf{Z}_S^T$ and $\mathbf{Z}_{f(\boldsymbol{\omega})}\mathbf{Z}_{f(\boldsymbol{\omega})}^T$ are principled randomized low-rank approximations to the corresponding kernel matrices. A substantial body of work has demonstrated that these approximations exhibit theoretically and empirically similar performance to full kernel matrices (Hamid et al., 2014; Rahimi & Recht, 2007; Yu et al., 2016).More discussion on the construction and drawing of these feature maps is also made in Appendix G

Approximation of the Fairness Regularizer: We utilize these randomized low-rank approximations to efficiently compute principled approximations to the regularizer $\psi(\boldsymbol{\omega})$ in Equation 8. Note that $\mathbf{Z}_{f(\boldsymbol{\omega})}$ and $\mathbf{Z}_S$ have dimensions $n \times D$ and that $D \ll n$ in size. Specifically, we utilize *the Orthogonal Random Feature Maps (ORFM)* of (Yu et al., 2016).

The first crucial observation is that by using RFMs, one need only communicate a $D \times D$ matrix to approximate $\psi(\boldsymbol{\omega})$ (see Theoram 2) and its gradient $\mathbf{g}(\boldsymbol{\omega})$ (see Corollary 1) , rather than communicating two $n \times n$ kernel matrices. For proof refer to the Appendix A.3 and A.5 respectively.

Theorem 2. *Let $\mathbf{Z}_f$ and $\mathbf{Z}_s$ be $n \times D$ matrices constructed using RFMs. Then $\psi(\boldsymbol{\omega}) = \mathbb{E}\left[\|\mathbf{G}(\boldsymbol{\omega})\|_F^2\right]$ where $\mathbf{G}(\boldsymbol{\omega}) = \mathbf{Z}_s^\top\mathbf{Z}_f(\boldsymbol{\omega}) - n\boldsymbol{\mu}_s\boldsymbol{\mu}_f^\top(\boldsymbol{\omega}) \in \mathbb{R}^{D\times D}$. Here, $\boldsymbol{\mu}_s$ is the mean over the rows of $\mathbf{Z}_s$ and $\boldsymbol{\mu}_f$ is the mean over the rows of $\mathbf{Z}_f(\boldsymbol{\omega})$.*

Corollary 1. *$\mathbf{g}(\boldsymbol{\omega}) = \nabla\|\mathbf{G}(\boldsymbol{\omega})\|_F^2$ is an unbiased stochastic estimate of $\nabla\psi(\boldsymbol{\omega})$.*

Distributed Computation in Federated Settings:

In a distributed setup the RFM matrices can be partitioned over the m workers as follows:

$$\mathbf{Z}_{f(X)} = \begin{pmatrix} \mathbf{Z}_{f(X),1} \\ \vdots \\ \mathbf{Z}_{f(X),m} \end{pmatrix} \quad \text{and} \quad \mathbf{Z}_S = \begin{pmatrix} \mathbf{Z}_{S,1} \\ \vdots \\ \mathbf{Z}_{S,m} \end{pmatrix}, \tag{10}$$

where $\mathbf{Z}_{f(X),i}, \mathbf{Z}_{S,i} \in \mathbb{R}^{n_i\times D}$. Here n_i is the number of data points on worker i, so $n = \sum_{i=1}^m n_i$. This observation allows each worker to efficiently compute its contribution to the feature interaction matrix $\mathbf{G}(\boldsymbol{\omega})$.

Lemma 1. *The global feature interaction matrix can be partitioned into local interactions as $\mathbf{G}(\boldsymbol{\omega}) = \sum_{i=1}^m \mathbf{Z}_{s,i}^\top\mathbf{Z}_{f,i} - n\left(\frac{1}{n}\sum_{i=1}^m n_i\boldsymbol{\mu}_{s,i}\right)\left(\frac{1}{n}\sum_{i=1}^m n_i\boldsymbol{\mu}_{f,i}\right)^\top$*

Therefore, using Lemma 1 (see Proof in A.4), to compute the global interaction term $\mathbf{G}(\boldsymbol{\omega})$, each worker only needs to transmit the local feature interaction matrix $\mathbf{Z}_{s,i}^\top\mathbf{Z}_{f,i} \in \mathbb{R}^{D_s\times D_f}$, along with its local average feature vectors $\mu_{s,i} \in \mathbb{R}^{D_s}$ and $\mu_{f,i} \in \mathbb{R}^{D_f}$ to the server. Since empirically we use the same size of D_s and D_f we

drop the subscript in our discussion. This approach facilitates the computation of $\mathbf{G}(\boldsymbol{\omega})$ and, consequently, $\mathbf{g}(\boldsymbol{\omega})$ using Lemma 2 (see Proof in A.6).

Lemma 2. *An unbiased estimate of the gradient of fairness regularizer ψ can be partitioned into local interactions as $\mathbf{g}(\boldsymbol{\omega}) = \sum_{i=1}^{m} \mathbf{J}_{\boldsymbol{\Omega}_i}(\boldsymbol{\omega})^T \mathbf{G}(\boldsymbol{\omega})$, where $\boldsymbol{\Omega}_i(\boldsymbol{\omega}) = \mathbf{Z}_s^\top \mathbf{Z}_{f,i}(\boldsymbol{\omega}) - n_i \boldsymbol{\mu}_s \boldsymbol{\mu}_f^\top$ and $\mathbf{J}_{\boldsymbol{\Omega}_i}(\boldsymbol{\omega})$ is the Jacobian of $\boldsymbol{\Omega}_i$ with respect to the model parameters $\boldsymbol{\omega}$.*

Communication Efficiency using RFMs: Therefore, instead of transmitting $n \times n$ matrices, only $D \times D$ matrices need be sent to compute unbiased approximations to ψ and $\nabla\psi$. This results in a substantial reduction in communication costs. For instance, consider training a fair model on the ADULT dataset ($n = 32K$). Choosing $D = 1024$ (numbers based on our experimental evaluations) with equal partitioning of data points, computing ψ exactly requires communicating $32K \times 32K$ matrices, compared to 1024×1024 matrices when RFMs are employed, so the communication costs are reduced by three orders of magnitude!

Table 1: Table that explains the notations used in `KFFL` Algorithm

Symbol	**Description**
$\mathbf{x}$	Data parameter vector
$\boldsymbol{\omega}$	Model parameter vector
$\mathbf{M}^i(\boldsymbol{\omega})$	Interaction matrix for client i
$\boldsymbol{\mu}_s$	Global mean sensitivity vector
$\boldsymbol{\mu}_f$	Global mean function vector
$\boldsymbol{\mu}_{s,i}$	Local mean sensitivity vector for client i
$\boldsymbol{\mu}_{f,i}$	Local mean function vector for client i
$\boldsymbol{\Omega}_i(\boldsymbol{\omega}_t)$	Local interaction matrix for client i
$f_i(\boldsymbol{\omega}_t)$	Local data fitting term for client i
γ_t	Parameter for regularization
$\mathbf{Z}$	Random Feature Map Matrix
D_s	Dimensions of the protected attribute feature map matrix
D_f	Dimensions of the logits feature map matrix
$\mathbf{J}_{\boldsymbol{\Omega}_i}(\boldsymbol{\omega})$	Jacobian of $\boldsymbol{\Omega}_i$ at $\boldsymbol{\omega}$
$\mathbf{G}(\boldsymbol{\omega})$	Gradient vector at $\boldsymbol{\omega}$

Equations used in `KFFL/KFFL-TD`

$$\mathbf{G}(\boldsymbol{\omega}) = \mathbf{Z}_s^\top \mathbf{Z}_f(\boldsymbol{\omega}) - n\boldsymbol{\mu}_s \boldsymbol{\mu}_f^\top(\boldsymbol{\omega}) \tag{11}$$

$$\mathbf{M}^i(\boldsymbol{\omega}) = \mathbf{Z}_s^\top \mathbf{Z}_{f,i}(\boldsymbol{\omega}) \tag{12}$$

$$\boldsymbol{\mu}_s = \left(\frac{1}{n}\sum_{i=1}^{m} n_i \boldsymbol{\mu}_{s,i}\right) \tag{13}$$

$$\boldsymbol{\mu}_f = \left(\frac{1}{n}\sum_{i=1}^{m} n_i \boldsymbol{\mu}_{f,i}\right)^\top \tag{14}$$

$$\boldsymbol{\mu}_{s,i}^\top = \frac{1}{n_i}\mathbf{1}^\top \mathbf{Z}_{s,i} \tag{15}$$

$$\boldsymbol{\mu}_{f,i}^\top = \frac{1}{n_i}\mathbf{1}^\top \mathbf{Z}_{f,i} \tag{16}$$

$$\boldsymbol{\omega}_{t+1} = \text{argmin}_{\boldsymbol{\omega}} \left[f_i(\boldsymbol{\omega_t}) + \frac{1}{2\alpha_t} \|\boldsymbol{\omega} - \boldsymbol{\omega}_t\|_2^2 \right] \tag{17}$$

$$\boldsymbol{\Omega}_i(\boldsymbol{\omega}) = \mathbf{M}^i(\boldsymbol{\omega}) - n_i \boldsymbol{\mu}_s \boldsymbol{\mu}_f^\top \tag{18}$$

$$\mathbf{g}^i(\omega) = \mathbf{J}_{\boldsymbol{\Omega}_i}(\boldsymbol{\omega})^T \mathbf{G}(\boldsymbol{\omega}) \tag{19}$$

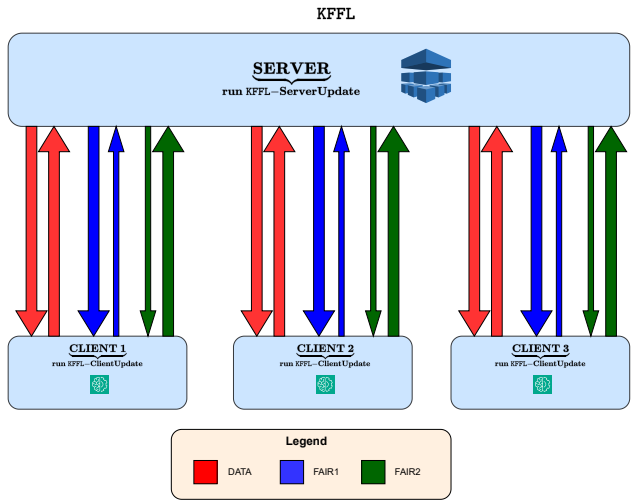

Figure 2: The communication pattern for KFFL. Different colors correspond to the **FAIR1**, **FAIR2** and **DATA** sub-rounds of KFFL. The direction of the arrows indicate an uplink or downlink communication and the width of each arrowhead highlights the communication cost in each sub-round. Thicker lines indicate (large) communication costs on the order of the size of $\boldsymbol{\omega}$, while thinner lines represent communication costs on the order of D^2.

Kernel Regularized Fair Federated Learning KFFL

We now introduce the KFFL algorithm, which uses the FedProxGrad method to implement kernel-regularized fair learning using the approximation to ψ introduced in the previous section. A time-delayed variant that uses stale fairness gradient information to incur one less round of communication per iteration, KFFL-TD, is presented in Appendix C.

Algorithm 2 KFFL – Client Side

Input: (**ROUND**, ..)

if ROUND = **FAIR1** **then**
  Clients compute $\mathbf{M}^i(\boldsymbol{\omega}_t)$ (see Equation 12) using shared random seed ζ to generate their RFM, $\boldsymbol{\mu}_{f,i}(t)$ using Equation 16 and $\boldsymbol{\mu}_{s,i}(t)$ using Equation 15.
  Combine terms $\Phi^i(\boldsymbol{\omega}_t) = \{\mathbf{M}^i(\boldsymbol{\omega}_t), \boldsymbol{\mu}_{s,i}, \boldsymbol{\mu}_{f,i}\}$
  Return: $\Phi^i(\boldsymbol{\omega}_t)$
else if ROUND = **FAIR2** **then**
  Client compute local interaction for gradients using Equation 18 to get $\boldsymbol{\Omega}_i(\boldsymbol{\omega}_t)$
  The clients then compute the local gradient $\mathbf{g}^i(\boldsymbol{\omega}_t)$ using Equation 19 from $\Lambda(\boldsymbol{\omega}_t)$
  Return: $\mathbf{g}^i(\boldsymbol{\omega}_t)$
else if ROUND = **DATA** **then**
  Clients do a local update on $\boldsymbol{\omega}_{t+1/2}$ following Equation 17 to get $\boldsymbol{\omega}^i_{t+1}$
  Return: $\boldsymbol{\omega}^i_{t+1}$
end if

The KFFL algorithm is detailed in Algorithm 2, which gives the client-side procedure, and Algorithm 3, which gives the server-side process. Equations 11 till 19 are used in these algorithms. At a high level, KFFL uses federated composite optimization to fit a fair model in three rounds:

FAIR1 At the start of the $t+1$-th iteration, the clients use the RFMs to compute the local terms $\Phi^i(\boldsymbol{\omega}_t)$ needed to compute the interaction matrix $\mathbf{G}(\boldsymbol{\omega}_t)$ for the current global model $\boldsymbol{\omega}_t$, and communicate them to the server. Clients also store the local interaction term for next round $\mathbf{M}^i(\boldsymbol{\omega}_t)$. The server combines the local terms to compute the interaction matrix $\mathbf{G}(\boldsymbol{\omega}_t)$ on the current global model, and returns these to the workers.

Algorithm 3 `KFFL` – Server Side

$\boldsymbol{\omega} = \boldsymbol{\omega}_0$ {This is the initial model}
$t \leftarrow 0$
while $\boldsymbol{\omega}$ not converged **do**
  for all $i = 1, \ldots, m$ in parallel **do**
    Generation of random seed ζ
    $\Phi^i(\boldsymbol{\omega}_t) =$ **Client Update**(**FAIR1**, $\boldsymbol{\omega}_t, \zeta$)
  end for
  $\Phi(\boldsymbol{\omega}_t) = \{\Phi^i(\boldsymbol{\omega}_t)\}_{i=1}^m$
  From $\Phi(\boldsymbol{\omega}_t)$ compute $\mathbf{G}(\boldsymbol{\omega}_t)$ using Equation 11 ; $\boldsymbol{\mu}_s(t)$ using Equation 13 and $\boldsymbol{\mu}_f(t)$ using Equation 14
  for all $i = 1, \ldots, m$ in parallel **do**
    $\Lambda(\boldsymbol{\omega}_t) = \{$**FAIR2**, $\mathbf{G}(\boldsymbol{\omega}_t), \boldsymbol{\mu}_s(t)\boldsymbol{\mu}_f(t)^\top\}$
    $\mathbf{g}^i(\boldsymbol{\omega}_t) =$ **Client Update**$(\Lambda(\boldsymbol{\omega}_t))$
  end for
  $\boldsymbol{\omega}_{t+1/2} \leftarrow \boldsymbol{\omega}_t - \sum_{i=1}^m \mathbf{g}^i(\boldsymbol{\omega}_t)$
  for all $i = 1, \ldots, m$ in parallel **do**
    $\boldsymbol{\omega}_{t+1}^i =$ **Client Update**(**DATA**, $\boldsymbol{\omega}_{t+1/2}$)
  end for
  $\boldsymbol{\omega}_{t+1} \leftarrow$ average $\left(\boldsymbol{\omega}_{t+1}^i\right)$
end while

FAIR2 The workers use $\mathbf{G}(\boldsymbol{\omega}_t)$,$\boldsymbol{\mu}_s(t)\boldsymbol{\mu}_f(t)^\top$ from the server and the local interaction term $\mathbf{M}^i(\boldsymbol{\omega}_t)$ previously computed to compute their contribution to the stochastic estimate of the fairness gradient and communicate these to the server.

DATA The rest of the iteration implements `FedProxGrad`: the server computes the gradient estimate for ψ and sends $\boldsymbol{\omega}_{t+1/2} = \boldsymbol{\omega}_t - \mathbf{g}_t$ to the clients, which locally update their models ω_{t+1}^i and send them back to the server, which then computes the next global model $\boldsymbol{\omega}_{t+1}$.

For brevity, the time-delay variant `KFFL-TD` and its associated algorithms (Algorithms 4 and 5) are detailed in Appendix C. The following section (5) evaluates the empirical performance of `KFFL` and `KFFL-TD` against baseline federated learning algorithms designed to mitigate demographic bias, utilizing the fairness metrics introduced in the Preliminaries. Furthermore, we examine the communication costs of `KFFL` and `KFFL-TD` relative to these baselines.

5 Experimental Section

In this section, we evaluate the performance of our methods, `KFFL` and `KFFL-TD`, in achieving statistical group fairness in a federated setting. Fairness is assessed using statistical parity and equalized odds for classification models (see Equation 1 2 and the Kolmogorov-Smirnov (KS) 4 distance for regression models. While most of the work in fair federated learning has explored fairness algorithms that aim to achieve *client parity / client fairness* (i.e., consistent performance across clients) , such as in Cui et al. (2021) and Du et al. (2021), it is essential to highlight that client parity algorithms **do not directly target statistical *group* fairness reduction**. Instead, they often achieve statistical *group* fairness as a byproduct rather

Table 2: Communication measured in terms of the number of rounds required for one global update using `KFFL`, `KFFL-TD`, `FedAvg`, and `FairFed`. Algorithms that incorporate fairness, such as `FairFed`, require a similar number of communication rounds as our methods `KFFL` and `KFFL-TD`

Method	Rounds of Communication
KFFL	3
KFFL-TD (Time Delay Variant)	2
Fair Fed Ezzeldin et al. (2023)	3
FedAvg	1

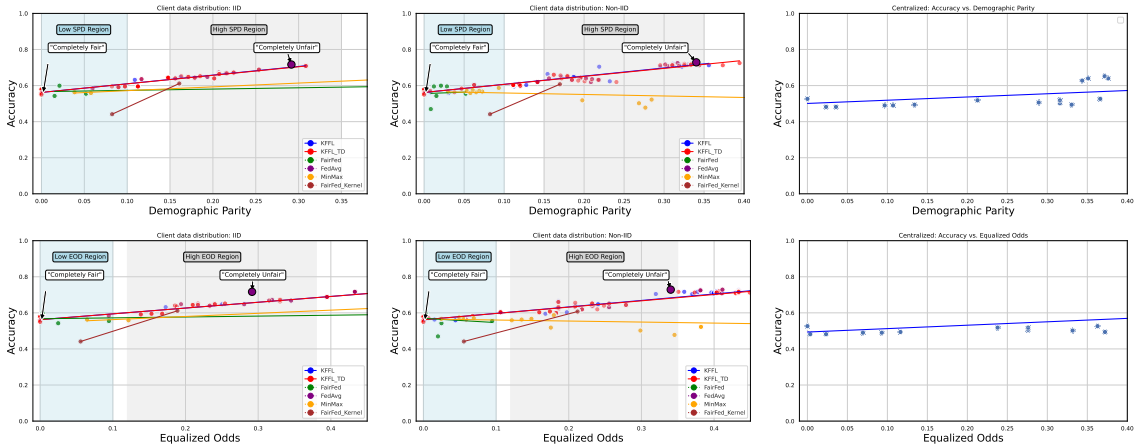

Figure 3: Accuracy versus Demographic Parity (DP) and Equalized Odds (EOD) for `KFFL` and its baselines under IID and Non-IID conditions on the `COMPAS` test dataset. Each point represents a different fairness weight λ ranging 0.01 to 123.16 for both `KFFL` and `KFFL-TD`. The blue region denotes **higher levels of group fairness** (low SPD and EOD), while the gray region indicates **lower levels of group fairness** (high SPD and EOD)

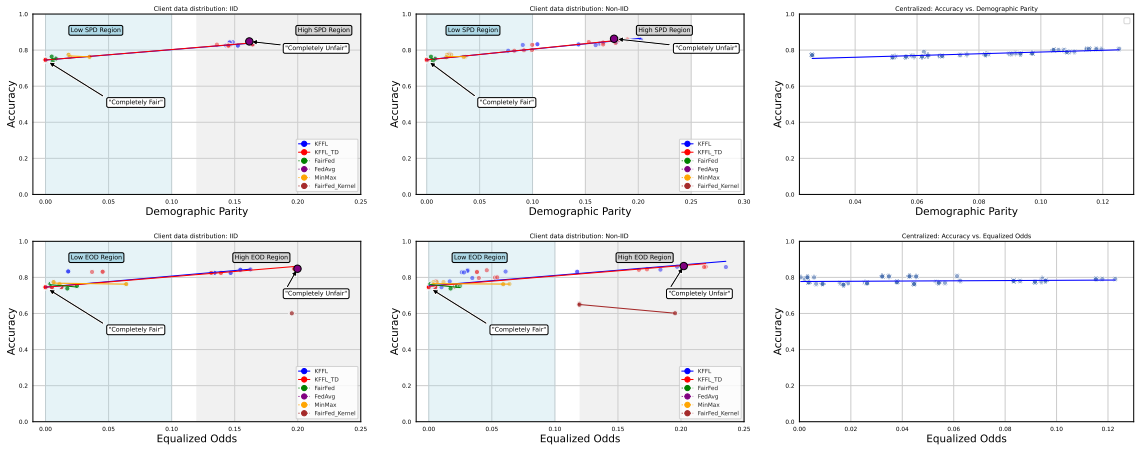

Figure 4: Accuracy versus Demographic Parity (DP) and Equalized Odds (EOD) for `KFFL` and its baselines under IID and Non-IID conditions on the `ADULT` test dataset. Each point represents a different fairness weight λ ranging from 20.00 to 1000.00 for both `KFFL` and `KFFL-TD`. The blue region denotes **higher levels of group fairness** (low SPD and EOD), while the gray region indicates **lower levels of group fairness** (high SPD and EOD).

than a primary objective. Thus, these algorithms are not suitable benchmarks for our approach, as we focus explicitly on addressing statistical group fairness in federated learning.

We compare `KFFL` against five baseline methods: `FedAvg` Li et al. (2020); the `MinMax` algorithm of Papadaki et al. (2022), which aims to optimize model performance for the worst-performing demographic; `FairFed` Ezzeldin et al. (2023), where clients convey localized fairness metrics and the server optimizes weighting coefficients by minimizing the contribution of the poorest-performing client for a chosen fairness metric (eg: Equalized Odds). The framework of `FairFed` allows for different local bias mitigation algorithms. We use `FairFed` with the best performing local bias mitigation algorithm, `FairBatch` Roh et al. (2020). The latter two baselines were chosen as these methods are **explicitly designed to mitigate demographic bias in federated learning**.To highlight the importance of global communication of fairness information, we also consider `Local-Kernel` or `FairFed-Kernel`, which uses `FedAvg` with each local client implementing a local bias mitigation algorithm (similar to Equation 8). Finally, `KFFL` is compared against its time delay variant `KFFL-TD`. The selection of various hyperparameters for `KFFL` and its variants is discussed in Appendix B.1. This section includes a discussion of both common hyperparameters (e.g., batch size, learning rate, local epochs, global rounds) and algorithm-specific hyperparameters, such as the feature map size $\mathbf{D}$ used in `KFFL` and `KFFL-TD`.

We evaluate the performance in two different federated learning settings:**IID** (Independent and Identically Distributed) and **Non-IID** (Non-Independent and Non-Identically Distributed).In the IID setting, each client is provided with an equal number of samples and a consistent data distribution for local training Li et al. (2020). In the Non-IID setting, each client has a different distribution of the protected attribute. Specifically, since the protected group $\mathcal{A}$ is binary with attributes $\mathcal{A}_0$ and $\mathcal{A}_1$, half of the clients have 90 % of $\mathcal{A}_0$ and 10% of $\mathcal{A}_1$, while the other half has 90% of $\mathcal{A}_1$ and 10% of $\mathcal{A}_0$ Li et al. (2020).

For the classification task, we used five datasets commonly applied in group fairness research Han et al. (2023): `Adult`, `COMPAS`, `Bank`, `ACS`, and `German`. The client models for evaluation include Logistic Regression Han et al. (2023) and Neural Networks. Additional results and details can be found in Appendix D.

When the underlying task is regression, we incorporate additional datasets into our evaluation. Beyond the `Adult` dataset, we also consider the `Law School` and `Communities and Crime` datasets, as utilized in the work Agarwal et al. (2019a). For more details on the dataset refer to B.2.

Note on Privacy: While privacy-preserving techniques like Secure Aggregation Bonawitz et al. (2017), as discussed in Ezzeldin et al. (2023), can be incorporated into our approach—enabling clients to send encrypted gradients through such protocols—our primary focus in this work is the development and evaluation of our novel algorithm. Specifically, we aim to explore the Pareto frontier of fairness-performance trade-offs. Integration of privacy-preserving protocols, such as DP-SGD Chua et al. (2024) and Secure Aggregation, is left as future work, as the current evaluation emphasizes statistical fairness over privacy concerns.Similar to prior works in federated learning that do not impose privacy constraints Crawshaw & Liu (2024); Cho et al. (2023); Gu et al. (2021); Malinovsky et al. (2023); Eichner et al. (2019), we assume no privacy-preserving mechanisms in this study.

Note on Full Participation: Furthermore, we consider **full client participation**, consistent with approaches such as `FairFed` Ezzeldin et al. (2023), the `MinMax` algorithm proposed by Papadaki et al. (2022),

Table 3: Communication Costs of `KFFL` and `KFFL-TD`, relative to `FedAvg`. Here, $\varepsilon \triangleq D^2/|\boldsymbol{\omega}|$.. One limitation of `KFFL` and `KFFL-TD` is the additional cost incurred by the exchange of model parameters.However, the use of Random Fourier Features see Algorithms 2 and 3 in our approach ensures that the cost of the additional matrices is effectively mitigated for large model sizes

Method	**Uplink**	**Downlink**
KFFL	$2+\varepsilon$	$2+\varepsilon$
KFFL-TD (Time Delay Variant)	$2+\varepsilon$	$2+\varepsilon$
Fair Fed Ezzeldin et al. (2023)	1	1
FedAvg	1	1

Fairness weight λ	RMSE $\downarrow$	KS Difference $\downarrow$
0.0	0.320332 ± 0.024803	0.604128 ± 0.166035
5.0	0.320936 ± 0.024753	0.618280 ± 0.186275
10.0	0.321824 ± 0.024351	0.551812 ± 0.179296
50.0	0.314788 ± 0.026521	0.391712 ± 0.157367
100.0	0.318976 ± 0.023506	0.290872 ± 0.099638

Table 4: RMSE and KS Difference with standard deviations for 5 runs with `KFFL` with **IID** for the Communities and Crime Dataset.Optimal points are those with lower RMSE (for accuracy) and KS (for fairness)

Fairness weight λ	RMSE $\downarrow$	KS Difference $\downarrow$
0.00	0.810188 ± 0.002067	0.439208 ± 0.175972
0.01	0.810068 ± 0.002152	0.424060 ± 0.164314
0.10	0.809940 ± 0.002189	0.487244 ± 0.139673
1.00	0.810052 ± 0.002201	0.200248 ± 0.170934
5.00	0.810100 ± 0.002022	0.212672 ± 0.065094

Table 5: RMSE and KS Difference with standard deviations for 5 runs KFFL with **Non-IID** for the Law School Dataset.Optimal points are those with lower RMSE (for accuracy) and KS (for fairness)

and other works in federated learning Zhang et al. (2023); Li et al. (2023a;b); Zakerinia et al. (2023); Huang et al. (2022).

6 Performance Evaluation of KFFL

Existing fair classifiers and regressors that balance fairness and accuracy typically optimize for a single point on the trade-off curve (see Figure 3 and Figure 4), whereas methods such as `KFFL` and `KFFL-TD` explore the Pareto frontier more comprehensively to provide a better *trade-off* between accuracy and statistical fairness. This narrow focus restricts the exploration of alternative solutions, potentially leading to suboptimal outcomes that fail to reflect diverse stakeholder preferences for fairness and performance. The `KFFL` and `KFFL-TD` methods proposed in the previous sections overcome this limitation by systematically exploring multiple points along the Pareto front, offering a broader spectrum of optimal trade-offs between fairness and accuracy tailored to specific needs for various stakeholders.

6.1 KFFL offers better Pareto Frontier

For the classification task, we evaluate the models' test accuracy and fairness metrics, focusing on Demographic Parity (SPD) 1 and Equalized Odds (EOD) 2, across five datasets: `BANK`, `ACS`, `COMPAS`, `ADULT`, and `GERMAN`. Each dataset considers a single protected binary sensitive attribute. For example, in `COMPAS`, the protected attribute is race (black/white), while in `ADULT`, it is sex (male/female). This subsection presents results for the Logistic Regression model, with the Appendix D covering the performance of the Neural Network model.

`KFFL` and its baselines are compared under both IID and Non-IID conditions on the `COMPAS` *test* dataset in Figure 3. The blue region, referred to as the 'Low SPD Region' and 'Low EOD Region,' represents low statistical parity discrepancy (SPD) (see Equation 1) and low equalized odds discrepancy (EOD) (see Equation 2). These regions correspond to **higher levels of group fairness**. In contrast, the gray region, labeled as the 'High SPD Region' and 'High EOD Region,' indicates **lower levels of group fairness**.

Points labeled 'completely fair' indicate trade off points where the model achieves no statistical parity or equalized odds discrepancies, as defined in Equations 1 and 2. In contrast, points labeled 'completely unfair' represent the performance of the standard `FedAvg` model, which is trained without any fairness objective (i.e., $\lambda = 0$ in the distributed setting of Equation 8). Each point reflects a different fairness weight λ, ranging from 20.00 to 1000.00 for both `KFFL` and `KFFL-TD` using the `ADULT` dataset and from 0.01 to 123.16 for `COMPAS`

	Method	COMPAS	BANK	ACS	GERMAN
Acc. (↑)	FedAvg	61.13 ± 1.25	91.21 ± 1.10	81.14 ± 1.20	72.50 ± 1.15
	KFFL	**60.30** ± 1.30	**90.23** ± 1.25	**79.12** ± 1.35	**72.50** ± 1.10
	KFFL-TD	**59.51** ± 1.45	**90.78** ± 1.20	**81.12** ± 1.05	**72.50** ± 1.40
	FairFed/FairBatch	55.47 ± 1.50	88.05 ± 1.30	58.77 ± 1.20	30.00 ± 1.25
	KHSIC-Local	44.13 ± 1.55	88.77 ± 1.15	58.77 ± 1.05	70.00 ± 1.35
	MinMax	56.68 ± 1.40	64.93 ± 1.20	58.77 ± 1.25	70.00 ± 1.10
SPD (↓)	FedAvg	0.16 ± 0.02	0.23 ± 0.01	0.08 ± 0.04	0.19 ± 0.03
	KFFL	**0.06** ± 0.01	**0.05** ± 0.03	**0.04** ± 0.02	**0.00** ± 0.04
	KFFL-TD	**0.05** ± 0.02	**0.06** ± 0.01	**0.00** ± 0.03	**0.00** ± 0.02
	FairFed/FairBatch	0.05 ± 0.03	0.00 ± 0.02	0.00 ± 0.01	0.00 ± 0.03
	KHSIC-Local	0.08 ± 0.02	0.00 ± 0.03	0.00 ± 0.02	0.00 ± 0.01
	MinMax	0.03 ± 0.01	0.21 ± 0.02	0.00 ± 0.03	0.00 ± 0.02
EOD (↓)	FedAvg	0.23 ± 0.02	0.16 ± 0.01	0.04 ± 0.02	0.02 ± 0.03
	KFFL	**0.07** ± 0.01	**0.04** ± 0.02	**0.02** ± 0.03	**0.00** ± 0.02
	KFFL-TD	**0.08** ± 0.02	**0.05** ± 0.01	**0.03** ± 0.02	**0.00** ± 0.03
	FairFed/FairBatch	0.09 ± 0.02	0.16 ± 0.03	0.00 ± 0.01	0.00 ± 0.02
	KHSIC-Local	0.05 ± 0.03	0.00 ± 0.02	0.00 ± 0.01	0.00 ± 0.03
	MinMax	0.03 ± 0.02	0.02 ± 0.01	0.00 ± 0.03	0.00 ± 0.02

Table 6: Trade-off points for datasets `BANK`, `ACS`, and `GERMAN`, alongside the previously analyzed `COMPAS`, under Non-IID client distributions. Models are trained and evaluated on test datasets after three runs. **KFFL** and **KFFL-TD** are showcased, demonstrating Pareto optimality across accuracy and fairness metrics. Lower SPD and EOD values indicate higher fairness; it is observed that `KFFL` and `KFFL-TD` provide better trade-off points, achieving a balance between accuracy and fairness with reduced group disparity.

dataset. More details on the choice of λ for other datasets is given in section Appendix B.5. As λ increases beyond this range, we observe non-optimal points to the right of the 'completely unfair' point.

It can be seen that other baselines, such as `FairFed` and `MinMax`, are not designed to produce a smooth Pareto Frontier between accuracy and fairness. For `FairFed` Ezzeldin et al. (2023), the tradeoff is controlled by a parameter called the "fairness budget" β, which varies from 0.1 to 5, based on the recommendations in Ezzeldin et al. (2023). However, optimal performance of the `FairFed` baseline requires a local debiasing mechanism, for which we use the `FairBatch` algorithm Roh et al. (2020) for comparison.

For `MinMax`, a "global adversary rate" Papadaki et al. (2022) is used to control the reduction of expected loss for the worst-performing demographic. To explore the accuracy-fairness tradeoff, we varied this parameter from 0.001 to 0.1 based on the setting in that paper.

In the right column, the 'Centralized' method refers to a non-distributed data setting, corresponding to Equation 8, where the full kernel is used as a regularizer. Similarly, Figure 4 provides an evaluation for the `ADULT` *test* dataset. Each point represents a different fairness weight λ, ranging from 0 to 0.01 for both `KFFL` and `KFFL-TD`.

The "Centralized" version for each dataset illustrates that using the full kernel allows for a clear tradeoff between accuracy and fairness. Our **distributed KHSIC approach `KFFL` preserves this tradeoff** in Figure 3 and 4. Specifically, `KFFL` and `KFFL-TD` maintain competitive accuracy (approximately 0.85 to 0.78) on the `ADULT` dataset and between 0.6 to 0.7 on the `COMPAS` dataset, **across different *fairness regions***. In contrast, methods like `FairFed` and `MinMax` perform similarly to our methods in **lower levels of group fairness** but significantly underperform in regions with **higher levels of group fairness** often failing to provide any tradeoff points in the areas identified as **high fairness regions**.

A key drawback of `FairFed` and `MinMax` is that their hyperparameter choices limit the smooth exploration of the Pareto frontier. In contrast, `KFFL` and `KFFL-TD` provide a more effective accuracy-fairness tradeoff, achieving a smoother and more optimal Pareto frontier across various fairness regions, consistently maintaining or surpassing the performance of these baselines. Notably, `FairFed` and `MinMax` fail to produce tradeoff points in the higher Equalized Odds (EOD) regions, with their performance consistently falling below Pareto-optimal levels across both evaluation metrics and distributions.

Moreover, the smooth exploration of the Pareto frontier by `KFFL` and `KFFL-TD` persists even under more challenging federated learning conditions, such as when the data distribution is Non-IID. Unlike `FairFed`

and MinMax, which require extensive tuning for specific evaluation metrics, KFFL and KFFL-TD **perform well without metric-specific hyperparameter optimization**.

However, the choice of the fairness weight hyperparameter λ is crucial for effectively utilizing KFFL and KFFL-TD. Increasing the fairness weight λ beyond a certain threshold can push the model's performance towards lower levels of group fairness Figure 3 and 4. While a tradeoff between accuracy and fairness is inevitable in group fairness problems, KFFL and KFFL-TD consistently offer a superior balance between accuracy and fairness.

Figures 4 and 3 illustrate the tradeoff achieved when KHSIC is applied as a local fairness regularizer in the methods Local-Kernel and FairFed-Kernel. In these methods, each client independently solves Equation 8 without global communication of fairness gradients. The results clearly show that local debiasing methods fail to achieve a tradeoff comparable to the centralized approach, where all data is consolidated and optimized by a single entity using Equation 8. These findings underscore the importance of the principled strategies employed by KFFL and KFFL-TD.

6.2 KFFL has better Tradeoff Points

Table 6 highlights key trade-off points for additional datasets, including BANK, ACS, and GERMAN, alongside the previously analyzed COMPAS dataset. The models were trained under **Non-IID** client distribution, and the trade-off points were computed based on test dataset performance after three runs.

To showcase the benefits of our approach, we selected specific trade-off points. For the COMPAS dataset, if the desired accuracy is around 60%, similar to the performance of FedAvg, KFFL and KFFL-TD achieve a small SPD of 0.06. In contrast, other baselines achieve similar SPD but with an approximately 5% drop in accuracy. A similar trend becomes evident with the ACS dataset, where KFFL-TD provides a trade-off point of relatively high 80% accuracy for low SPD value. For the baselines, a comparable fairness point results in a 22% drop in accuracy.

Additionally, KFFL is robust across different evaluation metrics,In the BANK dataset, KFFL and KFFL-TD offer a trade-off point of 90% accuracy with SPD values of 0.05 and 0.04, respectively. In comparison, FairFed achieves 0.00 SPD with 88% accuracy but exhibits a high EOD of 0.16, as it was optimized for reducing SPD in our experiments. This underscores the importance of carefully selecting an evaluation metric in the FairFed approach. Additionally, MinMax shows a high SPD but low EOD, indicating that one evaluation metric may be favored over the other, depending on the dataset with this approach. In contrast, KFFL and KFFL-TD provide a trade-off point that reduces unfairness across both evaluation metrics for this dataset.

6.3 KFFL offers reduced Communication Overhead

Table 2 compares the *communication rounds* required by our proposed methods, KFFL and its time-delay variant KFFL-TD see (C), with baseline methods such as FedAvg and FairFed Ezzeldin et al. (2023). While KFFL requires a similar number of communication rounds as FairFed, KFFL-TD reduces the number of rounds needed for a global model update by using stale fairness gradients, thereby lowering the total communication rounds needed per update

One limitation of KFFL and KFFL-TD is the additional *communication overhead* caused by the exchange of extra model parameters. Table 3 compares the relative communication costs (per client, per iteration) for the FedAvg and KFFL algorithms. Let $|\boldsymbol{\omega}|$ denote the size of the model. For each iteration, FedAvg incurs uplink (client-to-server) and downlink (server-to-client) communication costs of $|\boldsymbol{\omega}|$. In contrast, KFFL requires the client to transmit $\boldsymbol{\omega}_t^i, \Phi^i(\boldsymbol{\omega}_t), \mathbf{g}^i(\boldsymbol{\omega}_t)$ (see Algorithm 2), resulting in an uplink cost of $2|\boldsymbol{\omega}|+D^2$, and the server to transmit $\boldsymbol{\omega}_t, \Lambda(\boldsymbol{\omega}_t), \boldsymbol{\omega}_{t+1/2}$, incurring a similar downlink cost of $2|\boldsymbol{\omega}| + D^2$ (see Algorithm 3). Notably, D^2 is often smaller than $|\boldsymbol{\omega}|$, particularly for large models. KFFL-TD (Appendix C) incurs comparable communication costs but reduces the number of communication rounds per iteration (see Figure 5).

To quantify this overhead, Table 3 defines $\epsilon \equiv \frac{D^2}{|\boldsymbol{\omega}|}$, where D is the data dimensionality. This results in total uplink and downlink communication costs of approximately $2 + \epsilon$ per iteration for KFFL, compared to 1 for standard methods like FedAvg. Although FairFed also exchanges local and global accuracy and fairness

metrics during each global update, we exclude these scalar quantities from the comparison for simplicity in Table 3.

To mitigate the overhead from these additional matrices, we leveraged Random Fourier Features (RFF). RFF approximates kernel functions with finite-dimensional feature mappings, keeping the size of the extra matrices proportional to ϵ. Since ϵ is relatively small when $D^2 \ll |\omega|$, the additional communication cost becomes negligible in practice. This ensures that the overall communication costs of `KFFL` and `KFFL-TD` remains only one order of magnitude higher due to the additional exchange of model parameters, while providing the benefits of ensuring learning a model that is fair with respect to different demographic groups.

6.4 KFFL provides better tradeoff points for Regression Tasks

We conduct experiments using a linear regression model to evaluate its performance in terms of both accuracy and group fairness. Accuracy for regression tasks is measured using Root Mean Square Error (RMSE), which quantifies the average magnitude of errors between predicted and actual values; a lower RMSE indicates higher accuracy. Group fairness is assessed using the Kolmogorov-Smirnov (KS) distance (Equation 4), where a lower KS distance signifies greater statistical fairness.

Table 4 presents the performance of `KFFL` under varying fairness weights (λ) in the **IID** setting using the `Communities and Crime` dataset. Increasing λ emphasizes the fairness term in `KFFL`. Across different λ values, the model achieves solutions with significantly reduced KS distance while maintaining RMSE comparable to standard distributed regression tasks (e.g., $\lambda = 0$). Notably, the KS distance is reduced by half with minimal impact on RMSE, demonstrating a balanced trade-off between accuracy and fairness. A similar pattern emerges under **Non-IID** settings with the `Law School` dataset, as detailed in Table 5.

Additional results in the Appendix, including Tables 15, 13, 11, 12, 14, and 10, extend these findings to the `Law School`, `Communities and Crime`, and `Adult` datasets. These results underscore the effectiveness of `KFFL` in balancing accuracy and fairness by appropriately tuning the fairness weight λ in the regularization term for regression tasks.

7 Conclusions

This work introduces a systematic approach for training group-fair machine learning models in a federated setting. Our method leverages KHSIC as a fairness regularizer to capture complex, non-linear dependencies between model outputs and sensitive attributes, ensuring approximate statistical parity without requiring access to sensitive attributes during training. The proposed method, `KFFL`, significantly reduces the communication and computation costs of a naive implementation by employing Random Feature Maps and a novel federated proximal gradient algorithm, `FedProxGrad`, which accommodates the non-convexity of both the data-fitting term and the fairness regularizer.

Experimental results demonstrate that `KFFL` performs robustly across diverse client data distributions and standard datasets commonly used to evaluate fair learning methods. It achieves strong performance in both regression and classification tasks by thoroughly exploring the Pareto frontier, providing a smoother and more optimal trade-off between accuracy and fairness.

Limitations and Future Work

`KFFL` effectively enforces statistical parity but is specifically designed for this particular fairness definition. Expanding the method to include other notions of group fairness, such as conditional variants of KHSIC, could significantly enhance its versatility. However, the current framework assumes full client participation, making it less suitable for scenarios with partial participation or privacy constraints. To address these challenges, future work could incorporate differentially private gradient estimation techniques and develop more advanced stochastic analyses to support partial participation. Additionally, the convergence rate of `FedProxGrad` is influenced by the variance σ^2 of the stochastic gradient estimate for the fairness regularizer, which imposes a variance-dependent floor on the efficiency of the method. Future work will aim to overcome these challenges by exploring variance reduction techniques or alternative strategies to mitigate the impact of σ^2 on convergence rates.

References

pyrfm: A library for random feature maps in python. URL https://neonnnnn.github.io/pyrfm/random_feature.html.

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pp. 120–129. PMLR, 2019a.

Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pp. 120–129. PMLR, 2019b.

Sina Baharlouei, Maher Nouiehed, Ahmad Beirami, and Meisam Razaviyayn. R\'enyi fair inference. *arXiv preprint arXiv:1906.12005*, 2019a.

Sina Baharlouei, Maher Nouiehed, Ahmad Beirami, and Meisam Razaviyayn. R\'enyi fair inference. *arXiv preprint arXiv:1906.12005*, 2019b.

Yajie Bao, Michael Crawshaw, Shan Luo, and Mingrui Liu. Fast composite optimization and statistical recovery in federated learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 1508–1536. PMLR, 2022.

Matias Barenstein. Propublica's compas data revisited. *arXiv preprint arXiv:1906.04711*, 2019.

Barry Becker and Ronny Kohavi. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, 2017.

Hongyan Chang and Reza Shokri. Bias propagation in federated learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=V7CYzdruWdm.

Bhaskar Ray Chaudhury, Linyi Li, Mintong Kang, Bo Li, and Ruta Mehta. Fairness in federated learning via core-stability. *arXiv preprint arXiv:2211.02091*, 2022.

Yae Jee Cho, Pranay Sharma, Gauri Joshi, Zheng Xu, Satyen Kale, and Tong Zhang. On the convergence of federated averaging with cyclic client participation. In *International Conference on Artificial Intelligence and Statistics*, pp. 10351–10375, 2023. URL https://arxiv.org/abs/2302.03109.

Lynn Chua, Badih Ghazi, Pritish Kamath, Ravi Kumar, Pasin Manurangsi, Amer Sinha, and Chiyuan Zhang. How private is dp-sgd? *arXiv preprint arXiv:2403.17673*, 2024.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression with wasserstein barycenters. *Advances in Neural Information Processing Systems*, 33:7321–7331, 2020a.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Fair regression via plug-in estimator and recalibration with statistical guarantees. *Advances in Neural Information Processing Systems*, 33:19137–19148, 2020b.

Jose Correa, Andres Cristi, Paul Duetting, and Ashkan Norouzi-Fard. Fairness and bias in online selection. In *International conference on machine learning*, pp. 2112–2121. PMLR, 2021.

Michael Crawshaw and Mingrui Liu. Federated learning under periodic client participation and heterogeneous data: A new communication-efficient algorithm and analysis. *arXiv preprint arXiv:2410.23131*, 2024. URL https://arxiv.org/abs/2410.23131.

Sen Cui, Weishen Pan, Jian Liang, Changshui Zhang, and Fei Wang. Addressing algorithmic disparity and performance inconsistency in federated learning. *Advances in Neural Information Processing Systems*, 34: 26091–26102, 2021.

Kate Donahue and Jon Kleinberg. Models of fairness in federated learning. *arXiv preprint arXiv:2112.00818*, 2021.

Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science advances*, 4(1):eaao5580, 2018.

Wei Du, Depeng Xu, Xintao Wu, and Hanghang Tong. Fairness-aware agnostic federated learning. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pp. 181–189. SIAM, 2021.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012a.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012b.

Hubert Eichner, Tomer Koren, Brendan McMahan, Nathan Srebro, and Kunal Talwar. Semi-cyclic stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1764–1773, 2019. URL `https://arxiv.org/abs/1904.10120`.

Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and A Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7494–7502, 2023.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pp. 63–77. Springer, 2005a.

Arthur Gretton, Ralf Herbrich, Alexander Smola, Olivier Bousquet, Bernhard Schölkopf, et al. Kernel methods for measuring independence. 2005b.

Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. *Advances in Neural Information Processing Systems*, 34:12052–12064, 2021. URL `https://arxiv.org/abs/2106.07070`.

Raffay Hamid, Ying Xiao, Alex Gittens, and Dennis DeCoste. Compact random feature maps. In *International conference on machine learning*, pp. 19–27. PMLR, 2014.

Xiaotian Han, Jianfeng Chi, Yu Chen, Qifan Wang, Han Zhao, Na Zou, and Xia Hu. Ffb: A fair fairness benchmark for in-processing group fairness methods. *arXiv preprint arXiv:2306.09468*, 2023.

Shengyuan Hu, Steven Wu, and Virginia Smith. Fair federated learning via bounded group loss.

Kaixuan Huang, Xinran Gu, Jingzhao Zhang, and Longbo Huang. Federated stochastic primal-dual learning with differential privacy. *arXiv preprint arXiv:2204.12284*, 2022. URL `https://arxiv.org/abs/2204.12284`.

Ajil Jalal, Sushrut Karmalkar, Jessica Hoffmann, Alex Dimakis, and Eric Price. Fairness for image generation with uncertain sensitive attributes. In *International Conference on Machine Learning*, pp. 4721–4732. PMLR, 2021.

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*, pp. 5132–5143. PMLR, 2020.

Kevin Kim and Alex Gittens. Learning fair canonical polyadical decompositions using a kernel independence criterion. *arXiv preprint arXiv:2104.13504*, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

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.

Wei Li, Jian Wang, Xiaoyang Zhang, Yifan Zhang, and Yanfeng Wang. Privacy protection bottom-up hierarchical federated learning. In *International Conference on Artificial Intelligence and Security*, pp. 25–36, 2023a. URL `https://link.springer.com/chapter/10.1007/978-981-97-5562-2_3`.

Wei Li, Jian Wang, Xiaoyang Zhang, Yifan Zhang, and Yanfeng Wang. Anofel: Supporting anonymity for privacy-preserving federated learning. *arXiv preprint arXiv:2306.06825*, 2023b. URL `https://arxiv.org/abs/2306.06825`.

Grigory Malinovsky, Samuel Horváth, Konstantin Burlachenko, and Peter Richtárik. Improving accelerated federated learning with compression and importance sampling. *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023. URL `https://arxiv.org/abs/2302.03662`.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

Ninareh Mehrabi, Cyprien de Lichy, John McKay, Cynthia He, and William Campbell. Towards multi-objective statistically fair federated learning. *arXiv preprint arXiv:2201.09917*, 2022.

Omid Memarrast, Linh Vu, and Brian D Ziebart. Superhuman fairness. In *International Conference on Machine Learning*, pp. 24420–24435. PMLR, 2023.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Afroditi Papadaki, Natalia Martinez, Martin Bertran, Guillermo Sapiro, and Miguel Rodrigues. Minimax demographic group fairness in federated learning. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 142–159, 2022.

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.

Adrián Pérez-Suay, Valero Laparra, Gonzalo Mateo-García, Jordi Muñoz-Marí, Luis Gómez-Chova, and Gustau Camps-Valls. Fair kernel learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 339–355. Springer, 2017.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

Yuji Roh, Kangwook Lee, Steven Euijong Whang, and Changho Suh. Fairbatch: Batch selection for model fairness. *arXiv preprint arXiv:2012.01696*, 2020.

Yaniv Romano, Stephen Bates, and Emmanuel Candes. Achieving equalized odds by resampling sensitive attributes. *Advances in neural information processing systems*, 33:361–371, 2020.

Teresa Salazar, Miguel Fernandes, Helder Araujo, and Pedro Henriques Abreu. Fair-fate: Fair federated learning with momentum. *arXiv preprint arXiv:2209.13678*, 2022.

Teresa Salazar, Helder Araújo, Alberto Cano, and Pedro Henriques Abreu. A survey on group fairness in federated learning: Challenges, taxonomy of solutions and directions for future research. *arXiv preprint arXiv:2410.03855*, 2024.

Quoc Tran Dinh, Nhan H Pham, Dzung Phan, and Lam Nguyen. FedDR: Randomized Douglas-Rachford splitting algorithms for nonconvex federated composite optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 30326–30338. Curran Associates, Inc., 2021.

Ganghua Wang, Ali Payani, Myungjin Lee, and Ramana Kompella. Mitigating group bias in federated learning: Beyond local fairness. *arXiv preprint arXiv:2305.09931*, 2023.

Jia Wang and Jueyou Li. Federated dual averaging learning algorithm with delayed gradients for composite optimization. *Available at SSRN 4507875*, 2023.

Linda F Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *Advances in neural information processing systems*, 29, 2016.

Honglin Yuan, Manzil Zaheer, and Sashank Reddi. Federated composite optimization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 12253–12266. PMLR, 2021.

Hossein Zakerinia, Shayan Talaei, Giorgi Nadiradze, and Dan Alistarh. Blockchain-based optimized client selection and privacy preserved framework for federated learning. *arXiv preprint arXiv:2308.04442*, 2023. URL `https://arxiv.org/abs/2308.04442`.

Yuchen Zeng, Hongxu Chen, and Kangwook Lee. Improving fairness via federated learning. *arXiv preprint arXiv:2110.15545*, 2021.

Yuxin Zhang, Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Advancing personalized federated learning: Group privacy, fairness, and beyond. *SN Computer Science*, 4(5):1–13, 2023. URL `https://link.springer.com/article/10.1007/s42979-023-02292-0`.

A Appendix

A.1 Related Works

Methods for ensuring fairness within centralized machine learning are typically categorized into three distinct groups: pre-processing, in-processing, and post-processing Mehrabi et al. (2021). In federated learning, bias mitigation methods predominantly fall into in-processing approaches, although some work has also been done in post-processing methods.

For in-processing methods of bias mitigation in federated learning, Ezzeldin et al. (2023) is notable for its versatility and compatibility with various *local* bias mitigation techniques.In this approach, clients convey their localized fairness metrics to the server, which then optimizes weighting coefficients to minimize the contribution of the poorest-performing client with respect to a chosen fairness metric. Papadaki et al. (2022) optimize model performance on the worst-performing demographic by adopting a minimax optimization framework. Salazar et al. (2022) propose a fairness-aware momentum-based method to address bias in federated learning. The approach in Mehrabi et al. (2022) strives for fair federated learning but requires the server to maintain a validation dataset. Zeng et al. (2021) address the challenge of bias mitigation in federated learning through a bi-level optimization problem; their analysis predominantly pertains to specific loss functions. Pentyala et al. (2022) consider post-processing and pre-processing approaches to ensuring fairness. Cui et al. (2021) require the clients to achieve Pareto optimality with respect to both fairness and accuracy.

Other variants of group fairness have been explored in the federated setting. Hu et al. introduce the concept of bounded group loss as a facet of group fairness in federated learning, although their work does not specifically develop algorithms targeting bias mitigation. Chang & Shokri (2023) analyze how bias within participating clients can propagate during the training process but do not propose methods aimed at explicitly addressing group fairness.A comprehensive summary of additional approaches in group fairness federated learning is provided in Table 3 of Salazar et al. (2024). For baseline comparisons, we limit our focus to most **published works in group fairness federated learning** except the recent Wang et al. (2023).

Significant related works from the centralized setting that aim to ensure fair models include Pérez-Suay et al. (2017), which leverages the (non-kernel) Hilbert-Schmidt Independence Criterion (HSIC) to promote the learning of fair kernel machines, and Baharlouei et al. (2019a), which incorporates the Rényi correlation as a regularization term to achieve statistically fair models.

A.2 Proof of Theorem 1

In the following, $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \,|\, \boldsymbol{\omega}_t]$ denotes the expectation conditioned on all sources of randomness in the algorithm up to and including the calculation of $\boldsymbol{\omega}_t$, and $\mathbb{E}_i[\cdot] = \frac{1}{N}\sum_{i=1}^{N}[\cdot]$ denotes the average of a quantity over the workers.

The computations are complicated by the presence of a composite objective and stochasticity in our estimate of $\nabla\psi(\boldsymbol{\omega})$, but the conceptual outline of the proof of the convergence rate follows that of the proof of the convergence rate for FedProx in Li et al. (2020). Namely,

- First, we establish that the distance between consecutive iterates, $\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2$, is upper bounded by the quantity $\mathbb{E}_i\|\nabla\ell^i(\boldsymbol{\omega}_t) + \nabla\psi(\boldsymbol{\omega}_t)\|_2$, and use the bounded dissimilarity condition to upper bound the latter by a multiple of the composite objective gradient at $\boldsymbol{\omega}_t$, $\|\nabla F(\boldsymbol{\omega}_t)\|_2^2$, plus a term due to noise.

- Next, we use this result and the smoothness of the composite objective to establish that after one round of the algorithm, the composite objective satisfies $\mathbb{E}_t F(\boldsymbol{\omega}_{t+1}) \leq F(\boldsymbol{\omega}_t) - \alpha\mathbb{E}_t\|\nabla F(\boldsymbol{\omega}_t)\|_2^2 +$ (vanishing terms) + (noise).

- We conclude that if the step-size α is chosen appropriately, then the objective decreases in expectation at each iteration, up to the noise level. A standard argument with Jensen's inequality and telescoping sums delivers the claimed convergence rate.

Proof. We flesh out the preceeding outline.

Iterate proximity To bound the iterate proximity, begin by introducing the local exact minimizer,

$$\widehat{\boldsymbol{\omega}}_{t+1}^i = \operatorname{argmin}_{\boldsymbol{\omega}} \ell_t^i(\boldsymbol{\omega}) := \ell^i(\boldsymbol{\omega}) + \frac{1}{2\alpha}\|\boldsymbol{\omega} - \boldsymbol{\omega}_{t+1/2}\|_2^2.$$

The data-fitting term ℓ^i is L_--weakly convex and the quadratic regularizer is $\frac{1}{\alpha}$-strongly convex, so the local objective ℓ_t^i is μ strongly convex for $\mu = \frac{1}{\alpha} - L_-$.

The μ-strong convexity of ℓ_t^i implies that the iterate distance between $\boldsymbol{\omega}_t$ and $\boldsymbol{\omega}_{t+1}^i$ can be estimated using the size of the gradient of ℓ_t^i at those models:

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\|_2 &\leq \|\boldsymbol{\omega}_{t+1}^i - \widehat{\boldsymbol{\omega}}_{t+1}^i\|_2 + \|\boldsymbol{\omega}_t - \widehat{\boldsymbol{\omega}}_{t+1}^i\|_2 \\ &\leq \frac{1}{\mu}\left[\|\nabla \ell_t^i(\boldsymbol{\omega}_{t+1}^i)\|_2 + \|\nabla \ell_t^i(\boldsymbol{\omega}_t)\|_2\right].\end{aligned}$$

Employing the γ-suboptimality of $\boldsymbol{\omega}_{t+1}^i$ to estimate the size of $\|\nabla \ell_t^i(\boldsymbol{\omega}_{t+1}^i)\|_2$ refines this estimate to

$$\|\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\|_2 \leq \frac{1+\gamma}{\mu}\|\nabla \ell_t^i(\boldsymbol{\omega}_t)\|_2.$$

We note that

$$\nabla \ell_t^i(\boldsymbol{\omega}_t) = \nabla \ell^i(\boldsymbol{\omega}_t) + \frac{1}{\alpha}(\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t+1/2}) = \nabla \ell^i(\boldsymbol{\omega}_t) + \mathbf{g}_t,$$

and consequently

$$\|\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\|_2 \leq \frac{1+\gamma}{\mu}\Big[\|\nabla \ell^i(\boldsymbol{\omega}_t) + \nabla\psi(\boldsymbol{\omega}_t)\|_2 + \|\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t\|_2\Big].$$

Using Jensen's inequality delivers

$$\begin{aligned}\mathbb{E}_t\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2 &\leq \mathbb{E}_t\big[\mathbb{E}_i\|\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\|_2^2\big] \\ &\leq 2\left(\frac{1+\gamma}{\mu}\right)^2 \mathbb{E}_t\Big[\mathbb{E}_i\|\nabla \ell^i(\boldsymbol{\omega}_t) + \nabla\psi(\boldsymbol{\omega}_t)\|_2^2 + \|\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t\|_2^2\Big] \\ &\leq 2\left(\frac{1+\gamma}{\mu}\right)^2 \Big[B^2\|\nabla F(\boldsymbol{\omega}_t)\|_2^2 + G^2 + \sigma^2\Big]. \qquad (20)\end{aligned}$$

The last inequality holds because of the bounded dissimilarity condition and the upper bound on the variance of $\mathbf{g}_t$.

Similarly,

$$\begin{aligned}\mathbb{E}_t\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2 &\leq \mathbb{E}_t\big[\mathbb{E}_i\|\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\|_2\big] \\ &\leq \frac{1+\gamma}{\mu}\mathbb{E}_t\Big[\sqrt{\mathbb{E}_i\|\nabla \ell^i(\boldsymbol{\omega}_t) + \nabla\psi(\boldsymbol{\omega}_t)\|_2^2} + \sqrt{\|\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t\|_2^2}\Big] \\ &\leq \frac{1+\gamma}{\mu}\Big[B\|\nabla F(\boldsymbol{\omega}_t)\|_2 + G + \sigma\Big]. \qquad (21)\end{aligned}$$

Objective Decrease The L-smoothness of the composite objective implies that

$$\begin{aligned}\mathbb{E}_t F(\boldsymbol{\omega}_{t+1}) &\leq \mathbb{E}_t\left[F(\boldsymbol{\omega}_t) + \langle \nabla F(\boldsymbol{\omega}_t), \boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\rangle + \frac{L}{2}\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2\right] \\ &= F(\boldsymbol{\omega}_t) - \alpha\|\nabla F(\boldsymbol{\omega}_t)\|_2^2 + \mathbb{E}_t\left[\langle \nabla F(\boldsymbol{\omega}_t), \underbrace{\boldsymbol{\omega}_{t+1} - (\boldsymbol{\omega}_t - \alpha\nabla F(\boldsymbol{\omega}_t))}_{=\Delta_t}\rangle\right] \\ &\quad + \frac{L}{2}\mathbb{E}_t\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2.\end{aligned} \tag{22}$$

Equation 20 establishes that, up to noise terms, the term $\mathbb{E}_t\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2$ scales like $\alpha^2\|\nabla F(\boldsymbol{\omega}_t)\|_2^2$, because μ^{-2} is on the order of α^2. Now we develop a series of estimates to establish that the quantity $\mathbb{E}_t\langle F(\boldsymbol{\omega}_t), \Delta_t\rangle$ also scales like $\alpha^2\|\nabla F(\boldsymbol{\omega}_t)\|_2^2$, up to noise terms.

We begin by using the γ-suboptimality of $\boldsymbol{\omega}_{t+1}^i$ to find a useful expression for Δ_t. In particular, γ-suboptimality implies that

$$\begin{aligned}\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \frac{1}{\alpha}(\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_{t+1/2}) &= \nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \mathbf{g}_t + \frac{1}{\alpha}(\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t) \\ &= \big(\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \nabla\psi(\boldsymbol{\omega}_t)\big) - (\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t) + \frac{1}{\alpha}(\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t) \\ &= \mathbf{e}_{t+1}^i,\end{aligned}$$

where $\|\mathbf{e}_{t+1}^i\|_2 \leq \gamma\|\nabla\ell_t^i(\boldsymbol{\omega}_t)\|_2$. Consequently,

$$\begin{aligned}\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t &= \mathbb{E}_i\left[\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\right] \\ &= \alpha\mathbb{E}_i\left[\mathbf{e}_{t+1}^i - \big(\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \nabla\psi(\boldsymbol{\omega}_t)\big) + (\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t)\right]\end{aligned}$$

and, adding and subtracting terms judiciously yields

$$\begin{aligned}&= \alpha\mathbb{E}_i\big[\mathbf{e}_{t+1}^i - \nabla F(\boldsymbol{\omega}_{t+1}) + \nabla F(\boldsymbol{\omega}_{t+1}) \\ &\qquad - \big(\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \nabla\psi(\boldsymbol{\omega}_t)\big) + \nabla\psi(\boldsymbol{\omega}_{t+1}^i) - \nabla\psi(\boldsymbol{\omega}_{t+1}^i) \\ &\qquad + (\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t)\big] \\ &= -\alpha\nabla F(\boldsymbol{\omega}_{t+1}) - \alpha\mathbb{E}_i\big[(\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \nabla\psi(\boldsymbol{\omega}_{t+1}^i) - \nabla F(\boldsymbol{\omega}_{t+1})\big] \\ &\qquad - \alpha\mathbb{E}_i\left[\nabla\psi(\boldsymbol{\omega}_t) - \nabla\psi(\boldsymbol{\omega}_{t+1}^i)\right] + \alpha\left(\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t\right) + \alpha\mathbb{E}_i\mathbf{e}_{t+1}^i.\end{aligned}$$

It follows that

$$\begin{aligned}\Delta_t &= \boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t + \alpha\nabla F(\boldsymbol{\omega}_t) \\ &= -\alpha\underbrace{\left(\nabla F(\boldsymbol{\omega}_{t+1}) - \nabla F(\boldsymbol{\omega}_t)\right)}_{=\mathbf{t}_1} - \alpha\underbrace{\mathbb{E}_i\big[\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i) + \nabla\psi(\boldsymbol{\omega}_{t+1}^i) - \nabla F(\boldsymbol{\omega}_{t+1})\big]}_{=\mathbf{t}_2} \\ &\quad - \alpha\underbrace{\mathbb{E}_i\left[\nabla\psi(\boldsymbol{\omega}_t) - \nabla\psi(\boldsymbol{\omega}_{t+1}^i)\right]}_{=\mathbf{t}_3} + \alpha\underbrace{\left(\nabla\psi(\boldsymbol{\omega}_t) - \mathbf{g}_t\right)}_{=\mathbf{t}_4} + \alpha\underbrace{\mathbb{E}_i\mathbf{e}_{t+1}^i}_{=\mathbf{t}_5}.\end{aligned}$$

Consider the quantity $\mathbb{E}_t\left[\langle\nabla F(\boldsymbol{\omega}_t), \Delta_t\rangle\right]$:

$$\mathbb{E}_t\left[\langle\nabla F(\boldsymbol{\omega}_t), \Delta_t\rangle\right] \leq \alpha\mathbb{E}_t\left[\|\nabla F(\boldsymbol{\omega}_t)\|_2 \cdot \left(\|\mathbf{t}_1\|_2 + \|\mathbf{t}_2\|_2 + \|\mathbf{t}_3\|_2 + \|\mathbf{t}_4\|_2 + \|\mathbf{t}_5\|_2\right)\right]$$

Observe that because the composite objective is L-smooth,

$$\mathbb{E}_t\left[\|\mathbf{t}_1\|_2\right] \leq L\mathbb{E}_t\left[\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2\right] \leq L\mathbb{E}_t\left[\mathbb{E}_i\left\|\boldsymbol{\omega}_{t+1}^i - \boldsymbol{\omega}_t\right\|_2\right].$$

Similarly, the estimate for $\mathbf{t}_3$ uses the L-smoothness of the regularizer:

$$\mathbb{E}_t\left[\|\mathbf{t}_3\|_2\right] \leq \mathbb{E}_t\left[\mathbb{E}_i\left\|\nabla\psi(\boldsymbol{\omega}_t) - \nabla\psi(\boldsymbol{\omega}_{t+1}^i)\right\|_2\right] \leq L\,\mathbb{E}_t\left[\mathbb{E}_i\left\|\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t+1}^i\right\|_2\right].$$

The $\mathbf{t}_2$ term can also be bounded in terms of the iterate distance:

$$\begin{aligned}\mathbb{E}_t\left[\|\mathbf{t}_2\|_2\right] &\leq \mathbb{E}_t\left[\mathbb{E}_i\left\|\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i)+\nabla\psi(\boldsymbol{\omega}_{t+1}^i)-\nabla F(\boldsymbol{\omega}_{t+1})\right\|_2\right]\\ &\leq \mathbb{E}_t\left[\mathbb{E}_i\left\|\nabla\ell^i(\boldsymbol{\omega}_{t+1}^i)+\nabla\psi(\boldsymbol{\omega}_{t+1}^i)-\nabla\ell^i(\boldsymbol{\omega}_{t+1})-\nabla\psi(\boldsymbol{\omega}_{t+1})\right\|_2\right],\end{aligned}$$

where the last equality holds because $\nabla\ell(\boldsymbol{\omega}_{t+1}) = \mathbb{E}_i\nabla\ell^i(\boldsymbol{\omega}_{t+1})$. We use the triangle inequality and the L-smoothness of ψ and the functions ℓ^i to continue our estimation:

$$\begin{aligned}&\leq 2L\mathbb{E}_t\left[\mathbb{E}_i\left\|\boldsymbol{\omega}_{t+1}^i-\boldsymbol{\omega}_{t+1}\right\|_2\right] \leq 2L\mathbb{E}_t\left\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}_t\right\|_2 + 2L\mathbb{E}_t\left[\mathbb{E}_i\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}_{t+1}^i\right\|_2\right]\\ &\leq 4L\mathbb{E}_t\left[\mathbb{E}_i\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}_{t+1}^i\right\|_2\right].\end{aligned}$$

Thus we find that

$$\begin{aligned}\mathbb{E}_t\Big[\|\nabla F(\boldsymbol{\omega}_t)\|_2\cdot(\|\mathbf{t}_1\|_2+\|\mathbf{t}_2\|_2+\|\mathbf{t}_3\|_2)\Big] &\leq 6L\|\nabla F(\boldsymbol{\omega}_t)\|_2\cdot\mathbb{E}_t\left[\mathbb{E}_i\left\|\boldsymbol{\omega}_t-\boldsymbol{\omega}_{t+1}^i\right\|_2\right]\\ &\leq \frac{6L(1+\gamma)}{\mu}\left[B\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\|\nabla F(\boldsymbol{\omega}_t)\|_2\cdot(G+\sigma)\right]\\ &\leq \frac{6L(1+\gamma)}{\mu}\left((B+1)\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\sigma^2+G^2\right).\end{aligned}$$

The last two inequalities are justified by equation 21 and the fact that $|ab| \leq \frac{1}{2}(a^2+b^2)$ for any real numbers a and b.

The noise term $\mathbf{t}_4$ is controlled by the variance of the stochastic gradient estimate

$$\mathbb{E}_t\left[\|\mathbf{t}_4\|_2\right] \leq \sqrt{\mathbb{E}_t\left\|\nabla\psi(\boldsymbol{\omega}_t)-\mathbf{g}_t\right\|^2} \leq \sigma.$$

To control the $\mathbf{t}_5$ term, recall that $\|\mathbf{e}_{t+1}^i\|_2 \leq \gamma\|\nabla\ell_t^i(\boldsymbol{\omega}_t)\|_2$. This implies that

$$\begin{aligned}\mathbb{E}_t\left[\|\mathbf{t}_5\|_2\right] &= \mathbb{E}_t\left\|\mathbb{E}_i\mathbf{e}_{t+1}^i\right\|_2 \leq \mathbb{E}_t\left[\mathbb{E}_i\left\|\mathbf{e}_{t+1}^i\right\|_2\right]\\ &\leq \gamma\mathbb{E}_t\left[\mathbb{E}_i\left\|\nabla\ell^i(\boldsymbol{\omega}_t)+\mathbf{g}_t\right\|_2\right]\\ &\leq \gamma\mathbb{E}_t\left[\mathbb{E}_i\left\|\nabla\ell^i(\boldsymbol{\omega}_t)+\nabla\psi(\boldsymbol{\omega}_t)\right\|_2+\|\nabla\psi(\boldsymbol{\omega}_t)-\mathbf{g}_t\|_2\right]\\ &\leq \gamma\left(B\left\|\nabla F(\boldsymbol{\omega}_t)\right\|_2+G+\sigma\right).\end{aligned}$$

The last inequality holds because of Jensen's inequality, the bounded dissimilarity condition, and the bound on the variance of the stochastic gradient estimate.

From these last two estimates, we find that

$$\begin{aligned}\mathbb{E}_t\Big[\|\nabla F(\boldsymbol{\omega}_t)\|_2\cdot(\|\mathbf{t}_4\|_2+\|\mathbf{t}_5\|_2)\Big] &\leq \|\nabla F(\boldsymbol{\omega}_t)\|_2\cdot\sigma+\gamma\left(B\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\|\nabla F(\boldsymbol{\omega}_t)\|_2(G+\sigma)\right)\\ &\leq \frac{1}{2}\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\frac{1}{2}\sigma^2+\gamma\left((B+1)\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+G^2+\sigma^2\right),\end{aligned}$$

and therefore we conclude that

$$\begin{aligned}\mathbb{E}_t\left[\langle\nabla F(\boldsymbol{\omega}_t),\Delta_t\rangle\right] &\leq \alpha\frac{6L(1+\gamma)}{\mu}\left((B+1)\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+G^2+\sigma^2\right)\\ &\quad+\frac{\alpha}{2}\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\frac{\alpha}{2}\sigma^2+\alpha\gamma\left((B+1)\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+G^2+\sigma^2\right)\\ &\leq \alpha\cdot c_{\gamma,L,B,\mu}\cdot\left(\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+G^2+\sigma^2\right),\end{aligned} \tag{23}$$

where, for convenience, we defined

$$c_{\gamma,L,B,\mu} = \left(\frac{1}{2}+\gamma(B+1)+\frac{6L(1+\gamma)(B+1)}{\mu}\right).$$

Finally, consider the squared iterate distance in equation 22. In particular, equation 20 implies that

$$\frac{L}{2}\mathbb{E}_t\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}_t\|_2^2 \leq L\left(\frac{1+\gamma}{\mu}\right)^2\left[B^2\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+G^2+\sigma^2\right].$$

Using this estimate and equation 23 in equation 22 gives that

$$\begin{aligned}\mathbb{E}_t\left[F(\boldsymbol{\omega}_{t+1})\right] &\leq F(\boldsymbol{\omega}_t)-\alpha\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\mathbb{E}_t\Big[\langle\nabla F(\boldsymbol{\omega}_t),\Delta_t\rangle\Big]+\frac{L}{2}\mathbb{E}_t\left[\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}_t\|_2^2\right] \\ &\leq F(\boldsymbol{\omega}_t)-\alpha\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+c_{\gamma,L,B,\mu,\alpha}\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+c_{\gamma,L,B,\mu,\alpha}(G^2+\sigma^2),\end{aligned} \tag{24}$$

where, for convenience, we define

$$\begin{aligned}c_{\gamma,L,B,\mu,\alpha} &= \alpha\cdot c_{\gamma,L,B,\mu}+LB^2\cdot\left(\frac{1+\gamma}{\mu}\right)^2 \\ &= \alpha\left(\frac{1}{2}+\gamma(B+1)+\frac{6L(1+\gamma)(B+1)}{\mu}\right)+LB^2\cdot\left(\frac{1+\gamma}{\mu}\right)^2\end{aligned}$$

Because $\alpha<\frac{1}{2L_-}$,

$$\frac{1}{\mu}=\frac{\alpha}{1-\alpha L_{-1}}<2\alpha,$$

and consequently

$$c_{\gamma,L,B,\mu,\alpha}\leq\alpha\left(\frac{1}{2}+\gamma(B+1)+12\alpha L(1+\gamma)(B+1)\right)+4\alpha^2LB^2(1+\gamma)^2.$$

We also choose $\gamma<\frac{1}{8(B+1)}$ and $\gamma<\frac{1}{20}$, which further implies that

$$c_{\gamma,L,B,\mu,\alpha}\leq\alpha\left(\frac{5}{8}+13\alpha L(B+1)\right)+5\alpha^2LB^2,$$

and because $\alpha<\min\left\{\frac{1}{120L(B+1)},\frac{1}{5LB^2},\frac{1}{20}\right\}$, in fact

$$c_{\gamma,L,B,\mu,\alpha}\leq\frac{3}{4}\alpha+\alpha^2\leq\frac{4}{5}\alpha.$$

Thus, we conclude that the expected decrease in the composite objective at each iteration satisfies

$$\mathbb{E}_t\left[F(\boldsymbol{\omega}_{t+1})\right]\leq F(\boldsymbol{\omega}_t)-\frac{\alpha}{5}\|\nabla F(\boldsymbol{\omega}_t)\|_2^2+\frac{4\alpha}{5}(G^2+\sigma^2). \tag{25}$$

Convergence rate In the remainder of this proof, $\mathbb{E}[\cdot]$ denotes the expectation with respect to all sources of randomness and we take $G=0$. Using the tower rule of expectations and summing over the first T iterations, we find that

$$\begin{aligned}\frac{\alpha}{5}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\boldsymbol{\omega}_t)\|_2^2\right]-\frac{4\alpha T}{5}\sigma^2 &\leq\sum_{t=0}^{T}\mathbb{E}\left[F(\boldsymbol{\omega}_t)-F(\boldsymbol{\omega}_{t+1})\right] \\ &= F(\boldsymbol{\omega}_0)-\mathbb{E}\left[F(\boldsymbol{\omega}_{T+1})\right]\leq F(\boldsymbol{\omega}_0)-F^\star.\end{aligned}$$

Rearranging terms yields the claimed result:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left[\|\nabla F(\boldsymbol{\omega}_t)\|_2^2\right]\leq\frac{F(\boldsymbol{\omega}_0)-F^\star}{\alpha T}+4\sigma^2.$$

□

A.3 Proof of Theoram 2

Proof. We know from the definition of KHSIC that

$$\begin{aligned} r(\boldsymbol{\omega}) &= \frac{1}{(n-1)^2}\mathrm{Tr}(\mathbf{K}_s\mathbf{H}\mathbf{K}_f\mathbf{H}) \\ &= \frac{1}{(n-1)^2}\mathbb{E}\left[\mathrm{Tr}(\mathbf{Z}_s\mathbf{Z}_s^\top\mathbf{H}\mathbf{Z}_f\mathbf{Z}_f^\top\mathbf{H})\right] \\ &= \frac{1}{(n-1)^2}\mathbb{E}\left[\mathrm{Tr}(\mathbf{Z}_s^\top\mathbf{H}\mathbf{Z}_f\cdot\mathbf{Z}_f^\top\mathbf{H}\mathbf{Z}_s)\right] \\ &= \frac{1}{(n-1)^2}\mathbb{E}\left[\|\mathbf{Z}_s^\top\mathbf{H}\mathbf{Z}_f\|_F^2\right]. \end{aligned}$$

The first equality is the definition of the KHISC, and the second holds because the outer products of the random feature maps are the kernel matrices, in expectation. The third holds due to the cyclicity of the trace operator, and the final holds by the definition of the Frobenius norm. The rest of the proof follows Lemma 1 □

A.4 Proof of Lemma 1:

Consider the following:

Proof. The reduced size fairness interaction matrix can be computed efficiently, by noting that

$$\mathbf{Z}_s^\top\mathbf{H}\mathbf{Z}_f = \mathbf{Z}_s^\top\mathbf{H}\mathbf{H}\mathbf{Z}_f$$

Thus,

$$\begin{aligned} \mathbf{Z}_s^\top\mathbf{H} &= \mathbf{Z}_s^\top\left(\mathbf{I}-\frac{1}{n}\mathbf{1}\mathbf{1}^\top\right) = \mathbf{Z}_s^\top - \boldsymbol{\mu}_s\mathbf{1}^\top, \text{ and} \\ \mathbf{H}\mathbf{Z}_f &= \left(\mathbf{I}-\frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\mathbf{Z}_f = \mathbf{Z}_f - \mathbf{1}\boldsymbol{\mu}_f^\top. \end{aligned}$$

Putting these identities together and using the local paritition of the random feature matrices and their means, we have that

$$\begin{aligned} \mathbf{Z}_s^\top\mathbf{H}\mathbf{Z}_f &= \mathbf{Z}_s^\top\mathbf{Z}_f - \boldsymbol{\mu}_s\mathbf{1}^\top\mathbf{Z}_f - \mathbf{Z}_s^\top\mathbf{1}\boldsymbol{\mu}_f^\top + n\boldsymbol{\mu}_s\boldsymbol{\mu}_f^\top \\ &= \mathbf{Z}_s^\top\mathbf{Z}_f - n\boldsymbol{\mu}_s\boldsymbol{\mu}_f^\top \\ &= \sum_{i=1}^m \mathbf{Z}_{s,i}^\top\mathbf{Z}_{f,i} - n\left(\frac{1}{n}\sum_{i=1}^m n_i\boldsymbol{\mu}_{s,i}\right)\left(\frac{1}{n}\sum_{i=1}^m n_i\boldsymbol{\mu}_{f,i}\right)^\top, \end{aligned} \tag{26}$$

□

A.5 Proof of Corollary 1

Proof. We use the linearity of expectation to obtain an unbiased approximation of the gradient of the fairness regularizer:

$$\begin{aligned} \nabla_{\boldsymbol{\omega}}\psi(\boldsymbol{\omega}) &= \mathbb{E}\,\nabla_{\boldsymbol{\omega}}\tilde{\psi}(\boldsymbol{\omega}) \\ &= \frac{1}{(n-1)^2}\mathbb{E}\,\nabla_{\boldsymbol{\omega}}\|\mathbf{G}(\boldsymbol{\omega})\|_F^2. \end{aligned}$$

□

A.6 Proof of Lemma 2

The stochastic gradient can then be computed in terms of the Jacobians of the local interaction terms $\mathbf{\Omega}_i$ by a simple application of chain rule:

$$\begin{aligned}\frac{1}{(n-1)^2}\nabla_{\boldsymbol{\omega}}\|\mathbf{G}(\boldsymbol{\omega})\|_F^2 &= \frac{2}{(n-1)^2}\mathbf{J}_{\mathbf{G}}(\boldsymbol{\omega})^T\mathbf{G}(\boldsymbol{\omega}) \\ &= \frac{2}{(n-1)^2}\sum_{i=1}^{m}\mathbf{J}_{\mathbf{G}_i}(\boldsymbol{\omega})^T\mathbf{G}(\boldsymbol{\omega}).\end{aligned}$$

B Details of the experimental Evaluation

B.1 Choice of Hyperparameters

Common Hyperparameters: To ensure a fair comparison, all evaluated algorithms utilize a consistent set of common hyperparameters. The batch size is uniformly set to 64. Each algorithm undergoes a total of 10 global training rounds, with each round comprising 5 local epochs on every client. This choice of local epochs ensured that the global model converges within 10 rounds or fewer for all datasets and distribution, allowing us to limit the number of global rounds to 10. The experiments are conducted with 4 clients, a configuration determined based on recent studies. Ezzeldin et al. (2023); Papadaki et al. (2022). The learning rate α was set to 0.01 and Adam Optimizer Kingma & Ba (2014) was used for optimization

Algorithm Specific Hyperparameters: Some hyperparameters are specific to the algorithm being used. For example, the **KFFL** and **KFFL-TD** algorithms rely on feature maps $\mathbf{D}$ to estimate the kernel regularizer. In our experiments, we use the Pyrfm librarypyr to generate random feature maps based on Orthogonal Random Features Yu et al. (2016). The dimensionality of the feature maps used for kernel approximation, denoted as $\mathbf{D}$, is set to 10. While higher dimensions also yielded good results, we selected the smallest feature map size that ensured **KFFL** performed effectively..

For other baselines such as Ezzeldin et al. (2023), a tradeoff parameter called "fairness budget" β is used to control the effect of reweighing. This tradeoff parameter determines the balance between model accuracy and a specific evaluation metric, by varying β from 0.1 to 5 based on the suggestions provided in the paper. However, it should be noted that for the best performance of the Ezzeldin et al. (2023) baseline, a local debiasing mechanism is required. Based on the results from the paper, we used the Roh et al. (2020) algorithm as a local demographic bias mitigation algorithm to compare with our method. Papadaki et al. (2022) use a "global adversary rate" to control how the expected loss over the worst-performing demographic is reduced. To consider an accuracy-fairness tradeoff, we varied this parameter from 0.001 to 0.1.

To enable fine-grained control over the tradeoff between fairness and other performance metrics, our methods incorporate a controllable fairness weight λ. This weight can be fine-tuned based on the desired tradeoff.More on this in Section B.5

B.2 Dataset

Datasets for Classification Task

- **ADULT**: Becker & Kohavi (1996) is a binary classification dataset that contains up to 14 attributes used in predicting whether an individual would earn an income $\geq$ 50K or $\leq$ 50K. The features used in the prediction include continuous attributes such as age, hours per week worked, etc, and discrete attributes including relationship, race, sex, and education. For the purpose of our experimental evaluation, we train a Logistic Regression model Mohri et al. (2019) on this dataset and we consider **sex** as the protected sensitive variable.

- **COMPAS**:Barenstein (2019) is used for predicting criminal recidivism for individuals.The number of samples considered is 6,172 samples and the number of predictive features used in determining

recidivism is 52 including race, age, and previous criminal offenses. For the purpose of our experimental evaluation on this dataset, we consider **race** as the protected sensitive variable evaluated on a Logistic Regression model.

- **BANK**:Data from a Portuguese bank utilized to forecast client subscriptions to term deposits Han et al. (2023).Here we consider **age** as the sensitive attribute and Loan Approval as the target attribute.There are 64 predictive features (including the sensitive) and 41188 target samples.
- **ACS**: From the American Community Survey, utilized for various prediction tasks including income and employment Han et al. (2023). There are 910 (including the sensitive) predictive features and 195665 samples for this dataset. We consider income as the target variable with **sex** as the protected attribute.
- **GERMAN**:Dataon credit applicants from a German bank used for predicting credit risk ratings Han et al. (2023) where the sensitive attribute we consider is **sex** and the target attribute is Credit risk rating. There are 60 (including the sensitive) predictive features and 1000 target samples.

Datasets for Regression Task

- `Law School`: Sourced from the Law School Admissions Council's National Longitudinal Bar Passage Study Wightman (1998), this dataset contains 20,649 examples. The task is to predict a student's GPA—normalized to the range [0, 1]—using squared loss minimization. Race serves as the protected attribute, categorized as white versus non-white.
- `Communities and Crime`: This dataset comprises socio-economic, law enforcement, and crime statistics from various U.S. communities Redmond & Baveja (2002), totaling 1,994 examples. The objective is to predict the number of violent crimes per 100,000 inhabitants, normalized to [0, 1], through squared loss minimization. The protected attribute is race, defined by whether the community's majority population is white.By including these datasets, we aim to thoroughly evaluate the fair regression estimator's performance across different contexts where fairness with respect to sensitive attributes like race is crucial.

B.3 Data Distribution on Clients

We explore both the IID (Independent and Identically Distributed) and Non-IID (Non-Independent and Non-Identically Distributed) settings in our evaluation:

- **IID**: In this setting, each client is provided with an equal number of samples and a consistent data distribution for local training Li et al. (2020)
- **Non-IID**: In this setting, each of the clients has different distribution of the protected attribute. Particularly, in our case, since the protected group $\mathcal{A}$ is binary with attributes being $\mathcal{A}_0$ and $\mathcal{A}_1$, half of the clients have 90 % of $\mathcal{A}_0$ and 10 % of $\mathcal{A}_1$ while the other half has 90 % of $\mathcal{A}_1$ and 10 % of $\mathcal{A}_0$ Li et al. (2020)

B.4 Models

We consider two distinct types of models for classification and a **linear model for regression**, which clients use for local training. In the fairness literature, Logistic Regression is commonly employed for fair classification tasks Ezzeldin et al. (2023), while linear models are the standard choice for fair regression Chzhen et al. (2020a). Beyond these, we extend our evaluation to include a more complex, non-convex model for classification.

- **Logistic Regression**: This involves a binary logistic regression model with a sigmoid activation function Han et al. (2023)

- **Neural Network**: We also examine the performance of our algorithm using a neural network configuration. This neural network consists of a single hidden layer with 100 neurons and employs ReLU activation, culminating in an output layer.

B.5 Fairness weights

- **ADULT:** 0.00,20.00, 71.58, 123.16, 174.74, 226.32, 277.89, 329.47, 381.05, 432.63, 484.21, 535.79, 587.37, 690.53, 742.11, 793.68, 845.26, 896.84, 948.42, 1000.00
- **COMPAS:** 0.00, 0.01, 0.10, 20.00, 71.58, 123.16
- **BANK:** 0.00, 0.01, 0.10, 1.00
- **GERMAN:** 0.00, 0.01, 0.10, 1.00
- **ACS:** 0.00, 0.01, 0.10, 1.00

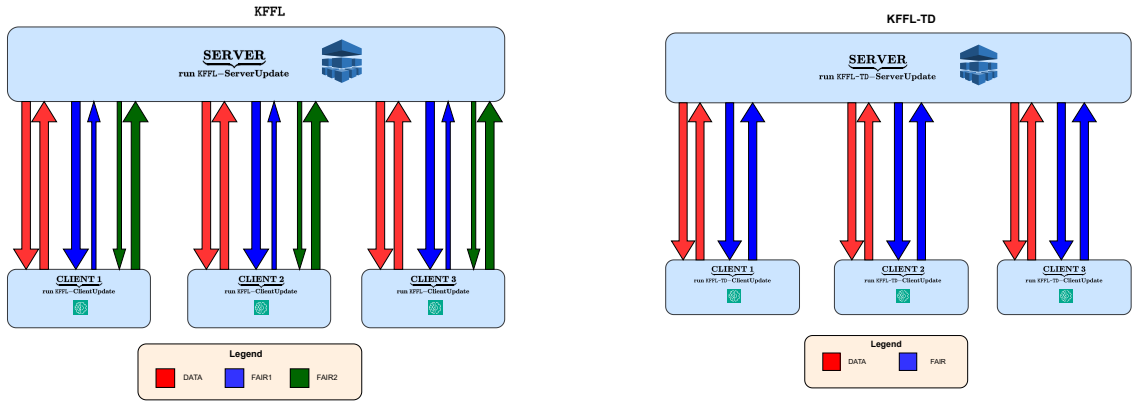

Figure 5: `KFFL` and `KFFL-TD` are illustrated in the figure. Different colors correspond to various segments of communication associated with either the **FAIR1**, **FAIR2** or **DATA** sub-rounds (see the `KFFL` Algorithm for detail on these flags). For `KFFL-TD` the relevant flags are **FAIR** and **DATA** (see see the `KFFL-TD` Algorithm for detail on these flags) The direction of the arrows indicate an uplink or downlink communication and the width of each arrowhead highlights the communication cost in each sub-round. Thicker lines indicate higher communication overhead, while thinner lines represent smaller overhead.

C KFFL-TD - Kernel Regualarized Fair Learning with time delay

The `KFFL-TD` variant optimizes communication efficiency by incorporating delayed information for the fair term. Assuming the training begins at round t with $t \geq 1$; the server transmits the current model Algorithm 5 ω_t and all global information from the preceding time step $\boldsymbol{\mu}_s(t-1), \boldsymbol{\mu}_f(t-1), \mathbf{G}(\boldsymbol{\omega}_{t-1})$ to the client for local gradient computation. This set of downlink information is denoted as $\Gamma(\boldsymbol{\omega}_t)$. In this case, the server also shares a common seed ζ to control the randomness in the generation of random feature maps generated. The client also receives a **FAIR** flag, indicating that no data-fitting operation is required. The clients leverage this information to calculate the fair gradient $\mathbf{g}^i(\boldsymbol{\omega}_{t-1})$ (if the global round is not zero) and relevant details $\Phi^i(\boldsymbol{\omega}_t)$ (see Algorithm 4), contributing to the computation of the global interaction $\mathbf{G}(\boldsymbol{\omega}_t)$, $\boldsymbol{\mu}_s(t)$, and $\boldsymbol{\mu}_f(t)$ which will be used in the subsequent round for the fair update (see Algorithm 5) . All of this information is $\Psi^i(\boldsymbol{\omega}_t)$ sent by each client.

With these outdated fair gradients, the server updates the global model see Algorithm 5 and compute the global fairness interaction terms $\mathbf{G}(\boldsymbol{\omega}_t)$,$\boldsymbol{\mu}_s(t)$ to be used by the clients in the next round. To steer the model towards the data-fit direction, the server sends **DATA** flag instructing the clients perform local optimization

Algorithm 4 `KFFL-TD` – Client Update

Input: (**ROUND**, ..)

if ROUND = **FAIR** **then**
  if $t \neq 0$ **then**
    Clients compute local interaction for gradients using Equation 18 at $\boldsymbol{\omega}_{t-1}$ to get $\boldsymbol{\Omega}_i(\boldsymbol{\omega}_{t-1})$
    The clients then compute the local gradient $\mathbf{g}^i(\boldsymbol{\omega}_{t-1})$ using Equation 19
  end if
  Clients compute $\mathbf{M}^i(\boldsymbol{\omega}_t)$ using Equation 12 with random seed ζ and RFMs (such as ORFMs Yu et al. (2016)) , $\boldsymbol{\mu}_{f,i}(t)$ using Equation 14 and $\boldsymbol{\mu}_{s,i}(t)$ using 13
  $\Phi^i(\boldsymbol{\omega}_t) = \{\mathbf{M}^i(\boldsymbol{\omega}_t), \boldsymbol{\mu}_{s,i}(t), \boldsymbol{\mu}_{f,i}(t)\}$
  if $t \neq 0$ **then**
    $\Psi^i(\boldsymbol{\omega}_t) = \{\mathbf{g}^i(\boldsymbol{\omega}_{t-1}), \Phi^i(\boldsymbol{\omega}_t)\}$
  else
    $\Psi^i(\boldsymbol{\omega}_t) = \Phi^i(\boldsymbol{\omega}_t)$
  end if
  Return: $\Psi^i(\boldsymbol{\omega}_t)$
else if ROUND = **DATA** **then**
  Clients do a local update on $\boldsymbol{\omega}_{t+1/2}$ following Equation 17 to get $\boldsymbol{\omega}^i_{t+1}$
  Return: $\boldsymbol{\omega}^i_{t+1}$
end if

Algorithm 5 `KFFL-TD` – Server Side

$\boldsymbol{\omega} = \boldsymbol{\omega}_0$ {This is the initial model}
$t \leftarrow 0$
while $\boldsymbol{\omega}$ not converge **do**
  for all $i = 1, \ldots, m$ in parallel **do**
    Generation of random seed ζ
    if $t \neq 0$ **then**
      $\Gamma(\boldsymbol{\omega}_t) = \{\boldsymbol{\omega}_t, \boldsymbol{\mu}_s(t-1), \boldsymbol{\mu}_f(t-1), \mathbf{G}(\boldsymbol{\omega}_{t-1}), \zeta, t\}$
    else
      $\Gamma(\boldsymbol{\omega}_t) = \{\boldsymbol{\omega}_t, \zeta, t\}$
    end if
    $\Psi(\boldsymbol{\omega}_t) =$ **Client Update**$(\Gamma(\boldsymbol{\omega}_t),$ **FAIR**$)$
  end for
  if $t \neq 0$ **then**
    $\boldsymbol{\omega}_{t+1/2} \leftarrow \boldsymbol{\omega}_t - \sum_{i=1}^m \mathbf{g}^i(\boldsymbol{\omega}_{t-1})$
  end if
  From $\Psi(\boldsymbol{\omega}_t)$ clients get $\Phi(\boldsymbol{\omega}_t) = \{\Phi^i(\boldsymbol{\omega}_t)\}_{i=1}^m$
  From $\Phi(\boldsymbol{\omega}_t)$ compute $\mathbf{G}(\boldsymbol{\omega}_t)$ using Equation 11; $\boldsymbol{\mu}_s(t)$ using Equation 13; and $\boldsymbol{\mu}_f(t)$ using Equation 14
  for all $i = 1, \ldots, m$ in parallel **do**
    if $t \neq 0$ **then**
      $\boldsymbol{\omega}^i_{t+1} =$ **Client Update**$(\boldsymbol{\omega}_{t+1/2},$ **DATA**$)$
    else
      $\boldsymbol{\omega}^i_{t+1} =$ **Client Update**$(\boldsymbol{\omega}_t,$ **DATA**$)$
    end if
  end for
  $\boldsymbol{\omega}_{t+1} \leftarrow$ `FedAvg`$(\boldsymbol{\omega}^i_{t+1})$
end while

(see Equation 17) and communicate the updated copy to the server ; the server conducts `Fed-Avg` after receiving these locally updated models, resulting in the generation of a the updated model. The process continues till the model $\boldsymbol{\omega}_t$ converges.

D Additional Empirical Evaluation of Classification

Table 6 and 8 show the results of KFFL and its baselines on Logistic Regression and Neural Network across different datasets.KFFL is most robust across datasets in the **Non-IID** setting and provides greater tradeoffs for accuracy and fairness.

	Method	Adult	COMPAS	BANK	ACS	GERMAN
Acc.	FedAvg	84.30 ± 1.41	61.13 ± 1.25	91.21 ± 1.10	81.14 ± 1.20	72.50 ± 1.15
	KFFL	83.14 ± 0.42	60.3 ± 1.30	90.23 ± 1.25	79.12 ± 1.35	72.50 ± 1.10
	KFFL-TD	83.97 ± 1.12	59.51 ± 1.45	90.78 ± 1.20	81.12 ± 1.05	72.50 ± 1.40
	FairFed/FairBatch	76.34 ± 0.1	55.47 ± 1.50	88.05 ± 1.30	58.77 ± 1.20	30.00 ± 1.25
	KHSIC-Local	63.40 ± 0.03	44.13 ± 1.55	88.77 ± 1.15	58.77 ± 1.05	70.00 ± 1.35
	MinMax	76.34 ± 0.1	56.68 ± 1.40	64.93 ± 1.20	58.77 ± 1.25	70.00 ± 1.10
SPD	FedAvg	0.18 ± 0.06	0.16 ± 0.02	0.23 ± 0.01	0.08 ± 0.04	0.19 ± 0.03
	KFFL	0.14 ± 0.02	0.06 ± 0.01	0.05 ± 0.03	0.04 ± 0.02	0.00 ± 0.04
	KFFL-TD	0.16 ± 0.03	0.05 ± 0.02	0.06 ± 0.01	0.00 ± 0.03	0.00 ± 0.02
	FairFed/FairBatch	0.004 ± 0.001	0.05 ± 0.03	0.00 ± 0.02	0.00 ± 0.01	0.00 ± 0.03
	KHSIC-Local	0.34 ± 0.00	0.08 ± 0.02	0.00 ± 0.03	0.00 ± 0.02	0.00 ± 0.01
	MinMax	0.004 ± 0.001	0.03 ± 0.01	0.21 ± 0.02	0.00 ± 0.03	0.00 ± 0.02
EOD	FedAvg	0.22 ± 0.03	0.23 ± 0.02	0.16 ± 0.01	0.04 ± 0.02	0.02 ± 0.03
	KFFL	0.12 ± 0.03	0.07 ± 0.01	0.04 ± 0.02	0.02 ± 0.03	0.00 ± 0.02
	KFFL-TD	0.04 ± 0.1	0.08 ± 0.02	0.05 ± 0.01	0.03 ± 0.02	0.00 ± 0.03
	FairFed/FairBatch	0.013 ± 0.001	0.09 ± 0.02	0.16 ± 0.03	0.00 ± 0.01	0.00 ± 0.02
	KHSIC-Local	0.12 ± 0.00	0.05 ± 0.03	0.00 ± 0.02	0.00 ± 0.01	0.00 ± 0.03
	MinMax	0.013 ± 0.001	0.03 ± 0.02	0.02 ± 0.01	0.00 ± 0.03	0.00 ± 0.02

Table 7: Comparison of Methods in the Non-IID environment with Logistic Regression for 3 seperate runs.Similar to the results in Table 6 under **Non-IID** conditions we have improved tradeoff points using KFFL

	Method	Adult	COMPAS	BANK	ACS	GERMAN
Acc.	FedAvg	84.35 ± 1.45	63.20 ± 1.30	94.25 ± 1.15	84.10 ± 1.25	72.55 ± 1.20
	KFFL	83.10 ± 0.45	62.35 ± 1.35	92.20 ± 1.30	81.15 ± 1.40	75.0 ± 0.01
	KFFL-TD	83.95 ± 1.10	59.55 ± 1.50	90.75 ± 1.25	81.15 ± 1.10	75.0 ± 0.01
	FairFed/FairBatch	76.30 ± 0.15	55.50 ± 1.55	88.10 ± 1.35	58.75 ± 1.25	37.50 ± 1.30
	KHSIC-Local	63.45 ± 0.05	44.15 ± 1.60	88.80 ± 1.20	58.80 ± 1.10	70.05 ± 1.40
	MinMax	76.30 ± 0.15	56.70 ± 1.45	64.95 ± 1.25	58.75 ± 1.30	70.05 ± 1.15
SPD	FedAvg	0.17 ± 0.05	0.15 ± 0.03	0.22 ± 0.02	0.09 ± 0.05	0.20 ± 0.04
	KFFL	0.13 ± 0.03	0.07 ± 0.02	0.06 ± 0.04	0.03 ± 0.03	0.01 ± 0.05
	KFFL-TD	0.17 ± 0.04	0.06 ± 0.03	0.07 ± 0.02	0.01 ± 0.04	0.01 ± 0.03
	FairFed/FairBatch	0.005 ± 0.002	0.06 ± 0.04	0.01 ± 0.03	0.01 ± 0.02	0.01 ± 0.04
	KHSIC-Local	0.35 ± 0.01	0.09 ± 0.03	0.01 ± 0.04	0.01 ± 0.03	0.01 ± 0.02
	MinMax	0.005 ± 0.002	0.04 ± 0.02	0.20 ± 0.03	0.01 ± 0.04	0.01 ± 0.03
EOD	FedAvg	0.23 ± 0.04	0.24 ± 0.03	0.17 ± 0.02	0.05 ± 0.03	0.03 ± 0.04
	KFFL	0.13 ± 0.04	0.08 ± 0.02	0.05 ± 0.03	0.03 ± 0.04	0.01 ± 0.03
	KFFL-TD	0.05 ± 0.11	0.09 ± 0.03	0.06 ± 0.02	0.04 ± 0.03	0.01 ± 0.04
	FairFed/FairBatch	0.014 ± 0.002	0.10 ± 0.03	0.17 ± 0.04	0.01 ± 0.02	0.01 ± 0.03
	KHSIC-Local	0.13 ± 0.01	0.06 ± 0.04	0.01 ± 0.03	0.01 ± 0.02	0.01 ± 0.04
	MinMax	0.014 ± 0.002	0.04 ± 0.03	0.03 ± 0.02	0.01 ± 0.04	0.01 ± 0.03

Table 8: Comparison of Methods in the Non-IID environment with Neural Network for 3 seperate runs. Similar to the results in Table 6 under Non-IID conditions we have improved tradeoff using KFFL

E Additional Experiments for Regression Task

Tables 15, 13, 11, 12, 14, and 10 show how KFFL performs under various regression tasks.

F Fair Regression

The KHSIC method facilitates the training of predictive models for regression tasks while ensuring fairness across various sensitive groups. Fair regression has been extensively explored in centralized settings, with significant contributions from studies such as Chzhen et al. (2020a); Agarwal et al. (2019a); Chzhen et al. (2020b).

Fairness weight λ	**RMSE $\downarrow$**	**KS Difference $\downarrow$**
0.0	0.322400 ± 0.015698	0.462100 ± 0.152637
1.0	0.326067 ± 0.018071	0.386633 ± 0.163709
50.0	0.311633 ± 0.024243	0.410500 ± 0.106058
100.0	0.322400 ± 0.016542	0.277200 ± 0.114275

Table 9: RMSE and KS Difference with standard deviations for 5 runs KFFL-TD with **IID** for the Communities and Crime Dataset

Fairness weight λ	**RMSE $\downarrow$**	**KS Difference $\downarrow$**
0.0	0.317528 ± 0.024826	0.671308 ± 0.202138
1.0	0.317952 ± 0.025052	0.620508 ± 0.251276
5.0	0.316936 ± 0.025075	0.662032 ± 0.159920
50.0	0.317576 ± 0.024314	0.444808 ± 0.163534
100.0	0.317592 ± 0.025111	0.395280 ± 0.141982

Table 10: RMSE and KS Difference with standard deviations for 5 runs `KFFL` with **Non-IID** for the Communities and Crime Dataset

In our framework, we employ the Root Mean Squared Error (RMSE) as the primary evaluation metric to assess the accuracy of the regression model. The objective function is defined as:

$$\ell(\mathbf{y}, f(\mathbf{x}; \boldsymbol{\omega})) = \frac{1}{n} \sum_{i=1}^{n} \left(f(\mathbf{x}_i; \boldsymbol{\omega}) - \mathbf{y}_i\right)^2 + \lambda \psi(\boldsymbol{\omega})$$

where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the input features, $\mathbf{y}_i \in \mathbb{R}$ is the target variable, and $f(\mathbf{x}_i; \boldsymbol{\omega})$ represents the model's prediction parameterized by $\boldsymbol{\omega}$. Consistent with the rest of this paper, we exclude the sensitive attribute $\mathbf{s}_i$ from the training process of the fair regressor.

Consider a regression task on a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{s}_i, \mathbf{y}_i)\}_{i=1}^n$, where $\mathbf{x}_i$ is the input feature vector, $\mathbf{s}_i \in \mathcal{S}$ is the sensitive attribute (e.g., gender, race), and $\mathbf{y}_i$ is the target variable (e.g., GPA, income). For each sensitive group $s \in \mathcal{S}$, we define the corresponding subset of data as:

$$\mathcal{D}^s = \{(\mathbf{x}, \mathbf{s}, \mathbf{y}) \in \mathcal{T} : \mathbf{s} = s\}$$

To evaluate fairness in regression tasks, we utilize the Kolmogorov-Smirnov (KS) distance Chzhen et al. (2020a), which measures the distributional differences between the model's predictions for different sensitive groups $s \in \mathcal{S}$. The KS distance is a widely adopted fairness metric in regression, enabling the assessment of disparities between groups. For instance, in a normalized GPA prediction task where the sensitive attribute $\mathcal{S}$ represents gender (e.g., male and female), the KS distance quantifies the difference in GPA predictions between these groups.

The KS distance between predictions for any two groups s and s' is defined as:

$$\mathrm{KS}(f(\mathbf{x}, \mathbf{s})) = \max_{s, s' \in \mathcal{S}} \sup_{t \in \mathbb{R}} \left| F^s(t) - F^{s'}(t) \right|$$

where $F^s(t)$ denotes the empirical cumulative distribution function (CDF) of the model's predictions for group s, calculated as:

$$F^s(t) = \frac{1}{|\mathcal{D}^s|} \sum_{(\mathbf{x}, \mathbf{s}, \mathbf{y}) \in \mathcal{D}^s} \mathbb{1}\{f(\mathbf{x}, \mathbf{s}) \leq t\}$$

Fairness weight λ	**RMSE** $\downarrow$	**KS Difference** $\downarrow$
0.00	0.800667 ± 0.002491	0.450867 ± 0.065047
0.01	0.800367 ± 0.002579	0.398433 ± 0.224212
0.10	0.800300 ± 0.002524	0.412733 ± 0.303714
1.00	0.799533 ± 0.001940	0.148967 ± 0.009340

Table 11: RMSE and KS Difference with standard deviations for 5 runs KFFL-TD with **IID** for the Law School Dataset

Fairness weight λ	**RMSE** $\downarrow$	**KS Difference** $\downarrow$
0.00	0.799293 ± 0.002132	0.514920 ± 0.124809
0.01	0.799253 ± 0.002306	0.452160 ± 0.163044
0.10	0.799187 ± 0.002101	0.467133 ± 0.143954
1.00	0.798800 ± 0.002080	0.140173 ± 0.032955

Table 12: RMSE and KS Difference with standard deviations for 5 runs KFFL with **IID** for the Law School Dataset

This formulation ensures that the regression model maintains fairness by minimizing the KS distance across all sensitive groups, thereby promoting equitable outcomes.

Fairness weight λ	**RMSE $\downarrow$**	**KS Difference $\downarrow$**
0.00	0.485100 ± 0.005092	0.410600 ± 0.031681
0.01	0.485100 ± 0.005260	0.410833 ± 0.026668
0.10	0.484967 ± 0.005424	0.408967 ± 0.013403
1.00	0.484767 ± 0.004964	0.381000 ± 0.009752
5.00	0.483967 ± 0.004944	0.282967 ± 0.010901

Table 13: RMSE and KS Difference with standard deviations for 5 runs KFFL-TD with **IID** for the Adult Dataset

Fairness weight λ	**RMSE $\downarrow$**	**KS Difference $\downarrow$**
0.00	0.491523 ± 0.006695	0.345782 ± 0.082513
0.01	0.493383 ± 0.008737	0.350317 ± 0.111856
0.10	0.493500 ± 0.008616	0.361175 ± 0.080166
1.00	0.493492 ± 0.008715	0.371725 ± 0.040675
5.00	0.493467 ± 0.009102	0.348567 ± 0.034269
50.00	0.494740 ± 0.009975	0.130180 ± 0.050417
100.00	0.498000 ± 0.011371	0.118580 ± 0.100111

Table 14: RMSE and KS Difference with standard deviations for 5 runs KFFL with **Non-IID** for the Adult Dataset

Fairness weight λ	**RMSE $\downarrow$**	**KS Difference $\downarrow$**
0.00	0.49088 ± 0.009294	0.38293 ± 0.016433
0.01	0.49101 ± 0.009303	0.39744 ± 0.027059
0.10	0.49053 ± 0.009130	0.37377 ± 0.030239
1.00	0.49071 ± 0.009039	0.38429 ± 0.053008
5.00	0.49068 ± 0.009474	0.26067 ± 0.027564
50.00	0.52160 ± 0.013804	0.06012 ± 0.018358
100.00	0.83521 ± 0.015799	0.34421 ± 0.018229

Table 15: RMSE and KS Difference with standard deviations for 5 runs KFFL with **IID** for the Adult Dataset

G Additional: Communication-Efficient Kernel Regularized Fair Learning

In this section, we address the computational and communication challenges of incorporating kernel-based fairness regularizers into federated learning. Specifically, we leverage Random Feature Maps (RFMs) to approximate kernel functions efficiently, reducing both computational complexity and communication overhead.

G.1 Random Feature Maps for Kernel Approximation

Kernel methods are powerful tools in machine learning but often suffer from high computational complexity, especially when dealing with large datasets. Computing kernel matrices requires $\mathcal{O}(n^2)$ memory and $\mathcal{O}(n^3)$ computational time, which is impractical for large n.

To overcome this, Rahimi & Recht (2007) introduced Random Feature Maps (RFMs) to approximate shift-invariant kernel functions. A shift-invariant kernel $\kappa(\mathbf{x}, \mathbf{y}) = \kappa(\mathbf{x} - \mathbf{y})$ can be represented using the Fourier transform via Bochner's theorem. Specifically, the kernel can be expressed as:

$$\kappa(\mathbf{x}, \mathbf{y}) = \int_{\mathbb{R}^d} p(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^\top(\mathbf{x}-\mathbf{y})} d\boldsymbol{\omega}, \tag{27}$$

where $p(\boldsymbol{\omega})$ is the spectral density function of the kernel κ.

Constructing Random Feature Maps To approximate $\kappa(\mathbf{x}, \mathbf{y})$, we draw D random samples $\{\boldsymbol{\omega}_k\}_{k=1}^D$ from $p(\boldsymbol{\omega})$ and define the random feature map $\phi : \mathbb{R}^d \to \mathbb{R}^D$ as:

$$\phi(\mathbf{x}) = \sqrt{\frac{2}{D}} \left[\cos(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1), \ldots, \cos(\boldsymbol{\omega}_D^\top \mathbf{x} + b_D)\right], \tag{28}$$

where $\{b_k\}_{k=1}^D$ are drawn uniformly from $[0, 2\pi]$.

With this mapping, the kernel function can be approximated as:

$$\kappa(\mathbf{x}, \mathbf{y}) \approx \phi(\mathbf{x})^\top \phi(\mathbf{y}). \tag{29}$$

Frequency of Drawing Random Features In our implementation, the random features are drawn **once at the beginning** of global training round and are fixed throughout the optimization process. This ensures consistency across iterations and clients, and avoids the overhead of regenerating random features at each iteration.

G.2 Computing $\mathbf{Z}_s$ and $\mathbf{Z}_f$

In the context of our fairness regularizer, we need to compute feature maps for both the sensitive attributes S and the model outputs $f_{\boldsymbol{\omega}}(\mathbf{X})$. Specifically:

- **Sensitive Attributes Feature Map** ($\mathbf{Z}_s$): For each data point i, we compute $\phi(s_i)$, where s_i is the sensitive attribute of the i-th sample. The matrix $\mathbf{Z}_s \in \mathbb{R}^{n \times D}$ has rows $\phi(s_i)^\top$.
- **Model Outputs Feature Map** ($\mathbf{Z}_f$): For each data point i, we compute $\phi(f_{\boldsymbol{\omega}}(\mathbf{x}_i))$, where $f_{\boldsymbol{\omega}}(\mathbf{x}_i)$ is the model output (e.g., logits) for the i-th sample. The matrix $\mathbf{Z}_f \in \mathbb{R}^{n \times D}$ has rows $\phi(f_{\boldsymbol{\omega}}(\mathbf{x}_i))^\top$.

Example: Gaussian Kernel Approximation As an example, consider the Gaussian (RBF) kernel:

$$\kappa(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_2^2}{2\sigma^2}\right). \tag{30}$$

The spectral density of the Gaussian kernel is $p(\boldsymbol{\omega}) = \mathcal{N}(\boldsymbol{\omega}; \mathbf{0}, \sigma^{-2}\mathbf{I})$. Therefore, to approximate the Gaussian kernel, we draw $\boldsymbol{\omega}_k \sim \mathcal{N}(\mathbf{0}, \sigma^{-2}\mathbf{I})$ and compute the feature maps as:

$$\phi(\mathbf{x}) = \sqrt{\frac{2}{D}} \left[\cos(\boldsymbol{\omega}_1^\top \mathbf{x} + b_1), \ldots, \cos(\boldsymbol{\omega}_D^\top \mathbf{x} + b_D)\right]. \tag{31}$$

Orthogonal Random Features To improve the quality of the approximation and reduce variance, we employ Orthogonal Random Features (ORF) as proposed by Yu et al. (2016). Instead of sampling $\boldsymbol{\omega}_k$ independently, we construct them to be orthogonal, which can lead to better kernel approximations with fewer features.

G.3 Approximation of the Fairness Regularizer

Using the random feature maps, we approximate the kernel matrices:

$$\mathbf{K}_s \approx \mathbf{Z}_s \mathbf{Z}_s^\top, \quad \mathbf{K}_f \approx \mathbf{Z}_f \mathbf{Z}_f^\top. \tag{32}$$

The variance of the approximation decreases as the number of random features D increases.

Theorem 2. *Let $\mathbf{Z}_f$ and $\mathbf{Z}_s$ be $n \times D$ matrices constructed using RFMs as described above. Then, the fairness regularizer can be approximated as:*

$$\psi(\boldsymbol{\omega}) = \mathbb{E}\left[\|\mathbf{G}(\boldsymbol{\omega})\|_F^2\right], \tag{33}$$

where $\mathbf{G}(\boldsymbol{\omega}) = \mathbf{Z}_s^\top \mathbf{Z}_f(\boldsymbol{\omega}) - n\boldsymbol{\mu}_s \boldsymbol{\mu}_f^\top(\boldsymbol{\omega}) \in \mathbb{R}^{D\times D}$. Here, $\boldsymbol{\mu}_s = \frac{1}{n}\sum_{i=1}^n \phi(s_i)$ and $\boldsymbol{\mu}_f = \frac{1}{n}\sum_{i=1}^n \phi(f_{\boldsymbol{\omega}}(\mathbf{x}_i))$ are the mean feature vectors.

Corollary 1. *The gradient of the fairness regularizer can be approximated as:*

$$\mathbf{g}(\boldsymbol{\omega}) = \nabla_{\boldsymbol{\omega}} \|\mathbf{G}(\boldsymbol{\omega})\|_F^2, \tag{34}$$

which is an unbiased stochastic estimate of $\nabla_{\boldsymbol{\omega}} \psi(\boldsymbol{\omega})$.

G.4 Distributed Computation in Federated Settings

In a federated learning setup with m clients, the data is partitioned across clients. The random feature matrices can be partitioned accordingly:

$$\mathbf{Z}_s = \begin{pmatrix} \mathbf{Z}_{s,1} \\ \vdots \\ \mathbf{Z}_{s,m} \end{pmatrix}, \quad \mathbf{Z}_f = \begin{pmatrix} \mathbf{Z}_{f,1} \\ \vdots \\ \mathbf{Z}_{f,m} \end{pmatrix}, \tag{35}$$

where $\mathbf{Z}_{s,i} \in \mathbb{R}^{n_i \times D}$ and $\mathbf{Z}_{f,i} \in \mathbb{R}^{n_i \times D}$ are the local random feature maps for client i, and n_i is the number of samples at client i.

Lemma 1. *The global feature interaction matrix $\mathbf{G}(\boldsymbol{\omega})$ can be computed as:*

$$\mathbf{G}(\boldsymbol{\omega}) = \sum_{i=1}^m \left(\mathbf{Z}_{s,i}^\top \mathbf{Z}_{f,i}(\boldsymbol{\omega}) - n_i \boldsymbol{\mu}_{s,i} \boldsymbol{\mu}_{f,i}^\top(\boldsymbol{\omega})\right), \tag{36}$$

where $\boldsymbol{\mu}_{s,i} = \frac{1}{n_i}\mathbf{1}^\top \mathbf{Z}_{s,i}$ and $\boldsymbol{\mu}_{f,i} = \frac{1}{n_i}\mathbf{1}^\top \mathbf{Z}_{f,i}$ are the local mean feature vectors at client i.

Lemma 2. *The gradient* $\mathbf{g}(\boldsymbol{\omega})$ *can be partitioned into local computations as:*

$$\mathbf{g}(\boldsymbol{\omega}) = \sum_{i=1}^{m} \mathbf{J}_{\boldsymbol{\Omega}_i}(\boldsymbol{\omega})^\top \mathbf{G}(\boldsymbol{\omega}), \tag{37}$$

where $\boldsymbol{\Omega}_i(\boldsymbol{\omega}) = \mathbf{Z}_s^\top \mathbf{Z}_{f,i}(\boldsymbol{\omega}) - n_i \boldsymbol{\mu}_s \boldsymbol{\mu}_{f,i}^\top(\boldsymbol{\omega})$ *and* $\mathbf{J}_{\boldsymbol{\Omega}_i}(\boldsymbol{\omega})$ *is the Jacobian of* $\boldsymbol{\Omega}_i$ *with respect to* $\boldsymbol{\omega}$*.*

Communication Efficiency By utilizing RFMs and the above partitioning, clients only need to transmit $D \times D$ matrices and D-dimensional vectors (the local interaction matrices and mean feature vectors) instead of $n \times n$ kernel matrices, significantly reducing communication costs.

For example, if we choose $D = 1024$ and $n = 32,000$ (as in the ADULT dataset), the communication cost is reduced by a factor of $(32,000/1024)^2 \approx 1000$.

Synchronizing Random Features To ensure consistency across clients, we use a shared random seed ζ to generate the random features $\{\boldsymbol{\omega}_k\}$ and $\{b_k\}$ at the beginning of training. All clients use the same random feature map ϕ.