# Pursuing Overall Welfare in Federated Learning through Sequential Decision Making

**Seok-Ju Hahn** [1] **Gi-Soo Kim** [1 2] **Junghye Lee** [3 4 5]

## Abstract

In traditional federated learning, a single global model cannot perform equally well for all clients. Therefore, the need to achieve the *client-level fairness* in federated system has been emphasized, which can be realized by modifying the static aggregation scheme for updating the global model to an adaptive one, in response to the local signals of the participating clients. Our work reveals that existing fairness-aware aggregation strategies can be unified into an online convex optimization framework, in other words, a central server's *sequential decision making* process. To enhance the decision making capability, we propose simple and intuitive improvements for suboptimal designs within existing methods, presenting `AAggFF`. Considering practical requirements, we further subdivide our method tailored for the *cross-device* and the *cross-silo* settings, respectively. Theoretical analyses guarantee sublinear regret upper bounds for both settings: $\mathcal{O}(\sqrt{T \log K})$ for the cross-device setting, and $\mathcal{O}(K \log T)$ for the cross-silo setting, with $K$ clients and $T$ federation rounds. Extensive experiments demonstrate that the federated system equipped with `AAggFF` achieves better degree of client-level fairness than existing methods in both practical settings. Code is available at https://github.com/vaseline555/AAggFF.

[1]Department of Industrial Enginnering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea [2]Artificial Intelligence Graduate School, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea [3]Technology Management, Economics and Policy Program, Seoul National University (SNU), Seoul, South Korea [4]Graduate School of Engineering Practice, Seoul National University (SNU), Seoul, South Korea [5]Institute of Engineering Research, Seoul National University (SNU), Seoul, South Korea. Correspondence to: Junghye Lee <junghye@snu.ac.kr>, Gi-Soo Kim <gisookim@unist.ac.kr>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. 

## 1. Introduction

Federated Learning (FL) has been posed as an effective strategy to acquire a global model without centralizing data, therefore with no compromise in privacy (McMahan et al., 2017). It is commonly assumed that the central server coordinates the whole FL procedure by repeatedly *aggregating* local updates from participating $K$ clients during $T$ rounds.

Since each client updates the copy of a global model with its own data, variability across clients' data distributions causes many problems (Kairouz et al., 2021; Li et al., 2020b). The *client-level fairness* (Chen et al., 2023) is one of the main problems affected by such a statistical heterogeneity (Kairouz et al., 2021; Li et al., 2020b; Mohri et al., 2019; Li et al., 2019). Although the performance of a global model is high in average, some clients may be more benefited than others, resulting in violation of the client-level fairness. In this situation, there inevitably exists a group of clients who cannot utilize the trained global model due to its poor performance. This is a critical problem in practice since the underperformed groups may lose motivation to participate in the federated system. To remedy this problem, previous works (Mohri et al., 2019; Li et al., 2019; 2020a; Hu et al., 2022; Zhang et al., 2022) proposed to modify the static aggregation scheme into an *adaptive aggregation* strategy, according to given local signals (e.g., losses or gradients). In detail, the server *re-weights local updates* by assigning larger *mixing coefficients* to higher local losses.

When updating the mixing coefficients, however, *only a few bits are provided to the server*, compared to the update of a model parameter. For example, suppose that there exist $K$ clients in the federated system, each of which has $N$ local samples. When all clients participate in each round, $KN$ samples are used effectively for updating a new global model $\boldsymbol{\theta}$. On the contrary, only $K$ bits (e.g., local losses: $F_1(\boldsymbol{\theta}), ..., F_K(\boldsymbol{\theta})$) are provided to the server for an update of mixing coefficients. This is aggravated in cases where $K$ is too large, thus client sampling is inevitably required. In this case, the server is provided with far less than $K$ signals, which hinders faithful update of the mixing coefficients.

For sequentially updating a status in this *sample-deficient* situation, the Online Convex Optimization (OCO) framework

is undoubtedly the best solution. Interestingly, we discovered that most existing adaptive aggregation strategies can be readily unified into the OCO framework. Starting from this unification result, we propose an improved design for a fair FL algorithm in the view of sequential decision making. Since there exist OCO algorithms specialized for the setting where the decision space is a simplex (i.e., same as the domain of the mixing coefficient), these may be adopted to FL setting with some modifications for practical constraints.

In practice, FL is subdivided into two settings: *cross-silo* setting and *cross-device* setting (Kairouz et al., 2021). For $K$ clients and $T$ training rounds, each setting requires a different dependency on $K$ and $T$. In the cross-silo setting, the number of clients (e.g., institutions) is small and usually less than the number of rounds (i.e., $K < T$). e.g., $K = 20$ institutions with $T = 200$ rounds (Dayan et al., 2021). On the other hand, in the cross-device setting, the number of clients (e.g., mobile devices) is larger than the number of rounds ($K > T$). e.g., $K = 1.5 \times 10^6$ with $T = 3,000$ rounds (Hard et al., 2018). In designing an FL algorithm, these conditions should be reflected for the sake of practicality.

**Contributions** We propose `AAggFF`, a sequential decision making framework for the central server in FL tailored for inducing client-level fairness in the system. The contributions of our work are summarized as follows.

- We unify existing fairness-aware adaptive aggregation methods into an OCO framework and propose better online decision making designs for pursuing client-level fairness by the central server. (Section 3)
- We propose `AAggFF`, which is designed to enhance the client-level fairness, and further specialize our method into two practical settings: `AAggFF-S` for cross-silo FL and `AAggFF-D` for cross-device FL. (Section 4)
- We provide regret analyses on the behavior of two algorithms, `AAggFF-S` and `AAggFF-D`, presenting sublinear regrets. (Section 5)
- We evaluate `AAggFF` on extensive benchmark datasets for realistic FL scenarios with other baselines. `AAggFF` not only boosts the worst-performing clients but also maintains overall performance. (Section 6)

## 2. Related Works

**Client-Level Fairness in Federated Learning** The statistical heterogeneity across clients often causes non-uniform performances of a single global model on participating clients, which is also known as the violation of client-level fairness. Fairness-aware FL algorithms aim to eliminate such inequality to achieve uniform performance across all clients. There are mainly two approaches to address the problem (Chen et al., 2023): a single model approach, and a personalization approach. This paper mainly focuses on the former, which is usually realized by modifying the FL objective, such as a minimax fairness objective (Mohri et al., 2019) (which is also solved by multi-objective optimization (Hu et al., 2022) and is also modified to save a communication cost (Deng et al., 2020)), alpha-fairness (Mo & Walrand, 2000) objective (Li et al., 2019), suppressing outliers (i.e., clients having high losses) by tilted objective (Li et al., 2020a), and adopting the concept of proportional fairness to reach Nash equilibrium in performance distributions (Zhang et al., 2022). While the objective can be directly aligned with existing notions of fairness, it is not always a standard for the design of a fair FL algorithm. Notably, most of works share a common underlying principle: *assigning more weights to a local update having larger losses*.

**Definition 2.1.** (Client-Level Fairness; Definition 1 of (Li et al., 2019), Section 4.2 of (Chen et al., 2023)) We informally define the notion of client-level fairness in FL as the status where a trained global model yields uniformly good performance across all participating clients. Note that uniformity can be measured by the spread of performances.

**Online Decision Making** The OCO framework is designed for making *sequential decisions* with the best utilities, having solid theoretical backgrounds. It aims to minimize the cumulative mistakes of a decision maker (e.g., central sever), given a response of the environment (e.g., losses from clients) for finite rounds $t \in [T]$. The cumulative mistakes of the learner are usually denoted as the cumulative regret (see (5)), and the learner can achieve sublinear regret in finite rounds using well-designed OCO algorithms (Shalev-Shwartz et al., 2012; McMahan, 2017; Orabona, 2019). In designing an OCO algorithm, two main frameworks are mainly considered: Online Mirror Descent (OMD) (Nemirovskij & Yudin, 1983; Warmuth et al., 1997; Beck & Teboulle, 2003) and Follow-The-Regularized-Leader (FTRL) (Abernethy et al., 2009; Hazan & Kale, 2010; Agarwal & Hazan, 2005; Shalev-Shwartz & Singer, 2006). One of popular instantiations of both frameworks is the Online Portfolio Selection (OPS) algorithm, of which decision space is restricted to a probability simplex. The universal portfolio algorithm is the first that yields an optimal theoretical regret, $\mathcal{O}(K \log T)$) despite its heavy computation ($\mathcal{O}(K^4 T^{14})$) (Cover, 1991), the Online Gradient Descent (Zinkevich, 2003) and the Exponentiated Gradient (EG) (Helmbold et al., 1998) show slightly worse regrets (both are $\mathcal{O}(\sqrt{T})$), but can be executed in linear runtime in practice ($\mathcal{O}(K)$). Plus, the Online Newton Step (ONS) (Agarwal et al., 2006; Hazan et al., 2007) presents logarithmic regret with quadratic runtime in $K$. Since these OPS algorithms are proven to perform well when the decision is a probability vector, we adopt them for finding adaptive

mixing coefficients to achieve performance fairness in FL. To the best of our knowledge, we are the first to consider fair FL algorithms under the OCO framework.

## 3. Backgrounds

### 3.1. Mixing Coefficients in Federated Learning

The canonical objective of FL is given as follows:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) = \sum\nolimits_{i=1}^{K} p_i F_i(\boldsymbol{\theta}). \tag{1}$$

For $K$ clients, the FL objective aims to minimize the composite objectives, where client $i$'s local objective is $F_i(\boldsymbol{\theta})$, weighted by a corresponding *mixing coefficient* $p_i \geq 0$ ($\sum_{i=1}^{K} p_i = 1$), which is usually set to be a *static* value proportional to the sample size $n_i$: e.g., $p_i = \frac{n_i}{n}, n = \sum_{j=1}^{K} n_j$. Each local objective, $F_i(\boldsymbol{\theta}) = \frac{1}{n_i}\sum_{k=1}^{n_i} \mathcal{L}(\xi_k; \boldsymbol{\theta})$, is defined as the average of per-sample training loss $\mathcal{L}(\cdot; \boldsymbol{\theta})$ calculated from the local dataset, $\mathcal{D}_i = \{\xi_k\}_{k=1}^{n_i}$. Denote $\|\cdot\|_p$ as an $L_p$-norm and $\Delta_{K-1}$ as a probability simplex where $\Delta_{K-1} = \{\boldsymbol{q} \in \mathbb{R}^K : q_i \geq 0, \|\boldsymbol{q}\|_1 = 1\}$. Note that the mixing coefficient is a member of $\Delta_{K-1}$.

In vanilla FL, the role of the server to solve (1) is to naively add up local updates into a new global model by weighting each update with the *static* mixing coefficient proportional to $n_i$. As the fixed scheme often violates the client-level fairness, the server should use *adaptive* mixing coefficients to pursue overall welfare across clients. This can be modeled as an optimization w.r.t. $\boldsymbol{p} \in \Delta_{K-1}$, apart from (1).

### 3.2. Online Convex Optimization as a Unified Language

To mitigate the performance inequalities across clients, adaptive mixing coefficients can be estimated in response to local signals (e.g., local losses of a global model). Intriguingly, the adaptive aggregation strategies in existing fair FL methods (McMahan et al., 2017; Mohri et al., 2019; Li et al., 2019; 2020a; Zhang et al., 2022) can be readily *unified into one framework*, borrowing the language of OCO.

*Remark* 3.1. Suppose we want to solve a minimization problem defined in (2). For all $t \in [T]$, it aims to minimize a *decision loss* $\ell^{(t)}(\boldsymbol{p}) = -\langle \boldsymbol{p}, \boldsymbol{r}^{(t)} \rangle$ (where $\langle \cdot, \cdot \rangle$ is an inner product) defined by a *response* $\boldsymbol{r}^{(t)} \in \mathbb{R}^K$ and a *decision* $\boldsymbol{p} \in \Delta_{K-1}$, with a *regularizer* $R(\boldsymbol{p})$ having a constant *step size* $\eta \in \mathbb{R}_{\geq 0}$.

$$\boldsymbol{p}^{(t+1)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} \ell^{(t)}(\boldsymbol{p}) + \eta R(\boldsymbol{p}) \tag{2}$$

As long as the regularizer $R(\boldsymbol{p})$ in the Remark 3.1 is fixed as the negative entropy, i.e., $R(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$, this subsumes aggregation strategies proposed in `FedAvg` (McMahan et al., 2017), `AFL` (Mohri et al., 2019), `q-FFL` (Li et al., 2019), `TERM` (Li et al., 2020a), and `PropFair` (Zhang et al., 2022). It has an update as follows.

$$p_i^{(t+1)} \propto p_i^{(t)} \exp\left(r_i^{(t)}/\eta\right) \tag{3}$$

This is widely known as EG (Helmbold et al., 1998), a special realization of OMD (Nemirovskij & Yudin, 1983; Warmuth et al., 1997; Beck & Teboulle, 2003). We summarize how existing methods can be unified under this OCO framework in Table 1. The detailed derivations of mixing coefficients from each method are provided in Appendix A.1.

To sum up, we can interpret the aggregation mechanism in FL is secretly a result of *the server's sequential decision making* behind the scene. Since the sequential learning scheme is *well-behaved in a sample-deficient setting*, adopting OCO is surely a suitable tactic for the server in that *only a few bits are provided to update the mixing coefficients* in each FL round, e.g, the number of local responses collected from the clients is at most $K$. However, existing methods have not been devised with sequential decision making in mind. Therefore, one can easily find suboptimal designs inherent in existing methods from an OCO perspective.

### 3.3. Sequential Probability Assignment

To address the client-level fairness, the server should make an adaptive mixing coefficient vector, $\boldsymbol{p}^{(t)} \in \Delta_{K-1}$, for each round $t \in [T]$. In other words, the server needs to assign appropriate probabilities sequentially to local updates in every FL communication round.

Notably, this fairly resembles OPS, which seeks to maximize an investor's cumulative profits on a set of $K$ assets during $T$ periods, by assigning his/her wealth $\boldsymbol{p} \in \Delta_{K-1}$ to each asset every time. In the OPS, the investor observes a price of all assets, $\boldsymbol{r}^{(t)} \in \mathbb{R}^K$ for each time $t \in [T]$ and accumulates corresponding wealth according to the portfolio $\boldsymbol{p}^{(t)} \in \Delta_{K-1}$. After $T$ periods, achieved cumulative profits is represented as $\prod_{t=1}^{T}\left(1 + \langle \boldsymbol{p}^{(t)}, \boldsymbol{r}^{(t)} \rangle\right)$, or in the form of logarithmic growth ratio, $\sum_{t=1}^{T} \log\left(1 + \langle \boldsymbol{p}^{(t)}, \boldsymbol{r}^{(t)} \rangle\right)$. In other words, one can view that OPS algorithms adopt negative logarithmic growth as a decision loss.

**Definition 3.2.** (Negative Logarithmic Growth as a Decision Loss) For all $t \in [T]$, define a decision loss $\ell^{(t)} : \Delta_{K-1} \times \mathbb{R}^K \to \mathbb{R}$ as follows.

$$\ell^{(t)}(\boldsymbol{p}) = -\log(1 + \langle \boldsymbol{p}, \boldsymbol{r}^{(t)} \rangle), \tag{4}$$

where $\boldsymbol{p}$ is a decision vector in $\Delta_{K-1}$ and $\boldsymbol{r}^{(t)}$ is a response vector given at time $t$.

Again, the OPS algorithm can serve as a metaphor for the central server's fairness-aware online decision making in FL. For example, one can regard a response (i.e., local losses) of $K$ clients at a specific round $t$ the same as returns of

Table 1: Summary of Unification Results of Existing Fair FL Methods into an OCO Framework (2), viz. *Remark* 3.1

| **Method** | **Original Objective** (w.r.t. $\boldsymbol{\theta}$) | **Response** $(r_i^{(t)})$ | **Last Decision** $(p_i^{(t)})$ | **Step Size** $(\eta)$ | **New Decision** $(p_i^{(t+1)})$ |
|---|---|---|---|---|---|
| `FedAvg` (McMahan et al., 2017) | $\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^K \frac{n_i}{n} F_i(\boldsymbol{\theta})$ | $0$ | $n_i/n$ | $1$ | $\propto n_i$ |
| `q-FedAvg` (Li et al., 2019) (`AFL` (Mohri et al., 2019)) | $\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \sum_{i=1}^K \frac{n_i/n}{q+1} F_i^{q+1}(\boldsymbol{\theta})$ (`AFL` if $q\to\infty$) | $q\log F_i(\boldsymbol{\theta}^{(t)})$ | $n_i/n$ | $1$ | $\propto n_i F_i^q\left(\boldsymbol{\theta}^{(t)}\right)$ |
| `TERM` (Li et al., 2020a) | $\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{1}{\lambda}\log(\sum_{i=1}^K \frac{n_i}{n}\exp(\lambda F_i(\boldsymbol{\theta})))$ | $F_i(\boldsymbol{\theta}^{(t)})$ | $n_i/n$ | $\frac{1}{\lambda}$ | $\propto n_i \exp\left(\lambda F_i\left(\boldsymbol{\theta}^{(t)}\right)\right)$ |
| `PropFair` (Zhang et al., 2022) | $\min_{\boldsymbol{\theta}\in\mathbb{R}^d} -\sum_{i=1}^K \frac{n_i}{n}\log(M-F_i(\boldsymbol{\theta}))$ | $-\log(M-F_i(\boldsymbol{\theta}^{(t)}))$ | $n_i/n$ | $1$ | $\propto \frac{n_i}{M-F_i(\boldsymbol{\theta}^{(t)})}$ |

assets on the day $t$. Similarly, by considering cumulative losses (i.e., cumulative wealth) achieved until $t$, the server can determine the next mixing coefficients (i.e., portfolio ratios) in $\Delta_{K-1}$. In the same context, the negative logarithmic growth can also be adopted as the decision loss. Accordingly, we can adopt well-established OPS strategies for achieving client-level fairness in FL.

Including OPS, a de facto standard objective for OCO is to minimize the regret defined in (5), with regard to the best decision in hindsight, $\boldsymbol{p}^\star \triangleq \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} \sum_{t=1}^T \ell^{(t)}(\boldsymbol{p})$, given all decisions $\{\boldsymbol{p}^{(1)}, ..., \boldsymbol{p}^{(T)}\}$. (Shalev-Shwartz et al., 2012; McMahan, 2017; Orabona, 2019)

$$\text{Regret}^{(T)}(\boldsymbol{p}^\star) = \sum_{t=1}^T \ell^{(t)}(\boldsymbol{p}^{(t)}) - \sum_{t=1}^T \ell^{(t)}(\boldsymbol{p}^\star) \quad (5)$$

In finite time $T$, an online decision making strategy should guarantee that the regret grows sublinearly. Therefore, when OPS strategies are modified for the fair FL, we should check if the strategy can guarantee vanishing regret upper bound in $T$. Besides, we should also consider *the dependency on* $K$ due to practical constraints of the federated system.

## 4. Proposed Methods

### 4.1. Improved Design for Better Decision Making

From the Remark 3.1 and Table 1, one can easily notice suboptimal designs of existing methods in terms of OCO, as follows.

a) Existing methods are *stateless* in making a new decision, $p_i^{(t+1)}$. The previous decision is ignored as a fixed value ($p_i^{(t)} = n_i/n$) in the subsequent decision making. This naive reliance on static coefficients still runs the risk of violating client-level fairness.

b) The decision maker sticks to a *fixed* and *arbitrary* step size $\eta$, or a *fixed* regularizer $R(\boldsymbol{p})$ across $t \in [T]$, which can significantly affect the performance of OCO algorithms and should be manually selected.

c) The decision loss is neither *Lipschitz continuous* nor *strictly convex*, which is related to achieving a sublinear regret.

As a remedy for handling a) and b), the OMD objective for the server (i.e., (2)) can be replaced as follows.

$$\boldsymbol{p}^{(t+1)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} \sum_{\tau=1}^t \ell^{(\tau)}(\boldsymbol{p}) + R^{(t+1)}(\boldsymbol{p}) \quad (6)$$

This is also known as FTRL objective (Abernethy et al., 2009; Hazan & Kale, 2010; Agarwal & Hazan, 2005; Shalev-Shwartz & Singer, 2006), which is inherently a *stateful* sequential decision making algorithm that adapts to histories of decision losses, $\sum_{\tau=1}^t \ell^{(\tau)}(\boldsymbol{p})$, where $\ell^{(t)} : \Delta_{K-1}\times\mathbb{R}^K \to \mathbb{R}$, with the *time-varying* regularizer $R^{(t)} : \Delta_{K-1}\to\mathbb{R}$. Note that the time-varying regularizer is sometimes represented as, $\eta^{(t+1)}R(\boldsymbol{p})$, a fixed regularizer $R(\boldsymbol{p})$ multiplied by a *time-varying* step size, $\eta^{(t+1)}\in\mathbb{R}_{\geq 0}$, which can later be automatically determined from the regret analysis (see e.g., Remark 4.5).

Additionally, when equipped with the negative logarithmic growth as a decision loss (i.e., (4)), the problem c) can be addressed due to its strict convexity and Lipscthiz continuity. (See Lemma 4.1) Note that when the loss function is convex, we can run the FTRL with a linearized loss (i.e., $\tilde{\ell}^{(t)}(\boldsymbol{p}) = \left\langle \boldsymbol{p}, \boldsymbol{g}^{(t)}\right\rangle$ where $\boldsymbol{g}^{(t)} = \nabla\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)$). This is useful in that a closed-form update can be obtained thanks to the properties of the Fenchel conjugate (see Remark 4.5).

### 4.2. AAggFF: Adaptive Aggregation for Fair Federated Learning

Based on the improved objective design derived from the FTRL, we now introduce our methods, `AAggFF`, an acronym of **A**daptive **Agg**regation for **F**air **F**ederated Learning). Mirroring the practical requirements of FL, we further subdivide into two algorithms: `AAggFF-S` for the cross-silo setting and `AAggFF-D` for the cross-device setting.

#### 4.2.1. AAggFF-S: Algorithm for the Cross-Silo Federated Learning

In the cross-silo setting, it is typically assumed that *all* $K$ clients participate in $T$ rounds, since there are a moderately small number of clients in the federated system. Therefore, the server's stateful decision making is beneficial for enhancing overall welfare across federation rounds. This is also favorable since existing OPS algorithms can be readily

adopted.

**Online Newton Step (Agarwal et al., 2006; Hazan et al., 2007)** The ONS algorithm updates a new decision as follows ($\alpha$ and $\beta$ are constants to be determined).

$$\boldsymbol{p}^{(t+1)} = \underset{\boldsymbol{p}\in\Delta_{K-1}}{\arg\min} \sum_{\tau=1}^{t} \tilde{\ell}^{(\tau)}(\boldsymbol{p}) + \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{\tau=1}^{t}(\langle \boldsymbol{g}^{(\tau)}, \boldsymbol{p}-\boldsymbol{p}^{(\tau)}\rangle)^2 \tag{7}$$

The ONS can be reduced to the FTRL objective introduced in (6). It can be retrieved when we use a linearized loss, $\tilde{\ell}^{(t)}(\boldsymbol{p}) = \langle \boldsymbol{p}, \boldsymbol{g}^{(t)}\rangle$, and the time-varying proximal regularizer, defined as $R^{(t+1)}(\boldsymbol{p}) = \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{\tau=1}^{t}\left(\langle \boldsymbol{g}^{(\tau)}, \boldsymbol{p}-\boldsymbol{p}^{(\tau)}\rangle\right)^2$.

We choose ONS in that its regret is optimal in $T$, which is also a dominating constant for the cross-silo FL setting: $\mathcal{O}(L_\infty K \log T)$ regret upper bound, where $L_\infty$ is the Lipschitz constant of decision loss w.r.t. $\|\cdot\|_\infty$. That is, $L_\infty$ should be *finite* for a vanishing regret (see Theorem 5.1).

**Necessity of Bounded Response** Note that the Lipchitz continuity of the negative logarithmic growth as a decision loss is determined as follows.

**Lemma 4.1.** *For all $t \in [T]$, suppose each entry of a response vector $\boldsymbol{r}^{(t)} \in \mathbb{R}^K$ is bounded as $r_i^{(t)} \in [C_1, C_2]$ for some constants $C_1$ and $C_2$ satisfying $0 < C_1 < C_2$. Then, the decision loss $\ell^{(t)}$ defined in (4) is $\frac{C_2}{1+C_1}$-Lipschitz continuous in $\Delta_{K-1}$ w.r.t. $\|\cdot\|_\infty$.*

From now on, all proofs are deferred to Appendix A.2. According to the Lemma 4.1, the Lipschitz constant of the decision loss, $L_\infty$, is dependent upon *the range of a response vector's element*. While from the unification result in Table 1, one can easily notice that the response is constructed from local losses collected in round $t$, $F_i(\boldsymbol{\theta}^{(t)}) \in \mathbb{R}_{\geq 0}, i \in [K]$.

This is a scalar value calculated from a local training set of each client, using the current model $\boldsymbol{\theta}^{(t)}$ *before its local update*. Since the local loss function is typically unbounded above (e.g., cross-entropy), it should be transformed into bounded values to satisfy the Lipschitz continuity. In existing fair FL methods, however, all responses are not bounded above, thus we cannot guarantee the Lipschitz continuity.

To ensure a bounded response, we propose to use a transformation denoted as $\rho^{(t)}(\cdot)$, inspired by the cumulative distribution function (CDF) as follows.

**Definition 4.2.** (CDF-driven Response Transformation) We define $r_i^{(t)} \equiv \rho^{(t)}\left(F_i(\boldsymbol{\theta}^{(t)})\right)$, each element of the response vector is defined from the corresponding entry of a local loss by an element-wise mapping $\rho^{(t)} : \mathbb{R}_{\geq 0} \to [C_1, C_2]$, given a pre-defined CDF as:

$$\rho^{(t)}\left(F_i\left(\boldsymbol{\theta}^{(t)}\right)\right) \triangleq C_1 + (C_2 - C_1)\mathrm{CDF}\left(\frac{F_i\left(\boldsymbol{\theta}^{(t)}\right)}{\bar{\mathrm{F}}^{(t)}}\right), \tag{8}$$

where $\bar{\mathrm{F}}^{(t)} = \frac{1}{|S^{(t)}|}\sum_{i\in S^{(t)}} F_i\left(\boldsymbol{\theta}^{(t)}\right)$, and $S^{(t)}$ is an index set of available clients in $t$.

Note again that the larger mixing coefficient should be assigned for the larger local loss. In such a perspective, using the CDF for transforming a loss value is an acceptable approach in that the CDF value is a good indicator for estimating "*how large a specific local loss is*", relative to local losses from other clients. To instill the comparative nature, local losses are divided by the average of observed losses in time $t$ before applying the transformation. As a result, all local losses are centered on 1 in expectation. See Appendix B.1 for detailed discussions.

In summary, the whole procedure of AAggFF-S is illustrated as a pseudocode in Algorithm 2.

#### 4.2.2. AAggFF-D: Algorithm for the Cross-Device Federated Learning

Unlike the cross-silo setting, we cannot be naïvely edopt existing OCO algorithms for finding adaptive mixing coefficients in the cross-device setting. It is attributed to *the large number of participating clients* in this special setting. Since the number of participating clients ($K$) is massive (e.g., Android users are over 3 billion (Curry, 2023)), the dependence on $K$ in terms of regret bound and algorithm runtime is as significant as a total communication round $T$.

**Linear Runtime OCO Algorithm** The ONS has regret proportional to $K$ and runs in $\mathcal{O}\left(K^2 + K^3\right)$[1] per round, which is *nearly impossible* to be adopted for the cross-device FL setting due to large $K$, even though the logarithmic regret is guaranteed in $T$. Instead, we can exploit the variant of EG adapted to FTRL (Orabona, 2019), which can be run in $\mathcal{O}(K)$ time per round.

$$\boldsymbol{p}^{(t+1)} = \underset{\boldsymbol{p}\in\Delta_{K-1}}{\arg\min} \sum_{\tau=1}^{t} \tilde{\ell}^{(\tau)}(\boldsymbol{p}) + \eta^{(t+1)} R(\boldsymbol{p}), \tag{9}$$

where $\eta^{(t)}$ is non-decreasing step size across $t \in [T]$, and $R(\boldsymbol{p}) = \sum_{i=1}^{K} p_i \log p_i$ is a negative entropy regularizer. Still, the regret bound gets worse than that of ONS, as $\mathcal{O}\left(L_\infty\sqrt{T\log K}\right)$ (see Theorem 5.2).

**Partially Observed Response** The large number of clients coerces the federated system to introduce the *client*

[1] A generalized projection required for the Online Newton Step can be solved in $\tilde{\mathcal{O}}\left(K^3\right)$ (Agarwal et al., 2006; Hazan et al., 2007).

*sampling scheme* in each round. Therefore, the decision maker (i.e., the central server) cannot always observe all entries of a response vector per round. This is problematic in terms of OCO, since OCO algorithms assume that they can acquire intact response vector for every round $t \in [T]$. Instead, when the client sampling is introduced, the learner can only observe entries of sampled client indices in the round $t$, denoted as $S^{(t)}$.

To make a new decision using a *partially observed* response vector, the effect of unobserved entries should be appropriately estimated. We solve this problem by adopting a doubly robust (DR) estimator (Robins et al., 1994; Bang & Robins, 2005) for the expectation of the response vector. The rationale behind the adoption of the DR estimator is the fact that the unobserved entries are *missing data*.

For handling the missingness problem, the DR estimator combines inverse probability weighting (IPW (Auer et al., 2002)) estimator and imputation mechanism, where the former is to adjust the weight of observed entries by the inverse of its observation probability (i.e., client sampling probability), and the latter is to fill unobserved entries with appropriate values specific to a given task.

Similar to the IPW estimator, the DR estimator is an unbiased estimator when the true observation probability is known. Since we sample clients uniformly at random without replacement, *the observation probability is known* (i.e., $C \in (0, 1)$) to the algorithm.

**Lemma 4.3.** *Denote $C = P\left(i \in S^{(t)}\right)$ as a client sampling probability in a cross-device FL setting for every round $t \in [T]$. The DR estimator $\breve{\boldsymbol{r}}^{(t)}$, of which element is defined in (10) is an unbiased estimator of given partially observed response vector $\boldsymbol{r}^{(t)}$. i.e., $\mathbb{E}\left[\breve{\boldsymbol{r}}^{(t)}\right] = \boldsymbol{r}^{(t)}$.*

$$\breve{r}_i^{(t)} = \left(1 - \frac{\mathbb{I}(i \in S^{(t)})}{C}\right) \bar{\mathrm{r}}^{(t)} + \frac{\mathbb{I}\left(i \in S^{(t)}\right)}{C} r_i^{(t)}, \quad (10)$$

*where* $\bar{\mathrm{r}}^{(t)} = \frac{1}{\left|S^{(t)}\right|} \sum_{i \in S^{(t)}} r_i^{(t)}$.

Still, it is required to guarantee that the gradient vector from the DR estimator is also an unbiased estimator of a true gradient of a decision loss. Unfortunately, the gradient of a decision loss is *not linear* in the response vector due to its fractional form: $\boldsymbol{g}^{(t)} = \nabla \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) = -\frac{\boldsymbol{r}^{(t)}}{1+\left\langle \boldsymbol{p}^{(t)}, \boldsymbol{r}^{(t)} \right\rangle}$.

Therefore, we instead use linearly approximated gradient *w.r.t. a response vector* as follows.

**Lemma 4.4.** *Denote the gradient of a decision loss in terms of a response vector as $\boldsymbol{g} \equiv \mathbf{h}(\boldsymbol{r}) = [h_1(\boldsymbol{r}), ..., h_K(\boldsymbol{r})]^\top = -\frac{\boldsymbol{r}}{1+\langle \boldsymbol{p}, \boldsymbol{r} \rangle}$. It can be linearized for the response vector into $\tilde{\boldsymbol{g}} \equiv \tilde{\mathbf{h}}(\boldsymbol{r})$, given a reference $\boldsymbol{r}_0$ as follows. (Note that the superscript $(t)$ is omitted for a brevity of notation)*

$$\boldsymbol{g} \approx \tilde{\boldsymbol{g}} \equiv \tilde{\mathbf{h}}(\boldsymbol{r}) = -\frac{\boldsymbol{r}}{1 + \langle \boldsymbol{p}, \boldsymbol{r}_0 \rangle} + \frac{\boldsymbol{r}_0 \boldsymbol{p}^\top (\boldsymbol{r} - \boldsymbol{r}_0)}{(1 + \langle \boldsymbol{p}, \boldsymbol{r}_0 \rangle)^2} \quad (11)$$

*Further denote $\breve{\boldsymbol{g}}$ as a gradient estimate from (11) using the DR estimator of a response vector according to (10), at an arbitrary reference $\boldsymbol{r}_0$. Then, $\breve{\boldsymbol{g}}$ is an unbiased estimator of the linearized gradient of a decision loss at $\boldsymbol{r}_0$, which is close to the true gradient: $\mathbb{E}[\breve{\boldsymbol{g}}] = \tilde{\boldsymbol{g}} \approx \boldsymbol{g}$.*

As suggested in (10), we similarly set the reference as an average of observed responses at round $t$, i.e., $\boldsymbol{r}_0^{(t)} = \bar{\mathrm{r}}^{(t)} \mathbf{1}_K$. It is a valid choice in that dominating unobserved entries are imputed by the average of observed responses as in (10).

To sum up, we can update a new decision using this unbiased and linearly approximated gradient estimator even if only a partially observed response vector is provided (i.e., mixing coefficients of unsampled clients can also be updated). Note that the linearized gradient calculated from the DR estimator, $\breve{\boldsymbol{g}}$, has finite norm w.r.t. $\|\cdot\|_\infty$ (see Lemma A.2).

**Closed-Form Update** Especially for the cross-device setting, we can obtain a closed-form update of the objective (9), which is due to the property of Fenchel conjugate.

*Remark* 4.5. The objective of `AAggFF-D` stated in (9) has a closed-form update formula as follows. (Orabona, 2019)

$$p_i^{(t+1)} \propto \exp\left(-\frac{\sqrt{\log K} \sum_{\tau=1}^{t} \breve{g}_i^{(\tau)}}{\breve{L}_\infty \sqrt{t+1}}\right) \quad (12)$$

It is equivalent to setting the time-varying step size as $\eta^{(t)} = \frac{\breve{L}_\infty \sqrt{t}}{\sqrt{\log K}}$. Note that $\breve{g}_i^{(t)}$ is an entry of gradient from DR estimator defined in Lemma 4.4 and $\breve{L}_\infty$ is a corresponding Lipschitz constant satisfying $\|\breve{\boldsymbol{g}}\|_\infty \leq \breve{L}_\infty$ stated in Lemma A.2. See Appendix A.6 for the derivation.

In summary, the whole procedure of `AAggFF-D` is illustrated in Algorithm 3.

## 5. Regret Analysis

In this section, we provide theoretical guarantees of our methods, `AAggFF-S` and `AAggFF-D` in terms of sequential decision making. The common objective for OCO algorithms is to *minimize the regret* across a sequence of decision losses in eq. (5). We provide sublinear regret upper bounds in terms of $T$ as follows.

**Theorem 5.1.** *(Regret Upper Bound for `AAggFF-S` (i.e., ONS (Agarwal et al., 2006; Hazan et al., 2007))) With the notation in eq. (7), suppose for every $\boldsymbol{p} \in \Delta_{K-1}$, and for every $t \in [T]$, let the decisions $\{\boldsymbol{p}^{(t)} : t \in [T]\}$ be derived by `AAggFF-S` for $K$ clients during $T$ rounds in*

Table 2: Comparison Results of AAggFF-S in the Cross-Silo Setting

| Dataset | Berka (AUROC) | | | | MQP (AUROC) | | | | ISIC (Acc. 5) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | Avg. (↑) | Worst (↑) | Best (↑) | Gini (↓) | Avg. (↑) | Worst (↑) | Best (↑) | Gini (↓) | Avg. (↑) | Worst (↑) | Best (↑) | Gini (↓) |
| FedAvg | 80.09 | 48.06 | **99.03** | 10.87 | 56.06 | 41.03 | 76.31 | 8.63 | 87.42 | 69.92 | 92.57 | 4.84 |
| (McMahan et al., 2017) | (2.45) | (25.15) | (1.37) | (4.11) | (0.06) | (4.33) | (8.42) | (0.91) | (2.11) | (6.78) | (2.56) | (1.17) |
| AFL | 79.70 | 49.02 | 98.55 | 10.58 | 56.01 | 41.28 | 75.54 | 8.56 | 87.39 | 68.17 | 93.33 | 4.80 |
| (Mohri et al., 2019) | (4.14) | (25.89) | (2.05) | (5.03) | (0.30) | (3.92) | (6.77) | (1.24) | (2.31) | (10.09) | (2.18) | (1.74) |
| q-FedAvg | 79.98 | 49.44 | 98.07 | 10.62 | **56.89** | 40.22 | **79.38** | 8.68 | 41.59 | 20.38 | 58.08 | 22.25 |
| (Li et al., 2019) | (3.89) | (26.15) | (2.73) | (5.22) | (0.42) | (3.06) | (9.09) | (0.57) | (16.22) | (23.24) | (28.52) | (10.02) |
| TERM | 80.11 | 48.96 | **99.03** | 10.86 | 56.47 | 40.73 | 76.80 | 8.67 | 87.89 | 77.32 | 96.00 | 3.77 |
| (Li et al., 2020a) | (3.08) | (25.79) | (1.37) | (4.73) | (0.19) | (4.36) | (8.30) | (1.43) | (1.69) | (5.84) | (3.27) | (0.94) |
| FedMGDA | 79.24 | 46.38 | **99.03** | 11.64 | 53.02 | 34.91 | 69.65 | 10.33 | 42.36 | 21.44 | 59.21 | 22.25 |
| (Hu et al., 2022) | (2.96) | (24.11) | (1.37) | (4.84) | (1.67) | (2.22) | (3.89) | (0.44) | (14.94) | (21.30) | (28.52) | (10.02) |
| PropFair | 79.61 | 49.44 | 98.07 | 10.47 | 56.60 | 41.71 | 79.09 | 8.74 | 83.88 | 58.36 | 91.35 | 7.91 |
| (Zhang et al., 2022) | (4.49) | (26.15) | (2.73) | (5.04) | (0.39) | (3.80) | (7.40) | (0.87) | (2.50) | (11.63) | (2.48) | (2.10) |
| AAggFF-S | **80.93** | **52.08** | **99.03** | **10.16** | 56.63 | **41.79** | 75.56 | **8.38** | **89.76** | **85.17** | **98.22** | **2.52** |
| (Proposed) | (2.96) | (23.59) | (1.37) | (3.80) | (0.54) | (4.43) | (6.53) | (0.77) | (1.03) | (3.87) | (1.66) | (0.38) |

Table 3: Comparison Results of AAggFF-D in the Cross-Device Setting

| Dataset | CelebA (Acc. 1) | | | | Reddit (Acc. 1) | | | | SpeechCommands (Acc. 5) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Method** | Avg. (↑) | Worst 10% (↑) | Best 10%(↑) | Gini (↓) | Avg. (↑) | Worst 10%(↑) | Best 10%(↑) | Gini (↓) | Avg. (↑) | Worst 10%(↑) | Best 10%(↑) | Gini (↓) |
| FedAvg | 90.79 | 55.76 | 100.00 | 7.86 | 10.76 | 2.50 | 20.86 | 25.66 | 75.51 | 7.93 | 100.00 | 24.58 |
| (McMahan et al., 2017) | (0.53) | (0.84) | (0.00) | (0.30) | (1.45) | (0.21) | (3.64) | (0.49) | (1.08) | (2.87) | (0.00) | (1.34) |
| q-FedAvg | 90.88 | 55.73 | 100.00 | 7.82 | 12.76 | 3.38 | 21.81 | 23.34 | 73.34 | 11.19 | 100.00 | 23.16 |
| (Li et al., 2019) | (0.19) | (0.85) | (0.00) | (0.21) | (0.32) | (0.20) | (0.19) | (0.34) | (0.47) | (0.47) | (0.00) | (0.13) |
| TERM | 90.71 | 55.66 | 100.00 | 7.90 | 12.02 | 2.85 | 20.74 | 24.15 | 70.90 | 5.98 | 100.00 | 26.37 |
| (Li et al., 2020a) | (0.65) | (0.93) | (0.00) | (0.38) | (0.16) | (0.41) | (0.65) | (1.05) | (2.96) | (1.10) | (0.00) | (1.32) |
| FedMGDA | 88.33 | 48.60 | 100.00 | 9.75 | 10.58 | 2.35 | 19.09 | 25.20 | 72.45 | 9.65 | 100.00 | 23.68 |
| (Hu et al., 2022) | (0.63) | (25.85) | (0.00) | (0.59) | (0.18) | (0.20) | (0.62) | (0.22) | (1.88) | (2.90) | (0.00) | (1.27) |
| PropFair | 87.25 | 48.11 | 100.00 | 10.39 | 11.26 | 1.95 | 21.33 | 25.97 | 73.64 | 7.30 | 100.00 | 24.97 |
| (Zhang et al., 2022) | (5.01) | (10.03) | (0.00) | (3.43) | (0.71) | (0.32) | (0.92) | (1.02) | (3.31) | (1.02) | (0.00) | (1.09) |
| AAggFF-D | **91.27** | **56.71** | 100.00 | **7.54** | **12.95** | **4.75** | **22.81** | **22.59** | **76.68** | **14.54** | 100.00 | **21.42** |
| (Proposed) | (0.07) | (0.08) | (0.00) | (0.04) | (0.39) | (0.76) | (1.36) | (0.28) | (0.80) | (2.58) | (0.00) | (0.81) |

*Algorithm 2. Then, the regret defined in eq. (5) is bounded above as follows, where $\alpha$ and $\beta$ are determined as $\alpha = 4KL_\infty, \beta = \frac{1}{4L_\infty}$.*

$$\text{Regret}^{(T)}\left(\boldsymbol{p}^\star\right) \leq 2L_\infty K\left(1+\log\left(1+\frac{T}{16K}\right)\right).$$

From the Theorem 5.1, we can enjoy $\mathcal{O}\left(L_\infty K \log T\right)$ regret upper bound, which is an acceptable result, considering a typical assumption in the cross-silo setting (i.e., $K < T$).

For the cross-device setting, we first present the full synchronization setting, which requires no extra adjustment.

**Theorem 5.2.** *(Regret Upper Bound for AAggFF-D with Full-Client Participation) With the notation in (9), suppose for every $\boldsymbol{p} \in \Delta_{K-1}$, and for every $t \in [T]$, let the decisions $\{\boldsymbol{p}^{(t)} : t \in [T]\}$ be derived by AAggFF-D for $K$ clients with client sampling probability $C = 1$ during $T$ rounds in Algorithm 3. Then, the regret defined in (5) is bounded above as follows.*

$$\text{Regret}^{(T)}\left(\boldsymbol{p}^\star\right) \leq 2L_\infty\sqrt{T\log K}.$$

When equipped with a client sampling, the randomness from the sampling should be considered. Due to local losses of selected clients can only be observed, AAggFF-D should be equipped with the unbiased estimator of a response vector (from Lemma 4.3) and a corresponding linearly approximated gradient vector (from Lemma 4.4). Since they are unbiased estimators, the expected regret is also the same.

**Corollary 5.3.** *(Regret Upper Bound for AAggFF-D with Partial-Client Participation) With the client sampling probability $C \in (0,1)$, the DR estimator of a partially observed response $\breve{\boldsymbol{r}}^{(t)}$ and corresponding linearized gradient $\breve{\boldsymbol{g}}^{(t)}$ for all $t \in [T]$, the regret defined in (5) is bounded above in expectation as follows.*

$$\mathbb{E}\left[\text{Regret}^{(T)}(\boldsymbol{p}^\star)\right] \leq \mathcal{O}\left(L_\infty\sqrt{T\log K}\right).$$

## 6. Experimental Results

We design experiments to evaluate empirical performances of our proposed framework AAggFF, composed of sub-methods AAggFF-S and AAggFF-D.

Table 4: Description of Federated Benchmarks for Cross-Silo and Cross-Device Settings

| Cross-Silo Dataset | $K$ | $T$ | Cross-Device Dataset | $K$ | $T$ |
|---|---|---|---|---|---|
| Berka | 7 | 100 | CelebA | 9,343 | 3,000 |
| MQP | 11 | 100 | Reddit | 817 | 300 |
| ISIC | 6 | 50 | SpeechCommands | 2,005 | 500 |

**Experimental Setup** We conduct experiments on datasets mirroring *realistic scenarios* in federated systems: multiple modalities (vision, text, speech, and tabular form) and natural data partitioning. We briefly summarize FL settings of each dataset in Table 4. For the *cross-silo* setting, we used Berka tabular dataset (Berka, 1999), MQP clinical text dataset (McCreery et al., 2020), and ISIC oncological image dataset (Codella et al., 2018) (also a part of FLamby benchmark (Ogier du Terrail et al., 2022)). For the *cross-device* setting, we used CelebA vision dataset (Liu et al., 2015), Reddit text dataset (both are parts of LEAF benchmark (Caldas et al., 2019)) and SpeechCommands audio dataset (Warden, 2018).

Instead of manually partitioning data to simulate statistical heterogeneity, we adopt natural client partitions inherent in each dataset. Each client dataset is split into an 80% training set and a 20% test set in a stratified manner where applicable. All experiments are run with 3 different random seeds after tuning hyperparameters. See Appendix C for full descriptions of the experimental setup.

**Improvement in the Client-Level Fairness** We compare our methods with existing fair FL methods including `FedAvg` (McMahan et al., 2017), `AFL` (Mohri et al., 2019), `q-FedAvg` (Li et al., 2019), `TERM` (Li et al., 2020a), `FedMGDA` (Hu et al., 2022), and `PropFair` (Zhang et al., 2022). Since `AFL` requires full synchronization of clients every round, it is only compared in the cross-silo setting.

In the *cross-silo* setting, we assume all $K$ clients participate in $T$ federation rounds (i.e., $C = 1$), and in the *cross-device* setting, the client participation rate $C \in (0, 1)$ is set to ensure 5 among $K$ clients are participating in each round. We evaluate each dataset using appropriate metrics for tasks as indicated under the dataset name in Table 2 and 3, where the average, the best (10%), the worst (10%), and Gini coefficient[2] of clients' performance distributions are reported with the standard deviation inside parentheses in gray color below the averaged metric.

From the results, we verify that `AAggFF` can lead to enhanced worst-case metric and Gini coefficient in both settings while retaining competitive average performance to other baselines. Remarkably, along with the improved worst-case performance, the small Gini coefficient indicates that performances of clients are close to each other, which is directly translated into the improved client-level fairness.

Table 5: Accuracy Parity Gap in the Cross-Silo Setting

| Dataset | Berka | MQP | ISIC |
|---|---|---|---|
| **Method** | $\Delta$AG ($\downarrow$) | | |
| `FedAvg` (McMahan et al., 2017) | 50.84 (23.98) | 35.30 (5.39) | 22.64 (4.50) |
| `AFL` (Mohri et al., 2019) | 50.98 (23.78) | 34.26 (5.16) | 25.16 (8.01) |
| `q-FedAvg` (Li et al., 2019) | 50.43 (22.15) | 39.16 (7.13) | 37.69 (5.52) |
| `TERM` (Li et al., 2020a) | 49.60 (23.74) | 36.07 (6.93) | 15.19 (9.26) |
| `FedMGDA` (Hu et al., 2022) | 44.46 (17.49) | 34.74 (1.74) | 37.69 (5.52) |
| `PropFair` (Zhang et al., 2022) | 49.05 (23.78) | 37.38 (4.35) | 32.99 (9.60) |
| `AAggFF-S` (Proposed) | **44.03** (17.55) | **33.77** (3.31) | **13.05** (2.23) |

Table 6: Accuracy Parity Gap in the Cross-Device Setting

| Dataset | CelebA | Reddit | Speech Commands |
|---|---|---|---|
| **Method** | $\Delta$AG ($\downarrow$) | | |
| `FedAvg` (McMahan et al., 2017) | 44.25 (0.84) | 18.36 (3.52) | 92.07 (2.87) |
| `q-FedAvg` (Li et al., 2019) | 44.27 (0.85) | 18.43 (0.09) | 88.81 (0.47) |
| `TERM` (Li et al., 2020a) | 44.34 (0.93) | 17.89 (0.75) | 94.02 (1.10) |
| `FedMGDA` (Hu et al., 2022) | 51.40 (2.59) | **16.74** (0.43) | 90.35 (2.90) |
| `PropFair` (Zhang et al., 2022) | 51.90 (10.03) | 19.39 (0.64) | 92.70 (1.02) |
| `AAggFF-D` (Proposed) | **43.29** (0.08) | 18.07 (0.70) | **85.46** (2.58) |

**Connection to Accuracy Parity** As discussed in (Li et al., 2019), the client-level fairness can be loosely connected to existing fairness notion, the *accuracy parity* (Zafar et al., 2017). It is guaranteed if the accuracies in protected groups are equal to each other. While the accuracy parity requires *equal* performances among specific groups having protected attributes (Zafar et al., 2017; Cotter et al., 2019), this is too restrictive to be directly applied to FL settings, since each client cannot always be exactly corresponded to the concept of 'a group', and each client's local distribution may not be partitioned by protected attributes in the federated system.

With a relaxation of the original concept, we adopt the

[2]The Gini coefficient is inflated by ($\times 10^2$) for readability.

notion of accuracy parity for measuring the degree of the client-level fairness in the federated system, i.e., we simply regard the group as each client. As a metric, we adopt the accuracy parity gap ($\Delta$AG) proposed by (Zhao et al., 2019; Chi et al., 2021), which is simply defined as an absolute difference between the performance of the best and the worst performing groups (clients). The results are in Table 5 and Table 6. It can be said that the smaller the $\Delta$AG, the more degree of the accuracy parity fairness (and therefore the client-level fairness) is achieved.

It should be noted that strictly achieving the accuracy parity can sometimes require sacrifice in the average performance. This is aligned with the result of Reddit dataset in Table 6, where FedMGDA (Hu et al., 2022) achieved the smallest $\Delta$AG, while its average performance is only 10.58 in Table 3. This is far lower than our proposed method's average performance, 12.95. Except this case, AAggFF consistently shows the smallest $\Delta$AG than other baseline methods, which is important in the perspective of striking a good balance between overall utility and the client-level fairness.

Table 7: Improved Performance of FL Algorithms after being Equipped with AAggFF

| Dataset | Heart (AUROC) | | TinyImageNet (Acc. 5) | |
|---|---|---|---|---|
| Method | Avg. (↑) | Worst (↑) | Avg. (↑) | Worst 10%(↑) |
| FedAvg (McMahan et al., 2017) | 84.42 (2.45) | 65.22 (9.78) | 85.93 (0.77) | 50.95 (0.15) |
| | **85.04**(2.86) | **66.56** (10.81) | **86.66** (0.63) | **51.50** (2.32) |
| FedProx (Li et al., 2020c) | 84.48(0.25) | 65.44(9.77) | **86.49** (0.72) | 51.64 (2.07) |
| | **85.72**(2.81) | **66.67**(10.71) | 86.11 (0.72) | **52.29** (2.16) |
| FedAdam (Reddi et al., 2020) | 84.34(2.78) | 65.44(10.12) | 87.04 (1.05) | 53.54 (2.63) |
| | **84.84**(2.85) | **67.00**(10.61) | **87.89** (0.90) | **55.92** (2.25) |
| FedYogi (Reddi et al., 2020) | 84.29(2.62) | 65.67(10.68) | 86.70 (1.40) | 52.81 (3.50) |
| | **84.86**(3.01) | **67.00**(11.09) | **87.42** (0.94) | **54.76** (3.11) |
| FedAdagrad (Reddi et al., 2020) | 84.61(2.96) | 65.67(10.68) | 83.52 (0.63) | 45.09 (1.79) |
| | **85.09**(2.91) | **66.67**(10.37) | **84.62** (0.51) | **47.88** (1.95) |

**Plug-and-Play Boosting** We additionally check if AAggFF can also boost other FL algorithms than FedAvg, such as FedAdam, FedAdagrad, FedYogi (Reddi et al., 2020) and FedProx (Li et al., 2020c). Since the sequential decision making procedure required in AAggFF is about finding a good mixing coefficient, $\boldsymbol{p}$, this is orthogonal to the minimization of $\boldsymbol{\theta}$. Thus, our method can be easily integrated into existing methods with no special modification, in a plug-and-play manner.

For the verification, we test with two more datasets, Heart (Janosi et al., 1988) and TinyImageNet (Le & Yang, 2015), each of which is suited for binary and multi-class classification (i.e., 200 classes in total). Since the Heart dataset is a part of FLamby benchmark (Ogier du Terrail et al., 2022), it has pre-defined $K = 4$ clients. For the TinyImageNet dataset, we simulate statistical heterogeneity for $K = 1,000$ clients using Dirichlet distribution with a concentration of 0.01, following (Hsu et al., 2019). The results are in Table 7, where the upper cell represents the performance of a naive FL algorithm, and the lower cell contains a performance of the FL algorithm with AAggFF. While the average performance remains comparable, the worst performance is consistently boosted in both cross-silo and cross-device settings. This underpins the efficacy and flexibility of AAggFF, which can strengthen the fairness perspective of existing FL algorithms.

## 7. Limitations and Future Works

Our work suggests interesting future directions for better federated systems, which may also be a limitation of the current work. First, we can exploit side information (e.g., parameters of local updates) to not preserve all clients' mixing coefficients, and filter out malicious signals for robustness. For example, the former can be realized by adopting other decision making schemes such as contextual Bayesian optimization (Char et al., 2019), and the latter can be addressed by clustered FL (Ghosh et al., 2020; Sattler et al., 2020) for a group-wise estimation of mixing coefficients. Both directions are promising and may improve the practicality of federated systems. Furthermore, the FTRL objective can be replaced by the Follow-The-Perturbed-Leader (FTPL) (Kalai & Vempala, 2005), of which random perturbation in decision making process can be directly linked to the differential privacy (DP (Dwork, 2006)) guarantee (McMillan, 2018), which is frequently considered for the cross-silo setting. Last but not least, further convergence analysis is required w.r.t. the parameter perspective along with mixing coefficients, e.g., using bi-level optimization formulation.

## 8. Conclusion

For improving the degree of the client-level fairness in FL, we first reveal the connection of existing fair FL methods with the OCO. To emphasize the sequential decision making perspective, we propose improved designs and further specialize them into two practical settings: cross-silo FL & cross-device FL. Our framework not only efficiently enhances a low-performing group of clients compared to existing baselines, but also maintains an acceptable average performance with theoretically guaranteed behaviors. It should also be noted that AAggFF requires *no extra communication* and *no added local computation*, which are significant constraints for serving FL-based services. With this scalability, our method can also improve the fairness of the performance distributions of existing FL algorithms without much modification to their original mechanism. By explicitly bringing the sequential decision making scheme to the front, we expect our work to open up new designs to promote the practicality and scalability of FL.

## Acknowledgements

This research was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korea Government (MSIT) (No. 2020R1C1C1011063), Basic Science Research Program through the NRF Grant funded by the Ministry of Education (No. NRF-2022R1I1A4069163) and the NRF (No. NRF-2020S1A3A2A02093277). Gi-Soo Kim was supported by the Institute of Information & communications Technology Planning & evaluation (IITP) grants funded by the Korea government (MSIT) (No. RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST); No. 2022-0-00469, Development of Core Technologies for Task-oriented Reinforcement Learning for Commercialization of Autonomous Drones).

## Impact Statement

Federated Learning (FL) poses a potential risk of skewed performance distributions toward partial groups of clients. We believe that our proposed method can mitigate such an unfair situation in the federated system, thereby enhancing the trustworthiness and social welfare. A potential concern is the situation where malicious clients deliberately inflate their local signals to cause global updates to be biased towards themselves. To prevent such a catastrophic situation, we aim to improve the current work by exploiting information other than local losses, as mentioned in the Limitations and Future Works section. The authors are not aware of any other critical ethical/social implications otherwise, but are open to discussing them if they exist.

## References

Abernethy, J. D., Hazan, E., and Rakhlin, A. Competing in the dark: An efficient algorithm for bandit linear optimization. 2009.

Agarwal, A. and Hazan, E. New algorithms for repeated play and universal portfolio management. Technical report, Princeton University Technical Report TR-740-05, 2005.

Agarwal, A., Hazan, E., Kale, S., and Schapire, R. E. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd international conference on Machine learning*, pp. 9–16, 2006.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Bang, H. and Robins, J. M. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61 (4):962–973, 2005.

Beck, A. and Teboulle, M. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Ben Abacha, A. and Demner-Fushman, D. A question-entailment approach to question answering. *BMC Bioinform.*, 20(1):511:1–511:23, 2019. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3119-4.

Berka, P. Workshop notes on discovery challenge pkdd'99. 1999. URL http://lisp.vse.cz/pkdd99/.

Caldas, S., Duddu, S. M. K., Wu, P., Li, T., Konečnỳ, J., McMahan, H. B., Smith, V., and Talwalkar, A. Leaf: A benchmark for federated settings. *Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

Cesa-Bianchi, N. and Lugosi, G. *Prediction, learning, and games*. Cambridge university press, 2006.

Char, I., Chung, Y., Neiswanger, W., Kandasamy, K., Nelson, A. O., Boyer, M., Kolemen, E., and Schneider, J. Offline contextual bayesian optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Chen, H., Zhu, T., Zhang, T., Zhou, W., and Yu, P. S. Privacy and fairness in federated learning: on the perspective of trade-off. *ACM Computing Surveys*, 2023.

Cheng, R. and Amin, N. Estimating parameters in continuous univariate distributions with a shifted origin. *Journal of the Royal Statistical Society: Series B (Methodological)*, 45(3):394–403, 1983.

Chi, J., Tian, Y., Gordon, G. J., and Zhao, H. Understanding and mitigating accuracy disparity in regression. In *International conference on machine learning*, pp. 1866–1876. PMLR, 2021.

Codella, N. C., Gutman, D., Celebi, M. E., Helba, B., Marchetti, M. A., Dusza, S. W., Kalloo, A., Liopyris, K., Mishra, N., Kittler, H., et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 international symposium on biomedical imaging (isbi), hosted by the international skin imaging collaboration (isic). In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pp. 168–172. IEEE, 2018.

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

Cover, T. M. Universal portfolios. *Mathematical finance*, 1 (1):1–29, 1991.

Curry, D. Android statistics (2023). *BusinessofApps*, 2023. URL https://www.businessofapps.com/data/android-statistics/.

Dai, W., Dai, C., Qu, S., Li, J., and Das, S. Very deep convolutional neural networks for raw waveforms. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 421–425. IEEE, 2017.

Dayan, I., Roth, H. R., Zhong, A., Harouni, A., Gentili, A., Abidin, A. Z., Liu, A., Costa, A. B., Wood, B. J., Tsai, C.-S., et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature medicine*, 27 (10):1735–1743, 2021.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Deng, Y., Kamani, M. M., and Mahdavi, M. Distributionally robust federated averaging. *Advances in neural information processing systems*, 33:15111–15122, 2020.

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

Fréchet, M. Sur la loi de probabilité de l'écart maximum. *Ann. de la Soc. Polonaise de Math.*, 1927.

Gauß, C. F. *Theoria Motus Corporvm Coelestivm In Sectionibvs Conicis Solem Ambientivm*. Perthes et Besser, 1809.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. An efficient framework for clustered federated learning. *Advances in Neural Information Processing Systems*, 33: 19586–19597, 2020.

Gumbel, E. J. Les valeurs extrêmes des distributions statistiques. In *Annales de l'institut Henri Poincaré*, volume 5, pp. 115–158, 1935.

Hahn, S.-J., Jeong, M., and Lee, J. Connecting low-loss subspace for personalized federated learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 505–515, 2022.

Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

Hazan, E. and Kale, S. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80:165–188, 2010.

Hazan, E., Agarwal, A., and Kale, S. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69:169–192, 2007.

Helmbold, D. P., Schapire, R. E., Singer, Y., and Warmuth, M. K. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Hsu, T.-M. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

Hu, Z., Shaloudegi, K., Zhang, G., and Yu, Y. Federated learning meets multi-objective optimization. *IEEE Transactions on Network Science and Engineering*, 9(4):2039–2051, 2022.

Janosi, A., Steinbrunn, W., Pfisterer, M., and Detrano, R. Heart disease. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C52P4X.

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

Kalai, A. and Vempala, S. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Kotz, S. and Nadarajah, S. *Extreme value distributions: theory and applications*. world scientific, 2000.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Li, T., Sanjabi, M., Beirami, A., and Smith, V. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

Li, T., Beirami, A., Sanjabi, M., and Smith, V. Tilted empirical risk minimization. *arXiv preprint arXiv:2007.01162*, 2020a.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020b.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020c.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

McCreery, C. H., Katariya, N., Kannan, A., Chablani, M., and Amatriain, X. Effective transfer learning for identifying similar questions: matching user questions to covid-19 faqs. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3458–3465, 2020.

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

McMahan, H. B. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

McMillan, A. *Differential Privacy, Property Testing, and Perturbations*. PhD thesis, 2018.

Mo, J. and Walrand, J. Fair end-to-end window-based congestion control. *IEEE/ACM Transactions on networking*, 8(5):556–567, 2000.

Mohri, M., Sivek, G., and Suresh, A. T. Agnostic federated learning. In *International Conference on Machine Learning*, pp. 4615–4625. PMLR, 2019.

Nemirovskij, A. S. and Yudin, D. B. Problem complexity and method efficiency in optimization. 1983.

Ogier du Terrail, J., Ayed, S.-S., Cyffers, E., Grimberg, F., He, C., Loeb, R., Mangold, P., Marchand, T., Marfoq, O., Mushtaq, E., et al. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

Orabona, F. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečnỳ, J., Kumar, S., and McMahan, H. B. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Robins, J. M., Rotnitzky, A., and Zhao, L. P. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, pp. 846–866, 1994.

Rusnock, P. and Kerr-Lawson, A. Bolzano and uniform continuity. *Historia Mathematica*, 32(3):303–311, 2005.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

Sattler, F., Müller, K.-R., and Samek, W. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Shalev-Shwartz, S. and Singer, Y. Online learning meets optimization in the dual. In *International Conference on Computational Learning Theory*, pp. 423–437. Springer, 2006.

Shalev-Shwartz, S. et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Warden, P. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Warmuth, M. K., Jagota, A. K., et al. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *Electronic proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, volume 326. Citeseer, 1997.

Weibull, W. A statistical distribution function of wide applicability. *Journal of applied mechanics*, 1951.

Zafar, M. B., Valera, I., Gomez Rodriguez, M., and Gummadi, K. P. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pp. 1171–1180, 2017.

Zhang, G., Malekmohammadi, S., Chen, X., and Yu, Y. Proportional fairness in federated learning. *arXiv preprint arXiv:2202.01666*, 2022.

Zhao, H., Coston, A., Adel, T., and Gordon, G. J. Conditional learning of fair representations. *arXiv preprint arXiv:1910.07162*, 2019.

Zinkevich, M. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pp. 928–936, 2003.

# A. Derivations & Proofs

## A.1. Derivation of Mixing Coefficients from Existing Methods

In this section, we provide details of the unification of existing methods in the OCO framework, introduced in Section 3. We assume full-client participation for derivation, and we denote $n = \sum_{i=1}^{K} n_i$ as a total sample size for the brevity of notation.

Suppose any FL algorithms follow the update formula in (13), where we define $\boldsymbol{p}^{(t+1)}$ as a *mixing coefficient* vector discussed in Section 3.2.

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \left( \sum_{i=1}^{K} p_i^{(t+1)} \left( \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)} \right) \right), \tag{13}$$

where $\boldsymbol{\theta}^{(t)}$ is a global model in a previous round $t$, $\boldsymbol{\theta}_i^{(t+1)}$ is a local update from $i$-th client starting from $\boldsymbol{\theta}^{(t)}$, and $\boldsymbol{\theta}^{(t+1)}$ is a new global model updated by averaging local updates with corresponding mixing coefficient $p_i^{(t+1)}$.

**FedAvg** (McMahan et al., 2017) The update of a global model from `FedAvg` is defined as follows.

$$\boldsymbol{\theta}^{(t+1)} \leftarrow \boldsymbol{\theta}^{(t)} - \left( \sum_{i=1}^{K} \frac{n_i}{n} \left( \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)} \right) \right), \tag{14}$$

where $n_i$ is the sample size of client $i$. Thus, we can regard $p_i^{(t+1)} \propto n_i$ in `FedAvg`.

**AFL & q-FedAvg** (Mohri et al., 2019; Li et al., 2019) The objective of `AFL` is a minimax objective defined as follows.

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\boldsymbol{v} \in \Delta_{K-1}} \sum_{i=1}^{K} v_i F_i \left( \boldsymbol{\theta} \right), \tag{15}$$

which is later subsumed by `q-FedAvg` as its special case for the algorithm-specific constant $q$, where $q \to 0$.

The objective of `q-FedAvg` is therefore defined with a nonnegative constant $q$ as follows.

$$\begin{aligned} &\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^{K} \frac{1}{q+1} \frac{n_i}{n} F_i^{q+1} \left( \boldsymbol{\theta} \right) \\ &= \min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^{K} \frac{n_i}{n} \tilde{F}_i \left( \boldsymbol{\theta} \right), \end{aligned} \tag{16}$$

which is reduced to `FedAvg` when $q = 0$.

The update of a global model from (16) has been proposed in the form of a Newton style update by assuming $L$-Lipschitz continuous gradient of each local objective (i.e., `q-FedSGD`) (Li et al., 2019).

$$\begin{aligned} \boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} - \left( \sum_{j=1}^{K} \frac{n_j}{n} \nabla^2 \tilde{F}_j \left( \boldsymbol{\theta}^{(t)} \right) \right)^{-1} \sum_{i=1}^{K} \frac{n_i}{n} \nabla \tilde{F}_i \left( \boldsymbol{\theta}^{(t)} \right) \\ &\preceq \boldsymbol{\theta}^{(t)} - \left( \sum_{j=1}^{K} \frac{n_j}{n} L_{q,j} \boldsymbol{I} \right)^{-1} \sum_{i=1}^{K} \frac{n_i}{n} F_i^q \left( \boldsymbol{\theta}^{(t)} \right) \nabla F_i \left( \boldsymbol{\theta}^{(t)} \right), \end{aligned} \tag{17}$$

where $L_{q,i} = q F_i^{q-1} \left( \boldsymbol{\theta}^{(t)} \right) \| \nabla F_i \left( \boldsymbol{\theta}^{(t)} \right) \|^2 + L F_i^q \left( \boldsymbol{\theta}^{(t)} \right)$ is an upper bound of the local Lipschitz gradient of $\tilde{F}_i \left( \boldsymbol{\theta}^{(t)} \right)$ (see Lemma 3 of (Li et al., 2019)). This can be extended to `q-FedAvg` by replacing $\nabla F_i \left( \boldsymbol{\theta}^{(t)} \right)$ into $L \left( \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)} \right)$. To sum up, the update formula of a global model from `q-FedAvg` (including `AFL` as a special case) is as follows.

$$\boldsymbol{\theta}^{(t+1)} \propto \boldsymbol{\theta}^{(t)} - \left( \sum_{i=1}^{K} \frac{\frac{n_i}{n} L F_i^q(\boldsymbol{\theta}^{(t)})}{\sum_{j=1}^{K} \frac{n_j}{n} L_{q,j}} \left( \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)} \right) \right), \tag{18}$$

which implies $p_i^{(t+1)} \propto n_i F_i^q \left(\boldsymbol{\theta}^{(t)}\right)$.

**TERM** (Li et al., 2020a) The objective of TERM is dependent upon a hyperparameter, a *tilting* constant $\lambda \in \mathbb{R}$.

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{1}{\lambda} \log\left(\sum_{i=1}^{K} \frac{n_i}{n} \exp\left(\lambda F_i(\boldsymbol{\theta}^{(t)})\right)\right) \tag{19}$$

The corresponding update formula is given as follows.

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \left(\sum_{i=1}^{K} \frac{(n_i/n)\exp\left(\lambda F_i\left(\boldsymbol{\theta}^{(t)}\right)\right)}{\sum_{j=1}^{K}(n_j/n)\exp\left(\lambda F_j\left(\boldsymbol{\theta}^{(t)}\right)\right)}\left(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)}\right)\right) \tag{20}$$

From the update formula, we can conclude that $p_i^{(t+1)} \propto n_i \exp\left(\lambda F_i\left(\boldsymbol{\theta}^{(t)}\right)\right)$.

**PropFair** (Zhang et al., 2022) The objective of PropFair is to maximize Nash social welfare by regarding a negative local loss as an achieved utility as follows.

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} -\sum_{i=1}^{K} p_i \log\left(M - F_i\left(\boldsymbol{\theta}\right)\right), \tag{21}$$

where $M \geq 1$ is a problem-specific constant.

The corresponding update formula is given as follows.

$$\boldsymbol{\theta}^{(t+1)} \propto \boldsymbol{\theta}^{(t)} + \left(\sum_{i=1}^{K} \frac{n_i}{n} \nabla \log\left(M - F_i\left(\boldsymbol{\theta}^{(t)}\right)\right)\right) = \boldsymbol{\theta}^{(t)} - \left(\sum_{i=1}^{K} \frac{n_i}{n} \frac{\nabla F_i\left(\boldsymbol{\theta}^{(t)}\right)}{M - F_i\left(\boldsymbol{\theta}^{(t)}\right)}\right). \tag{22}$$

Similar to q-FedAvg, by replacing the gradient $\nabla F_i\left(\boldsymbol{\theta}^{(t)}\right)$ into $\left(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)}\right)$, the update formula finally becomes:

$$\boldsymbol{\theta}^{(t+1)} \propto \boldsymbol{\theta}^{(t)} - \left(\sum_{i=1}^{K} \frac{n_i/n}{M - F_i\left(\boldsymbol{\theta}^{(t)}\right)}\left(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)}\right)\right), \tag{23}$$

which implies $p_i^{(t+1)} \propto \frac{n_i}{M - F_i\left(\boldsymbol{\theta}^{(t)}\right)}$.

### A.2. Technical Lemmas

In this section, we provide technical lemmas and proofs (including deferred ones in the main text) required for proving Theorem 5.1, Theorem 5.2, and Corollary 5.3.

#### A.2.1. Strict Convexity of Decision Loss

**Lemma A.1.** *For all $t \in [T]$, the decision loss $\ell^{(t)}$ defined in (4) satisfies following for $\gamma \in (0,1)$, i.e., the decision loss is a strictly convex function of its first argument.*

$$\ell^{(t)}\left(\gamma\boldsymbol{p} + (1-\gamma)\boldsymbol{q}\right) < \gamma\ell^{(t)}\left(\boldsymbol{p}\right) + (1-\gamma)\ell^{(t)}\left(\boldsymbol{q}\right), \forall \boldsymbol{p}, \boldsymbol{q} \in \Delta_{K-1}, \boldsymbol{p} \neq \boldsymbol{q}. \tag{24}$$

*Proof.* From the left-hand side, we have

$$\begin{aligned} &\ell^{(t)}\left(\gamma\boldsymbol{p} + (1-\gamma)\boldsymbol{q}\right) \\ =& -\log\left(1 + \langle \gamma\boldsymbol{p} + (1-\gamma)\boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle\right) \\ =& -\log\left(1 + \langle \gamma\boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle + \langle (1-\gamma)\boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle\right) \\ =& -\log\left(\gamma(1 + \langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle) + (1-\gamma)(1 + \langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle)\right). \end{aligned} \tag{25}$$

Since the negative of logarithm is strictly convex, the last term becomes

$$-\log\left(\gamma(1+\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle)+(1-\gamma)(1+\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle)\right) < \gamma\left(-\log(1+\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle)\right)+(1-\gamma)\left(-\log(1+\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle)\right), \quad (26)$$

which satisfies the definition of the strict convexity, thereby concludes the proof. ■

### A.2.2. Lipschitz Continuity of Decision Loss (Lemma 4.1)

From the definition of the Lipschitz continuity w.r.t. $\|\cdot\|$, we need to check if the decision loss $\ell^{(t)}$ satisfies following inequality for the constant $L_\infty$.

$$\left|\ell^{(t)}(\boldsymbol{p}) - \ell^{(t)}(\boldsymbol{q})\right| \leq L_\infty \|\boldsymbol{p}-\boldsymbol{q}\|_\infty . \quad (27)$$

*Proof.* From Lemma A.1, we have the following inequality from the convexity of the decision loss.

$$\begin{aligned}\left|\ell^{(t)}(\boldsymbol{p}) - \ell^{(t)}(\boldsymbol{q})\right| &\leq \left|\langle \nabla \ell^{(t)}(\boldsymbol{p}), \boldsymbol{p}-\boldsymbol{q}\rangle\right| \\ &= \left|-\frac{\langle \boldsymbol{p}-\boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle}{1+\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle}\right| \\ &= \frac{1}{1+\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle}\left|\left\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\right\rangle - \left\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\right\rangle\right|\end{aligned} \quad (28)$$

Setting the denominator to be the minimum value, $\left\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\right\rangle$ is $C_1$, we have the upper bound as follows.

$$\begin{aligned}&\frac{1}{1+\left\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\right\rangle}\left|\left\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\right\rangle - \left\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\right\rangle\right| \\ &\leq \frac{1}{1+\left\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\right\rangle}\max\left(\left\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\right\rangle, \left\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\right\rangle\right) \\ &\leq \frac{1}{1+C_1}\max\left(\left\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\right\rangle, C_1\right),\end{aligned} \quad (29)$$

where the first inequality is from the fact that both $\langle \boldsymbol{p}, \boldsymbol{r}^{(t)}\rangle$ and $\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle$ are nonnegative, and the second inequality is due to the minimized denominator achieving the upper bound. Since $\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle$ can achieve its maximum as $C_2$, we can further bound as follows.

$$\frac{1}{1+C_1}\max(\langle \boldsymbol{q}, \boldsymbol{r}^{(t)}\rangle, C_1) \leq \frac{1}{1+C_1}\max(C_2, C_1) = \frac{C_2}{1+C_1} \quad (30)$$

Finally, using the fact that $\|\boldsymbol{p}-\boldsymbol{q}\|_\infty = \max_i |p_i - q_i| = 1$, we can conclude the statement by setting $L_\infty = \frac{C_2}{1+C_1}$. ■

### A.2.3. Unbiasedness of Doubly Robust Estimator (Lemma 4.3)

*Proof.* Denote the client sampling probability $C \in [0,1]$ in time $t$ as $P(i \in S^{(t)}) = C$. Taking expectation on the doubly robust estimator of partially observed response defined in (10), we have

$$\begin{aligned}\mathbb{E}\left[\breve{r}_i^{(t)}\right] &= \mathbb{E}\left[\left(1-\frac{\mathbb{I}(i\in S^{(t)})}{C}\right)\bar{\mathrm{r}}^{(t)}\right] + \mathbb{E}\left[\frac{\mathbb{I}(i\in S^{(t)})}{C} r_i^{(t)}\right] \\ &= \left(1-\frac{\mathbb{E}[\mathbb{I}(i\in S^{(t)})]}{C}\right)\bar{\mathrm{r}}^{(t)} + \frac{\mathbb{E}\left[\mathbb{I}(i\in S^{(t)})\right]}{C} r_i^{(t)} \\ &= \left(1-\frac{P(i\in S^{(t)})}{C}\right)\bar{\mathrm{r}}^{(t)} + \frac{P(i\in S^{(t)})}{C} r_i^{(t)} \\ &= r_i^{(t)},\end{aligned} \quad (31)$$

where $\mathbb{I}(\cdot)$ is an indicator function.

Note that the randomness of the doubly robust estimator comes from the random sampling of client indices $i \in S^{(t)}$ in round $t$, thus the expectation is with respect to $i \in S^{(t)}$. Thus, we can conclude that $\mathbb{E}\left[\breve{\boldsymbol{r}}^{(t)}\right] = \boldsymbol{r}^{(t)}$. See also (Robins et al., 1994; Bang & Robins, 2005). ■

### A.2.4. UNBIASEDNESS OF LINEARLY APPROXIMATED GRADIENT (LEMMA 4.4)

*Proof.* The gradient of a decision loss in terms of a response, $\boldsymbol{g} \equiv \mathbf{h}(\boldsymbol{r}) = [h_1(\boldsymbol{r}), ..., h_K(\boldsymbol{r})]^\top = -\frac{\boldsymbol{r}}{1+\langle \boldsymbol{p}, \boldsymbol{r}\rangle}$ can be linearly approximated at reference $\boldsymbol{r}_0$ as follows.

$$\tilde{\mathbf{h}}(\boldsymbol{r}) = \mathbf{h}(\boldsymbol{r}_0) + \mathbf{J}_{\mathbf{h}}(\boldsymbol{r}_0)(\boldsymbol{r} - \boldsymbol{r}_0) \tag{32}$$

The Jacobian $\mathbf{J}_{\mathbf{h}}(\boldsymbol{r}) \in \mathbb{R}^{K\times K}$ is defined as follows.

$$\begin{aligned}
\mathbf{J}_{\mathbf{h}}(\boldsymbol{r}) &= \left[\frac{\partial \mathbf{h}}{\partial r_1}, ..., \frac{\partial \mathbf{h}}{\partial r_K}\right] \\
&= \begin{bmatrix} \frac{\partial h_1}{\partial r_1} & \cdots & \frac{\partial h_1}{\partial r_K} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_K}{\partial r_1} & \cdots & \frac{\partial h_K}{\partial r_K} \end{bmatrix} \\
&= \begin{bmatrix} -\frac{1}{1+\langle \boldsymbol{p},\boldsymbol{r}\rangle} + \frac{p_1 r_1}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} & \frac{p_2 r_1}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} & \cdots & \frac{p_K r_1}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} \\ \frac{p_1 r_2}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} & -\frac{1}{1+\langle \boldsymbol{p},\boldsymbol{r}\rangle} + \frac{p_2 r_2}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} & \cdots & \frac{p_K r_2}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{p_1 r_K}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} & \frac{p_2 r_K}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} & \cdots & -\frac{1}{1+\langle \boldsymbol{p},\boldsymbol{r}\rangle} + \frac{p_K r_K}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2} \end{bmatrix} \\
&= -\frac{1}{1+\langle \boldsymbol{p},\boldsymbol{r}\rangle}\boldsymbol{I}_K + \frac{1}{(1+\langle \boldsymbol{p},\boldsymbol{r}\rangle)^2}\boldsymbol{r}\boldsymbol{p}^\top
\end{aligned} \tag{33}$$

Plugging (33) into (32) with respect to arbitrary reference $\boldsymbol{r}_0$, we have a linearized gradient of a decision loss as follows.

$$\begin{aligned}
\tilde{\boldsymbol{g}} \triangleq \tilde{\mathbf{h}}(\boldsymbol{r}) &= -\frac{\boldsymbol{r}_0}{1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle} - \frac{(\boldsymbol{r}-\boldsymbol{r}_0)}{1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle} + \frac{\boldsymbol{r}_0\boldsymbol{p}^\top(\boldsymbol{r}-\boldsymbol{r}_0)}{(1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle)^2} \\
&= -\frac{\boldsymbol{r}}{1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle} + \frac{\boldsymbol{r}_0\boldsymbol{p}^\top(\boldsymbol{r}-\boldsymbol{r}_0)}{(1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle)^2}.
\end{aligned} \tag{34}$$

From the statement of Lemma 4.4, plugging the doubly robust estimator of the partially observed response, $\breve{\boldsymbol{r}}$ from Lemma 4.3 into above, we have gradient estimate $\breve{\boldsymbol{g}}$ as follows.

$$\breve{\boldsymbol{g}} = \tilde{\mathbf{h}}(\breve{\boldsymbol{r}}) = -\frac{\breve{\boldsymbol{r}}}{1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle} + \frac{\boldsymbol{r}_0\boldsymbol{p}^\top(\breve{\boldsymbol{r}}-\boldsymbol{r}_0)}{(1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle)^2}. \tag{35}$$

Taking an expectation, we have

$$\begin{aligned}
\mathbb{E}\left[\breve{\boldsymbol{g}}\right] = \mathbb{E}\left[\tilde{\mathbf{h}}(\breve{\boldsymbol{r}})\right] &= -\frac{\mathbb{E}\left[\breve{\boldsymbol{r}}\right]}{1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle} + \frac{\boldsymbol{r}_0\boldsymbol{p}^\top(\mathbb{E}\left[\breve{\boldsymbol{r}}\right]-\boldsymbol{r}_0)}{(1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle)^2} = -\frac{\boldsymbol{r}}{1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle} + \frac{\boldsymbol{r}_0\boldsymbol{p}^\top(\boldsymbol{r}-\boldsymbol{r}_0)}{(1+\langle \boldsymbol{p},\boldsymbol{r}_0\rangle)^2} \\
&= \tilde{\mathbf{h}}(\boldsymbol{r}) = \tilde{\boldsymbol{g}} \approx \boldsymbol{g}.
\end{aligned} \tag{36}$$

■

### A.2.5. LIPSCHITZ CONTINUITY OF LINEARLY APPROXIMATED GRADIENT FROM DOUBLY ROBUST ESTIMATOR

**Lemma A.2.** *Denote $\breve{\boldsymbol{g}}^{(t)}$ as the linearized gradient calculated from the doubly robust estimator of a response vector, $\boldsymbol{r}^{(t)}$, with reference $\boldsymbol{r}_0^{(t)} = \bar{\boldsymbol{r}} = \bar{\mathrm{r}}^{(t)}\mathbf{1}_K$ where $\bar{\mathrm{r}}^{(t)} = \frac{1}{|S^{(t)}|}\sum_{i\in S^{(t)}} r_i^{(t)}$. When $S^{(t)}$ is a randomly selected client indices in round $t$ and $C = P\left(i \in S^{(t)}\right)$ is a client sampling probability, then $\left\|\breve{\boldsymbol{g}}^{(t)}\right\|_\infty \leq \breve{L}_\infty = \frac{C_2}{1+C_1} + \frac{2(C_2-C_1)}{C(1+C_1)}$ for $r_i^{(t)} \in [C_1, C_2], \forall i \in S^{(t)}$.*

*Proof.* Note that we intentionally omit superscript $^{(t)}$ from now on for the brevity of notation. The linearized gradient constructed from the doubly robust estimator of a response vector has a form as follows, according to (34).

$$\breve{\boldsymbol{g}} = -\frac{\breve{\boldsymbol{r}}}{1+\langle \boldsymbol{p},\bar{\boldsymbol{r}}\rangle} + \frac{\bar{\boldsymbol{r}}\boldsymbol{p}^\top(\breve{\boldsymbol{r}}-\bar{\boldsymbol{r}})}{(1+\langle \boldsymbol{p},\bar{\boldsymbol{r}}\rangle)^2}, \tag{37}$$

where we used $\bar{\boldsymbol{r}} = \bar{\mathrm{r}}\mathbf{1}_K$ as a reference $\boldsymbol{r}_0$, therefore $\|\bar{\boldsymbol{r}}\|_\infty = \bar{\mathrm{r}} \leq C_2$.

Thus, we have

$$\begin{aligned}
&\|\breve{\boldsymbol{g}}\|_\infty \\
&= \left\| -\frac{\breve{\boldsymbol{r}}}{1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle} + \frac{\bar{\boldsymbol{r}}\boldsymbol{p}^\top(\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}})}{(1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle)^2} \right\|_\infty \\
&\leq \left\| -\frac{\breve{\boldsymbol{r}}}{1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle} \right\|_\infty + \left\| \frac{\bar{\boldsymbol{r}}\boldsymbol{p}^\top(\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}})}{(1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle)^2} \right\|_\infty \\
&= \frac{\|\breve{\boldsymbol{r}}\|_\infty}{1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle} + \frac{1}{(1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle)^2} \left\| \bar{\boldsymbol{r}}\boldsymbol{p}^\top(\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}) \right\|_\infty \\
&\leq \frac{\|\breve{\boldsymbol{r}}\|_\infty}{1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle} + \frac{1}{(1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle)^2} \left\| \bar{\boldsymbol{r}}\boldsymbol{p}^\top \right\|_\infty \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \\
&= \frac{\|\breve{\boldsymbol{r}}\|_\infty}{1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle} + \frac{1}{(1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle)^2} \|\bar{\boldsymbol{r}}\|_\infty \|\boldsymbol{p}\|_1 \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \\
&= \frac{\|\breve{\boldsymbol{r}}\|_\infty}{1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle} + \frac{1}{(1+\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle)^2} \|\bar{\boldsymbol{r}}\|_\infty \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty
\end{aligned}$$

, where the first inequality is due to triangle inequality, the second inequality is due to the property that $\|\boldsymbol{A}\boldsymbol{x}\|_\infty \leq \|\boldsymbol{A}\|_\infty \|\boldsymbol{x}\|_\infty$ for a matrix $\boldsymbol{A} \in \mathbb{R}^{K\times K}$ and a vector $\boldsymbol{x} \in \mathbb{R}^K, \boldsymbol{x} \neq \mathbf{0}_K$, the very next equality is due to $|\boldsymbol{x}\boldsymbol{y}^\top|_\infty = \max_i \|x_i \boldsymbol{y}^\top\|_1 = \max_i |x_i| \|\boldsymbol{y}\|_1 = \|\boldsymbol{x}\|_\infty \|\boldsymbol{y}\|_1$, and the last equality is trivial since $\boldsymbol{p} \in \Delta_{K-1}$.

Since $\langle \boldsymbol{p}, \bar{\boldsymbol{r}}\rangle = \sum_{i=1}^K (p_i \bar{\mathrm{r}}) = \bar{\mathrm{r}}$, this can be further bounded as follows.

$$\begin{aligned}
&= \frac{1}{1+\bar{\mathrm{r}}} \|\breve{\boldsymbol{r}}\|_\infty + \frac{\bar{\mathrm{r}}}{(1+\bar{\mathrm{r}})^2} \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \\
&= \frac{1+\bar{\mathrm{r}}}{(1+\bar{\mathrm{r}})^2} \|\breve{\boldsymbol{r}}\|_\infty + \frac{\bar{\mathrm{r}}}{(1+\bar{\mathrm{r}})^2} \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \\
&\leq \frac{1}{1+\bar{\mathrm{r}}} \left( \|\breve{\boldsymbol{r}}\|_\infty + \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \right),
\end{aligned} \tag{38}$$

Since $\frac{1}{1+\bar{\mathrm{r}}} \leq \frac{1}{1+C_1}$, we can further upper bound as follows.

$$\frac{1}{1+\bar{\mathrm{r}}} \left( \|\breve{\boldsymbol{r}}\|_\infty + \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \right) \leq \frac{1}{1+C_1} \left( \|\breve{\boldsymbol{r}}\|_\infty + \|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \right) \tag{39}$$

To upper bound each term, let us look into $\breve{\boldsymbol{r}}$ first. By the definition in (10), we have

$$\breve{\boldsymbol{r}} = \begin{cases} \bar{\mathrm{r}}\mathbf{1}_K, & i \notin S^{(t)} \\ \left(1 - \frac{1}{C}\right) \bar{\mathrm{r}}\mathbf{1}_K + \frac{1}{C}\boldsymbol{r}, & i \in S^{(t)} \end{cases}. \tag{40}$$

For each case, $\|\breve{\boldsymbol{r}}\|_\infty$ becomes

$$\|\breve{\boldsymbol{r}}\|_\infty = \begin{cases} \bar{\mathrm{r}}, & i \notin S^{(t)} \\ \sup_i \left| \frac{1}{C}(r_i - \bar{\mathrm{r}}) + \bar{\mathrm{r}} \right|, & i \in S^{(t)}. \end{cases} \tag{41}$$

For the first case, the average is smaller than its maximum, thus $\bar{\mathrm{r}} \leq C_2$. For the second case, it can be upper bounded as $\sup_i \left| \frac{1}{C}(r_i - \bar{\mathrm{r}}) + \bar{\mathrm{r}} \right| \leq \frac{1}{C} \sup_i |r_i - \bar{\mathrm{r}}| + C_2$ by the triangle inequality.

From the trivial fact that the deviation from the average is always smaller than its range,

$$\frac{1}{C} \sup_i |r_i - \bar{\mathrm{r}}| \leq \frac{1}{C}(C_2 - C_1). \tag{42}$$

Combined, we have the following upper bounds.

$$\|\breve{\boldsymbol{r}}\|_\infty \leq \begin{cases} C_2, & i \notin S^{(t)} \\ \frac{C_2 - C_1}{C} + C_2, & i \in S^{(t)} \end{cases} \tag{43}$$

Similarly, for the second term inside in (39), we have:

$$\|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty = \begin{cases} 0, & i \notin S^{(t)} \\ \frac{1}{C} \sup_i |r_i - \bar{\mathrm{r}}|, & i \in S^{(t)}. \end{cases} \tag{44}$$

Corresponding upper bounds are:

$$\|\breve{\boldsymbol{r}} - \bar{\boldsymbol{r}}\|_\infty \leq \begin{cases} 0, & i \notin S^{(t)} \\ \frac{C_2 - C_1}{C}, & i \in S^{(t)} \end{cases} \tag{45}$$

Finally, adding (43) and (45) to have (39), we have:

$$\left\|\breve{\boldsymbol{g}}^{(t)}\right\|_\infty \leq \begin{cases} \frac{C_2}{1+C_1}, & i \notin S^{(t)} \\ \frac{C_2}{1+C_1} + \frac{2(C_2 - C_1)}{C(1+C_1)}, & i \in S^{(t)}. \end{cases} \tag{46}$$

Finally, it suffices to set $\breve{L}_\infty = \frac{C_2}{1+C_1} + \frac{2(C_2-C_1)}{C(1+C_1)}$ to conclude the proof. ■

### A.2.6. Regret from a Linearized Loss

**Corollary A.3.** *From the convexity of a decision loss $\ell^{(t)}$ (Lemma A.1), the regret defined in (5) satisfies*

$$\text{Regret}^{(T)}(\boldsymbol{p}^\star) = \sum_{t=1}^T \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^T \ell^{(t)}(\boldsymbol{p}^\star) \leq \sum_{t=1}^T \tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^T \tilde{\ell}^{(t)}(\boldsymbol{p}^\star), \tag{47}$$

*where $\tilde{\ell}^{(t)}$ is a linearized loss defined as $\tilde{\ell}^{(t)}(\boldsymbol{p}) = \langle \boldsymbol{p}, \boldsymbol{g}^{(t)} \rangle$ and $\boldsymbol{g}^{(t)} = \nabla \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)$.*

*Proof.* It is straightforward from the convexity of the decision loss.

$$\ell^{(t)}(\boldsymbol{p}^{(t)}) - \ell^{(t)}(\boldsymbol{p}^\star) \leq \left\langle \boldsymbol{g}^{(t)}, \boldsymbol{p}^{(t)} - \boldsymbol{p}^\star \right\rangle. \tag{48}$$

Summing up both sides for $t \in [T]$, we proved the statement. ■

### A.2.7. Equality for the Regret

**Lemma A.4.** *(Lemma 7.1 of (McMahan, 2017); Lemma 5 of (Orabona, 2019)) Let us define $L^{(t)}(\boldsymbol{p}) \triangleq \sum_{\tau=1}^{t-1} \ell^{(\tau)}(\boldsymbol{p}) + R^{(t)}(\boldsymbol{p})$, where $\ell : \Delta_{K-1} \times \mathbb{R}^d \to \mathbb{R}$ is an arbitrary loss function and $R^{(t)} : \Delta_{K-1} \to \mathbb{R}$ is an arbitrary regularizer, non-decreasing across $t \in [T]$. Assume further that $\boldsymbol{p}^{(t)} = \arg\min_{\boldsymbol{p} \in \Delta_{K-1}} L^{(t)}(\boldsymbol{p})$. Then, for any $\boldsymbol{p}^\star \in \Delta_{K-1}$, we have*

$$\begin{aligned} \text{Regret}^{(T)}(\boldsymbol{p}^\star) &= \sum_{t=1}^T \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^T \ell^{(t)}(\boldsymbol{p}^\star) \\ &= R^{(T+1)}(\boldsymbol{p}^\star) - R^{(1)}\left(\boldsymbol{p}^{(1)}\right) + L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right) - L^{(T+1)}(\boldsymbol{p}^\star) \\ &+ \sum_{t=1}^T \left[ L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) \right]. \end{aligned} \tag{49}$$

*Proof.* Since $\sum_{t=1}^T \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)$ appears in both sides, we only need to show that

$$-\sum_{t=1}^T \ell^{(t)}(\boldsymbol{p}^\star) = R^{(T+1)}(\boldsymbol{p}^\star) - R^{(1)}\left(\boldsymbol{p}^{(1)}\right) + L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right) - L^{(T+1)}(\boldsymbol{p}^\star) + \sum_{t=1}^T \left[ L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) \right] \tag{50}$$

First, note that

$$\ell^{(t)}\left(\boldsymbol{p}^{\star}\right)=\sum_{\tau=1}^{t}\ell^{(\tau)}\left(\boldsymbol{p}^{\star}\right)-\sum_{\tau=1}^{t-1}\ell^{(\tau)}\left(\boldsymbol{p}^{\star}\right)=\left(L^{(t+1)}\left(\boldsymbol{p}^{\star}\right)-R^{(t+1)}\left(\boldsymbol{p}^{\star}\right)\right)-\left(L^{(t)}\left(\boldsymbol{p}^{\star}\right)-R^{(t)}\left(\boldsymbol{p}^{\star}\right)\right). \tag{51}$$

Summing up the right-hand side of the above from $t=1,...,T$, we have

$$\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{\star}\right)=\left(L^{(T+1)}\left(\boldsymbol{p}^{\star}\right)-R^{(T+1)}\left(\boldsymbol{p}^{\star}\right)\right)-\left(L^{(1)}\left(\boldsymbol{p}^{\star}\right)-R^{(1)}\left(\boldsymbol{p}^{\star}\right)\right)=L^{(T+1)}\left(\boldsymbol{p}^{\star}\right)-R^{(T+1)}\left(\boldsymbol{p}^{\star}\right), \tag{52}$$

by telescoping summation. Thus,

$$-\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{\star}\right)=R^{(T+1)}(\boldsymbol{p}^{\star})-L^{(T+1)}(\boldsymbol{p}^{\star}). \tag{53}$$

Similarly,

$$\sum_{t=1}^{T}\left[L^{(t)}\left(\boldsymbol{p}^{(t)}\right)-L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right]=L^{(1)}\left(\boldsymbol{p}^{(1)}\right)-L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right)=R^{(1)}\left(\boldsymbol{p}^{(1)}\right)-L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right). \tag{54}$$

Rearranging,

$$0=L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right)-R^{(1)}\left(\boldsymbol{p}^{(1)}\right)+\sum_{t=1}^{T}\left[L^{(t)}\left(\boldsymbol{p}^{(t)}\right)-L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right] \tag{55}$$

Adding (53) and (55), we have

$$-\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{\star}\right)=R^{(T+1)}\left(\boldsymbol{p}^{\star}\right)-R^{(1)}\left(\boldsymbol{p}^{(1)}\right)+L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right)-L^{(T+1)}\left(\boldsymbol{p}^{\star}\right)+\sum_{t=1}^{T}\left[L^{(t)}\left(\boldsymbol{p}^{(t)}\right)-L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right] \tag{56}$$

Finally, by adding $\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)$ to both sides, we prove the statement. Note that the left hand side of the main statement does not depend on $R^{(t)}$, thus we can replace $R^{(T+1)}\left(\boldsymbol{p}^{\star}\right)$ into $R^{(T)}\left(\boldsymbol{p}^{\star}\right)$. (Remark 7.3 of (Orabona, 2019)) ■

### A.2.8. Upper Bound to the Suboptimality Gap

**Lemma A.5.** *(Oracle Gap; Corollary 7.7 of (Orabona, 2019)) Let $f:\mathbb{R}^{K}\to\mathbb{R}$ be a $\mu$-strongly convex w.r.t. $\|\cdot\|$ over its domain. Let $\boldsymbol{x}^{\star}=\arg\min_{\boldsymbol{x}}f(\boldsymbol{x})$. Then, for all $\boldsymbol{x}\in\operatorname{dom}\partial f(\boldsymbol{x})$, and $\boldsymbol{g}\in\partial f(\boldsymbol{x})$, we have:*

$$f(\boldsymbol{x})-f(\boldsymbol{x}^{\star})\leq\frac{1}{2\mu}\|\boldsymbol{g}\|_{\star}^{2}, \tag{57}$$

*where $\|\cdot\|_{\star}$ is a dual norm of $\|\cdot\|$.*

### A.2.9. Progress Bound

**Lemma A.6.** *(Progress Bound of FTRL with Proximal Regularizer) With a slight abuse of notation, assume $L^{(t)}$ is closed, subdifferentiable and $\mu^{(t)}$-strongly convex w.r.t. $\|\cdot\|$ in $\Delta_{K-1}$. First assume that $\boldsymbol{p}^{(t+1)}=\arg\min_{\boldsymbol{p}\in\Delta_{K-1}}L^{(t+1)}\left(\boldsymbol{p}\right)$. Assume further that the regularizer satisfies $\boldsymbol{p}^{(t)}=\arg\min_{\boldsymbol{p}\in\Delta_{K-1}}\left(R^{(t+1)}\left(\boldsymbol{p}\right)-R^{(t)}\left(\boldsymbol{p}\right)\right)$, and $\boldsymbol{g}^{(t)}\in\partial L^{(t+1)}(\boldsymbol{p}^{(t)})$. Then, we have the following inequality:*

$$L^{(t)}\left(\boldsymbol{p}^{(t)}\right)-L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)+\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)\leq\frac{\left\|\boldsymbol{g}^{(t)}\right\|_{\star}^{2}}{2\mu^{(t+1)}}+\left(R^{(t)}\left(\boldsymbol{p}^{(t)}\right)-R^{(t+1)}\left(\boldsymbol{p}^{(t)}\right)\right), \tag{58}$$

*where $\|\cdot\|_{\star}$ is a dual norm of $\|\cdot\|$.*

*Proof.*

$$\begin{aligned}
&L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) \\
&= \left(L^{(t)}\left(\boldsymbol{p}^{(t)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) + R^{(t+1)}\left(\boldsymbol{p}^{(t)}\right) - R^{(t)}\left(\boldsymbol{p}^{(t)}\right)\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t+1)}\left(\boldsymbol{p}^{(t)}\right) + R^{(t)}\left(\boldsymbol{p}^{(t)}\right) \\
&= L^{(t+1)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t+1)}\left(\boldsymbol{p}^{(t)}\right) + R^{(t)}\left(\boldsymbol{p}^{(t)}\right) \\
&\leq \frac{\left\|\boldsymbol{g}^{(t)}\right\|_{\star}^{2}}{2\mu^{(t+1)}} - R^{(t+1)}\left(\boldsymbol{p}^{(t)}\right) + R^{(t)}\left(\boldsymbol{p}^{(t)}\right),
\end{aligned} \tag{59}$$

where the first inequality is due the assumption that $\boldsymbol{p}^{(t+1)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} L^{(t+1)}(\boldsymbol{p})$, $\boldsymbol{g}^{(t)} \in \partial L^{(t+1)}(\boldsymbol{p}^{(t)})$, and the result from Lemma A.5. See also Lemma 7.23 of (Orabona, 2019). ■

**Lemma A.7.** *(Progress Bound of FTRL with Non-Decreasing Regularizer) With a slight abuse of notation, assume $L^{(t)}$ to be closed and sub-differentiable in $\Delta_K$, and $\left(L^{(t)} + \ell^{(t)}\right)$ to be closed, differentiable and $\nu^{(t)}$-strongly convex w.r.t. $\|\cdot\|_1$ in $\Delta_{K-1}$. Further define with an abuse of notation again that $\boldsymbol{g}^{(t)} = \nabla\ell^{(t)}(\boldsymbol{p}^{(t)}) \in \partial\left(L^{(t)} + \ell^{(t)}\right)\left(\boldsymbol{p}^{(t)}\right)$, and define further that $\boldsymbol{p}^{(t)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} L^{(t)}(\boldsymbol{p})$. Then, we have the following inequality:*

$$L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) \leq \frac{\left\|\boldsymbol{g}^{(t)}\right\|_{\infty}^{2}}{2\nu^{(t)}} + \left(R^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right). \tag{60}$$

*Proof.* Let us first assume that $\boldsymbol{p}_*^{(t)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}}\left(L^{(t)}(\boldsymbol{p}) + \ell^{(t)}(\boldsymbol{p})\right)$. Observe that

$$L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) = L^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) + R^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right), \tag{61}$$

we have

$$\begin{aligned}
&L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) \\
&= L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \left(L^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) + R^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) \\
&= \left(L^{(t)}\left(\boldsymbol{p}^{(t)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)\right) - \left(L^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t+1)}\right)\right) + \left(R^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right) \\
&\leq \left(L^{(t)}\left(\boldsymbol{p}^{(t)}\right) + \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)\right) - \left(L^{(t)}\left(\boldsymbol{p}_*^{(t)}\right) + \ell^{(t)}\left(\boldsymbol{p}_*^{(t)}\right)\right) + \left(R^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right) \\
&\leq \frac{\left\|\boldsymbol{g}^{(t)}\right\|_{\infty}^{2}}{2\nu^{(t)}} + \left(R^{(t)}\left(\boldsymbol{p}^{(t+1)}\right) - R^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)\right),
\end{aligned} \tag{62}$$

where the first inequality is due the assumption that $\boldsymbol{p}_*^{(t)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}}\left(L^{(t)} + \ell^{(t)}\right)(\boldsymbol{p})$, $\boldsymbol{g}^{(t)} \in \partial\left(L^{(t)} + \ell^{(t)}\right)$. Lastly, the second inequality is the direct result from Lemma A.5. See also Lemma 7.8 of (Orabona, 2019). ■

### A.2.10. EXP-CONCAVITY

**Definition A.8.** A function $f : X \to \mathbb{R}$ is $\gamma$-exp-concave if $\exp\left(-\gamma f\left(\boldsymbol{x}\right)\right)$ is concave for $\boldsymbol{x} \in X$.

*Remark* A.9. The decision loss defined in (4) is 1-exp-concave.

**Lemma A.10.** *For an $\gamma$-exp-concave function $f : X \to \mathbb{R}$, let the domain $X$ is bounded, and choose $\delta \leq \frac{\gamma}{2}$ such that $|\delta\langle\nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle| \leq \frac{1}{2}$ and for all $\boldsymbol{x}, \boldsymbol{y} \in X$, the following statement holds.*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle\nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\delta}{2}\left(\langle\nabla f(\boldsymbol{y}), \boldsymbol{y} - \boldsymbol{x}\rangle\right)^2 \tag{63}$$

*Proof.* For all such that $\delta \leq \frac{\gamma}{2}$, a function $g(\boldsymbol{x}) = \exp\left(-2\delta f(\boldsymbol{x})\right)$ is concave. From the concavity of $g$, we have:

$$g(\boldsymbol{x}) \leq g(\boldsymbol{y}) + \langle\nabla g(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\rangle. \tag{64}$$

By taking logarithm on both sides, we have:

$$f(\boldsymbol{x}) \geq f(\boldsymbol{y}) - \frac{1}{2\delta}\log\left(1 - 2\delta\left\langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y}\right\rangle\right). \tag{65}$$

From the assumption, we have $|\delta\left\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\right\rangle| \leq \frac{1}{2}$, and using the fact that $\log(1+x) \leq x - \frac{x^2}{4}$ for $|x| \leq 1$, we can conclude the proof. ■

*Remark* A.11. (Remark 7.27 from (Orabona, 2019)) For a positive definite matrix $\boldsymbol{B} \in \mathbb{R}^{K\times K}$, $\|\boldsymbol{p}\|_{\boldsymbol{B}}$ is a norm induced by $\boldsymbol{B}$, defined as $\|\boldsymbol{p}\|_{\boldsymbol{B}} \triangleq \sqrt{\boldsymbol{p}^\top \boldsymbol{B}\boldsymbol{p}}$ for $\boldsymbol{p} \in \mathbb{R}^K$. A function $f(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{p}^\top \boldsymbol{B}\boldsymbol{p}$ is therefore 1-strongly convex w.r.t. $\|\cdot\|_{\boldsymbol{B}}$. Note that the dual norm of $\|\cdot\|_{\boldsymbol{B}}$ is $\|\cdot\|_{\boldsymbol{B}^{-1}}$.

### A.3. Regret Bound of AAggFF-S: Proof of Theorem 5.1

*Remark* A.12. The regularizer of AAggFF-S (i.e., ONS) is proximal since it has a quadratic form.

*Proof.* Since Lemma A.4 holds for arbitrary loss function, let us set $L^{(t)}(\boldsymbol{p}) \triangleq \sum_{\tau=1}^{t-1}\tilde{\ell}^{(\tau)}(\boldsymbol{p}) + \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{\tau=1}^{t-1}\left(\left\langle \boldsymbol{g}^{(\tau)}, \boldsymbol{p} - \boldsymbol{p}^{(\tau)}\right\rangle\right)^2$ as in (7) with a slight abuse of notation. Note that we set $R^{(t)}(\boldsymbol{p}) = \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{\tau=1}^{t-1}\left(\left\langle \boldsymbol{g}^{(\tau)}, \boldsymbol{p} - \boldsymbol{p}^{(\tau)}\right\rangle\right)^2$, which is often called a proximal regularizer.

From the regret, we have:

$$\begin{aligned}
&\text{Regret}^{(T)}(\boldsymbol{p}^\star)\\
&= \sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T}\ell^{(t)}(\boldsymbol{p}^\star)\\
&\leq \sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T}\tilde{\ell}^{(t)}(\boldsymbol{p}^\star)\\
&= \underbrace{R^{(T+1)}(\boldsymbol{p}^\star) - R^{(1)}\left(\boldsymbol{p}^{(1)}\right)}_{\text{(i)}} + \underbrace{L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right) - L^{(T+1)}(\boldsymbol{p}^\star)}_{\text{(ii)}}\\
&+ \underbrace{\sum_{t=1}^{T}\left[L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right)\right]}_{\text{(iii)}}
\end{aligned} \tag{66}$$

Let us first denote $\boldsymbol{A}^{(t)} \triangleq \boldsymbol{g}^{(t)}\boldsymbol{g}^{(t)\top}$ for $\boldsymbol{g}^{(t)} = \nabla\ell^{(t)}(\boldsymbol{p}^{(t)})$ as in the main text, and further denote that $\boldsymbol{B}^{(t)} \triangleq \alpha\boldsymbol{I}_K + \beta\sum_{\tau=1}^{t}\boldsymbol{A}^{(\tau)}$. Then, we can rewrite the regularizer of AAggFF-D as follows.

$$R^{(t)}(\boldsymbol{p}) = \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{\tau=1}^{t-1}\left(\left\langle \boldsymbol{g}^{(\tau)}, \boldsymbol{p} - \boldsymbol{p}^{(\tau)}\right\rangle\right)^2 = \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{\tau=1}^{t-1}\left\|\boldsymbol{p}^{(\tau)} - \boldsymbol{p}\right\|_{\boldsymbol{A}^{(\tau)}}^2 \tag{67}$$

That is, $R^{(t)}(\boldsymbol{p})$, as well as $L^{(t)}(\boldsymbol{p})$ is $\beta$-strongly convex function w.r.t. $\|\cdot\|_{\boldsymbol{B}^{(t-1)}}, t \in [T]$.

For (i), since the regularizer $R^{(t)}(\boldsymbol{p})$ is nonnegative for all $t \in [T]$, it can be upper bounded as $R^{(T+1)}(\boldsymbol{p}^\star)$. Using (67), we have:

$$R^{(T+1)}(\boldsymbol{p}^\star) = \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\sum_{t=1}^{T}\left\|\boldsymbol{p}^{(t)} - \boldsymbol{p}^\star\right\|_{\boldsymbol{A}^{(t)}}^2. \tag{68}$$

For (ii), we use the assumption in Lemma A.7, where $\boldsymbol{p}^{(t)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} L^{(t)}(\boldsymbol{p})$. From the assumption, since $\boldsymbol{p}^{(T+1)}$ is the minimizer of $L^{(T+1)}$, (ii) becomes negative. Thus, we can exclude it in further upper bounding.

For (iii), we can directly use the result of Lemma A.6.

$$\sum_{t=1}^{T}\left[L^{(t)}\left(\boldsymbol{p}^{(t)}\right)-L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right)+\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right)\right] \tag{69}$$

$$\leq \frac{1}{2\beta}\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|^2_{\boldsymbol{B}^{(t)^{-1}}}+\sum_{t=1}^{T}\left(R^{(t)}\left(\boldsymbol{p}^{(t)}\right)-R^{(t+1)}\left(\boldsymbol{p}^{(t)}\right)\right) \tag{70}$$

$$= \frac{1}{2\beta}\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|^2_{\boldsymbol{B}^{(t)^{-1}}}. \tag{71}$$

Combining all, we have regret upper bound as follows.

$$\begin{aligned}&\text{Regret}^{(T)}\left(\boldsymbol{p}^{\star}\right)=\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)-\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{\star}\right)\\&\leq\sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right)-\sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{\star}\right)\\&\leq\frac{\alpha}{2}\|\boldsymbol{p}\|_2^2+\frac{\beta}{2}\sum_{t=1}^{T}\left\|\boldsymbol{p}^{(t)}-\boldsymbol{p}^{\star}\right\|^2_{\boldsymbol{A}^{(t)}}+\frac{1}{2\beta}\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|^2_{\boldsymbol{B}^{(t)^{-1}}}.\end{aligned} \tag{72}$$

Lastly, from the result of Lemma A.10, we have

$$\begin{aligned}&\text{Regret}^{(T)}\left(\boldsymbol{p}^{\star}\right)=\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)-\sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{\star}\right)\\&\leq\sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right)-\sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{\star}\right)-\frac{\beta}{2}\sum_{t=1}^{T}\left\|\boldsymbol{p}^{(t)}-\boldsymbol{p}^{\star}\right\|^2_{\boldsymbol{A}^{(t)}}\\&\leq\frac{\alpha}{2}\|\boldsymbol{p}\|_2^2+\frac{1}{2\beta}\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|^2_{\boldsymbol{B}^{(t)^{-1}}},\end{aligned} \tag{73}$$

where we need $\left|\beta\left\langle\boldsymbol{g}^{(t)},\boldsymbol{p}-\boldsymbol{p}^{(t)}\right\rangle\right|\leq\frac{1}{2}$ due to the assumption.

To meet the assumption, we have

$$\left|\beta\left\langle\boldsymbol{g}^{(t)},\boldsymbol{p}-\boldsymbol{p}^{(t)}\right\rangle\right|\leq\beta\|\boldsymbol{p}-\boldsymbol{p}^{(t)}\|_1\|\boldsymbol{g}^{(t)}\|_\infty\leq 2\beta L_\infty, \tag{74}$$

where the first inequality is due to Hölder's inequality and the second inequality is due to Lemma 4.1 and the fact that a diameter of the simplex is 2. Thus, we can set $\beta=\frac{1}{4L_\infty}$ to satisfy the assumption.

For the first term, using the fact that $\|\cdot\|_2\leq\|\cdot\|_1$, we have

$$\frac{\alpha}{2}\|\boldsymbol{p}\|_2^2\leq\frac{\alpha}{2}\|\boldsymbol{p}\|_1^2\leq\frac{\alpha}{2}, \tag{75}$$

where the last equality is due to $\boldsymbol{p}\in\Delta_{K-1}$.

For the second term, denote $\lambda_1,...,\lambda_K$ as the eigenvalues of $\boldsymbol{B}^{(T)}-\alpha\boldsymbol{I}_K$, then we have:

$$\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|^2_{\boldsymbol{B}^{(t)^{-1}}}\leq\sum_{i=1}^{K}\log\left(1+\frac{\lambda_i}{\alpha}\right), \tag{76}$$

which is the direct result of Lemma 11.11 and Theorem 11.7 of (Cesa-Bianchi & Lugosi, 2006). This can be further bounded by AM-GM inequality as follows.

$$\sum_{i=1}^{K}\log\left(1+\frac{\lambda_i}{\alpha}\right)\leq K\log\left(1+\frac{1}{K\alpha}\sum_{i=1}^{K}\lambda_i\right) \tag{77}$$

Since we have

$$\begin{aligned}\sum_{i=1}^{K}\lambda_i = \operatorname{trace}\left(\boldsymbol{B}^{(T)} - \alpha\boldsymbol{I}_K\right) = \operatorname{trace}\left(\beta\sum_{t=1}^{T}\boldsymbol{g}^{(t)}\boldsymbol{g}^{(t)\top}\right) = \beta\sum_{t=1}^{T}\|\boldsymbol{g}^{(t)}\|_2^2 \\ \leq \beta KT\|\boldsymbol{g}^{(t)}\|_\infty^2 \leq \beta KTL_\infty^2 = \frac{KTL_\infty}{4},\end{aligned} \tag{78}$$

where $L_\infty$ is a Lipschitz constant w.r.t. $\|\cdot\|_\infty$ from Lemma 4.1, thus the inequality is due to $\|\boldsymbol{p}\|_2 \leq \sqrt{K}\|\boldsymbol{p}\|_\infty, \forall \boldsymbol{p} \in \Delta_{K-1}$.

Followingly, we can upper-bound the second term as

$$\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|_{\boldsymbol{B}^{(t)-1}}^2 \leq K\log\left(1+\frac{TL_\infty}{4\alpha}\right). \tag{79}$$

Putting them all together, we have:

$$\begin{aligned}&\text{Regret}^{(T)}\left(\boldsymbol{p}^\star\right) = \sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^\star\right) \\ &\leq \sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^\star\right) - \frac{1}{2}\sum_{t=1}^{T}\left\|\boldsymbol{p}^{(t)} - \boldsymbol{p}^\star\right\|_{\boldsymbol{A}^{(t)}}^2 \\ &\leq \frac{\alpha}{2}\|\boldsymbol{p}\|_2^2 + \frac{1}{2\beta}\sum_{t=1}^{T}\left\|\boldsymbol{g}^{(t)}\right\|_{\boldsymbol{B}^{(t)-1}}^2 \\ &\leq \frac{\alpha}{2} + \frac{K}{2\beta}\log\left(1+\frac{TL_\infty}{4\alpha}\right) \\ &= \frac{\alpha}{2} + 2L_\infty K\log\left(1+\frac{TL_\infty}{4\alpha}\right).\end{aligned} \tag{80}$$

If we further set $\alpha = 4L_\infty K$, we finally have

$$\text{Regret}^{(T)}\left(\boldsymbol{p}^\star\right) \leq 2L_\infty K\left(1+\log\left(1+\frac{T}{16K}\right)\right). \tag{81}$$

■

### A.4. Regret Bound of AAggFF-D with Full Client Participation: Proof of Theorem 5.2

*Proof.* Again, since Lemma A.4 holds for arbitrary loss function, let us set $L^{(t)}\left(\boldsymbol{p}\right) \triangleq \sum_{\tau=1}^{t-1}\tilde{\ell}^{(\tau)}(\boldsymbol{p}) + \eta^{(t+1)}\sum_{i=1}^{K}p_i\log p_i$ with a slight abuse of notation. Note that we set $R^{(t)}\left(\boldsymbol{p}\right) = \eta^{(t)}\sum_{i=1}^{K}p_i\log p_i$ is a negative entropy regularizer with non-decreasing time-varying step size $\eta^{(t)}$, thus $L^{(t)}\left(\boldsymbol{p}\right)$ is $\eta^{(t)}$-strongly convex w.r.t. $\|\cdot\|_1$. (Proposition 5.1 from (Beck & Teboulle, 2003)) Then, we have an upper bound of the regret of AAggFF-D (with full-client participation setting) as follows.

$$\begin{aligned}&\text{Regret}^{(T)}\left(\boldsymbol{p}^\star\right) \\ &= \sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T}\ell^{(t)}\left(\boldsymbol{p}^\star\right) \\ &\leq \sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T}\tilde{\ell}^{(t)}\left(\boldsymbol{p}^\star\right) \\ &= \underbrace{R^{(T+1)}\left(\boldsymbol{p}^\star\right) - R^{(1)}\left(\boldsymbol{p}^{(1)}\right)}_{\text{(i)}} + \underbrace{L^{(T+1)}\left(\boldsymbol{p}^{(T+1)}\right) - L^{(T+1)}\left(\boldsymbol{p}^\star\right)}_{\text{(ii)}} \\ &+ \underbrace{\sum_{t=1}^{T}\left[L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right)\right]}_{\text{(iii)}},\end{aligned} \tag{82}$$

where the inequality is due to Corollary A.3.

For (i), recall from Lemma A.4 that the regret does not depend on the regularizer, we can bound it after changing from $R^{(T+1)}(\boldsymbol{p}^\star)$ to $R^{(T)}(\boldsymbol{p}^\star)$.

$$R^{(T)}(\boldsymbol{p}^\star) - R^{(1)}(\boldsymbol{p}) \leq \eta^{(T)} \sum_{i=1}^{K} p_i^\star \log p_i^\star + \eta^{(1)} \log K \leq \eta^{(T)} \sum_{i=1}^{K} p_i^\star \log p_i^\star + \eta^{(T)} \log K \leq \eta^{(T)} \log K. \tag{83}$$

For (ii), we use the assumption in Lemma A.7, where $\boldsymbol{p}^{(t)} = \arg\min_{\boldsymbol{p}\in\Delta_{K-1}} L^{(t)}(\boldsymbol{p})$. From the assumption, since $\boldsymbol{p}^{(T+1)}$ is the minimizer of $L^{(T+1)}$, (ii) becomes negative. Thus, we can exclude it from the upper bound.

For (iii), we directly use the result of Lemma A.7 as follows.

$$\sum_{t=1}^{T} \left[ L^{(t)}\left(\boldsymbol{p}^{(t)}\right) - L^{(t+1)}\left(\boldsymbol{p}^{(t+1)}\right) + \tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) \right] \leq \sum_{t=1}^{T} \frac{\left\|\boldsymbol{g}^{(t)}\right\|_\infty^2}{2\eta^{(t)}} \leq \sum_{t=1}^{T} \frac{L_\infty^2}{2\eta^{(t)}}, \tag{84}$$

where the additional terms are removed due to the non-decreasing property of regularizer thanks to the assumption of $\eta^{(t)}$, and the last inequality is due to Lemma 4.1. Note that $\boldsymbol{g}^{(t)} = \nabla \ell^{(t)}(\boldsymbol{p}^{(t)})$.

Combining all, we have regret upper bound as follows.

$$\text{Regret}^{(T)}(\boldsymbol{p}^\star) = \sum_{t=1}^{T} \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T} \ell^{(t)}(\boldsymbol{p}^\star) \leq \sum_{t=1}^{T} \tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \sum_{t=1}^{T} \tilde{\ell}^{(t)}(\boldsymbol{p}^\star) \leq \eta^{(T)} \log K + \sum_{t=1}^{T} \frac{L_\infty^2}{2\eta^{(t)}}. \tag{85}$$

Finally, by setting $\eta^{(t)} = \frac{L_\infty \sqrt{t}}{\sqrt{\log K}}$, we have

$$\leq L_\infty \sqrt{T \log K} + \frac{L_\infty \sqrt{\log K}}{2} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2 L_\infty \sqrt{T \log K}, \tag{86}$$

where the inequality is due to $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq \int_0^T \frac{\mathrm{d}x}{\sqrt{x}} = 2\sqrt{T}$. See also equation (7.3) of (Orabona, 2019). ■

### A.5. Regret Bound of AAggFF-D with Partial Client Participation: Proof of Corollary 5.3

*Proof.* Denote $\breve{\ell}^{(t)}$ as a linearized loss constructed from $\breve{\boldsymbol{r}}^{(t)}$ and $\breve{\boldsymbol{g}}^{(t)}$. i.e.,

$$\breve{\ell}^{(t)}(\boldsymbol{p}) = \left\langle \boldsymbol{p}, \breve{\boldsymbol{g}}^{(t)} \right\rangle = \left\langle \boldsymbol{p}, \frac{\breve{\boldsymbol{r}}^{(t)}}{1 + \left\langle \boldsymbol{p}^{(t)}, \bar{\boldsymbol{r}} \mathbf{1}_K \right\rangle} + \frac{\bar{\boldsymbol{r}} \mathbf{1}_K \boldsymbol{p}^{(t)\top} (\breve{\boldsymbol{r}}^{(t)} - \bar{\boldsymbol{r}} \mathbf{1}_K)}{(1 + \left\langle \boldsymbol{p}^{(t)}, \bar{\boldsymbol{r}} \mathbf{1}_K \right\rangle)^2} \right\rangle \tag{87}$$

The expected regret is

$$\begin{aligned}
&\mathbb{E}\left[\text{Regret}^{(T)}(\boldsymbol{p}^\star)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \left(\ell^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \ell^{(t)}(\boldsymbol{p}^\star)\right)\right] \\
&\leq \mathbb{E}\left[\sum_{t=1}^{T} \left(\breve{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \breve{\ell}^{(t)}(\boldsymbol{p}^\star)\right)\right] = \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \breve{\boldsymbol{g}}^{(t)}, \boldsymbol{p}^{(t)} - \boldsymbol{p}^\star \right\rangle\right] \\
&= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_{i\in S^{(t)}}\left[\left\langle \breve{\boldsymbol{g}}^{(t)}, \boldsymbol{p}^{(t)} - \boldsymbol{p}^\star \right\rangle\right]\right] (\because \text{Law of Total Expectation}) \\
&\approx \mathbb{E}\left[\sum_{t=1}^{T} \left\langle \boldsymbol{g}^{(t)}, \boldsymbol{p}^{(t)} - \boldsymbol{p}^\star \right\rangle\right] (\because \text{Lemma 4.4}) \\
&= \sum_{t=1}^{T} \left(\tilde{\ell}^{(t)}\left(\boldsymbol{p}^{(t)}\right) - \tilde{\ell}^{(t)}(\boldsymbol{p}^\star)\right) \leq \mathcal{O}\left(L_\infty K \sqrt{T \log K}\right)
\end{aligned} \tag{88}$$

■

*Remark* A.13. Even though we can enjoy the same regret upper bound *in expectation* from Corollary 5.3, it should be noted that the raw regret (i.e., regret without expectation) from $\breve{\boldsymbol{g}}^{(t)}$ may inflate the regret upper bound from $\mathcal{O}\left(L_\infty K\sqrt{T\log K}\right)$ to $\mathcal{O}\left(\breve{L}_\infty K\sqrt{T\log K}\right)$, where $\breve{L}_\infty$ is a Lipschitz constant from Lemma A.2, upper bounding $\left\|\breve{\boldsymbol{g}}^{(t)}\right\|_\infty \leq \breve{L}_\infty$. It is because $\breve{L}_\infty$ is dominated by $1/C \approx \mathcal{O}(K)$, which can be a *huge number* when if $C$ is a tiny constant. Although this inflation hinders proper update of `AAggFF-D` empirically, this can be easily eliminated in `AAggFF-D` through an appropriate choice of a range ($C_1$ and $C_2$) of the response vector, which ensures practicality of `AAggFF-D`. See Appendix B.2 for a detail.

### A.6. Derivation of Closed-From Update of `AAggFF-D`

The objective of `AAggFF-D` in (9) can be written in the following form.

$$\begin{aligned}&\min_{\boldsymbol{p}\in\Delta_{K-1}} \sum_{\tau=1}^{t} \tilde{\ell}^{(\tau)}(\boldsymbol{p}) + \eta^{(t+1)} \sum_{i=1}^{K} p_i \log p_i = \min_{\boldsymbol{p}\in\Delta_{K-1}} \left\langle \boldsymbol{p}, \sum_{\tau=1}^{t-1} \breve{\boldsymbol{g}}^{(\tau)} \right\rangle + \eta^{(t+1)} \sum_{i=1}^{K} p_i \log p_i \\ &= \min_{\boldsymbol{p}\in\Delta_{K-1}} \left\langle \sum_{\tau=1}^{t} \breve{\boldsymbol{g}}^{(\tau)}, \boldsymbol{p} \right\rangle + R^{(t+1)}(\boldsymbol{p}) = \max_{\boldsymbol{p}\in\Delta_{K-1}} \left\langle -\sum_{\tau=1}^{t} \breve{\boldsymbol{g}}^{(\tau)}, \boldsymbol{p} \right\rangle - R^{(t+1)}(\boldsymbol{p}).\end{aligned} \tag{89}$$

It exactly corresponds to the form of the Fenchel conjugate $R_*^{(t+1)}$, which is defined as follows.

$$R_*^{(t+1)}(\boldsymbol{p}) = \max_{\boldsymbol{p}\in\Delta_{K-1}} \left\langle -\sum_{\tau=1}^{t} \breve{\boldsymbol{g}}^{(\tau)}, \boldsymbol{p} \right\rangle - R^{(t+1)}(\boldsymbol{p}). \tag{90}$$

Thus, we can enjoy the property of Fenchel conjugate, which is

$$\boldsymbol{p}^{(t+1)} = \nabla R_*^{(t+1)} \left( -\sum_{\tau=1}^{t} \breve{\boldsymbol{g}}^{(\tau)} \right) \tag{91}$$

Since we can derive the log-sum-exp form by solving (90) as follows,

$$R_*^{(t+1)}(\boldsymbol{u}) = \log\left(\sum_{i=1}^{K} \exp(u_i)\right), \tag{92}$$

we have the closed-form solution for the new decision update.

$$p_i^{(t+1)} = \frac{\exp\left(-\sum_{\tau=1}^{t} \breve{g}_i^{(\tau)}/\eta^{(t+1)}\right)}{\sum_{j=1}^{K} \exp\left(-\sum_{\tau=1}^{t} \breve{g}_j^{(\tau)}/\eta^{(t+1)}\right)}. \tag{93}$$

Note that $\eta^{(t+1)}$ is already determined in Theorem 5.2 and Corollary 5.3 as $\frac{\breve{L}_\infty\sqrt{t+1}}{\sqrt{\log K}}$, with the reflection of modified Lipschitz constant from $L_\infty$ to $\breve{L}_\infty$ (see Remark A.13). See also (Helmbold et al., 1998) and Chapter 6.6 of (Orabona, 2019).

## B. Detailed Designs of AAggFF

### B.1. Cumulative Distribution Function for Response Transformation

**Choice of Distributions** We used the CDF to transform unbounded responses from clients (i.e., local losses of clients) into bounded values. Among diverse options, we used one of the following 6 CDFs in this work. (Note that $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-y^2} \mathrm{d}y$ is the Gauss error function)

1. Weibull (Weibull, 1951): $\texttt{CDF}(x) = 1 - e^{-(x/\alpha)^\beta}$; we set $\alpha = 1$ (scale) and $\beta = 2$ (shape).
2. Frechet (Fréchet, 1927): $\texttt{CDF}(x) = e^{-(\frac{x}{\alpha})^{-\beta}}$; we set $\alpha = 1$ (scale) and $\beta = 1$ (shape).
3. Gumbel (Gumbel, 1935): $\texttt{CDF}(x) = e^{-e^{-(x-\alpha)/\beta}}$; we set $\alpha = 1$ (scale) and $\beta = 1$ (shape).
4. Exponential: $\texttt{CDF}(x) = 1 - e^{-\alpha x}$; we set $\alpha = 1$ (scale).
5. Logistic: $\texttt{CDF}(x) = \frac{1}{1+e^{-(x-\alpha)/\beta)}}$; we set $\alpha = 1$ (scale) and $\beta = 1$ (shape).
6. Normal (Gauß, 1809): $\texttt{CDF}(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-\alpha}{\beta\sqrt{2}}\right)\right]$; we set $\alpha = 1$ (scale) and $\beta = 1$ (shape).

Commonly, the scale parameter of all distributions is set to 1, since in (8) we centered inputs to 1 in expectation. Although we fixed the parameters of each CDF, they can be statistically estimated in practice, such as using maximum spacing estimation (Cheng & Amin, 1983).

For imposing larger mixing coefficients for larger losses, the transformation should (i) preserve the relative difference between responses, as well as (ii) not too sensitive for outliers. While other heuristics (e.g., clipping values, subtracting from arbitrary large constant (Zhang et al., 2022)) for the transformation are also viable options for (i), additional efforts are still required to address (ii).

On the contrary, CDFs can address both conditions with ease. As CDFs are increasing functions, (i) can be easily satisfied. For (ii), it can be intrinsically addressed by the nature of CDF itself. Let us start with a simple example.

Suppose we have $K = 3$ local losses: $F_1(\boldsymbol{\theta}) = 0.01, F_2(\boldsymbol{\theta}) = 0.10, F_3(\boldsymbol{\theta}) = 0.02$. Since the average is $\bar{\mathrm{F}} = \frac{0.01+0.10+0.02}{3} \approx 0.043$, we have inputs of CDF as follows: $F_1(\boldsymbol{\theta})/\bar{\mathrm{F}} = 0.23, F_2(\boldsymbol{\theta})/\bar{\mathrm{F}} = 2.31, F_3(\boldsymbol{\theta})/\bar{\mathrm{F}} = 0.46$. These centered inputs are finally transformed into bounded values as in Table A1.

Table A1: Example: Effects of CDF Transformations

| | Transformed Responses |
|---|---|
| Weibull CDF $\texttt{CDF}(x) = 1 - e^{(-x^2)}$ | 0.05 / <u>1.00</u> / 0.19 |
| Frechet CDF $\texttt{CDF}(x) = e^{(-1/x)}$ | 0.01 / <u>0.65</u> / 0.11 |
| Gumbel CDF $\texttt{CDF}(x) = e^{\left(-e^{(-(x-1))}\right)}$ | 0.12 / <u>0.76</u> / 0.18 |
| Exponential CDF $\texttt{CDF}(x) = 1 - e^{(-x)}$ | 0.21 / <u>0.90</u> / 0.37 |
| Logistic CDF $\texttt{CDF}(x) = \frac{1}{1+e^{(-(x-1))}}$ | 0.32 / <u>0.79</u> / 0.37 |
| Normal CDF $\texttt{CDF}(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x-1}{\sqrt{2}}\right)\right]$ | 0.22 / <u>0.90</u> / 0.29 |

While all losses become bounded values in $[0, 1]$, the maximum local loss (i.e., $F_2(\boldsymbol{\theta}) = 0.10$) is transformed into different values by each CDF (see underlined figures in the 'Transformed Responses' column of Table A1). When using the Weibull CDF, the maximum local loss is translated into 1.00, which means that there may be no value greater than 0.10 (i.e., 0.10 is the largest one in 100% probability) given current local losses. Meanwhile, when using the Frechet CDF, the maximum local

loss is translated into 0.65, which means that there still is a 35% chance that some other local losses are greater than 0.10 when provided with other losses similar to 0.01 and 0.02. This implies that each CDF *treats a maximum value differently*. When a transformation is easily inclined to the maximum value, thereby returning 1 (i.e., maximum of CDF), it may yield a degenerate decision, e.g., $\boldsymbol{p} \approx [0, 1, 0]^\top$.

Fortunately, most of the listed CDFs are designed for *modeling maximum values*. For example, the three distributions, Gumbel, Frechet, and Weibull, are grouped as the Extreme Value Distribution (EVD) (Kotz & Nadarajah, 2000). As its name suggests, it models the behavior of extreme events, and it is well known that any density modeling a minimum or a maximum of independent and identically distributed (IID) samples follows the shape of one of these three distributions (by the Extreme Value Theorem (Rusnock & Kerr-Lawson, 2005)). In other words, EVDs can reasonably measure *how a certain value is close to a maximum*. Thus, they can estimate whether a certain value is relatively large or small. Otherwise, the Exponential distribution is a special case of Weibull distribution, and the logistic distribution is also related to the Gumbel distribution. Last but not least, although it is not a family of EVD, the Normal distribution is also considered due to the central limit theorem, since the local loss is the sum of errors from IID local samples. We expect the CDF transformation can appropriately measure a relative magnitude of local losses, and it should be helpful for decision making.

**Effects of Response Transformation** We also illustrated that the response should be bounded (i.e., Lipschitz continuous) in section 4.2.1, to have non-vacuous regret upper bound. To acquire bounded response, we compare the cumulative values of a global objective in (1), i.e., $\sum_{t=1}^{T} \sum_{i=1}^{K} p_i^{(t)} F_i(\boldsymbol{\theta}^{(t)})$ for the cross-silo setting, and $\sum_{t=1}^{T} \sum_{i \in S^{(t)}} p_i^{(t)} F_i(\boldsymbol{\theta}^{(t)})$ for the cross-device setting.

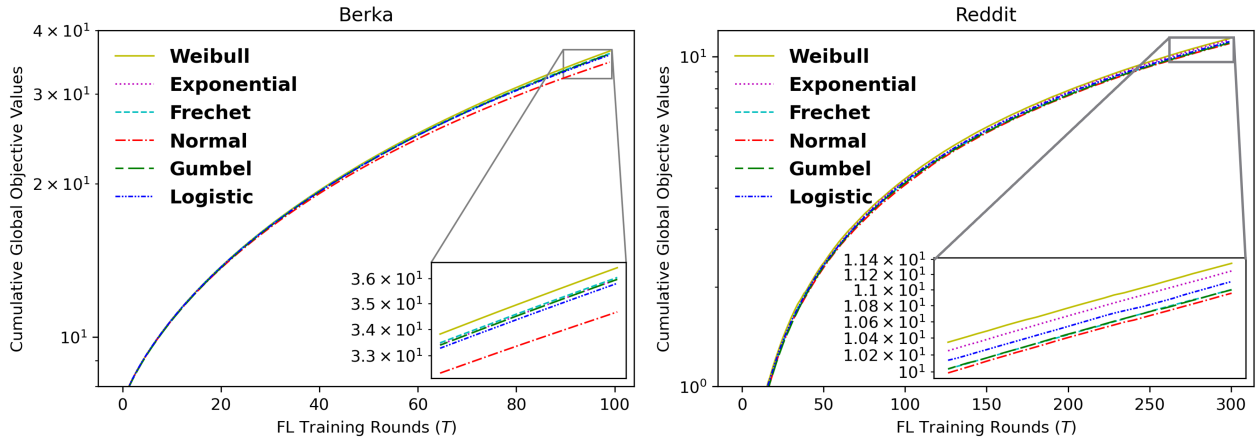


Figure A1: Cumulative values of a global objective according to different CDFs (smaller is better). (Left) Berka dataset (cross-silo setting; $K = 7, T = 100$). (Right) Reddit dataset (cross-device setting; $K = 817, T = 300, C = 0.00612$)

In the cross-silo experiment with the Berka dataset, the Normal CDF shows the smallest cumulative values, while in the cross-device experiment with the Reddit dataset, the Weibull CDF yields the smallest value. For the Berka dataset, the Normal CDF yields an average performance (AUROC) of 79.37 with the worst performance of 43.75, but the Weibull CDF shows an average performance of 73.02 with the worst performance of 25.00. The same tendency is also observed in the Reddit dataset. The Normal CDF presents an average performance (Acc. 1) of 14.05 and the worst performance of 4.26, while the Weibull CDF shows an average of 12.62 with the worst of 3.35. From these observations, we can conclude that an appropriate choice of CDF is necessary for better sequential decision making, and suitable transformation helps minimize a global objective of FL. Note also that these behaviors are also directly related to the global convergence of the algorithm w.r.t. $\boldsymbol{\theta}$.

### B.2. Choice of a Response Range

In regard to determining the range of a response vector, i.e., $[C_1, C_2]$, we can refer to the Lipschitz continuity in Lemma 4.1 and Lemma A.2. For the cross-silo setting, we can set arbitrary constants so that $L_\infty = \mathcal{O}(1)$ according to Lemma 4.1. Thus, for all experiments of `AAggFF-S`, we set $C_1 = 0, C_2 = \frac{1}{K}$.

For the cross-device setting, the Lipschitz constant is changed into $\breve{L}_\infty$, since it is influenced by the client sampling probability ($C$). In detail, $C$ is located in the denominator of the Lipschitz constant, $\breve{L}_\infty$, which inflates the resulting gradient value as $C$ is a constant close to 0. (e.g., when 10 among 100,000 clients are participating in each round, $1/C = 10^4$) This is problematic and even causes an overflow problem empirically in updating a new decision. Thus, we propose a simple remedy — setting $C_1$ and $C_2$ to be *a multiple of* $C$, so that the $C$ in the denominator is to be canceled out, according to Lemma A.2. For instance, when $C_1 = 0, C_2 = C$, resulting Lipschitz constant simply becomes $\|\breve{\boldsymbol{g}}^{(t)}\|_\infty \leq \breve{L}_\infty = \frac{C}{1+0} + \frac{2(C-0)}{C(1+0)} = C + 2 \approx 2$, which is a constant far smaller than $T$ and $K$, typically assumed in the practical cross-device FL setting. Therefore, for all experiments of `AAggFF-D`, we set $C_1 = 0, C_2 = C$.

# C. Experimental Details

Table A2: Statistics of Federated Benchmarks

| Dataset | Clients | Samples | Avg. | Std. |
|---|---|---|---|---|
| Berka | 7 | 621 | 88.71 | 24.78 |
| MQP | 11 | 3,048 | 277.09 | 63.25 |
| ISIC | 6 | 21,310 | 3551.67 | 3976.16 |
| CelebA | 9,343 | 200,288 | 21.44 | 7.63 |
| Reddit | 817 | 97,961 | 119.90 | 229.85 |
| SpeechCommands | 2,005 | 84,700 | 42.24 | 36.69 |

## C.1. Datasets

For the main experiments in Table 2 and Table 3, we used 3 datasets for the cross-silo setting (Berka (Berka, 1999), MQP (McCreery et al., 2020), and ISIC (Codella et al., 2018; Ogier du Terrail et al., 2022)), and 3 other datasets for the cross-device setting (CelebA, Reddit (Caldas et al., 2019), and SpeechCommands (Warden, 2018)). In this section, we describe details of each dataset about its task, metrics, and client partitioning. The statistics of all federated benchmarks are summarized in Table A2. Note that **Avg.** and **Std.** in the table refer to the average and a standard deviation of a sample size of each client in the federated system.

First, we present details of the federated benchmark for the cross-silo setting.

- **Berka** is a tabular dataset containing bank transaction records collected from Czech bank (Berka, 1999). Berka accompanies the loan default prediction task (i.e., binary classification) of each bank's customers. It is fully anonymized and is originally composed of 8 relational tables: accounts, clients, disposition, loans, permanent orders, transactions, demographics, and credit cards. We merged all 8 tables into one dataset by joining the primary keys of each table, and finally have 15 input features. From the demographics table, we obtain information on the region: Prague, Central Bohemia, South Bohemia, West Bohemia, North Bohemia, East Bohemia, South Moravia, and North Moravia. We split each client according to the region and excluded all samples of North Bohemia since it has only one record of loan default, thus we finally have 7 clients (i.e., banks). Finally, we used the area under the receiver operating characteristic (ROC) curve for the evaluation metric.
- **MQP** is a clinical question pair dataset crawled from medical question answering dataset (Ben Abacha & Demner-Fushman, 2019), and labeled by 11 doctors (McCreery et al., 2020). All paired sentences are labeled as either similar or dissimilar, thereby it is suitable for the binary classification task. As a pre-processing, we merge two paired sentences into one sentence by adding special tokens: `[SEP]`, `[PAD]`, and `[UNK]`. We set the maximum token length to 200, thus merged sentences less than 200 are filled with `[PAD]` tokens, and otherwise are truncated. Then, merged sentences are tokenized using pre-trained DistilBERT tokenizer (Sanh et al., 2019). We regard each doctor as a separate client and thus have 11 clients. Finally, we used the area under the ROC curve for the evaluation metric.
- **ISIC** is a dermoscopic image dataset for a skin cancer classification, collected from 4 hospitals. (Codella et al., 2018; Ogier du Terrail et al., 2022) The task contains 8 distinct melanoma classes, thus designed for the multi-class classification task. Following (Ogier du Terrail et al., 2022), as one hospital has three different imaging apparatus, its samples are further divided into 3 clients, thus we finally have 6 clients in total. Finally, we used top-5 accuracy for the evaluation metric.

Next, we illustrate details of the federated benchmark for the cross-device setting.

- **CelebA** is a vision dataset containing the facial images of celebrities (Liu et al., 2015). It is curated for federated setting in LEAF benchmark (Caldas et al., 2019), and is targeted for the binary classification task (presence of smile). We follow the processing of (Caldas et al., 2019), thereby each client corresponds to each celebrity, having 9,343 total clients in the federated system. Finally, we used top-1 accuracy for the evaluation metric following (Caldas et al., 2019).

- **Reddit** is a text dataset containing the comments of community users of Reddit in December 2017, and a part of LEAF benchmark (Caldas et al., 2019). Following (Caldas et al., 2019), we build a dictionary of vocabularies of size 10,000 from tokenized sentences and set the maximum sequence length to 10. The main task is tailored for language modeling, i.e., next token prediction, given word embeddings in each sentence of clients. Each client corresponds to one of the community users, thus 817 clients are presented in total. Finally, we used the top-1 accuracy for the evaluation metric following (Caldas et al., 2019).
- **SpeechCommands** is designed for a short-length speech recognition task that includes one second 35 short-length words, such as "Up", "Down", "Left", and "Right" (Warden, 2018). It is accordingly a multi-class classification task for 35 different classes. While it is collected from 2,618 speakers, we rule out all samples of speakers having too few samples when splitting the dataset into training and test sets. (e.g., exclude speakers whose total sample counts are less than 3) As a result, we have 2,005 clients, and each client has 16,000-length time-domain waveform samples. Finally, we used top-5 accuracy for the evaluation metric.

### C.2. Models

For each dataset, we used task-specific model architectures which are already used in previous works, or widely used in reality, to simulate the practical FL scenario as much as possible. For the experiment of the cross-silo setting, we used the following models.

- *Logistic Regression* is used for the Berka dataset. We used a simple logistic regression model with a bias term, and the output (i.e., logit vector) is transformed into predicted class probabilities by the softmax function.
- *DistilBERT (Sanh et al., 2019)* is used for the MQP dataset. We used a pre-trained DistilBERT model, from BookCorpus and English Wikipedia (Sanh et al., 2019). We also used the corresponding DistilBERT tokenizer for the pre-processing of raw clinical sentences. For a fine-tuning of the pre-trained DistilBERT model, we attach a classifier head next to the last layer of the DistilBERT's encoder, which outputs an embedding of 768 dimension. The classifier is in detail processing the embedding as follows: (768-ReLU-Dropout-2), where each figure is an output dimension of a fully connected layer with a bias term, ReLU is a rectified linear unit activation layer, and Dropout (Srivastava et al., 2014) is a dropout layer having probability of $0.1$. In the experiment, we trained all layers including pre-trained weights.
- *EfficientNet (Tan & Le, 2019)* is used for the ISIC dataset. We also used the pre-trained EfficientNet-B0 model from ImageNet benchmark dataset (Deng et al., 2009). For fine-tuning, we attach a classifier head after the convolution layers of EfficientNet. The classifier is composed of the following components: (AdaptiveAvgPool$(7, 7)$-Dropout-$8$), where AdaptiveAvgPool$(cdot, cdot)$ is a 2D adaptive average pooling layer outputs a feature map of size $7 \times 7$ (which are flattened thereafter), Dropout is a dropout layer with a probability of 0.1, and the last linear layer outputs an 8-dimensional vector, which is the total number of classes.

Next, for the cross-device setting, we used the following models.

- *ConvNet* model used in LEAF benchmark (Caldas et al., 2019) is used for the CelebA dataset. It is composed of four convolution layers, of which components are: 2D convolution layer without bias term with 32 filters of size $3 \times 3$ (stride=1, padding=1), group normalization layer (the number of groups is decreased from 32 by a factor of 2: 32, 16, 8, 4), 2D max pooling layer with $2 \times 2$ filters, and a ReLU nonlinear activation layer. Plus, a classifier comes after the consecutive convolution layers, which are composed of: (AdaptiveAvgPool$(5, 5)$-1), which are a 2D adaptive average pooling layer that outputs a feature map of size $5 \times 5$ (which are flattened thereafter), and a linear layer with a bias term outputs a scalar value since it is a binary classification task.
- *StackedLSTM* model used in LEAF benchmark (Caldas et al., 2019) is used for the Reddit dataset. It is composed of an embedding layer of which the number of embeddings is 200, and outputs an embedding vector of 256 dimensions. It is processed by consecutive 2 LSTM (Hochreiter & Schmidhuber, 1997) layers with the hidden size of 256, and enters the last linear layer with a bias term, which outputs a logit vector of 10,000 dimensions, which corresponds to the vocabulary size.
- *M5 (Dai et al., 2017)* model is used for the SpeechCommands dataset. It is composed of four 1D convolution layers followed by a 1D batch normalization layer, ReLU nonlinear activation, and a 1D max pooling layer with a filter of

size 4. All convolution layers EXCEPT the input layer have a filter of size 3, and the numbers of filters are 64, 128, and 256 (all with stride=1 and padding=1). The input convolution layer has 64 filters with a filter of size 80, and stride of 4. Lastly, one linear layer outputs a logit vector of 35 dimensions.

### C.3. Hyperparameters

Before the main experiment, we first tuned the hyperparameter of all baseline fair FL algorithms from a separate random seed. The hyperparameter of each fair algorithm is listed as follows.

- `AFL` (Mohri et al., 2019) — a step size of a mixing coefficient $\in\{0.01, 0.1, 1.0\}$
- `q-FedAvg` (Li et al., 2019) — a magnitude of fairness, $\in\{0.1, 1.0, 5.0\}$
- `TERM` (Li et al., 2020a) — a tilting constant, $\lambda$, $\in\{0.1, 1.0, 10.0\}$
- `FedMGDA` (Hu et al., 2022) — a deviation from static mixing coefficient $\in\{0.1, 0.5, 1.0\}$
- `PropFair` (Zhang et al., 2022) — a baseline constant $\in\{2, 3, 5\}$

Each candidate value is selected according to the original paper, and we fix the number of local epochs, $E = 1$ (following the set up in (Li et al., 2019)), along with the number of local batch size $B = 20$ in all experiments. For each dataset, a weight decay (L2 penalty) factor ($\psi$), a local learning rate ($\zeta$), and variables related to a learning rate scheduling (i.e., learning rate decay factor ($\phi$), and a decay step ($s$)) are tuned first with `FedAvg` (McMahan et al., 2017) as follows.

- **Berka**: $\psi = 10^{-3}, \zeta = 10^{0}, \phi = 0.99, s = 10$
- **MQP**: $\psi = 10^{-2}, \zeta = 10^{-\frac{5}{2}}, \phi = 0.99, s = 15$
- **ISIC**: $\psi = 10^{-2}, \zeta = 10^{-4}, \phi = 0.95, s = 5$
- **CelebA**: $\psi = 10^{-4}, \zeta = 10^{-1}, \phi = 0.96, s = 300$
- **Reddit**: $\psi = 10^{-6}, \zeta = 10^{\frac{7}{8}}, \phi = 0.95, s = 20$
- **SpeechCommands**: $\psi = 0, \zeta = 10^{-1}, \phi = 0.999, s = 10$

This is intended under the assumption that all fair FL algorithms should at least be effective in the same setting of the FL algorithm with the static aggregation scheme (i.e., `FedAvg`). Note that client-side optimization in all experiments is done by the Stochastic Gradient Descent (SGD) optimizer.

### C.4. Implementation Details

All code is implemented in PyTorch (Paszke et al., 2019), simulating a central parameter server that orchestrates a whole FL procedure and operates `AAggFF`. We further simulate $K$ participating clients having their own local samples, and a communication scheme with the central server. All experiments are conducted on a server with 2 Intel®Xeon®Gold 6226R CPUs (@ 2.90GHz) and 2 NVIDIA®Tesla®V100-PCIE-32GB GPUs.

# D. Pseudocode for AAggFF

## D.1. Pseudocode for ClientUpdate

**Algorithm 1** ClientUpdate

**Input:** number of local epochs $E$, local batch size $B$, local learning rate $\zeta$, global model $\boldsymbol{\theta}$
**Procedure:**
Evaluate the received global model on training set according to eq. (94) to yield $F_i(\boldsymbol{\theta})$.
Set local model $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}$
**for** $e = 0$ **to** $E - 1$ **do**
  $\mathcal{B}_e \leftarrow$ Split the client training dataset into batches of size $B$.
  **for** mini-batch $\Xi$ in $\mathcal{B}_e$ **do**
    Update the model $\boldsymbol{\theta}^{(e)} \leftarrow \boldsymbol{\theta}^{(e)} - \frac{\zeta}{B} \sum_{j=1}^{B} \nabla_{\boldsymbol{\theta}} \mathcal{L}\left(\Xi; \boldsymbol{\theta}^{(e)}\right)$.
  **end for**
  Set $\boldsymbol{\theta}^{(e+1)} \leftarrow \boldsymbol{\theta}^{(e)}$
**end for**
**Return:** $F_i(\boldsymbol{\theta}), \boldsymbol{\theta} - \boldsymbol{\theta}^{(E)}$.

For generating a response vector, each client is requested to evaluate the received global model on its *training samples*, $\{\xi_k\}_{k=1}^{n_i}$, before the local update. As a result of the evaluation, the local loss of client $i$ at round $t$, i.e., $F_i\left(\boldsymbol{\theta}^{(t)}\right)$, is calculated as follows.

$$F_i\left(\boldsymbol{\theta}^{(t)}\right) = \frac{1}{n_i} \sum_{k=1}^{n_i} \mathcal{L}\left(\xi_k; \boldsymbol{\theta}^{(t)}\right), \tag{94}$$

where $n_i$ is the total sample size of client $i$, $l$ is a loss function specific to the task, and $h_{\boldsymbol{\theta}}$ is a hypothesis realized by the parameter $\boldsymbol{\theta}$.

## D.2. Pseudocode for AAggFF-S

**Algorithm 2** AAggFF-S

**Input:** number of clients $K$, total rounds $T$, transformation $\rho$, minimum and maximum of a response $[C_1, C_2]$.
**Initialize:** mixing coefficients $\boldsymbol{p}^{(1)} = \frac{1}{K}\mathbf{1}_K$, global model $\boldsymbol{\theta}^{(1)} \in \mathbb{R}^d$
**Procedure:**
**for** $t = 0$ **to** $T - 1$ **do**
  **for** each client $i = 1, ..., K$ **in parallel do**
    $F_i\left(\boldsymbol{\theta}^{(t)}\right), \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)} \leftarrow$ ClientUpdate $\left(\boldsymbol{\theta}^{(t)}\right)$
  **end for**
  Return $\boldsymbol{r}^{(t)}$ according to eq. (8) and $C_1, C_2$.
  Suffer decision loss $\ell^{(t)}(\boldsymbol{p}^{(t)})$ according to eq. (4).
  Return a gradient $\boldsymbol{g}^{(t)} = \nabla \ell^{(t)}\left(\boldsymbol{p}^{(t)}\right)$.
  Return a mixing coefficient $\boldsymbol{p}^{(t+1)}$ according to eq. (7).
  Update a global model $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \sum_{i=1}^{K} p_i^{(t+1)} \left(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)}\right)$.
**end for**
**Return:** $\boldsymbol{\theta}^{(T)}$

## D.3. Pseudocode for AAggFF-D

After updating whole entries of a decision variable, the server only exploits mixing coefficients of which indices correspond

**Algorithm 3** AAggFF-D

**Input:** number of clients $K$, client sampling ratio $C \in (0, 1)$, total rounds $T$, transformation $\rho$, range of a response $[C_1, C_2]$.
**Initialize:** mixing coefficients $\boldsymbol{p}^{(0)} = \frac{1}{K}\mathbf{1}_K$, global model $\boldsymbol{\theta}^{(0)} \in \mathbb{R}^d$
**Procedure:**
**for** $t = 0$ **to** $T - 1$ **do**
  $S^{(t)} \leftarrow$ Wait until $\min(1, \lfloor C \cdot K \rfloor)$ clients are active in a network.
  **for** each client $i \in S^{(t)}$ **in parallel do**
    $F_i\left(\boldsymbol{\theta}^{(t)}\right), \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)} \leftarrow \texttt{ClientUpdate}\left(\boldsymbol{\theta}^{(t)}\right)$
  **end for**
  Return $\breve{\boldsymbol{r}}^{(t)}$ according to eq. (8), eq. (10), and $C_1, C_2$.
  Suffer decision loss $\ell^{(t)}(\boldsymbol{p}^{(t)})$ according to eq. (4).
  Return a gradient estimate $\breve{\boldsymbol{g}}^{(t)}$ according to eq. (11).
  Return mixing coefficients $\boldsymbol{p}^{(t+1)}$ according to eq. (12)).
  Acquire selected coefficients $\tilde{\boldsymbol{p}}^{(t+1)}$ according to eq. (95).
  Update a global model $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \sum_{i \in S^{(t)}} \tilde{p}_i^{(t+1)} \left(\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}_i^{(t+1)}\right)$.
**end for**
**Return:** $\boldsymbol{\theta}^{(T)}$

to selected clients. Denoting as $\tilde{\boldsymbol{p}}^{(t+1)} \in \Delta_{|S^{(t)}|-1}$, each selected entry is normalized as follows.

$$\tilde{p}_i^{(t+1)} = \frac{p_i^{(t+1)}}{\sum_{j \in S^{(t)}} p_j^{(t+1)}}, i \in S^{(t)} \tag{95}$$

This ensures that selected coefficients also satisfy the condition for being a probability vector (i.e., sum up to 1).