The Cost of Local and Global Fairness in Federated Learning

Yuying Duan **Gelei Xu** **Yiyu Shi** **Michael Lemmon**

University of Notre Dame

Abstract

With the emerging application of Federated Learning (FL) in finance, hiring and healthcare, FL models are regulated to be fair, preventing disparities with respect to legally protected attributes such as race or gender. Two concepts of fairness are important in FL: global and local fairness. Global fairness addresses the disparity across the entire population and local fairness is concerned with the disparity within each client. Prior fair FL frameworks have improved either global or local fairness without considering both. Furthermore, while the majority of studies on fair FL focuses on binary settings, many real-world applications are multi-class problems. This paper proposes a framework that investigates the minimum accuracy lost for enforcing a specified level of global and local fairness in multi-class FL settings. Our framework leads to a simple post-processing algorithm that derives fair outcome predictors from the Bayesian optimal score functions. Experimental results show that our algorithm outperforms the current state of the art (SOTA) with regard to the accuracy-fairness tradoffs, computational and communication costs. Codes are available at: https://github.com/papersubmission678/The-cost-of-local-and-global-fairness-in-FL.

1 Introduction

Federated Learning (FL) [McMahan et al., 2017] is a distributed machine learning framework that uses data collected from a group of community *clients* to learn a global model that can be used by all clients in the group. The communities served by these clients are formed from sets of remote users (e.g. mobile phones) or organizations (e.g. medical clinics and hospitals) that all share some defining attribute such as a similar geographical location. FL algorithms such as `FedAvg` [McMahan et al., 2017] train the global model in a distributed manner by first having each client use its local data to train a local model. This local model is then sent to the cloud server who averages these models and sends the averaged model back to the community clients who then retrain that model with their local data. This interaction continues for several update cycles until it converges on a global model that is agreeable to all clients. FL is now the dominant framework for distributed machine learning [Kairouz et al., 2021], particularly in smart city [Pandya et al., 2023, Qolomany et al., 2020] and smart healthcare applications [Antunes et al., 2022, Brisimi et al., 2018].

Group fairness [Dwork et al., 2012, Hardt et al., 2016] is a critical concern of FL applications such as healthcare. Group fairness requires that a model's decisions do not favor any particular group with legally protected (a.k.a sensitive) attributes such as race, gender, or age. There are two concepts of global fairness in FL: *global fairness* and *local fairness*. Global fairness [Ezzeldin et al., 2023] addresses disparities over the entire data distribution. Local fairness [Cui et al., 2021] addresses disparity within each local client's data distribution. Ensuring both fairness concepts is necessary in FL applications such as hospital networks. The network is legally required [Congress, 2022] to be fair to all individuals it serves, while also ensuring fairness in each client hospital.

Most existing fair FL approaches focus on enforcing either global fairness [Abay et al., 2020, Du et al., 2021, Ezzeldin et al., 2023] or local fairness [Cui et al., 2021] in FL. It has been empirically observed [Abay et al., 2020, Cui et al., 2021, Du et al., 2021, Ezzeldin et al., 2023, Makhija et al., 2024] that there is an underlying loss of accuracy when enforcing either local

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

Table 1: Comparison between our work and existing group fair FL frameworks

	Frameworks	Binary-Class	Multi-Class	Statistical Parity	Equal Opportunity	Equalized Odds	Approach
Local Fairness	`FCFL` [Cui et al., 2021]	✓	×	✓	✓	×	in-processing
Global Fairness	`FairFed` [Ezzeldin et al., 2023]	✓	×	✓	✓	×	in-processing
	`Fair-FATE` [Salazar et al., 2023]	✓	×	✓	✓	✓	in-processing
	`FedFB` [Zeng et al., 2021]	✓	×	✓	✓	✓	in-processing
	`Reweighing-FL` [Abay et al., 2020]	✓	×	✓	✓	×	pre-processing
Local and Global	`EquiFL` [Makhija et al., 2024]	✓	×	✓	✓	×	in-processing
	Ours	✓	✓	✓	✓	✓	post-processing

or global fairness. However, the exact minimum accuracy lost required to enforce both fairness remains unexplored. Furthermore, most studies on fairness in FL [Abay et al., 2020, Cui et al., 2021, Ezzeldin et al., 2023, Hamman and Dutta, 2023, Makhija et al., 2024, Salazar et al., 2023, Zeng et al., 2021] have focused on binary-class settings, despite the fact that many real-world FL applications [Arunkumar and Karthigaikumar, 2017] are multi-class problems. Most existing fair FL frameworks primarily address fairness w.r.t Statistical Parity (SP) [Dwork et al., 2012] and Equal Opportunity (EOp) [Hardt et al., 2016]. Equalized Odds (EO) [Hardt et al., 2016] has been neglected. EO is important in multi-class medical diagnosis since it ensures that individuals, regardless of their sensitive attributes, have an equal likelihood of being correctly diagnosed and receiving appropriate treatment across all diagnostic classes.

This paper provides a framework that investigates the minimum accuracy lost in enforcing both local and global fairness (specifically Equalized Odds) in multi-class FL. We formulate the problem of finding an optimal outcome predictor that satisfies fairness constraints as a convex program. The convex program's feasible set is a region under a Receiver Operating Characteristic (ROC) surface defined in [Landgrebe and Duin, 2008]. Solving the convex program is not practical due to the computational complexity of estimating the ROC surface, so we approximate that surface using a simplex, which allows us to reformulate the problem as a linear program (LP). The training algorithm of our framework is a post-processing approach that first trains a model using `FedAvg` [McMahan et al., 2017] and then enforces fairness through the LP. Our algorithm supports all commonly used fairness metrics, including SP, EOp and EO. Experiments show that, compared to SOTA, our framework achieves a comparable level of fairness with a smaller accuracy cost while reducing communication costs by 20% and computational costs by 33%.

2 Related Work

While the **cost of group fairness in centralized ML** has been well-studied [Chzhen et al., 2019, Kim et al., 2020, Menon and Williamson, 2018, Sabato and Yom-Tov, 2020, Tang and Zhang, 2022, Zhao and Gordon, 2022, Zhong and Tandon, 2024], the notion of local fairness is absent from the centralized settings. Existing literature thus focuses on the cost of global fairness. The accuracy lost for achieving both fairness concepts remains unexplored. Our FL framework fixes this gap and studies the accuracy lost for improving both fairness concepts.

Group fairness in FL requires that models do not discriminate individuals based on sensitive attributes. Existing methods for global or local fairness, summarized in Table 1, fall into three categories: pre-, in-, and post-processing. Pre-processing techniques [Abay et al., 2020] cannot address both global and local fairness. In-processing methods [Cui et al., 2021, Ezzeldin et al., 2023, Makhija et al., 2024, Salazar et al., 2023, Zeng et al., 2021] complicate the FL pipeline, lack convergence guarantees, and cannot precisely control fairness levels. Post-processing techniques [Duan et al., 2024] modify pre-trained models to a fair model. Our approach is a post-processing framework that optimally enforces both local and global fairness while allowing precise control of fairness in a multi-class setting across all common fairness metrics.

Existing **post-processing frameworks** [Chzhen et al., 2019, Denis et al., 2021, Gaucher et al., 2023, Hardt et al., 2016, Jiang et al., 2020, Xian and Zhao, 2023, 2024, Xian et al., 2023, Zeng et al., 2024] primarily focuses on centralized ML and specific fairness metrics. For example, [Denis et al., 2021, Gaucher et al., 2023, Jiang et al., 2020, Xian et al., 2023] only support Statistical Parity. [Chzhen et al., 2019, Hardt et al., 2016] supports Equalized Odds and Equal Opportunity, but only in binary settings. Our approach differs by being a distributed framework that achieves both local and global fairness, works in binary and multi-class settings, and supports all common group fairness metrics.

3 Preliminary Definitions

Notational Conventions: This paper will denote random variables using upper case letters, X, and lower case letters, x will denote instances of those random variables. A random variable's distribution will be denoted as F_X. Bold face lower case symbols will be reserved for vectors and bold face upper case symbols will be reserved for matrices.

To formalize our statistical setup, we define the notion of a **client group**.

Definition 3.1 *A **client group** consists of K geographically distinct clients that we formally represent as a tuple of jointly random variables, $D = (X, A, C, Y)$ with probability distribution $F_D : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \times \mathcal{Y} \to [0, 1]$. An instance of the client group, (x, a, c, y), is called an **individual** where $x \in \mathcal{X}$ is the individual's **private data vector**, $a \in \mathcal{A} = \{0, 1\}$ denotes the individual's **protected sensitive attribute**, and $c \in \mathcal{C} = \{1, 2, \ldots, K\}$ denotes which **client** the individual belongs to. The other value, $y \in \mathcal{Y} = \{1, \cdots, N\}$ denotes the individual's **qualified outcome**.*

Definition 3.1 has a concrete interpretation in a community health application. The client group D represents a city's clinic network, where each client is a distinct neighborhood served by a single health clinic. For an individual $(x, a, c, y) \sim F_D$, the variable x represents their private health data, a is a protected attribute (e.g., race or gender), c is their local clinic, and y indicates the individual has a certain illness and is therefore "qualified" to access corresponding medical resources to treat that illness.

We are interested in selecting an *outcome predictor*, $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ for client group D such that for any individual $(x, a, c, y) \sim F_D$ we have $\widetilde{Y}(x, a, c) = y$ with a high probability. With these definitions and notational conventions in place, we formally define global fairness and local fairness.

Definition 3.2 *(ϵ_0-Global Fairness) Let $\epsilon_0 \in [0, 1]$, the outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ for client group D satisfies **ϵ_0-global equalized odds** if and only if for all $y \in \mathcal{Y}$*

$$\begin{aligned} &|\mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = y \,|\, Y = y, A = 1 \right\} \\ &- \mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = y \,|\, Y = y, A = 0 \right\} | \leq \epsilon_0. \end{aligned} \tag{1}$$

Definition 3.3 *($\boldsymbol{\epsilon}_l$-Local Fairness) Let $\boldsymbol{\epsilon}_l = [\epsilon_1 \cdots, \epsilon_K] \in [0, 1]^K$, the outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ satisfies **$\boldsymbol{\epsilon}_l$-local equalized odds** if and only if for all $c \in \mathcal{C}$ and $y \in \mathcal{Y}$*

$$\begin{aligned} &|\mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = y \,|\, Y = y, A = 1, C = c \right\} \\ &- \mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = y \,|\, Y = y, A = 0, C = c \right\} | \leq \epsilon_c. \end{aligned} \tag{2}$$

Consider the *true positive* of outcome predictor $\widetilde{Y}$ for class y, group a and client c:

$$\mathrm{TP}_{ac}^{y}(\widetilde{Y}) = \mathbb{E}_{\mathrm{Pr}_{X|Y,A,C}} \left[\mathbf{1}(\widetilde{Y}(X, a, c) = y) \right] \tag{3}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Fairness requires equal true positives between sensitive and non-sensitive groups. In the following, we use the right lower indices a and c to represent the group and community, respectively. The right upper index y denotes the class. $(\cdot)_{ac}^{y}$ thus represents $(\cdot)$ for class y, group a, and client c.

The predictor's *accuracy* is a linear combination of true positives:

$$\mathrm{acc}(\widetilde{Y}) = \sum_{y,a,c \in \mathcal{Y},\mathcal{A},\mathcal{C}} \left[\mathrm{Pr}_D(Y = y, A = a, C = c) \cdot \mathrm{TP}_{ac}^{y}(\widetilde{Y}) \right]$$

The ideal fair outcome predictor is one that achieves a true positive rate of 1 for classes, groups and clients. However, such a predictor in general is not achievable due to the noise in the data. The following section formally defines the feasible set of true positives for multi-class problem.

4 Region under the ROC Surface

This section defines ROC surfaces used in the next section to formulate the optimal multi-class fair predictor. To define the ROC surface, we first extend the score function in [Hardt et al., 2016] to multi-class setting.

Definition 4.1 (Bayesian Optimal Score Function) *Let the function $R : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to [0, 1]^N$ have components $R(x, a, c) = [r_1(x, a, c), r_2(x, a, c), \cdots r_N(x, a, c)]$ This function is a Bayesian Optimal Score Function if $\sum_{y=1}^{N} r_y(x, a, c) = 1$, and for all $y \in \mathcal{Y}$, its components satisfy: $r_y(x, a, c) = \mathrm{Pr}_D(Y = y | X = x, A = a, C = c)$*

The Bayesian Optimal Score function maps inputs to a probability distribution over $\mathcal{Y}$, with each element representing the probability that an individual (x, a, c) belongs to each class. For simplicity, we drop the input arguments and write $R(X, A, C)$ or $\widetilde{Y}(X, A, C)$ as R or $\widetilde{Y}$.

Definition 4.2 (Derived Outcome Predictor) *Let $R : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to [0, 1]^N$ be the Bayesian Optimal Score Function and $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_N] \in \mathbb{R}_{\geq 0}^N$ be a given vector. The outcome predictor $\widetilde{Y}_{\boldsymbol{\theta}} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ that takes value of:*

$$\widetilde{Y}_{\boldsymbol{\theta}} = y, \;\; \textit{if: } \theta_y r_y(x, a, c) = \max_{i=1}^{N} \theta_i r_i(x, a, c) \tag{4}$$

is said to be derived from the Bayesian Optimal Score via $\boldsymbol{\theta}$*. The set of all derived outcome predictors is* $\{\widetilde{Y}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}}$.

A derived outcome predictor returns the highest value of the score weighted by $\boldsymbol{\theta}$. The derived outcome predictor follows the same format as in [Landgrebe and Duin, 2008]. It generalizes the threshold test predictor from [Hardt et al., 2016] used in binary classification.

The following proposition generalizes the NP Lemma [Neyman and Pearson, 1933] to multi-class settings.

Proposition 4.3 *(Appendix E.1) Consider client* c *and group* a*, let* $\{\widetilde{Y}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}}$ *be the set of all derived outcome predictors, and let* $\{\phi_g\}_{g=1}^{N-1}, \in [0,1]^N$ *be a set of specified values of true positives. If* $\widetilde{Y}^* : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$ *is the solution to the following optimization problem:*

$$\begin{array}{ll} \textit{maximize} & \mathrm{TP}^N_{ac}(\widetilde{Y}) \\ \textit{with respect to} & \widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y} \\ \textit{subject to} & \forall g \in \{1, 2, \cdots, N-1\}, \\ & \mathrm{TP}^g_{ac}(\widetilde{Y}) = \phi_g \end{array}$$

then $\widetilde{Y}^* \in \{\widetilde{Y}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}}$.

Proposition 4.3 states that, given the true positive rates for the first $(N-1)$ classes, the predictor that maximizes the true positive rate for class N is the *derived outcome predictor.*

Consider the vector in $[0,1]^N$ that represents the true positives for client c and group a of the *derived outcome predictor* for a given $\boldsymbol{\theta}$. We denote the vector of true positives of the predictor $\widetilde{Y}_{\boldsymbol{\theta}}$ as $\mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})$:

$$\mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}) = [\mathrm{TP}^1_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}), \ldots, \mathrm{TP}^N_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})]^T$$

The ROC hypersurface for the multi-class setting is composed of the true positive vectors of all derived outcome predictors, $\{\widetilde{Y}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}}$.

The ROC hypersurface for client c and group a is thus defined as:

$$\mathrm{ROC}_{ac} := \{\mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}) : \forall \boldsymbol{\theta} \in \mathbb{R}^N_{\geq 0}\} \tag{5}$$

The ROC hypersurfaces differ across clients as the data distributions vary across different clients in the FL system. We consider the region under the ROC hypersurface (RUS). The RUS is defined with respect to the separating hyperplanes that distinguishes the region under the ROC hypersurface from other regions.

Definition 4.4 *Let* $\boldsymbol{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})$ *be the point representing the true positive of the derived outcome predictor for a given* $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_N]$*, the hyperplane of* $\boldsymbol{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})$ *is defined as:*

$$\left\{\mathbf{x} \in \mathbb{R}^N | \mathbf{v}^T_{\boldsymbol{\theta}}\mathbf{x} = \mathbf{v}^T_{\boldsymbol{\theta}}\boldsymbol{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})\right\}$$

where,

$$\mathbf{v}^T_{\boldsymbol{\theta}} = [\theta_y \mathrm{Pr}_D(Y = y | A = a, C = c), \forall y \in \mathcal{Y}]$$

For any given $\boldsymbol{\theta} \in \mathbb{R}^N \geq \mathbf{0}$, it has a separating hyperplane such that the set $\{\mathbf{x} \in \mathbb{R}^N | \mathbf{v}^T_{\boldsymbol{\theta}}\mathbf{x} > \mathbf{v}^T_{\boldsymbol{\theta}}\mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})\}$ excludes the RUS.

Definition 4.5 *The region under* ROC_{ac} *is:*

$$D_{ac} = \bigcap_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}} \left\{\mathbf{x} \in [0,1]^N | \mathbf{v}^T_{\boldsymbol{\theta}}\mathbf{x} \leq \mathbf{v}^T_{\boldsymbol{\theta}}\boldsymbol{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})\right\}$$

Proposition 4.6 *(Appendix E.2) Let* D_{ac} *be the region defined in Def.4.5. Then,* D_{ac} *is a convex set.*

Proposition 4.7 *(Appendix E.3) Let* D_{ac} *be the region as defined in Def.4.5. For any predictor* $\widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$*, let the point representing true positives of* $\widetilde{Y}$ *be:* $\boldsymbol{TP}_{ac}(\widetilde{Y}) = [\mathrm{TP}^1_{ac}(\widetilde{Y}), \cdots, \mathrm{TP}^N_{ac}(\widetilde{Y})]$*. Then,* $\boldsymbol{TP}_{ac}(\widetilde{Y})$ *lies in* D_{ac}.

Proposition 4.7 asserts that for group a in client c, any outcome predictor that is a map $\widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$, has its true positives lying in D_{ac}. Thus, D_{ac} is the set representing the feasible region of true positives.

5 The Minimum Cost of Fairness

This section presents a framework that returns an optimal fair outcome predictor. The framework is implemented by solving an LP with polynomial computational complexity. It is applicable to large-scale distributed learning systems.

Consider the loss function $\ell : \mathcal{Y}\times\mathcal{Y}\to\{0,1\}$ that takes values

$$\ell(\widetilde{y}, y) = \mathbf{1}(\widetilde{y} \neq y) \tag{6}$$

for any $\widetilde{y}, y \in \mathcal{Y}$, where $\mathbf{1}(\cdot)$ is the indicator function. Let $\boldsymbol{\epsilon} = [\epsilon_0, \epsilon_1, \epsilon_2, \cdots, \epsilon_K] \in [0,1]^{K+1}$ be a prespecified fairness level. Any predictor $\widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$ that satisfies the following optimization problem is an **$\boldsymbol{\epsilon}$-fair optimal outcome predictor**

$$\begin{array}{ll} \text{minimize} & \mathbb{E}_D\left[\ell(\widetilde{Y}(X, A, C), Y)\right] \\ \text{with respect to} & \widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y} \\ \text{subject to} & \text{eq.(1) and eq.(2)} \end{array} \tag{7}$$

Optimization (7) selects an outcome predictor that satisfies the specified local and global fairness constraints while minimizing the classifier's error rate.

Optimization (7) can be recast as the following convex program.

For notational convenience, let z^y_{ac} be the variables for the outcome predictor $\widetilde{Y}$:

$$z^y_{ac} = TP^y_{ac}(\widetilde{Y}) = \Pr_D(\widetilde{Y} = y | Y = y, A = a, C = c)$$

that represents the true positives of $\widetilde{Y}$ for class y, group a and client c. The following proposition asserts that if $\{z^y_{ac}\}_{\mathcal{Y},\mathcal{A},\mathcal{C}}$ satisfies the following convex program, then $\widetilde{Y}$ is a $\epsilon-$fair optimal outcome predictor.

Proposition 5.1 *(Appendix E.4) Let the vector* $\mathbf{z} \in \mathbb{R}^{2NK}$*:*

$$\mathbf{z}^T \;=\; \begin{bmatrix} \mathbf{z}^T_{01} & \mathbf{z}^T_{11} & \mathbf{z}^T_{02} & \mathbf{z}^T_{12}\cdots & \mathbf{z}^T_{0K} & \mathbf{z}^T_{1K} \end{bmatrix}$$

where $\mathbf{z}_{ac}$ *are true positives of all classes:* $y \in \mathcal{Y}$ *for group* a *in client* c*:*

$$\mathbf{z}^T_{ac} \;=\; \begin{bmatrix} z^1_{ac} & z^2_{ac} & z^3_{ac} & \cdots & z^N_{ac} \end{bmatrix} \in \mathbb{R}^N$$

satisfy the following convex program

$$\begin{array}{ll} \textit{minimize:} & \mathbf{c}^T\mathbf{z} \\ \textit{with respect to:} & \mathbf{z} \in \mathbb{R}^{2NK} \\ \textit{subject to:} & -\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b} \\ & \mathbf{z}_{ac} \in D_{ac}, \forall a \in \mathcal{A}, c \in \mathcal{C} \end{array} \tag{8}$$

then, the outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ *that satisfies eq.* (9) *for all* $y \in \mathcal{Y}, a \in \mathcal{A}, c \in \mathcal{C}$

$$\Pr(\widetilde{Y} = y | Y = y, A = a, C = c) = z^y_{ac} \tag{9}$$

is an $\boldsymbol{\epsilon}$*-fair optimal outcome predictor.The optimal accuracy for a* ϵ*-fair outcome predictor is* $-\mathbf{c}^T\mathbf{z}$.

The parameters of (8), $\mathbf{A} \in \mathbb{R}^{N(K+1)\times 2NK}$, $\mathbf{c} \in \mathbb{R}^{2NK}$ and $\mathbf{b} \in \mathbb{R}^{N(K+1)}$ are detailed in Appendix A. D_{ac} is the region under the ROC hypersurface as defined in Def. 4.5. Optimization (8) is a convex program since D_{ac} is a convex set (Proposition 4.6). The first N inequalities of $-\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b}$ represent global fairness constraints and the last (NK) inequalities represent local fairness constraints.

Since D_{ac} is the feasible region for the true positives of all outcome predictors (Proposition 4.7), the predictor $\widetilde{Y}$ is fair and has optimal accuracy. The convex program (8) thus presents the *inherent trade-off* between global fairness, local fairness and accuracy and characterizes the minimum accuracy lost in enforcing specified level of global and local fairness.

The computational complexity of generating the convex set D_{ac} increases exponentially w.r.t the number of classes [Landgrebe and Duin, 2008]. Instead of solving the convex program (8) directly, our implementation approximates the D_{ac} using a simplex $\widehat{D_{ac}}$, reformulating the problem as a linear program.

Let us consider the *derived outcome predictor* $\widetilde{Y}_{\boldsymbol{\theta}}$ in Def 4.2 with $\boldsymbol{\theta} = \mathbf{1}$. $\widetilde{Y}_{\mathbf{1}} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ takes values of:

$$\widetilde{Y}_{\mathbf{1}}(x, a, c) = \arg\max_{y \in \mathcal{Y}} r_y(x, a, c) \tag{10}$$

The point representing true positives for $\widetilde{Y}_{\mathbf{1}}$ is: $\mathbf{TP}_{ac}(\widetilde{Y}_{\mathbf{1}}) = [\mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}}), \cdots, \mathrm{TP}^N_{ac}(\widetilde{Y}_{\mathbf{1}})]$. The simplex we use to approximate the convex set D_{ac} is the one whose vertices are $\{\mathbf{e}_y\}_{\forall y \in \mathcal{Y}}$ and $\mathbf{TP}_{ac}(\widetilde{Y}_{\mathbf{1}})$:

$$\widehat{D_{ac}} = \{f_0\mathbf{TP}_{ac}(\widetilde{Y}_{\mathbf{1}}) + \sum_{y=1}^{N} f_y\mathbf{e}_y | \sum_{i=0}^{N} f_i = 1 \text{ and } \forall i, f_i \geq 0\}$$

where, $\mathbf{e}_y$ are elementary basis vectors. We choose $\widehat{D_{ac}}$ to be an inner approximation to D_{ac} ($\widehat{D_{ac}} \subset D_{ac}$). $\widetilde{Y}_{\mathbf{1}}$ is the predictor that maximizes the accuracy for the distribution D (Appendix E.6).

The polytope $\widehat{D_{ac}}$ can be represented using $(N+1)$ inequalities. Any point $\mathbf{u} \in \mathbb{R}^N$ that lies in $\widehat{D_{ac}}$ must satisfy:

$$\mathbf{K}_{ac}\mathbf{u} \leq \mathbf{l}_{ac} \tag{11}$$

where $\mathbf{K}_{ac} \in \mathbb{R}^{(N+1)\times N}$ and $\mathbf{l}_{ac} \in \mathbb{R}^{(N+1)}$ are detailed in Appendix B.

We now reformulate the problem (8) as LP (12):

$$\begin{array}{ll} \text{minimize:} & \mathbf{c}^T\mathbf{z} \\ \text{with respect to:} & \mathbf{z} \in \mathbb{R}^{2NK} \\ \text{subject to:} & -\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b} \\ & \mathbf{K}_{ac}\mathbf{z}_{ac} \leq \mathbf{l}_{ac}, \forall a \in \mathcal{A}, c \in \mathcal{C} \end{array} \tag{12}$$

The LP has the same parameters as the convex program, except for the final linear constraint: $\mathbf{K_{ac}}\mathbf{z}_{ac} \leq \mathbf{l}_{ac}$. LP (12) can be extended to other fairness metrics (see Appendix D).

LP (12) identifies the true positives for the fair outcome predictor. The next step is to find a classifier that achieves these true positives. The following proposition demonstrates the existence and uniqueness of the fair outcome predictor given the LP solution.

Proposition 5.2 *(Appendix E.5) Let* $\mathbf{z} \in \mathbb{R}^{2NK}$ *be the solution of the LP (12)*

$$\mathbf{z}^T \;=\; \begin{bmatrix} \mathbf{z}^T_{01} & \mathbf{z}^T_{11} & \mathbf{z}^T_{02} & \mathbf{z}^T_{12}\cdots & \mathbf{z}^T_{0K} & \mathbf{z}^T_{1K} \end{bmatrix}$$

and $\mathrm{TP}^y_{ac}(\widetilde{Y}_{\mathbf{1}})$ *be the true positive of the predictor defined in Eq.* (10)*. For all* $a \in \mathcal{A}, c \in \mathcal{C}$*, let* $\boldsymbol{\beta}_{ac} = [\beta^0_{ac}, \beta^1_{ac}, \cdots, \beta^N_{ac}]$ *be the solution of the following linear algebraic equation (LAE),*

$$\mathbf{G}_{ac}\boldsymbol{\beta}_{ac} = \boldsymbol{\gamma}_{ac} \tag{13}$$

where, the parameter $\mathbf{G}_{ac} \in \mathbb{R}^{(N+1)\times(N+1)}, \boldsymbol{\gamma}_{ac} \in \mathbb{R}^{N+1}$*, are detailed in Appendix C. Then. the predictor* $\widetilde{Y}_{\boldsymbol{\beta}_{ac}}$ *that takes value,*

$$\widetilde{Y}_{\boldsymbol{\beta}_{ac}}(x, a, c) = \begin{cases} \widetilde{Y}_{\mathbf{1}}(x, a, c), & \textit{with the probability } \beta^0_{ac} \\ y, & \textit{with the probability } \beta^y_{ac}, \forall y \in \mathcal{Y} \end{cases} \tag{14}$$

```
Algorithm 1 Fair Outcome Predictor
Input: The outcome predictor: $\widetilde{Y}_{\mathbf{1}} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$, the client $c$'s $\boldsymbol{\beta}_{ac}, \forall a \in \mathcal{A}$
    $\boldsymbol{\beta}_{ac}^T = [\beta_{ac}^0, \beta_{ac}^1, \beta_{ac}^2, \cdots \beta_{ac}^N] \in [0,1]^{N+1}$
Output: Fair outcome predictor $\widetilde{Y}_{\boldsymbol{\beta}_{ac}} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$
  1. randomly sample $s \sim U(0,1)$, the uniform distribution between (0,1)
  2. Construct $\widetilde{Y}_{\boldsymbol{\beta}_{ac}}(x,a,c)$ as
     $\widetilde{Y}_{\boldsymbol{\beta}_{ac}}(x,a,c) = \begin{cases} \widetilde{Y}_{\mathbf{1}}(x,a,c), \text{ if } s \le \beta_{ac}^0 \\ 1, \text{ if } \beta_{ac}^0 < s \le \sum_{i=0}^{i=1} \beta_{ac}^i \\ 2, \text{ if } \sum_{i=0}^{i=1} \beta_{ac}^i < s \le \sum_{i=0}^{i=2} \beta_{ac}^i \\ 3, \text{ if } \sum_{i=0}^{i=2} \beta_{ac}^i < s \le \sum_{i=0}^{i=3} \beta_{ac}^i \\ \cdots \\ N, \text{ if } \beta_{ac} + \sum_{i=0}^{i=N-1} \beta_{ac}^i < s \le \beta_{ac} + \sum_{i=0}^{i=N} \beta_{ac}^i \end{cases}$
return $\widetilde{Y}_{\boldsymbol{\beta}_{ac}}$
```

is a fair outcome predictor. There always exists a unique set of parameters $\{\boldsymbol{\beta}_{ac}\}_{\mathcal{A},\mathcal{C}}$, where $\boldsymbol{\beta}_{ac} \in [0,1]^{N+1}$ and $|\boldsymbol{\beta}_{ac}|_{\ell_1} = 1$ that satisfies the LAE.

Proposition 5.2 gives the fair outcome predictor from the LP solution. Combining the LP (12) and LAE (13), our framework first solves the LP (12) that identifies the true positives for the fair outcome predictor. Then, the fair outcome predictor (14) can be uniquely determined by solving the LAE (13).

6 Training Fair Predictor in FL

This section shows how LP (12) and LAE (13) can be used within a FL framework to construct a fair outcome predictor. The training steps of our framework are enumerated below:

Step 1: **Training an Optimal Score Function Using FedAvg**: The server and clients collaboratively train a Bayesian optimal score function $R : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to [0,1]^N$ that minimizes the categorical cross-entropy loss using the FedAvg algorithm [McMahan et al., 2017].

The FedAvg algorithm converges to the Bayesian Optimal Score at a rate of $\mathcal{O}(1/T)$, where T is the number of communication round [Li et al., 2019].

Step 2: **Computing Client Statistics**: Each local client generates the output $\widetilde{Y}_{\mathbf{1}}(X,A,C)$ using Eq. 10 , computes the following empirical statistics (15) and transmits these probabilities to the global server.

$$\begin{aligned} &\Pr_D(\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a \mid C = c) \\ =&\frac{\#\text{ of samples with } (\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a) \text{ in client } c}{\#\text{ of samples in client c}} \\ &\Pr_D(Y = y, A = a \mid C = c) \\ =&\frac{\#\text{ of samples with } (Y = y, A = a) \text{ in client } c}{\#\text{ of samples in client c}} \end{aligned} \tag{15}$$

In this step, the client transmits statistics to the serve the individual's private data $x \in \mathcal{X}$, such as health profiles remains stored locally.

Step 3: **Constructing and Solving the Linear Program**: The global server computes the parameters of the LP (12) using the statistics sent by the clients. The parameters of the LP, as detailed in Appendix A, $p_{ac}^y, \alpha_a^y, \mathrm{TP}_{ac}^y(\widetilde{Y}_{\mathbf{1}})$ are computed as:

$$\begin{aligned} &p_{ac}^y = \Pr_D(Y = y, A = a | C = c) \cdot p_c, \quad \alpha_a^y = \sum_{c \in \mathcal{C}} p_{ac}^y \\ &\mathrm{TP}_{ac}^y(\widetilde{Y}_{\mathbf{1}}) = \frac{\Pr_D(\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a | C = c)}{\Pr_D(Y = y, A = a | C = c)} \\ &\quad \text{with, } p_c = \frac{\#\text{ of samples in client } c}{\#\text{ of total samples}} \end{aligned} \tag{16}$$

$\Pr_D(\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a \mid C = c), \Pr_D(Y = y, A = a \mid C = c)$ are the statistics transmitted from local clients.

Using the above parameters, the global server constructs the linear program (12), finds the minimizer $\mathbf{z}^T = \begin{bmatrix} \mathbf{z}_{01}^T & \mathbf{z}_{11}^T & \mathbf{z}_{02}^T & \mathbf{z}_{12}^T \cdots & \mathbf{z}_{0K}^T & \mathbf{z}_{1K}^T \end{bmatrix}$ and then sends the corresponding minimizer $\mathbf{z}_{0c}^T, \mathbf{z}_{1c}^T$ to client c, where $c = 1, 2, \cdots, K$.

Step 4: **Solving the LAE:** Local clients compute the parameters of the LAE (13). The parameter $\mathrm{TP}_{ac}^y(\widetilde{Y}_{\mathbf{1}})$ is computed in the same way as in equation (17), and $\mathbf{z}_{ac}$ is sent by the server to the clients. The local clients then construct and solve the LAE (13). The solution, $\boldsymbol{\beta}_{ac}$, is used in Algorithm 1 to make fair predictions.

Algorithm 1 provides a client-dependent randomized function that returns a model, which takes the value $\widetilde{Y}_{\mathbf{1}}$ with probability β_{ac}^0 and y with probability β_{ac}^y.

7 Experiments

We conduct experiments on real-world datasets. Results shows that our framework significantly reduces both local and global disparity, achieving competitive accuracy while ensuring fairness compared to SOTA.

Table 2: Accuracy, local and global disparity of all algorithms. All measurement is computed over 5 runs.

Dataset	Frameworks	Local Disparity (↓)	Global Disparity (↓)	Avg-Acc (%)(↑)
Adult (SP)	*FedAvg*	0.287 ± 0.037	0.226 ± 0.033	$\mathbf{84.5 \pm 0.6}$
	FCFL	0.052 ± 0.011	0.055 ± 0.020	79.5 ± 0.7
	FairFed	$\mathbf{0.031 \pm 0.022}$	0.030 ± 0.009	78.4 ± 2.1
	Fair-FATE	0.041 ± 0.039	0.037 ± 0.098	79.7 ± 3.4
	EquiFL	0.053 ± 0.064	0.039 ± 0.062	77.0 ± 2.7
	Ours	0.039 ± 0.012	$\mathbf{0.003 \pm 0.001}$	81.0 ± 0.3
	Ours (only local)	0.044 ± 0.015	0.023 ± 0.013	81.0 ± 0.2
	Ours (only global)	0.155 ± 0.001	$\mathbf{0.003 \pm 0.002}$	81.0 ± 0.3
PublicCoverage (EOp)	*FedAvg*	0.103 ± 0.003	0.062 ± 0.004	$\mathbf{77.8 \pm 0.4}$
	FCFL	0.051 ± 0.020	0.032 ± 0.02	76.5 ± 0.4
	FairFed	0.091 ± 0.031	0.003 ± 0.001	76.4 ± 0.4
	Fair-FATE	0.077 ± 0.050	0.049 ± 0.066	69.9 ± 6.1
	EquiFL	0.051 ± 0.001	0.042 ± 0.002	76.5 ± 0.1
	Ours	$\mathbf{0.041 \pm 0.001}$	$\mathbf{0.001 \pm 0.002}$	76.4 ± 0.2
	Ours (only local)	0.048 ± 0.002	0.020 ± 0.001	76.5 ± 0.6
	Ours (only global)	0.163 ± 0.001	$\mathbf{0.001 \pm 0.000}$	76.6 ± 0.0
HM10000 (EO)	*FedAvg*	0.421 ± 0.040	0.198 ± 0.024	$\mathbf{75.4 \pm 0.6}$
	Ours	$\mathbf{0.113 \pm 0.021}$	0.056 ± 0.025	69.4 ± 0.2
	Ours (only local)	0.116 ± 0.050	0.161 ± 0.031	69.5 ± 0.2
	Ours (only global)	0.241 ± 0.020	$\mathbf{0.048 \pm 0.017}$	74.9 ± 0.1

Table 3: The number of communication round and total time for convergence for all algorithms

	Adult					PublicCoverage				
Frameworks	*FCFL*	*FairFed*	*Fair-FATE*	*EquiFL*	*Ours*	*FCFL*	*FairFed*	*Fair-FATE*	*EquiFL*	*Ours*
# of convergence rounds	> 1000	≈ 15	≈ 12	≈ 30	**≈ 10**	> 1000	≈ 15	≈ 15	≈ 30	**≈ 10**
Total time for convergence (s) ≈	220.1	27.9	40.6	54.3	**18.8**	1080.0	484.8	563.8	852.2	**227.8**

Furthermore, our framework outperforms SOTA in communication efficiency and computational cost.

Dataset. We evaluate classification tasks on three datasets: Adult Dataset, ACSPublicCoverage Dataset and HM10000 Dataset.

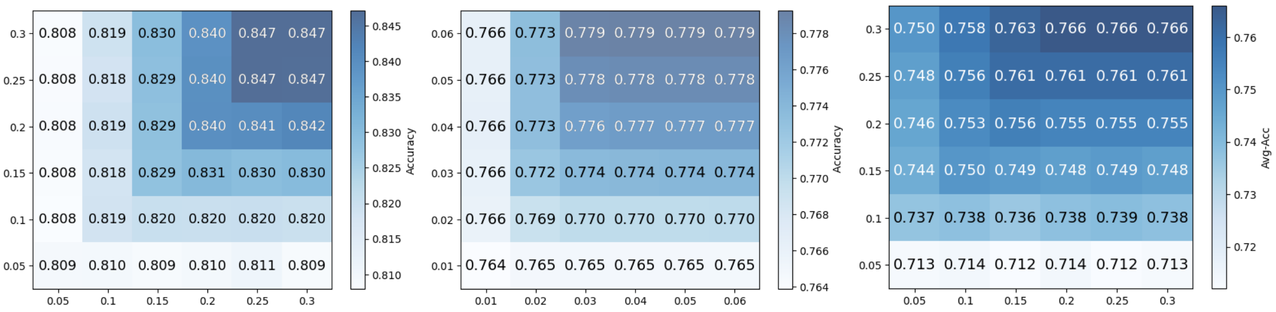
(1) **Adult** [Asuncion and Newman, 2007] predicts whether an individual earns over 50K/year. Following [Cui et al., 2021], the dataset is split into two clients based on PhD status, with gender as the sensitive attribute. The fairness metric is Statistical Parity (SP), ensuring equal likelihood of earning over 50K/year for males and females.

(2) **ACSPublicCoverage** [Ding et al., 2021] predicts whether an individual has public health insurance. It is divided by 50 U.S. states, with race (white/non-white) as the sensitive attribute. The fairness metric is Equal Opportunity (EOp), ensuring equal probability of coverage for eligible individuals, regardless of race.

(3) **HM10000** [Tschandl et al., 2018] includes 10,000 dermatoscopic images and we use it to predict 4 diagnostic class. The dataset is split into five age-based communities, with gender as the sensitive attribute. The fairness metric is Equalized Odds (EO), ensuring equal likelihood of correct diagnosis and treatment across all classes, regardless of gender.

Evaluation. We assess the model from three perspectives: model performance, local fairness and global fairness. (1) Model performance is measured by accuracy over all samples. For different datasets, we enforce fairness w.r.t different metrics and measure it accordingly. We use SP Difference (difference in positive rate between the sensitive and the non-sensitive) for Adult, EOp Difference (difference in true positives for class 1) for ACSPublicCoverage and EO Difference (maximum difference in true positives across all classes) for HM10000. (2) Local fairness is measured by average disparity within each client. (3) Global fairness, measured by disparity across the entire dataset.

Our Implementation and Baselines. We configure our implementation to enforce only local fairness ($\epsilon_0 = 1, \epsilon_c = 0.01$), only global fairness ($\epsilon_0 = 0.01, \epsilon_c = 1$) and both ($\epsilon_0 = 0.01, \epsilon_c = 0.01$) by adjusting ϵ_0 and ϵ_c in the LP (12). We compare ours to existing baelines that are designed to enforce local fairness: `FCFL` [Cui et al., 2021], global fairness: `Fair-Fate` [Salazar et al., 2023] `FairFed` [Ezzeldin et al., 2023] and both `EquiFL` [Makhija et al., 2024]. These baselines focus on a binary setting and most of them cannot be extended to enforce EO in multi-class. We compare ours with them in two binary classification tasks: Adult and ACSPublicCoverage. Full experimental details, including



(a) Adult

(b) ACSPublicCoverage

(c) HM10000

Figure 1: The accuracy under different levels of local fairness (y-axis) and global fairness (x-axis). The numbers inside the blocks are the accuracy, with each value being the average of 10 runs.

models, baselines and hyperparameters are detailed in Appendix F. We summarize the main experimental results in the following:

Fairness of Our Implementation. Table 2 shows accuracy, local disparity, and global disparity for three datasets. Compared to `Fedavg`, our method reduces local disparity by 86.4% (0.287 → 0.044) for Adult, 60.2% (0.103 → 0.041) for ACSPublicCoverage, and 73.2% (0.421 → 0.113) for HM10000. Global disparity decreases by 89.8% (0.226 → 0.023) for Adult, 98.4% (0.062 → 0.001) for ACSPublicCoverage, and 71.7% (0.198 → 0.056) for HM10000. The statistical fluctuation of our implementation is less than 1% relative to the reduction. These results demonstrate that our implementation significantly improves both fairness concepts in FL.

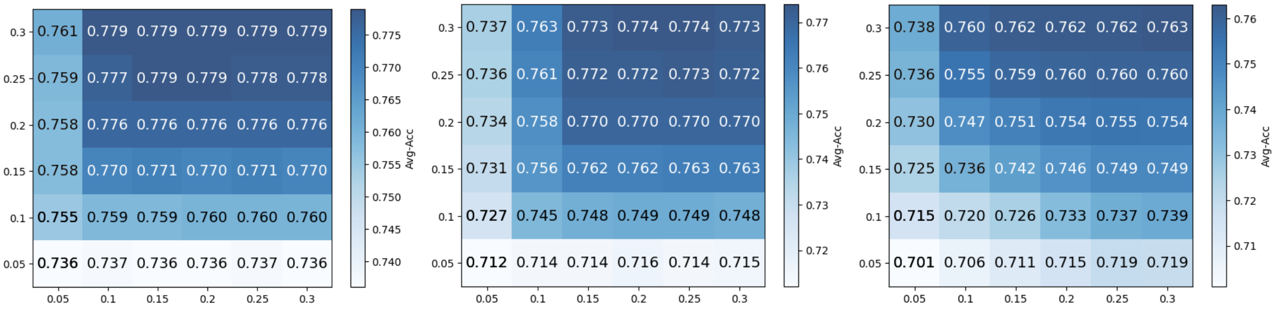
Comparing ours with local constraints and Ours with global constraints in Table 2, enforcing local fairness also reduces global disparity by 89.8% for Adult, 67.7% for ACSPublicCoverage, and 20% for HM10000. Conversely, enforcing global fairness reduces local disparity by 45% for Adult and 42% for HM10000 but increases it by 58% for ACSPublicCoverage. It shows that enforcing one type of fairness impacts the other.

Flexibility of Adjusting the Level of Fairness. Fig. 1 shows the accuracy under varying levels of local fairness (y-axis) and global fairness (x-axis). It demonstrates that by changing the parameters in LP (12), the trade-off between accuracy and fairness can be flexibly adjusted. The ratio of accuracy lost for local disparity reduction to global disparity reduction is 0.98, indicating that two reductions have a close impact on accuracy. For the ACSPublicCoverage, the ratio is 1.12, indicating that reducing local disparity has a slightly higher impact on accuracy. In the HM10000 dataset, the accuracy lost of reducing local disparity is significantly higher than reducing global disparity. These results shows that the impact of local and global fairness on accuracy varies across different datasets.

Accuracy Cost of Fairness. Table 2 shows the accuracy lost for our implementation of achieving both fairness objectives is 3.5% (84.5 → 81) for Adult, 1.4% (77.8 → 76.5) for PublicCoverage and 6% (75.4 → 69.4) for HM10000. Compared to other baselines, our framework achieves comparable or better fairness levels, the accuracy cost of ours is 0.5% – 4% less for the Adult dataset and 0% – 6% less for PublicCoverage and HM10000. This supports the assertion that our approach has the smaller or comparable accuracy lost for improving fairness.

Communication and Computational Cost. We use the number of communication rounds required as a measure of communication efficiency and the time to convergence as a measure of computational cost. The computation time is obtained on a 16-Core 4.00 GHz AMD Pro 5955WX Processor. The results in Table 3 show that the number of communication rounds for our method is 20% smaller for the Adult and 33% smaller for PublicCoverage compared with baselines that has smallest communication round (`Fair-Fate`) . The computational time is 32.6% less than for Adult and 53% less for PubCoverage than the fastest baseline (`FairFed`). These results show that our implementation improves fairness in FL with significantly lower communication and computational cost.

Impact of Data Heterogeneity We investigate how data heterogeneity impacts the trade-off between global fairness, local fairness and accuracy. Data heterogeneity in FL refers to the scenarios that the data across clients in FL are non-i.i.d. Experiments were conducted on the HM10000 dataset, where the client makeup with respect to the sensitive attribute was varied. The results for three different data heterogeneity scenarios are shown in Fig. 2. In Scenario 1, the proportion of sensitive (female) samples across the 5 clients is $[0.4, 0.4, 0.4, 0.4, 0.4]$. The the



(a) Scenario 1

(b) Scenario 2

(c) Scenario 3

Figure 2: The accuracy under different levels of local (y-axis) and global fairness (x-axis) for different data heterogeneity setup of HM10000. (each value is the average of 10 runs.)

sensitive attribute is i.i.d. across clients. In Scenario 2, the proportion of sensitive samples of each client is $[0.2, 0.3, 0.4, 0.5, 0.6]$. In Scenario 3, it is $[0.1, 0.3, 0.5, 0.7, 0.9]$. It is clear that from Scenario 1 to Scenario 3, the level of data heterogeneity increases.

Fig. 2 shows the accuracy lost for improving local and global fairness. The accuracy loss for reducing both local and global disparity from 0.3 to 0.05 in Scenario 1 is 0.045, which is 5.5% of the initial accuracy. The accuracy loss is 7.9% of the initial accuracy for Scenario 2 and 8.1% for Scenario 3. The result indicate that the accuracy loss for achieving both fairness increases as the level of data heterogeneity increases.

When local disparity is bounded by 0.05, we observe that as the global disparity is reduced from 0.3 to 0.05, the change of accuracy is 0% ($0.736 \rightarrow 0.736$) of the initial accuracy for Scenario 1, 0.4% ($0.715 \rightarrow 0.712$) of the initial accuracy for Scenario 2, and 2.5% ($0.719 \rightarrow 0.701$) for Scenario 3. This suggests that in scenarios with higher data heterogeneity, the accuracy loss for achieving global fairness is higher.

Privacy Protection of the Local Statistics Local differential privacy (DP) mechanism can be applied when communicating local statistics in step 2 of algorithm in Section 6 to protect client-level privacy. Experimental results in Table 4 shows how DP mechanisms affect the fairness and accuracy of our algorithms on UCI Adult. We add the Laplace mechanism on local statistics in Eq. (15) so those local statistics satisfies ϵ- differential privacy privacy, a smaller ϵ indicate better privacy protection. The detail of Laplace mechanism is in Appendix F.3. The local disparity, global disparity and Avg-Acc after post-processing are reported in Table 4

Compared to the initial FedAvg with $\epsilon = 0.05$, our framework reduces local disparity by 74% ($0.287 \rightarrow 0.076$) and global disparity by 66% ($0.215 \rightarrow 0.072$). It demonstrates our framework's effectiveness in mitigating both local and global disparities under a 0.05-differentially private setting. As ϵ decreases (i.e., better privacy protection), both local and global disparity increases. It is clear that there is a trade-off between privacy and fairness.

Table 4: Local Disparity (Local), Global Disparity (Global) and Avg-Acc under ϵ- Differential Privacy

ϵ	**Local**	**Global**	**Avg-Acc**
∞	0.019	0.015	81.30
1	0.024	0.017	81.69
0.5	0.043	0.038	81.32
0.1	0.057	0.072	82.00
0.05	0.076	0.072	81.54
FedAvg	0.287	0.215	84.83

8 Conclusions

This paper presents a convex program to identify the optimal accuracy under varying levels of local and global fairness in multi-class FL. Our framework is implemented by solving an LP with polynomial time complexity. Experiments show it outperforms baselines in accuracy lost for fairness, communication efficiency, and computational cost for achieving fairness.

9 Acknowledgments

This work was supported by the National Science Foundation under Grant No. CNS-2228092.

References

Annie Abay, Yi Zhou, Nathalie Baracaldo, Shashank Rajamoni, Ebube Chuba, and Heiko Ludwig. Mitigating bias in federated learning. *arXiv preprint arXiv:2012.02447*, 2020.

Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

R Arunkumar and Palanivel Karthigaikumar. Multi-retinal disease classification by reduced deep learning features. *Neural Computing and Applications*, 28:329–334, 2017.

Arthur Asuncion and David Newman. Uci machine learning repository, 2007.

Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. Leveraging labeled and unlabeled data for consistent fair binary classification. *Advances in Neural Information Processing Systems*, 32, 2019.

U.S. Congress. Health equity and accountability act of 2022, 2022. URL `https://www.congress.gov/bill/117th-congress/house-bill/7585`. H.R. 7585, 117th Cong.

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.

Christophe Denis, Romuald Elie, Mohamed Hebiri, and François Hu. Fairness guarantee in multi-class classification. *arXiv preprint arXiv:2109.13642*, 2021.

Frances Ding, Moritz Hardt, John Miller, and Ludwig Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490, 2021.

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)*, pages 181–189. SIAM, 2021.

Yuying Duan, Yijun Tian, Nitesh Chawla, and Michael Lemmon. Post-fair federated learning: Achieving group and community fairness in federated learning via post-processing, 2024. URL `https://arxiv.org/abs/2405.17782`.

Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

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*, pages 214–226, 2012.

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, pages 7494–7502, 2023.

Solenne Gaucher, Nicolas Schreuder, and Evgenii Chzhen. Fair learning with wasserstein barycenters for non-decomposable performance measures. In *International Conference on Artificial Intelligence and Statistics*, pages 2436–2459. PMLR, 2023.

Faisal Hamman and Sanghamitra Dutta. Demystifying local and global fairness trade-offs in federated learning using partial information decomposition. *arXiv preprint arXiv:2307.11333*, 2023.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In *Uncertainty in artificial intelligence*, pages 862–872. PMLR, 2020.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Joon Sik Kim, Jiahao Chen, and Ameet Talwalkar. Fact: A diagnostic for group fairness trade-offs. In *International Conference on Machine Learning*, pages 5264–5274. PMLR, 2020.

Thomas C. Landgrebe and Robert P. Duin. Efficient multiclass roc approximation by decomposition via confusion matrix perturbation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):810–822, May 2008. doi: 10.1109/TPAMI.2007.70740.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Disha Makhija, Xing Han, Joydeep Ghosh, and Yejin Kim. Achieving fairness across local and global models in federated learning. *arXiv preprint arXiv:2406.17102*, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

Aditya Krishna Menon and Robert C. Williamson. The cost of fairness in binary classification. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 107–118. PMLR, 23–24 Feb 2018. URL `https://proceedings.mlr.press/v81/menon18a.html`.

Jerzy Neyman and Egon Sharpe Pearson. Ix. on the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 231:289–337, 1933.

Sharnil Pandya, Gautam Srivastava, Rutvij Jhaveri, M Rajasekhara Babu, Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Spyridon Mastorakis, Md Jalil Piran, and Thippa Reddy Gadekallu. Federated learning for smart cities: A comprehensive survey. *Sustainable Energy Technologies and Assessments*, 55:102987, 2023.

Basheer Qolomany, Kashif Ahmad, Ala Al-Fuqaha, and Junaid Qadir. Particle swarm optimized federated learning for industrial iot and smart city services. In *GLOBECOM 2020-2020 IEEE Global Communications Conference*, pages 1–6. IEEE, 2020.

Sivan Sabato and Elad Yom-Tov. Bounding the fairness and accuracy of classifiers from population statistics. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8316–8325. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/sabato20a.html`.

Teresa Salazar, Miguel Fernandes, Helder Araújo, and Pedro Henriques Abreu. Fair-fate: Fair federated learning with momentum. In *International Conference on Computational Science*, pages 524–538. Springer, 2023.

Zeyu Tang and Kun Zhang. Attainability and optimality: The equalized odds fairness revisited. In *Conference on Causal Learning and Reasoning*, pages 754–786. PMLR, 2022.

Ye Tian and Yang Feng. Neyman-pearson multi-class classification via cost-sensitive learning. *Journal of the American Statistical Association*, (just-accepted):1–23, 2024.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.

Ruicheng Xian and Han Zhao. Efficient post-processing for equal opportunity in fair multi-class classification, 2023. URL `https://openreview.net/forum?id=zKjSmbYFZe`.

Ruicheng Xian and Han Zhao. Optimal group fair classifiers from linear post-processing. *arXiv preprint arXiv:2405.04025*, 2024.

Ruicheng Xian, Lang Yin, and Han Zhao. Fair and optimal classification via post-processing. In *International Conference on Machine Learning*, pages 37977–38012. PMLR, 2023.

Xianli Zeng, Guang Cheng, and Edgar Dobriban. Minimax optimal fair classification with bounded demographic disparity. *arXiv preprint arXiv:2403.18216*, 2024.

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

Han Zhao and Geoffrey J Gordon. Inherent tradeoffs in learning fair representations. *The Journal of Machine Learning Research*, 23(1):2527–2552, 2022.

Meiyu Zhong and Ravi Tandon. Intrinsic fairness-accuracy tradeoffs under equalized odds. *arXiv preprint arXiv:2405.07393*, 2024.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   The mathematical settings are in Section 3
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   The time complexity of the implementation is analyzed in Section 5
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
   The code is included in supplementary of the paper submission.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes]
   All proofs are in Appendix.
   (c) Clear explanations of any assumptions. [Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   The code is included in supplementary of the paper submission.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   Those details are in Appendix
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
   The GPU is talked in Experiments part of this paper.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

The Cost of Local and Global Fairness in Federated Learning: Supplementary Materials

A The Parameters of Convex Program in Proposition 5.1

This section provides the parameters of the convex program in Proposition 5.1. We first define the following statistics for the client group D:

$$\begin{aligned}
p^y_{ac} &= \Pr_D(Y = y, A = a, C = c) \\
\alpha^y_a &= \Pr_D(Y = y, A = a)
\end{aligned}$$

It is convenient to define the following matrix based on these statistics:

$$\begin{aligned}
\mathbf{c}^T_{ac} &= \begin{bmatrix} -p^1_{ac} & -p^2_{ac} & \cdots & -p^N_{ac} \end{bmatrix} \in \mathbb{R}^N \\
\mathbf{n}^i_{ac} &= \frac{1}{\alpha^i_a}\mathbf{E}_i\mathbf{c}_{ac} \in \mathbb{R}^N \\
\mathbf{m}^i &= \mathbf{e}^i \in \mathbb{R}^N \\
\mathbf{N}_i &= \begin{bmatrix} (\mathbf{n}^1_{0i})^T & -(\mathbf{n}^1_{1i})^T \\ (\mathbf{n}^2_{0i})^T & -(\mathbf{n}^2_{1i})^T \\ \vdots & \vdots \\ (\mathbf{n}^N_{0i})^T & -(\mathbf{n}^N_{1i})^T \end{bmatrix} \in \mathbb{R}^{N\times 2N} \\
\mathbf{M} &= \begin{bmatrix} (\mathbf{m}^1)^T & -(\mathbf{m}^1)^T \\ (\mathbf{m}^2)^T & -(\mathbf{m}^2)^T \\ \vdots & \vdots \\ (\mathbf{m}^N)^T & -(\mathbf{m}^N)^T \end{bmatrix} \in \mathbb{R}^{N\times 2N}
\end{aligned}$$

where $\mathbf{E}_i \in \mathbb{R}^{N\times N}$ is a matrix with the element at (i, i) (row i, column i) equal to 1 and all other elements are 0. $\mathbf{e}^i \in \mathbb{R}^N$ is the elementary basis vector in Euclidean space, i.e., $e^1 = [1, 0, 0, \ldots, 0]^T$.

The variables of the convex program, which represent true positives of the outcome predictor are:

$$\mathbf{z}^T_{ac} = \begin{bmatrix} z^1_{ac} & z^2_{ac} & \cdots & z^N_{ac} \end{bmatrix} \in \mathbb{R}^N$$

The convex program (8) is:

$$\begin{array}{ll}
\text{minimize:} & \mathbf{c}^T\mathbf{z} \\
\text{with respect to:} & \mathbf{z} \in \mathbb{R}^{2NK} \\
\text{subject to:} & -\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b} \\
& \mathbf{z}_{ac} \in D_{ac}
\end{array} \tag{17}$$

where $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ are composed of matrices we defined above:

$$\mathbf{c}^T = \begin{bmatrix} \mathbf{c}^T_{01} & \mathbf{c}^T_{11} & \mathbf{c}^T_{02} & \mathbf{c}^T_{12} & \cdots & \mathbf{c}^T_{0K} & \mathbf{c}^T_{1K} \end{bmatrix} \in \mathbb{R}^{2NK}$$

$$\mathbf{z}^T = \begin{bmatrix} \mathbf{z}^T_{01} & \mathbf{z}^T_{11} & \mathbf{z}^T_{02} & \mathbf{z}^T_{12} & \cdots & \mathbf{z}^T_{0K} & \mathbf{z}^T_{1K} \end{bmatrix} \in \mathbb{R}^{2NK}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \cdots & \mathbf{N}_K \\ \mathbf{M} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{M} \end{bmatrix} \in \mathbb{R}^{(N(K+1))\times 2NK}$$

$$\mathbf{b}^T = [\epsilon_0, \dots, \epsilon_0, \epsilon_1, \dots, \epsilon_1, \dots, \epsilon_K, \dots, \epsilon_K] \in \mathbb{R}^{N(K+1)}$$

B The Parameters of Simplex $\widehat{D_{ac}}$

The N-dimensional polytope $\widehat{D_{ac}}$ can be defined using (N+1) inequalities. Any single point $\mathbf{u} \in \mathbb{R}^N$ lies in the D_{ac} must have:

$$\mathbf{K}_{ac}\mathbf{u} \leq \mathbf{l}_{ac} \tag{18}$$

The true positives for predictor $\widetilde{Y}_{\mathbf{1}}$ that takes value of Eq. 10 are denoted as $\mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}})$.

then,

$$\mathbf{K}_{ac} = \begin{bmatrix} -1 & -1 & -1 & \cdots & -1 \\ 1 - \sum_{i\in\mathcal{Y}, i\neq 1} \mathrm{TP}^i_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}}) & \cdots & \mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}}) \\ \mathrm{TP}^2_{ac}(\widetilde{Y}_{\mathbf{1}}) & 1 - \sum_{i\in\mathcal{Y}, i\neq 2} \mathrm{TP}^i_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^2_{ac}(\widetilde{Y}_{\mathbf{1}}) & \cdots & \mathrm{TP}^2_{ac}(\widetilde{Y}_{\mathbf{1}}) \\ \mathrm{TP}^3_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^3_{ac}(\widetilde{Y}_{\mathbf{1}}) & 1 - \sum_{i\in\mathcal{Y}, i\neq 3} \mathrm{TP}^i_{ac}(\widetilde{Y}_{\mathbf{1}}) & \cdots & \mathrm{TP}^3_{ac}(\widetilde{Y}_{\mathbf{1}}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{TP}^N_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^N_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^N_{ac}(\widetilde{Y}_{\mathbf{1}}) & \cdots & 1 - \sum_{i\in\mathcal{Y}, i\neq N} \mathrm{TP}^i_{ac}(\widetilde{Y}_{\mathbf{1}}) \end{bmatrix} \in \mathbb{R}^{(N+1)\times N}$$

$$\mathbf{l}_{ac} = \begin{bmatrix} -1 & \mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^2_{ac}(\widetilde{Y}_{\mathbf{1}}) & \mathrm{TP}^3_{ac}(\widetilde{Y}_{\mathbf{1}}) & \cdots & \mathrm{TP}^N_{ac}(\widetilde{Y}_{\mathbf{1}}) \end{bmatrix}^T \in \mathbb{R}^{N+1}$$

C The Parameters of LAE in Proposition 5.2

The true positives for predictor $\widetilde{Y}_{\mathbf{1}}$ that takes value of Eq. 10 are $\mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}})$ and $\mathbf{z}_{ac}$ are the solution of the LP (12). The parameters of LAE (13): $\mathbf{G}_{ac}\boldsymbol{\beta}_{ac} = \boldsymbol{\gamma}_{ac}$ are:

$$\mathbf{G}_{ac} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ \mathrm{TP}^1_{ac}(\widetilde{Y}_{\mathbf{1}}) & 1 & 0 & \cdots & 0 & 0 \\ \mathrm{TP}^2_{ac}(\widetilde{Y}_{\mathbf{1}}) & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathrm{TP}^{N-1}_{ac}(\widetilde{Y}_{\mathbf{1}}) & 0 & 0 & \cdots & 1 & 0 \\ \mathrm{TP}^N_{ac}(\widetilde{Y}_{\mathbf{1}}) & 0 & 0 & \cdots & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\gamma}_{ac} = \begin{bmatrix} 1 \\ \mathbf{z}_{ac} \end{bmatrix} \tag{19}$$

D Extension to Other Fairness Metrics (Equal Opportunity and Statistical Parity)

D.1 Equal Opportunity

Equal Opportunity is a special case of Equalized Odds. It requires equal true positive rates between the sensitive and non-sensitive for a given class y, typically $y = 1$, rather than for all classes. The definitions of local and global Equal Opportunity are as follows:

Definition D.1 *Let $\epsilon_0 \in [0, 1]$. The outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ for client group D satisfies ϵ_0-**global equal opportunity** if*

$$\begin{aligned}&|\mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = 1 \,|\, Y = 1, A = 1 \right\} \\ &-\mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = 1 \,|\, Y = 1, A = 0 \right\} | \le \epsilon_0.\end{aligned} \tag{20}$$

Definition D.2 *($\boldsymbol{\epsilon_l}$-Local Fairness) Let $\boldsymbol{\epsilon}_l = [\epsilon_1, \cdots, \epsilon_K] \in [0, 1]^K$. The outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ satisfies $\boldsymbol{\epsilon}_l$-**local equal opportunity** if*

$$\begin{aligned}&|\mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = 1 \,|\, Y = 1, A = 1, C = c \right\} \\ &-\mathrm{Pr}_D \left\{ \widetilde{Y}(X, A, C) = 1 \,|\, Y = 1, A = 0, C = c \right\} | \le \epsilon_c.\end{aligned} \tag{21}$$

The LP for Equal Opportunity is a special case of LP (12), where only the constraints for class $y = 1$ are retained. The LP for Equal Opportunity is:

$$\begin{aligned} &\text{minimize:} && \mathbf{c}^T \mathbf{z} \\ &\text{with respect to:} && \mathbf{z} \in \mathbb{R}^{2NK} \\ &\text{subject to:} && -\mathbf{b} \le \mathbf{A}\mathbf{z} \le \mathbf{b} \\ & && \mathbf{K}_{ac}\mathbf{z}_{ac} \le \mathbf{l}_{ac} \end{aligned} \tag{22}$$

with

$$\begin{aligned} \mathbf{c}^T &= \begin{bmatrix} \mathbf{c}_{01}^T & \mathbf{c}_{11}^T & \mathbf{c}_{02}^T & \mathbf{c}_{12}^T & \cdots & \mathbf{c}_{0K}^T & \mathbf{c}_{1K}^T \end{bmatrix} \in \mathbb{R}^{2NK} \\ \mathbf{z}^T &= \begin{bmatrix} \mathbf{z}_{01}^T & \mathbf{z}_{11}^T & \mathbf{z}_{02}^T & \mathbf{z}_{12}^T & \cdots & \mathbf{z}_{0K}^T & \mathbf{z}_{1K}^T \end{bmatrix} \in \mathbb{R}^{2NK} \end{aligned}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{N}_1 & \mathbf{N}_2 & \cdots & \mathbf{N}_K \\ \mathbf{M} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{M} \end{bmatrix} \in \mathbb{R}^{(K+1) \times 2NK}$$

$$\mathbf{b}^T = [\epsilon_0, \epsilon_1, \cdots, \epsilon_K] \in \mathbb{R}^{(K+1)}$$

The parameters are the same as those for Equalized Odds in Appendix A, except for Equal Opportunity, where $\mathbf{N}_i$ and $\mathbf{M}$ are defined as:

$$\begin{aligned}\mathbf{N}_i &= \begin{bmatrix} (\mathbf{n}_{0i}^1)^T & -(\mathbf{n}_{1i}^1)^T \end{bmatrix} \in \mathbb{R}^{1\times 2N} \\ \mathbf{M} &= \begin{bmatrix} (\mathbf{m}^1)^T & -(\mathbf{m}^1)^T \end{bmatrix} \in \mathbb{R}^{1\times 2N}\end{aligned}$$

$\mathbf{K}_{ac}, \mathbf{l}_{ac}$ is the same as the LP (12) for Equalized Odds, which are in Appendix B.

D.2 Statistical Parity

Definition D.3 *Let $\epsilon_0 \in [0,1]$, the outcome predictor $\widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$ for client group D satisfies ϵ_0-**global statistical parity** if for all $y\in\mathcal{Y}$,*

$$\begin{aligned}&|\mathrm{Pr}_D\left\{\widetilde{Y}(X,A,C)=y\,|\;A=1\right\}\\ &-\mathrm{Pr}_D\left\{\widetilde{Y}(X,A,C)=y\,|\;A=0\right\}|\le\epsilon_0.\end{aligned}\tag{23}$$

Definition D.4 *($\boldsymbol{\epsilon_l}$-Local Fairness) Let $\boldsymbol{\epsilon}_l=[\epsilon_1\cdots,\epsilon_K]\in[0,1]^K$, the outcome predictor $\widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$ satisfies $\boldsymbol{\epsilon}_l$-**local statistical parity** if*

$$\begin{aligned}&|\mathrm{Pr}_D\left\{\widetilde{Y}(X,A,C)=y\,|\,Y=y,A=1,C=c\right\}\\ &-\mathrm{Pr}_D\left\{\widetilde{Y}(X,A,C)=y\,|\,Y=y,A=0,C=c\right\}|\le\epsilon_c.\end{aligned}\tag{24}$$

The variables we use to characterize the LP for statistical parity differ from those used for Equalized Odds and Equal Opportunity. Statistical Parity requires that the predictor's positive rate be the same for both sensitive and non-sensitive groups. Therefore, we need to focus on both true positives and false positives for a given class.

In the statistical parity setting, the variables we use to characterize the outcome predictor $\widetilde{Y} : \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$ are:

$$z_{ac}^{yj}=\mathrm{Pr}(\widetilde{Y}=y|\widetilde{Y}_{\mathbf{1}}=j,A=a,C=c)\tag{25}$$

The value of $\widetilde{Y}$ only depends on the output of the predictor $\widetilde{Y}_{\mathbf{1}}$ that is defined in (10), client label c and group a.

We find it is convenient to define the following probability of client group D for formulating the LP:

$$\begin{aligned}u_{ac}^{yj}=&\mathrm{Pr}_D(Y=y,\widetilde{Y}_{\mathbf{1}}=j,A=a,C=c)\\ u_{ac}^{j}=&\mathrm{Pr}_D(\widetilde{Y}_{\mathbf{1}}=j,A=a,C=c)\\ u_{ac}=&\mathrm{Pr}_D(A=a,C=c)\\ u_a=&\mathrm{Pr}_D(A=a)\end{aligned}$$

The LP for statistical Parity is:

$$\begin{aligned}&\text{maximize:} && \sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{Y}}\sum_{y\in\mathcal{Y}}u_{ac}^{yj}z_{ac}^{yj}\\ &\text{with respect to:} && z_{ac}^{yj}\in\mathbb{R},\forall y,j\in\mathcal{Y},a\in\mathcal{A},c\in\mathcal{C}\\ &\text{subject to:} && \forall y\in\mathcal{Y},-\epsilon_0\le\sum_{c\in\mathcal{C}}\sum_{j\in\mathcal{Y}}\frac{z_{0c}^{yj}\cdot u_{0c}^{j}}{u_0}-\sum_{c\in\mathcal{C}}\sum_{j\in\mathcal{Y}}\frac{z_{1c}^{yj}\cdot u_{1c}^{j}}{u_1}\le\epsilon_0\\ &&& \forall y\in\mathcal{Y},c\in\mathcal{C},-\epsilon_c\le\sum_{j\in\mathcal{Y}}\frac{z_{0c}^{yj}\cdot u_{0c}^{j}}{u_{0c}}-\sum_{j\in\mathcal{Y}}\frac{z_{1c}^{yj}\cdot u_{1c}^{j}}{u_{1c}}\le\epsilon_c\\ &&& \forall c\in\mathcal{C},a\in\mathcal{A},j\in\mathcal{Y},\sum_{y\in\mathcal{Y}}z_{ac}^{yj}=1\\ &&& \forall y,j\in\mathcal{Y},a\in\mathcal{A},c\in\mathcal{C},0\le z_{ac}^{yj}\le1\end{aligned}\tag{26}$$

Then, the outcome predictor $\widetilde{Y}: \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ that takes values:

$$\widetilde{Y}_{\mathbf{1}} = j, A = a, C = c : \widetilde{Y} = y \text{ with probability } z_{ac}^{yj} \tag{27}$$

satisfies local and global Statistical Parity. The proof of this assertion is given below.

Proof: We first show that the outcome predictor that takes value of (27) satisfies global statistical parity:

The first constraint of LP (26) is, $\forall y \in \mathcal{Y}$:

$$\begin{aligned}
&-\epsilon_0 \le \sum_{c\in\mathcal{C}}\sum_{j\in\mathcal{Y}} \frac{z_{0c}^{yj} \cdot u_{0c}^{j}}{u_0} - \sum_{c\in\mathcal{C}}\sum_{j\in\mathcal{Y}} \frac{z_{1c}^{yj} \cdot u_{1c}^{j}}{u_1} \le \epsilon_0 \\
\iff &-\epsilon_0 \le \sum_{c\in\mathcal{C}}\sum_{j\in\mathcal{Y}} \frac{\Pr_D(\widetilde{Y} = y | \widetilde{Y}_{\mathbf{1}} = j, C = c, A = 0) \cdot \Pr_D(\widetilde{Y}_{\mathbf{1}} = j, A = 0, C = c)}{\Pr_D(A = 0)} \\
&\qquad - \sum_{c\in\mathcal{C}}\sum_{j\in\mathcal{Y}} \frac{\Pr_D(\widetilde{Y} = y | \widetilde{Y}_{\mathbf{1}} = j, C = c, A = 1) \cdot \Pr_D(\widetilde{Y}_{\mathbf{1}} = j, A = 1, C = c)}{\Pr_D(A = 1)} \le \epsilon_0 \\
\iff &-\epsilon_0 \le \frac{\Pr_D(\widetilde{Y} = y, A = 0)}{\Pr_D(A = 0)} - \frac{\Pr_D(\widetilde{Y} = y, A = 1)}{\Pr_D(A = 1)} \le \epsilon_0 \\
\iff &-\epsilon_0 \le \Pr_D(\widetilde{Y} = y | A = 0) - \Pr_D(\widetilde{Y} = y | A = 1) \le \epsilon_0
\end{aligned} \tag{28}$$

The first constraint of the LP is equivalent to the global statistical parity. The outcome predictor that satisfies the LP will satisfies the global ϵ_0- statistical parity.

Then, we show that the outcome predictor (27) satisfies local statistical parity:

The second constraint of the LP (26) is, $\forall y \in \mathcal{Y}, c \in \mathcal{C}$:

$$\begin{aligned}
&-\epsilon_c \le \sum_{j\in\mathcal{Y}} \frac{z_{0c}^{yj} \cdot u_{0c}^{j}}{u_{0c}} - \sum_{j\in\mathcal{Y}} \frac{z_{1c}^{yj} \cdot u_{1c}^{j}}{u_{1c}} \le \epsilon_c \\
\iff &-\epsilon_c \le \sum_{j\in\mathcal{Y}} \frac{\Pr_D(\widetilde{Y} = y | \widetilde{Y}_{\mathbf{1}} = j, C = c, A = 0) \cdot \Pr_D(\widetilde{Y}_{\mathbf{1}} = j, A = 0, C = c)}{\Pr_D(A = 0, C = c)} \\
&\qquad - \sum_{j\in\mathcal{Y}} \frac{\Pr_D(\widetilde{Y} = y | \widetilde{Y}_{\mathbf{1}} = j, C = c, A = 1) \cdot \Pr_D(\widetilde{Y}_{\mathbf{1}} = j, A = 1, C = c)}{\Pr_D(A = 1, C = c)} \le \epsilon_c \\
\iff &-\epsilon_c \le \frac{\Pr_D(\widetilde{Y} = y, A = 0, C = c)}{\Pr_D(A = 0, C = c)} - \frac{\Pr_D(\widetilde{Y} = y, A = 1, C = c)}{\Pr_D(A = 1, C = c)} \le \epsilon_c \\
\iff &-\epsilon_c \le \Pr_D(\widetilde{Y} = y | A = 0, C = c) - \Pr_D(\widetilde{Y} = y | A = 1, C = c) \le \epsilon_c
\end{aligned} \tag{29}$$

The second constraints of the LP is equivalent to local statistical parity, thus, it will lead a outcome predictor that satisfies $\boldsymbol{\epsilon}_l-$ local statistical parity.

Since the variables z_{ac}^{yj} represent the probability, it satisfies the third and fourth constraints.

The objective function we maximize is:

$$\begin{aligned}
&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{Y}}\sum_{y\in\mathcal{Y}} u_{ac}^{yj} z_{ac}^{yj} \\
=&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{Y}}\sum_{y\in\mathcal{Y}} \Pr(\widetilde{Y}=y|\widetilde{Y}_{\mathbf{1}}=j, A=a, C=c)\cdot \Pr_D(Y=y, \widetilde{Y}_{\mathbf{1}}=j, A=a, C=c) \\
=&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{Y}}\sum_{y\in\mathcal{Y}} \Pr(\widetilde{Y}=y|\widetilde{Y}_{\mathbf{1}}=j, A=a, C=c, Y=y)\cdot \Pr_D(Y=y, \widetilde{Y}_{\mathbf{1}}=j, A=a, C=c) \\
=&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{Y}}\sum_{y\in\mathcal{Y}} \Pr(\widetilde{Y}=y, \widetilde{Y}_{\mathbf{1}}=j, A=a, C=c, Y=y) \\
=&\sum_{y\in\mathcal{Y}} \Pr(\widetilde{Y}=y, Y=y) \\
=&\Pr_D(\widetilde{Y}=Y)
\end{aligned} \tag{30}$$

The second equation holds because $\widetilde{Y}$ depends only on $\widetilde{Y}_{\mathbf{1}}$, A, and C. $\widetilde{Y}$ is independent of Y given $\widetilde{Y}_{\mathbf{1}}$, A, and C. The objective we maximize represents the accuracy of the outcome predictor.

Thus, the outcome predictor in (27) satisfies local and global fairness and maximizes the accuracy under our LP.

E Theoretical Proofs

E.1 Proof of Proposition 4.3

Proposition 4.3 Consider client c and group a, let $\{\widetilde{Y}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}}$ be the set of all derived outcome predictors, and let $\{\phi_g\}_{g=1}^{N-1}, \in [0,1]^N$ be a set of specified values of true positives. If $\widetilde{Y}^*: \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y}$ is the solution to the following optimization problem:

$$\begin{aligned}
&\text{maximize} && \mathrm{TP}_{ac}^{N}(\widetilde{Y}) \\
&\text{with respect to} && \widetilde{Y}: \mathcal{X}\times\mathcal{A}\times\mathcal{C}\to\mathcal{Y} \\
&\text{subject to} && \forall g\in\{1,2,\cdots,N-1\}, \\
&&& \mathrm{TP}_{ac}^{g}(\widetilde{Y}) = \phi_g
\end{aligned} \tag{31}$$

then $\widetilde{Y}^* \in \{\widetilde{Y}_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}\in\mathbb{R}^N_{\geq 0}}$.

Proof: Our proposition is a special case of the NP multi-class classification problem defined in [Tian and Feng, 2024], where the number of constraints on true positives is chosen to be an arbitrary value less than N, and the objective is to maximize the weighted sum of the remaining true positives. In our case, the number of constraints is fixed at $N-1$, and we maximize the true positive rate for class N. Our proof follows the proof provided in [Tian and Feng, 2024].

We first consider the Lagrange function associated with problem (31):

$$L(\widetilde{Y},\boldsymbol{\lambda}) = \mathrm{TP}_{ac}^{N}(\widetilde{Y}) + \sum_{g=1}^{N-1}\lambda_g\left(\mathrm{TP}_{ac}^{g}(\widetilde{Y}) - \phi_g\right)$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_{N-1}] \in \mathbb{R}^{N-1}$, with λ_g as the Lagrange multiplier corresponding to the g-th constraint: $\mathrm{TP}_{ac}^{g}(\widetilde{Y}) = \phi_g$.

The dual problem of (31) can be written as:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}^{N-1}} \min_{\widetilde{Y}} L(\widetilde{Y},\boldsymbol{\lambda}) \tag{32}$$

Problem (32) can be solved analytically. We consider $\widetilde{Y}^{\star}_{\boldsymbol{\lambda}} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ that minimizes the Lagrange function $L(\widetilde{Y}, \boldsymbol{\lambda})$ for any $\boldsymbol{\lambda} \in \mathbb{R}^{N-1}$. We find it convenient to represent the following probability as:

$$s^{y}_{ac} = \Pr_D(Y = y \mid A = a, C = c)$$

The term $\mathrm{TP}^{y}_{ac}(\widetilde{Y})$ is equal to:

$$\begin{aligned}
\mathrm{TP}^{y}_{ac}(\widetilde{Y}) &= \mathbb{E}_{\Pr_{X|Y,A,C}}[\mathbf{1}(\widetilde{Y} = y)] \\
&= \mathbb{E}_{\Pr_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot r_y(X, a, c)] \cdot \frac{\Pr_D(A = a, C = c)}{\Pr_D(Y = y, A = a, C = c)} \\
&= \mathbb{E}_{\Pr_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot r_y(X, a, c)] \cdot \frac{1}{\Pr_D(Y = y \mid A = a, C = c)} \\
&= \mathbb{E}_{\Pr_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot r_y(X, a, c)] \cdot \frac{1}{s^{y}_{ac}}
\end{aligned} \tag{33}$$

where $r_y(x, a, c)$ is the y-th element of the Bayesian optimal score function in Definition 4.1. Substituting Eq. (33) into the Lagrange function, we can express it as:

$$\begin{aligned}
L(\widetilde{Y}, \boldsymbol{\lambda}) = &\sum_{g=1}^{N-1} \frac{\lambda_g}{s^{g}_{ac}} \mathbb{E}_{\Pr_{X|A,C}}[\mathbf{1}(\widetilde{Y} = g) \cdot r_g(X, a, c)] \\
&+ \mathbb{E}_{\Pr_{X|A,C}}[\mathbf{1}(\widetilde{Y} = N) \cdot r_N(X, a, c)] - \sum_{g=1}^{N-1} \lambda_g \phi_g
\end{aligned}$$

For any given $\boldsymbol{\lambda}$, the predictor $\widetilde{Y}^{*}_{\boldsymbol{\lambda}}$ that minimizes the Lagrange function $L(\widetilde{Y}, \boldsymbol{\lambda})$ takes the values:

$$\begin{aligned}
\widetilde{Y}^{*}_{\boldsymbol{\lambda}} = g, \quad &\text{if } \frac{\lambda_g}{s^{g}_{ac}} r_g(x, a, c) = \max\left\{\frac{\lambda_1}{s^{1}_{ac}} r_1(x, a, c), \ldots, r_N(x, a, c)\right\} \\
\text{or, } \widetilde{Y}^{*}_{\boldsymbol{\lambda}} = N, \quad &\text{if } r_N(x, a, c) = \max\left\{\frac{\lambda_1}{s^{1}_{ac}} r_1(x, a, c), \ldots, r_N(x, a, c)\right\}
\end{aligned} \tag{34}$$

We then consider the optimal $\boldsymbol{\lambda}^{*}$ that maximizes $L(\widetilde{Y}^{*}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda})$. The first-order optimality conditions are:

$$\begin{aligned}
\forall g \in \{1, 2, \ldots, N-1\}, \quad &\left.\frac{\partial L(\widetilde{Y}^{*}_{\boldsymbol{\lambda}}, \boldsymbol{\lambda})}{\partial \lambda_g}\right|_{\boldsymbol{\lambda}^{*}} = 0 \\
\iff \quad &\mathrm{TP}^{g}_{ac}(\widetilde{Y}_{\boldsymbol{\lambda}^{*}}) = \phi_g
\end{aligned} \tag{35}$$

Consider the Bayesian optimal score function $r_y(X, A, C)$ with continuous probability density. Then there exist $\boldsymbol{\lambda}^{*}$ that satisfies Eq. (35). The solution must satisfy:

$$\frac{\lambda^{*}_{g}}{s^{g}_{ac}} \geq 0, \quad \forall g \in \{1, \ldots, N-1\}$$

If $\frac{\lambda_g}{s^{g}_{ac}} < 0$, the predictor will always output N, violating constraints when $\phi_g > 0$.

Under strong duality [Tian and Feng, 2024], the solution of the dual problem (32) is equivalent to the solution of the primal problem (31). Based on [Tian and Feng, 2024], in this problem, strong duality holds. Therefore, the solution of problem (31) is $\widetilde{Y}^{\star}_{\boldsymbol{\lambda}^{*}}$, which is a derived outcome predictor as defined in Definition 4.2, with:

$$\theta_g = \frac{\lambda^{*}_{g}}{s^{g}_{ac}}, \quad \forall g \in \{1, \ldots, N-1\}, \quad \text{and } \theta_N = 1.$$

E.2 Proof of Proposition 4.6

Proposition 4.6: Let D_{ac} be the region defined in Def. 4.5. Then, D_{ac} is a convex set.

Proof: Consider any $\mathbf{r}_0, \mathbf{r}_1 \in [0,1]^N$, from Def. 4.5, if $\mathbf{r}_0, \mathbf{r}_1 \in D_{ac}$, then:

$$\begin{aligned} \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{r}_0 \leq \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}), \quad \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N, \\ \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{r}_1 \leq \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}), \quad \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N. \end{aligned} \tag{36}$$

Since $\lambda \in [0,1]$, we must have:

$$\begin{aligned} \mathbf{v}_{\boldsymbol{\theta}}^T (\lambda \mathbf{r}_0) \leq \mathbf{v}_{\boldsymbol{\theta}}^T (\lambda \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})), \quad \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N, \\ \mathbf{v}_{\boldsymbol{\theta}}^T ((1-\lambda) \mathbf{r}_1) \leq \mathbf{v}_{\boldsymbol{\theta}}^T ((1-\lambda) \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})), \quad \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N, \\ \Rightarrow \mathbf{v}_{\boldsymbol{\theta}}^T (\lambda \mathbf{r}_0 + (1-\lambda) \mathbf{r}_1) \leq \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}), \quad \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N, \end{aligned} \tag{37}$$

which implies: $\lambda \mathbf{r}_0 + (1-\lambda)\mathbf{r}_1 \in D_{ac}$.

We choose $\mathbf{r}_0, \mathbf{r}_1 \in D_{ac}$ arbitrarily. Thus, for all $\mathbf{r}_0, \mathbf{r}_1 \in D_{ac}$ and $\lambda \in [0,1]$, it holds that $\lambda \mathbf{r}_0 + (1-\lambda)\mathbf{r}_1 \in D_{ac}$. Therefore, D_{ac} is a convex set.

E.3 Proof of Proposition 4.7

Proposition 4.7: Let D_{ac} be the region as defined in Def.4.5. For any predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$, let the point representing true positives of $\widetilde{Y}$ be: $\mathbf{TP}_{ac}(\widetilde{Y}) = [\mathrm{TP}_{ac}^y(\widetilde{Y}), \cdots, \mathrm{TP}_{ac}^N(\widetilde{Y})]^T$. Then, $\mathbf{TP}_{ac}(\widetilde{Y})$ lies in D_{ac}.

Proof: Proposition 4.7 states that for any predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$, the following condition must hold:

$$\begin{aligned} \mathbf{TP}_{ac}(\widetilde{Y}) \in D_{ac} &\iff \mathbf{TP}_{ac}(\widetilde{Y}) \in \bigcap_{\boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N} \left\{ \mathbf{x} \in [0,1]^N | \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{x} \leq \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}) \right\} \\ &\iff \forall \boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N, \quad \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}) \leq \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}}) \end{aligned} \tag{38}$$

To prove this, we consider the value of $\mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y})$:

$$\begin{aligned} \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}) &= \sum_{y \in \mathcal{Y}} \theta_y \mathrm{Pr}_D(Y = y \mid A = a, C = c) \mathrm{TP}_{ac}^y(\widetilde{Y}) \\ &= \sum_{y \in \mathcal{Y}} \theta_y \mathrm{Pr}_D(Y = y \mid A = a, C = c) \mathbb{E}_{\mathrm{Pr}_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot r_y(X, a, c)] \cdot \frac{1}{\mathrm{Pr}_D(Y = y \mid A = a, C = c)} \\ &= \sum_{y \in \mathcal{Y}} \theta_y \mathbb{E}_{\mathrm{Pr}_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot r_y(X, a, c)] \\ &= \sum_{y \in \mathcal{Y}} \mathbb{E}_{\mathrm{Pr}_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot \theta_y r_y(X, a, c)] \end{aligned} \tag{39}$$

The second equation is from Eq. (33) in Appendix E.1. Equation (39) achieves the maximum value if:

$$\widetilde{Y} = y, \quad \text{if } \theta_y r_y(x, a, c) = \max_{i=1}^N \theta_i r_i(x, a, c) \tag{40}$$

The predictor that takes the value of Eq. 40 is a derived outcome predictor $\widetilde{Y}_{\boldsymbol{\theta}}$ as defined in Def. 4.2. The derived outcome predictor $\widetilde{Y}_{\boldsymbol{\theta}}$ maximizes the value of $\mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y})$ for all $\boldsymbol{\theta} \in \mathbb{R}_{\geq 0}^N$. Therefore, any predictor must satisfy $\mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}) \leq \mathbf{v}_{\boldsymbol{\theta}}^T \mathbf{TP}_{ac}(\widetilde{Y}_{\boldsymbol{\theta}})$, which is equivalent to $\mathbf{TP}_{ac}(\widetilde{Y}) \in D_{ac}$.

E.4 Proof of Proposition 5.1

Proposition 5.1: Let the vector $\mathbf{z} \in \mathbb{R}^{2NK}$:

$$\mathbf{z}^T = \begin{bmatrix} \mathbf{z}_{01}^T & \mathbf{z}_{11}^T & \mathbf{z}_{02}^T & \mathbf{z}_{12}^T \cdots & \mathbf{z}_{0K}^T & \mathbf{z}_{1K}^T \end{bmatrix}$$

with

$$\mathbf{z}_{ac}^T = \begin{bmatrix} z_{ac}^1 & z_{ac}^2 & z_{ac}^3 & \cdots & z_{ac}^N \end{bmatrix} \in \mathbb{R}^N$$

satisfy the following convex program

$$\begin{aligned} &\text{minimize:} && \mathbf{c}^T\mathbf{z} \\ &\text{with respect to:} && \mathbf{z} \in \mathbb{R}^{2NK} \\ &\text{subject to:} && -\mathbf{b} \le \mathbf{A}\mathbf{z} \le \mathbf{b} \\ &&& \mathbf{z}_{ac} \in D_{ac}, \forall a \in \mathcal{A}, c \in \mathcal{C} \end{aligned} \tag{41}$$

then, the outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ that satisfies eq. (42) for all $y \in \mathcal{Y}, a \in \mathcal{A}, c \in \mathcal{C}$

$$\Pr(\widetilde{Y} = y | Y = y, A = a, C = c) = z_{ac}^y \tag{42}$$

is a $\boldsymbol{\epsilon}$-fair optimal outcome predictor.The optimal accuracy for a ϵ-fair outcome predictor is $-\mathbf{c}^T\mathbf{z}$.

Proof: We first show that the outcome predictor $\widetilde{Y}$ in Proposition 5.1 satisfies global Equalized Odds. The probability $\Pr_D(\widetilde{Y} = y|Y = y, A = a)$ can be extended as follows:

$$\begin{aligned} \Pr_D(\widetilde{Y} = y|Y = y, A = a) =& \sum_{c\in\mathcal{C}} \Pr_D(\widetilde{Y} = y, C = c|Y = y, A = a) \\ =& \sum_{c\in\mathcal{C}} \frac{\Pr_D(\widetilde{Y} = y|Y = y, A = a, C = c) \cdot \Pr_D(Y = y, A = a, C = c)}{\Pr_D(Y = y, A = a)} \\ =& \sum_{c\in\mathcal{C}} \frac{z_{ac}^y \cdot p_{ac}^y}{\alpha_a^y} \end{aligned} \tag{43}$$

where, $z_{ac}^y = \Pr_D(\widetilde{Y} = y|Y = y, A = a, C = c)$. As defined in Appendix A, p_{ac}^y, α_a^y are the probability for distribution D, $p_{ac}^y = \Pr_D(Y = y, A = a, C = c), \alpha_a^y = \Pr_D(Y = y, A = a)$.

The global *Equalized Odds Difference* of predictor $\widetilde{Y}$ for class y is defined as $\Pr_D(\widetilde{Y} = y|Y = y, A = 0) - \Pr_D(\widetilde{Y} = y|Y = y, A = 1)$, which can be expanded as:

$$\begin{aligned} &\Pr_D(\widetilde{Y} = y|Y = y, A = 0) - \Pr_D(\widetilde{Y} = y|Y = y, A = 1) \\ =& \sum_{c\in\mathcal{C}} \frac{z_{0c}^y \cdot p_{0c}^y}{\alpha_0^y} - \sum_{c\in\mathcal{C}} \frac{z_{1c}^y \cdot p_{1c}^y}{\alpha_1^y} \end{aligned} \tag{44}$$

The first N linear equations of $-\mathbf{b} \le \mathbf{A}\mathbf{z} \le \mathbf{b}$ are: $\forall y \in \mathcal{Y}$,

$$-\epsilon_0 \le \sum_{c\in\mathcal{C}} \mathbf{n}_{0c}^y - \sum_{c\in\mathcal{C}} \mathbf{n}_{1c}^y \le \epsilon_0 \tag{45}$$

Substituting the values of $\mathbf{n}_{0c}^y$ and $\mathbf{n}_{1c}^y$ into Eq. (45), Eq. (45) is:

$$\begin{aligned} &-\epsilon_0 \le \sum_{c\in\mathcal{C}} \frac{z_{0c}^y \cdot p_{0c}^y}{\alpha_0^y} - \sum_{c\in\mathcal{C}} \frac{z_{1c}^y \cdot p_{1c}^y}{\alpha_1^y} \le \epsilon_0 \\ \iff& -\epsilon_0 \le \Pr_D(\widetilde{Y} = y|Y = y, A = 0) - \Pr_D(\widetilde{Y} = y|Y = y, A = 1) \le \epsilon_0 \end{aligned} \tag{46}$$

The first N linear equations of $-\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b}$ are equivalent to global ϵ_0-global Equalized Odds. Thus, the predictor $\widetilde{Y}$ that satisfies the first N constraints of the convex program also satisfies ϵ_0-global Equalized Odds.

Next, we show that the outcome predictor $\widetilde{Y}$ from Proposition 5.1 satisfies local Equalized Odds.

The local *Equalized Odds Difference* of predictor $\widetilde{Y}$ for client c and class y is defined as $\Pr_D(\widetilde{Y} = y|Y = y, A = 0, C = c) - \Pr_D(\widetilde{Y} = y|Y = y, A = 1, C = c)$, which is:

$$\begin{aligned}&\Pr_D(\widetilde{Y} = y|Y = y, A = 0, C = c) - \Pr_D(\widetilde{Y} = y|Y = y, A = 1, C = c)\\ =&z_{0c}^y - z_{1c}^y\end{aligned} \tag{47}$$

The last NK equations of $-\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b}$ are: $\forall y \in \mathcal{Y}, c \in \mathcal{C}$,

$$\begin{aligned}&-\epsilon_c \leq z_{0c}^y - z_{1c}^y \leq \epsilon_c\\ \iff& -\epsilon_c \leq \Pr_D(\widetilde{Y} = y|Y = y, A = 0, C = c) - \Pr_D(\widetilde{Y} = y|Y = y, A = 1, C = c) \leq \epsilon_c\end{aligned} \tag{48}$$

The last NK linear equations of $-\mathbf{b} \leq \mathbf{A}\mathbf{z} \leq \mathbf{b}$ are equivalent to ϵ_c-local Equalized Odds. Thus, the predictor $\widetilde{Y}$ satisfies the last NK equations of the convex program, meaning it satisfies ϵ_c-local Equalized Odds.

The constraint $\mathbf{z}_{ac} \in D_{ac}$ is the condition that any outcome predictor must satisfy. (Proposition 4.7).

The objective function of the convex program is:

$$\begin{aligned}&\mathbf{c}^T\mathbf{z}\\ =&-\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{y\in\mathcal{Y}} z_{ac}^y p_{ac}^y\\ =&-\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{y\in\mathcal{Y}} \Pr_D(\widetilde{Y} = y|Y = y, A = a, C = c)\Pr_D(Y = y, A = a, C = c)\\ =&-\sum_{y\in\mathcal{Y}} \Pr_D(\widetilde{Y} = y, Y = y)\\ =&-\Pr_D(\widetilde{Y} = Y)\end{aligned}$$

The predictor $\widetilde{Y}$ that minimizes $\mathbf{c}^T\mathbf{z}$ corresponds to maximum accuracy. The maximum accuracy is $-\mathbf{c}^T\mathbf{z}$.

Overall, the predictor $\widetilde{Y}$ satisfies both ϵ_0-global Equalized Odds and ϵ_c-local Equalized Odds while maximizing accuracy. Therefore, the predictor $\widetilde{Y}$ that satisfies the convex program is an optimal $\boldsymbol{\epsilon}$-outcome predictor. The accuracy of the optimal $\boldsymbol{\epsilon}$-outcome predictor is $-\mathbf{c}^T\mathbf{z}$.

E.5 Proof of Proposition 5.2

Proposition 5.2: Let $\mathbf{z} \in \mathbb{R}^{2NK}$ be the solution of the LP (12)

$$\mathbf{z}^T \quad = \quad \begin{bmatrix} \mathbf{z}_{01}^T & \mathbf{z}_{11}^T & \mathbf{z}_{02}^T & \mathbf{z}_{12}^T \cdots & \mathbf{z}_{0K}^T & \mathbf{z}_{1K}^T \end{bmatrix}$$

and $\text{TP}_{ac}^y(\widetilde{Y}_{\mathbf{1}})$ be the true positive of the predictor defined in Eq. (10). For all $a \in \mathcal{A}, c \in \mathcal{C}$, let $\boldsymbol{\beta}_{ac}$ be the solution of the following linear algebraic equation (LAE),

$$\mathbf{G}_{ac}\boldsymbol{\beta}_{ac} = \boldsymbol{\gamma}_{ac} \tag{49}$$

where, the parameter $\mathbf{G}_{ac} \in \mathbb{R}^{(N+1)\times(N+1)}, \boldsymbol{\gamma}_{ac} \in \mathbb{R}^{N+1}$, are detailed in Appendix C. Then. the predictor $\widetilde{Y}_{\boldsymbol{\beta}_{ac}}$ that takes value,

$$\widetilde{Y}_{\boldsymbol{\beta}_{ac}}(x, a, c) = \begin{cases} \widetilde{Y}_{\mathbf{1}}(x, a, c), & \text{with the probability } \beta_{ac}^0 \\ y, & \text{with the probability } \beta_{ac}^y, \forall y \in \mathcal{Y} \end{cases} \tag{50}$$

is a fair outcome predictor. There always exists a unique set of parameters $\{\boldsymbol{\beta}_{ac}\}_{\mathcal{A},\mathcal{C}}$, where $\boldsymbol{\beta}_{ac} \in [0, 1]^{N+1}$ and $|\boldsymbol{\beta}_{ac}|_{\ell_1} = 1$ that satisfies the LAE.

Proof: The true positive of class y for the outcome predictor $\widetilde{Y}_{\boldsymbol{\beta}_{ac}} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$ that takes value of Eq. (50) is:

$$\mathrm{TP}^{y}_{ac}(\widetilde{Y}_{\boldsymbol{\beta}_{ac}}) = \mathrm{TP}^{y}_{ac}(\widetilde{Y}_{\mathbf{1}})\beta^{0}_{ac} + \beta^{y}_{ac} \tag{51}$$

Since $\boldsymbol{\beta}_{ac}$ is the solution of LAE (49), its elements satisfies that: $\forall y \in \mathcal{Y}$,

$$\begin{aligned} &\mathrm{TP}^{y}_{ac}(\widetilde{Y}_{\mathbf{1}})\beta^{0}_{ac} + \beta^{y}_{ac} = z^{y}_{ac} \\ \iff &\mathrm{TP}^{y}_{ac}(\widetilde{Y}_{\boldsymbol{\beta}_{ac}}) = z^{y}_{ac} \end{aligned} \tag{52}$$

The true positives of the predictor $\widetilde{Y}_{\boldsymbol{\beta}ac}$ for class y, client c, and group a are z^{y}_{ac}. Since $\{z^{y}_{ac}\}_{\mathcal{A},\mathcal{C},\mathcal{Y}}$ is the solution of LP (12), it satisfies the fairness constraints of LP (12). Thus, the predictor $\widetilde{Y}_{\boldsymbol{\beta}_{ac}}$ is a fair outcome predictor.

$\mathbf{G}_{ac}$ is a full-rank matrix for all clients and groups, thus, the LAE has a unique solution in all clients and groups.

E.6 $\widetilde{Y}_{\mathbf{1}}$ maximizes accuracy for distribution D

For any outcome predictor $\widetilde{Y} : \mathcal{X} \times \mathcal{A} \times \mathcal{C} \to \mathcal{Y}$, the accuracy for distribution D is:

$$\begin{aligned} &\mathrm{Pr}_{D}(\widetilde{Y} = Y) \\ =&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{y\in\mathcal{Y}} \mathrm{Pr}_{D}(\widetilde{Y} = y|Y = y, A = a, C = c)\mathrm{Pr}_{D}(Y = y, A = a, C = c) \\ =&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{y\in\mathcal{Y}} \mathrm{TP}^{y}_{ac}(\widetilde{Y})\mathrm{Pr}_{D}(Y = y, A = a, C = c) \\ =&\sum_{c\in\mathcal{C}}\sum_{a\in\mathcal{A}}\sum_{y\in\mathcal{Y}} \mathbb{E}_{\mathrm{Pr}_{X|A,C}}[\mathbf{1}(\widetilde{Y} = y) \cdot r_{y}(X, a, c)] \end{aligned} \tag{53}$$

The predictor that maximizes the accuracy is $\widetilde{Y}_{\mathbf{1}}$ that takes values of:

$$\widetilde{Y}_{\mathbf{1}} = y, \text{if: } r_{y}(x, a, c) = \max_{i\in\mathcal{Y}} r_{i}(x, a, c) \tag{54}$$

F Experimental Details

F.1 Data, Models, and Hyperparameters

We provide the models and hyperparameters of our implementation for each dataset. We run the experiments on our local Linux server, equipped with a 16-core 4.00 GHz AMD RYZEN Threadripper Pro 5955WX Processor. All code is implemented in TensorFlow, simulating a federated network with one server and several local clients.

For the Adult dataset, we randomly divide the data in each local client into three subsets: 60% for the training set, 20% for the validation set, and 20% for the test set. We first implement the `FedAvg` algorithm. For each communication round in `FedAvg`, the number of participating communities is set to $N = 2$. The number of local update epochs is set to $E = 1$, with a batch size of $B = 512$. The local models are logistic regression classifiers with two layers, containing 64 and 32 nodes, respectively. We use *ReLU* as the activation function for each hidden layer. These models are trained using the Adam optimizer with a learning rate of $\eta = 0.001$. We select the number of rounds that minimizes the validation loss . We construct and solve the LP (12) and LAE (13) using the validation data. Finally, we apply post-processing to the test dataset based on the solution from LAE and report its evaluation metrics.

For the PublicCoverage dataset, we similarly split the local dataset into 60% for training, 20% for validation, and 20% for testing. The number of participating communities for *FedAvg* is set to $N = 50$. The number of local update epochs remains at $E = 1$, with a batch size of $B = 256$. We follow the same model structure, optimization algorithm, and evaluation process as with the UCI Adult dataset and report the evaluation metrics.

For the HM 10000 dataset, we similarly split the local dataset into 60% for training, 20% for validation, and 20% for testing. The four diagnostic classes are: class 1 is Pre-cancerous and cancerous lesions, which include 'akiec', 'bcc', and 'mel'; class 2 is Benign lesions, which include 'bkl' and 'df'; class 3 is Nevus-like lesions, which include 'nv'; and class 4 is Vascular lesions, which include 'vasc'. We resize the images to $28 \times 28 \times 3$. The number of local update epochs is set to $E = 1$, with a batch size of $B = 32$. The local models are CNNs that consist of three convolutional layers (with 32, 64, and 128 filters, respectively), followed by global average pooling and two dense layers (with 128 and 32 nodes, respectively). The loss function used for the multi-class classification task is sparse categorical cross-entropy. The CNN models are trained using the Adam optimizer with a learning rate of $\eta = 0.0001$. We select the number of rounds that minimizes the validation loss. We construct the LP and LAE using the validation data, and finally, we apply post-processing to the test dataset based on the solution from the LAE and report its evaluation metrics.

F.2 Baselines

We introduce the baselines used in the experimental section. All baselines are in-processing techniques that modify the optimization algorithms in FL.

1. `FCFL` [Cui et al., 2021] is designed to achieve local fairness and performance consistency in FL. It achieves fairness by solving a constrained min-max program. We obtained our results by directly running their implementation at: https://github.com/cuis15/FCFL.

2. `Fair-Fate` [Salazar et al., 2023] is designed to achieve global fairness. It enforces fairness by adding a momentum term that helps to overcome the oscillations of non-fair gradients. Following the recommendations in [Salazar et al., 2023], the parameters of the algorithm $\{\lambda_0, \rho, \text{MAX}, \beta_0\}$ are set to $\{0.5, 0.05, 1, 0.99\}$ for the Adult dataset and $\{0.5, 0.05, 1, 0.9\}$ for ACSPublicCoverage.

3. `FairFed` [Ezzeldin et al., 2023] is designed to achieve global fairness. It enforces fairness by adaptively adjusting the aggregation weights of different clients based on their local fairness metrics. The parameter that controls the fairness budget for each update, β, is set to 1 for both the Adult and ACSPublicCoverage datasets.

4. `EquiFL` [Makhija et al., 2024] is designed to achieve both local and global fairness. It enforces fairness by adding a regularization term to the local loss function. The weight for fairness regularization, w, is set to 10^2 for the Adult dataset and 10^5 for ACSPublicCoverage.

F.3 Differential Privacy

The statistics computed and transmitted by the client c, as described in Eq.(15) of the paper, are:

$$\begin{aligned}
&\Pr_D(\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a \mid C = c) \\
&= \frac{\#\text{ of samples with } (\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a) \text{ in client } c}{\#\text{ of samples in client } c}, \\
&\Pr_D(Y = y, A = a \mid C = c) \\
&= \frac{\#\text{ of samples with } (Y = y, A = a) \text{ in client } c}{\#\text{ of samples in client } c}.
\end{aligned} \tag{55}$$

The statistics sent by the client c after applying the Laplace Mechanism are:

$$\begin{aligned}
&\Pr_D(\widetilde{Y}_{\mathbf{1}} = y, Y = y, A = a \mid C = c) + \text{Lap}(0|b_c), \\
&\Pr_D(Y = y, A = a \mid C = c) + \text{Lap}(0|b_c),
\end{aligned} \tag{56}$$

where $\text{Lap}(x|b) = \frac{1}{2b}e^{-\frac{|x|}{b}}$ is the density function of the Laplace distribution, and b_c is the scale parameter of the Laplace distribution. The larger b_c, the greater the variance of the added noise.

The statistics will satisfy ϵ-differential privacy [Dwork, 2006], if we set the scale parameter b_c as:

$$b_c = \frac{\Delta f_c}{\epsilon} \tag{57}$$

where Δf_c represents the sensitivity for client c, which is the maximum difference in the statistics sent by the client when a single data point is added or removed.

In our setting, the sensitivity is:

$$\Delta f_c = \frac{1}{\#\text{ of samples in client } c}. \tag{58}$$

Thus, for each client c, the scale parameter b_c is given by:

$$b_c = \frac{1}{(\#\text{ of samples in client } c) \cdot \epsilon}. \tag{59}$$

A larger b_c corresponds to a smaller ϵ, providing better privacy protection. To maintain the same level of privacy across all clients, clients with fewer samples will be given a larger b_c.