Federated Learning with Heterogeneous Data: A Superquantile Optimization Approach

Krishna Pillutla, Yassine Laguel, Jérôme Malick, Zaid Harchaoui

Federated Learning with Heterogeneous Data: A Superquantile Optimization Approach

Krishna Pillutla*[1], Yassine Laguel*[2], Jérôme Malick[3], Zaid Harchaoui[1]
[1]University of Washington, Seattle, WA, USA
[2]Univ. Grenoble Alpes, Grenoble, France
[3]CNRS, Grenoble, France

Abstract

We present a federated learning framework that is designed to robustly deliver good predictive performance across individual clients with heterogeneous data. The proposed approach hinges upon a superquantile-based learning objective that captures the tail statistics of the error distribution over heterogeneous clients. We present a stochastic training algorithm which interleaves differentially private client reweighting steps with federated averaging steps. The proposed algorithm is supported with finite time convergence guarantees that cover both convex and non-convex settings. Experimental results on benchmark datasets for federated learning demonstrate that our approach is competitive with classical ones in terms of average error and outperforms them in terms of tail statistics of the error.

1 Introduction

Federated learning is a distributed machine learning framework where many clients (e.g. mobile devices) collaboratively train a model under the orchestration of a central server (e.g. service provider), while keeping the training data private and local to the client throughout the training process [54, 37]. It has found widespread adoption across industry [6, 61] for applications ranging from applications on smart devices [83, 31] to healthcare [8, 34].

A key feature of federated learning is statistical heterogeneity, i.e., client data distributions are *not* identically distributed [37, 49]. In typical cross-device federated learning scenarios, each client corresponds to a user. The diversity in the data they generate reflects the diversity in their unique personal, cultural, regional and geographical characteristics.

This data heterogeneity in federated learning manifests itself as a train-test distributional shift. Indeed, the usual approach minimizes the prediction error of the model on average over the population of clients available for training [54], while at test time, the same model is deployed on individual clients. This approach can be liable to fail on clients whose data distribution is far from most of the population or who may have less data than most of the population. It is highly desirable, therefore, to have a federated learning method that can robustly deliver good predictive performance across a wide variety of natural distribution shifts posed by individual clients.

We present in this paper a robust approach to federated learning that guarantees a minimal level of predictive performance to all clients even in situations where the population is heterogeneous. The approach we develop addresses these issues by minimizing a learning objective based on the notion of a superquantile [67, 70], a risk measure that captures the tail behavior of a random variable.

Training models with a learning objective involving the superquantile raises challenges. The superquantile is a non-smooth functional with sophisticated properties. Furthermore, the superquantile function can be seen as a kind of nonlinear expectation that we would like to blend well with averaging mechanisms. We show how to address the former by leveraging the dual formulation and the latter by leveraging the tail-domain viewpoint. As a result, we can obtain an algorithm that can be implemented in a similar way to FedAvg [54] yet offers important benefits to heterogeneous populations.

The approach we propose, Δ-FL, allows one to control higher percentiles of the distribution of errors over the heterogeneous population of clients. We show in the experiments that our approach is more efficient than a direct

*These authors contributed equally to this work.

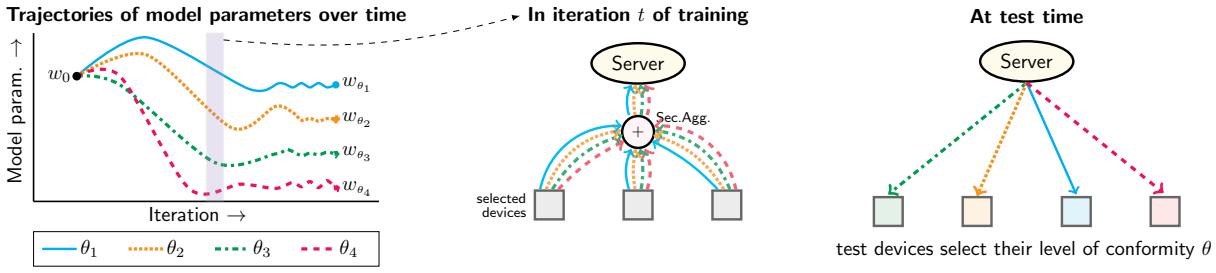

Figure 1: Schematic summary of the Δ-FL framework. **Left**: The server maintains multiple models w_{θ_j}, one for each level of conformity θ_j. **Middle**: During training, selected clients participate in training *each* model w_{θ_j}. Individual updates are securely aggregated to update the server model. **Right**: Each test user is allowed to select their level of conformity θ, and are served the corresponding model w_θ.

approach simply seeking to minimize the worst error over the population of clients. Compared to FedAvg, Δ-FL delivers improved prediction to data-poor or non-conforming clients. We present finite time theoretical convergence guarantees for the Δ-FL algorithm when used to train additive models or deep networks and show how to implement it in a way that is compatible and modular with secure aggregation and distributed differential privacy.

1.1 Contributions

We make the following concrete contributions.

- *The Δ-FL Framework*: We introduce the Δ-FL framework, summarized in Figure 1, which seeks to guarantee a minimal level of predictive performance on nonconforming clients. The framework relies on a nonsmooth superquantile-based objective to minimize the tail statistics of the prediction errors on the client data distributions. The objective is parameterized by the conformity level, which is a scalar summary of how closely a client conforms to the population.
- *Optimization Algorithm, Convergence and Privacy Analysis*: To optimize the Δ-FL objective, we present a federated optimization algorithm which interleaves differentially private client reweighting steps with federated averaging steps. We establish bounds on its rate of convergence in the convex and nonconvex cases. Further, we provided an analysis of the differential privacy of the proposed algorithm.
- *Numerical Experiments*: We perform numerical experiments using neural networks and linear models, on tasks including image classification, and sentiment analysis based on public datasets. The experiments demonstrate superior performance of Δ-FL over state-of-the-art baselines on the upper quantiles of the error on test clients, with particular improvements on data-poor clients, while being competitive on the mean error.

Outline. We start with Section 2 to describe the related work. Section 3 describes the general setup, recalls the FedAvg algorithm for federated learning, and formalizes the notions of conformity and heterogeneity. Section 4 presents a federated optimization algorithm for Δ-FL. We analyze its convergence in the convex and non-convex cases, as well as its differential privacy properties in Section 5. We discuss extension to other risk measures and relations to fair allocation in Section 6. Section 7 presents experimental results, comparing the proposed approach to existing ones, on benchmark datasets for federated learning. Detailed proofs and additional details are deferred to the supplement. The code and the scripts to reproduce results are made publicly available at `https://github.com/krishnap25/simplicial-fl`.

An early version of this work was presented at IEEE CISS [45]. This paper extends and improves upon it in several respects. First, we give an improved and sharp convergence analysis in both the general nonconvex as well as strongly convex cases. Second, we augment our algorithm with differential privacy and analyze its privacy and utility. Finally, we conduct an expanded numerical study, including comparing with baselines such as Tilted-ERM [52] that were

published after our conference paper.

Notation. The norm $\|\cdot\|$ denote the Euclidean norm $\|\cdot\|_2$ in $\mathbb{R}^d$. We use $\Delta^{N-1} = \left\{\pi \in \mathbb{R}_+^N : \sum_{k=1}^N \pi_k = 1\right\}$ to denote the probability simplex in $\mathbb{R}^N$.

2 Related Work

Federated learning was introduced by [54] to handle distributed on-client learning [37, 49, 28]. A plethora of recent extensions have also been proposed [84, 72, 55, 81, 56, 74, 35, 73, 16]. Our approach of addressing the statistical heterogeneity by proposing a new objective, which is broadly applicable in these settings.

Distributionally robust optimization [4], which aims train models that perform uniformly well across all subgroups instead of just on average, has witnessed a flurry of recent research [46, 23, 42]. This approach is closely related to the risk measures studied in economics and finance [1, 66, 3, 27]. The recent works [43, 47, 17] study optimization algorithms for risk measures. More broadly, risk measures have been successfully utilized in problems ranging from bandits [71, 12], reinforcement learning [13, 77, 14], and fairness in machine learning [82, 65]. The federated learning method we here is based on the superquantile [67], a popular risk measure. We propose a stochastic optimization algorithm adapted to the federated setting and prove the convergence.

Addressing statistical heterogeneity in federated learning has led to two lines of work. The first includes algorithmic advances to alleviate the effect of heterogeneity on convergence rates while still minimizing the classical expectation-based objective function of empirical risk minimization. These techniques include the use of proximal terms [50], control variates [38] or augmenting the server updates [78, 63]; we refer to the recent survey [79] for details. More generally, the framework of local SGD has been used to study federated optimization algorithms [76, 85, 30, 21, 53, 39, 40]. Compared to these works which study federated optimization algorithms in the smooth case, we tackle in our analysis the added challenge of nonsmoothness of the superquantile-based objective in both the general nonconvex and strongly convex cases.

The second line of work addressing heterogeneity involves designing new objective functions by modeling statistical heterogeneity and designing optimization algorithms. The AFL framework to minimize the worst-case error across all training clients and associated generalization bounds were given in [57]. The concurrent work of [51] proposes the q-FFL framework whose objective is inspired by fair resource allocation to minimize the L^p norm of the per-client losses. Several related works were also published following the initial presentation of the current work [44]. A federated optimization algorithm for AFL was proposed and its convergence was analyzed in [19]. Distributional robustness to affine shifts in the data was considered in [64] along with convergence guarantees. Finally, a classical risk measure, namely the entropic risk measure, was considered in [52]. We note that no convergence guarantees are currently known for the stochastic optimization algorithms of [51].

3 Problem Setup

We begin this section by recalling the standard setup of federated learning in Section 3.1. We then describe the standard approach to federated learning and its associated optimization, FedAvg [54] in Section 3.2. We then describe the statistical heterogeneity in some detail in Section 3.3.

3.1 Federated Learning Setup

Federated learning consists of heterogeneous clients which collaboratively train a machine learning model under the orchestration of a central server. The model is then deployed on all clients, including those not seen during training.

Let the vector $w \in \mathbb{R}^d$ denote the d model parameters. We assume that each client has a distribution q over some data space such that the data on the client is sampled i.i.d. from q. The loss incurred by the model $w \in \mathbb{R}^d$ on this client is $F(w; q) := \mathbb{E}_{\xi\sim q}[f(w; \xi)]$, where $f(w; \xi)$ is the chosen loss function, such as the logistic loss, on input-output pair ξ under the model w. The expectation above is assumed to be well-defined and finite. For a given distribution q, smaller values of $F(\cdot; q)$ denote a better fit of the model to the data.

There are N clients available for training. We number these clients as $1, \ldots, N$ and denote the distribution on training client k by q_k. We denote the loss on client k by $F_k(w) := F(w; q_k)$. The goal of federated learning is to train a model w so that it achieves good performance when deployed *each* test client, including those which are unseen during training. Owing to statistical heterogeneity of federated learning, the distribution p of a specific test client could be different from the average distribution $(1/N) \sum_{k=1}^{N} q_k$ that the model is trained on.

Each federated learning method is characterized by an objective function and the federated optimization algorithm used to minimize it. It is not possible to achieve good performance on each client *simultaneously* with a single model w, as it would be a difficult multiobjective optimization problem. The usual approach is combine the per-client losses into a scalar and minimize this objective. The choice of the objective function and optimization algorithm are primarily determined by the three key aspects of federated learning [37, 49]:

(1) *Communication Bottleneck*: The repeated exchange of massive models between the server and clients over resource-limited wireless networks makes communication an important bottleneck. Therefore, training algorithms should be able to trade-off more local computation for lower communication cost.

(2) *Statistical Heterogeneity*: The training distribution q_k and a specific test distribution p are likely to be different from each other. Therefore, a model which works well *on average* over all test clients might not work well on *each individual* test client.

(3) *Privacy*: The data on each client is extremely privacy-sensitive. Federated learning is designed to protect data privacy since no user data is transferred to a data center. This privacy is enhanced by *secure aggregation* of model parameters, which refers to aggregating client updates such that no client update is directly revealed to any other client or the server. This is achieved by cryptographic protocols based on secure multiparty communication [5].

3.2 Federated Learning and the FedAvg algorithm

Analogous to the classical expectation-based objective function in empirical risk minimization approach, the standard objective in federated learning is to minimize the average loss on the training clients

$$\min_{w \in \mathbb{R}^d} \frac{1}{N} \sum_{k=1}^{N} F_k(w) + \frac{\lambda}{2} \|w\|^2 , \tag{1}$$

where $\lambda \geq 0$ is a regularization parameter. We will call this objective as the *vanilla FL* objective.

The de facto standard training algorithm is FedAvg [54]. Each round of the algorithm consists in following steps:

(a) The server samples a set S of m clients from $[N]$ and broadcasts the current model $w^{(t)}$ to these clients.

(b) Staring from $w_{k,0}^{(t)} = w^{(t)}$, each client $k \in S$ makes τ local gradient or stochastic gradient descent steps[1] with a learning rate γ:

$$w_{k,j+1}^{(t)} = w_{k,j}^{(t)} - \gamma \nabla F_k(w_{k,j}^{(t)}) .$$

(c) The models from the selected clients are sent to the server and aggregated to update the server model

$$w^{(t+1)} = \frac{1}{m} \sum_{k \in S} w_{k,\tau}^{(t)} .$$

FedAvg addresses the communication bottleneck by using $\tau > 1$ local computation steps as opposed to $\tau = 1$ local steps in minibatch SGD. It also performs the averaging step (c) securely to enhance data privacy. However, the vanilla FL objective places a limit on how well statistical heterogeneity can be addressed. By minimizing the average training loss, the resulting model w can sacrifice performance on "difficult" clients in order to perform well on average. In other words, it is not guaranteed to perform well on *individual* test clients, whose distribution p might be quite different from the average training distribution $(1/N) \sum_{k=1}^{N} q_k$. Our goal in this work is to design an objective function, different from the vanilla FL objective (1) to better handle statistical heterogeneity and the associated train-test mismatch. We also design a federated optimization algorithm similar to FedAvg to optimize it.

[1] For simplicity, we consider full gradient steps on each client.

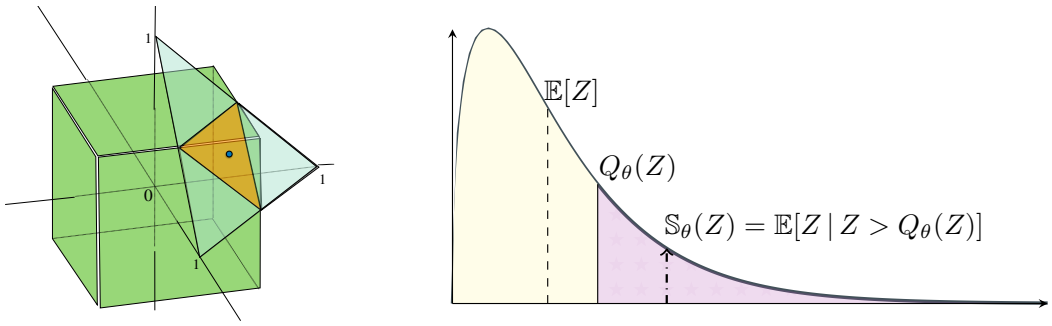

Figure 2: **Left**: The set of mixture weights $\pi = (\pi_1, \pi_2, \pi_3)$ of conformity $\text{conf}(p_\pi) \geq \theta$ is given by the intersection of the box constraints $0 \leq \pi_k \leq (3\theta)^{-1}$ for $k = 1, 2, 3$, with the simplex constraint $\pi_1 + \pi_2 + \pi_3 = 1$. **Right**: $(1-\theta)$-quantile $Q_\theta(Z)$ and superquantile $\mathbb{S}_\theta(Z)$ of a continuous r.v. Z.

3.3 Problem Formulation: Conformity and Heterogeneity

In this work, we consider test clients whose distribution p can be written as a mixture $p_\pi := \sum_{k=1}^N \pi_k q_k$ of the training distribution q_k of the clients with weights $\pi \in \Delta^{N-1}$. Here, Δ^{N-1} denotes the probability simplex in $\mathbb{R}^N$. The test distribution p_π is different from the average training distribution $p_{\text{train}} = (1/N) \sum_{k=1}^N q_k$ if the mixture weights π are different from $1/N$ at training time.

We now define *conformity* of a mixture p_π to the training distribution p_{train}, as a measure of the degree of similarity between p_π and p_{train}.

Definition 1. *The* conformity $\text{conf}(p_\pi) \in [N^{-1}, 1]$ *of a mixture* p_π *with weights* π *is defined as* $\left(N \max_{k \in [N]} \pi_k\right)^{-1}$. *The conformity of a client refers to the conformity of its data distribution.*

When π and α coincide, we have that $\text{conf}(p_\pi) = 1$, and this is the largest possible value of $\text{conf}(p_\pi)$. On the other extreme, suppose that $\pi_k = 1$ for some k, so that π is very different from α. Here, $\text{conf}(p_\pi) = 1/N$, which is the smallest value it takes. In other words, the conformity measures how similar the mixture weights π of p_π are to the original weights α.

More generally, a mixture distribution p_π with $\text{conf}(p_\pi) \geq \theta$ must satisfy $\pi_k \leq 1/(\theta N)$ for each k. In other words, the set of all mixture weights $\{\pi \in \Delta^{N-1} : \text{conf}(p_\pi) \geq \theta\}$ lie in an axis-parallel box around $(1/N, \cdots, 1/N)$, as shown in Figure 2 (left). We do not directly impose a lower bound on π_k because it is not realistic to assume that the distribution on a test client must necessarily contain a component of every training distribution q_k.

Interpretation. Assuming that the training clients are a representative sample of the population of clients, every client's distribution can be well-approximated by a mixture p_π for some $\pi \in \Delta^{N-1}$. The conformity of a client is *a scalar summary of how close it is to the population*. A test client with conformity $\theta \approx 1$ closely conforms to the population. Then, a model trained on the population p_{train} is expected to have a high predictive power. In contrast, a test client with $\theta \approx 0$ would be vastly different from the population p_{train}, and the predictive power of a model trained on p_{train} could be poor. The inverse of the conformity $1/\text{conf}(p_\pi)$ is a measure of how much p_π is shifted relative to p_{train}.

There is a trade-off between the fitting to the population and supporting non-conforming test clients. The conformity θ presents a natural way to encapsulate this tradeoff in a scalar parameter. That is, given a conformity $\theta \in (0, 1)$, we choose to only support test distributions p_π with $\text{conf}(p_\pi) \geq \theta$.

4 Handling Heterogeneity with Δ-FL

In this section, we introduce the Δ-FL framework in Section 4.1, and propose an algorithm to optimize in the federated setting in Section 4.2.

4.1 The Δ-FL Framework

The Δ-FL framework aims to address the train-test distributional mismatch by supplying each test client with a model appropriate to its conformity. Given a discretization $\{\theta_1, \ldots, \theta_r\}$ of $(0, 1]$, Δ-FL maintains r models, one for each conformity level θ_j. The local data is not allowed to leave a client due to privacy restrictions; hence, the conformity of a test client cannot be measured. Instead, we allow each test client to tune their conformity. See the schematic in Figure 1 for an illustration.

To train a model for a conformity level θ, we aim to do well on *all* distributions p_π with $\mathrm{conf}(p_\pi) \geq \theta$:

$$\min_{w \in \mathbb{R}^d} \left[F_\theta(w) := \max_{\pi \in \mathcal{P}_\theta} F(w; p_\pi) + \frac{\lambda}{2}\|w\|^2 \right], \quad \text{where,} \quad \mathcal{P}_\theta := \left\{ \pi \in \Delta^{N-1} \,:\, \mathrm{conf}(p_\pi) \geq \theta \right\}. \tag{2}$$

In contrast, the vanilla FL objective optimizes $F(w; p_{\mathrm{train}})$, which is defined on the basis of the training distribution p_{train}. We observe that Δ-FL is designed to be robust on all test clients with conformity at least θ.

Connection to the Superquantile. The objective function of (2) brings the notion of superquantile into play. For $\theta \in (0, 1)$, the $(1-\theta)$-superquantile $\mathbb{S}_\theta(Z)$ of a continuous random variable Z is simply its tail expectation $\mathbb{S}_\theta(Z) = \mathbb{E}[Z \mid Z > Q_\theta(Z)]$, where $Q_\theta(Z)$ is the $(1-\theta)$-quantile of Z. The superquantile, also known as the conditional value at risk (CVaR), thus quantifies the worst-case or tail behavior of a random variable Z; see Figure 2. More generally, the following definition is applicable to both discrete and continuous random variables [66]

$$\mathbb{S}_\theta(Z) = \min_{\eta \in \mathbb{R}} \left\{ \eta + \frac{1}{\theta}\mathbb{E}\left[\max\{0, Z - \eta\}\right] \right\}.$$

Next, we show that the Δ-FL objective is the superquantile of a discrete random variable with the per-client losses.

Property 1. *Let $Z(w)$ be a discrete random variable which takes the value $F_k(w)$ with probability $1/N$ for $k = 1, \ldots, N$. Then, we have that $F_\theta(w) = \mathbb{S}_\theta(Z(w)) + (\lambda/2)\|w\|^2$.*

Proof. The proof follows from the following equality, which holds due to linear programming duality:

$$\max_{\pi \in \mathcal{P}_\theta} \sum_{k=1}^{N} \pi_k x_k = \min_{(\eta, \mu) \in M} \left\{ \eta + \frac{1}{\theta} \sum_{k=1}^{N} \frac{\mu_k}{N} \right\}$$

with $M = \{(\eta, \mu) \in \mathbb{R} \times \mathbb{R}^N_+ \,:\, \mu_k \geq x_k - \eta \text{ for } k \in [N]\}$. □

4.2 Federated Optimization for Δ-FL

We now propose a federated optimization algorithm for the Δ-FL objective (2). While there could be many approaches to opimizing (2), we consider algorithms similar to FedAvg for their ability to avoid communication bottlenecks and preserve the privacy of user data.

The objective function (2) is effectively the average loss of the clients in the tail, as visualized in Figure 2. Therefore, a natural algorithm to minimize it first evaluates the loss on all the clients, and only performs gradient updates on those clients in the tail above the $(1-\theta)$-quantile. However, a practical algorithm cannot assume that all the clients are available at a given point of time. Therefore, we perform the same operation on a subsample of clients.

Concretely, the optimization algorithm for the Δ-FL objective (2) is given in Algorithm 1. It has four steps:

(a) *Model Broadcast* (line 2): The server samples a set S of m clients from $[N]$ and sends the current model $w^{(t)}$.

(b) *Quantile Computation and Reweighting* (lines 3 and 5): Selected clients $k \in S$ and the server collaborate to estimate the $(1-\theta)$-quantile of the losses $F_k(w^{(t)})$ with differential privacy. The clients then update their weights to be zero if their loss is smaller than the estimated quantile, and leave them unchanged otherwise. This ensures that model updates are only aggregated from the tail clients; cf. Figure 2.

(c) *Local Updates* (loop of line 7): Staring from $w^{(t)}_{k,0} = w^{(t)}$, each client $k \in S$ makes τ local gradient or stochastic gradient descent steps with a learning rate γ.

Algorithm 1 The Δ-FL Algorithm

Input:

Input: Initial iterate $w^{(0)}$, number of communication rounds T, number of clients per round m, number of local updates τ, local step size γ

1: **for** $t = 0, 1, \cdots, T-1$ **do**
2: Sample m clients from $[N]$ without replacement in S
3: Estimate the $(1-\theta)$-quantile of $F_k(w^{(t)})$ for $k \in S$ with distributed differential privacy (Algorithm 2); call this $Q^{(t)}$
4: **for** each selected client $k \in S$ in parallel **do**
5: Set $\tilde{\pi}_k^{(t)} = \mathbb{I}\left(F_k(w^{(t)}) \geq Q^{(t)}\right)$
6: Initialize $w_{k,0}^{(t)} = w^{(t)}$
7: **for** $j = 0, \cdots, \tau - 1$ **do**
8: $w_{k,j+1}^{(t)} = (1 - \gamma\lambda) w_{k,j}^{(t)} - \gamma \nabla F_k(w_{k,j}^{(t)})$
9: **end for**
10: **end for**
11: $w^{(t+1)} = \sum_{k \in S} \tilde{\pi}_k^{(t)} w_{k,\tau}^{(t)} / \sum_{k \in S} \tilde{\pi}_k^{(t)}$
12: **end for**
13: **return** w_T

(d) *Update Aggregation* (line 11): The models from the selected clients are sent to the server and aggregated to update the server model, with weights from line 5).

Compared to FedAvg, Δ-FL has the additional step of computing the quantile and new weights $\tilde{\pi}_k^{(t)}$ for each selected client $k \in S$ in lines 3 and 5. Let us consider Δ-FL in relation to the three keys aspects of federated learning which we introduced in Section 3.1.

(1) *Communication Bottleneck*: Identical to FedAvg, Δ-FL algorithm performs multiple computation rounds per communication round.

(2) *Statistical Heterogeneity*: The Δ-FL objective (2) is designed to better handle the statistical heterogeneity by minimizing the worst-case over all test distributions with conformity at least θ, while the vanilla FL objective cannot handle non-conforming clients.

(3) *Privacy*: Identical to FedAvg, Δ-FL does not require any data transfer and the aggregation of line 11 can be securely performed using secure multiparty communication. The extra step of quantile computation is also performed with distributed differential privacy, as we describe next.

Quantile Estimation with Distributed Differential Privacy. The naïve way to compute the quantile of the per-client losses in line 3 of Algorithm 1 is to have the clients send their losses to the server. To avoid the privacy risk of leakage of information about the clients to the server, we compute the quantile with distributed differential privacy [36] using the discrete Gaussian mechanism [11]. The key idea behind differential privacy [24, 25] is to ensure that the addition or removal of the data from one client does not lead to a substantial change in the output of an algorithm. A large change in the output would give a privacy adversary enough signal to learn about the client which was added or removed.

Our algorithm is given in Algorithm 2. All computations are performed on the ring $\mathbb{Z}_M$ of integers modulo M.[2] Each client k first computes a local histogram x_k on $[0, B]$ based on edges $0 \leq l_0 < l_1 < \cdots < l_n = B$, where the losses assumed to bounded as $F_k(w) \in [0, B]$. Each client then adds random discrete Gaussian[3] noise $\xi_k \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2 I_n)$ with scale parameter σ^2, and finally sums them up using secure multiparty computation [5]. We abstract out the details of the secure summation oracle and only require that it return the sum $\left(\sum_{k \in S} x_k\right) \mod M$ without revealing any further information to a privacy adversary.

At the end of all these steps, the server has a histogram $\hat{h} \in \mathbb{R}^n$ which approximates the true histogram $h = \sum_{k \in S} x_k$ of per-client losses. Finally, Algorithm 2 returns the bin edge $l_{j_\theta^*(\hat{h})}$ nearest to the $(1-\theta)$-quantile of the histogram $\hat{h}$

[2] For ease of handling negative integers, we perform modular arithmetic over the ring $\{-M/2 + 1, \cdots, -1, 0, 1, 2, \cdots, M/2\}$ rather than $\{0, 1, \cdots, M-1\}$.

[3] See Appendix B for a formal definition.

Algorithm 2 Quantile Computation with Distributed Differential Privacy

Input: Ring size M, set S of clients where each client k has a scalar $\ell_k \in [0, B]$, target quantile $1-\theta \in (0,1)$, discretization $l_0, l_1, \cdots, l_n$ of $[0, B]$, variance proxy σ^2, scaling factor $c \in \mathbb{Z}_+$

1: Each client k computes local histogram $x_k = \left(\mathbb{I}(l_{j-1} \leq \ell_k < l_j)\right)_{j=1}^n$
2: Each client k samples $\xi_k \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2 I_n)$ and sets $\tilde{x}_k = (cx_k + \xi_k) \mod M$
3: Compute $s = (\sum_{k \in S} \tilde{x}_k) \mod M$ securely
4: Set histogram $\hat{h} = s/c$
5: **return** Quantile estimate $l_{j^*_\theta(\hat{h})}$ corresponding to index $j^*_\theta(\hat{h})$; cf. Eq. (3)

as:

$$j^*_\theta(\hat{h}) = \arg\min_{j \in [n]} \left| \frac{\sum_{i=1}^{j} \hat{h}_i}{\sum_{i=1}^{n} \hat{h}_i} - (1-\theta) \right| . \tag{3}$$

5 Theoretical Analysis

In this section, we analyze the convergence analysis of Δ-FL (Section 5.1) and study the differential privacy properties of the quantile computation (Section 5.2).

5.1 Convergence Analysis

We study the convergence of Algorithm 1 with respect to the objective (2) in two cases: (i) the general non-convex case, and, (ii) when each $F_k(w)$ is convex.

Assumptions. We make some assumptions on the per-client losses F_k, which are assumed to hold throughout this section. For each client $k \in [N]$, the objective F_k is

(a) B-bounded, i.e., $0 \leq F_k(w) \leq B$ for all $w \in \mathbb{R}^d$,
(b) G-Lipschitz, i.e., $|F_k(w) - F_k(w')| \leq G \|w - w'\|$ for all $w, w' \in \mathbb{R}^d$, and,
(c) L-smooth, i.e., F_k is continuously differentiable and its gradient ∇F_k is L-Lipschitz.

Equivalent Algorithm. Algorithm 1 is not amenable to theoretical analysis as it is stated because the quantile function of discrete random variables computed in line 3 is piecewise constant and discontinuous. To overcome this obstacle, we introduce a near-equivalent algorithm in Algorithm 3, which replaces the reweighting step of Algorithm 1 (lines 3 and lines 5) with the ideal reweighting suggested by the objective (2).

Let us start with the case of $S = [N]$. Ideally, we wish the weights $\pi^{(t)}$ to achieve the maximum over π in the objective (2). It then follows by the chain rule [69, Thm. 10.6] that $\sum_{k=1}^{N} \pi_k^{(t)} \nabla F_k(w^{(t)}) \in \partial F_\theta(w^{(t)})$, where ∂F_θ is the regular subdifferential of F_θ. This allows us to derive convergence guarantees.

Algorithm 3 extends this intuition to the setting where only a subsample $S \subset [N]$ of clients are available in each round. We define the counterpart of the constraint set $\mathcal{P}_\theta$ from (2) defined on a subset $S \subset [N]$ of m clients as:

$$\mathcal{P}_{\theta,S} = \left\{ \pi \in \Delta^{|S|-1} \,:\, \pi_k \leq \frac{1}{\theta m}, \text{ for } k \in S \right\} , \tag{4}$$

where we denote $(\pi_k)_{k \in S} \in \mathbb{R}^{|S|}$ by π with slight abuse of notation. With this notation, Algorithm 3 computes the new weights of the clients as

$$\pi^{(t)} = \arg\max_{\pi \in \mathcal{P}_{\theta,S}} \sum_{k \in S} \pi_k F_k(w^{(t)}) .$$

We now analyze how close Algorithm 3 is to Algorithm 1. Let $Z(w)$ be a discrete random variable which takes the value $F_k(w)$ with probability $1/N$ for $k = 1, \ldots, N$, and let $Q_\theta(Z(w))$ denote its $(1-\theta)$-quantile. The weights

Algorithm 3 The Δ-FL Algorithm with Exact Reweighting

Input: Same as Algorithm 1

for $t = 0, 1, \cdots, T-1$ **do**
  Sample m clients from $[N]$ without replacement in S
  Compute $\pi^{(t)} = \arg\max_{\pi \in \mathcal{P}_{\theta,S}} \sum_{k \in S} \pi_k F_k(w^{(t)})$
  for each selected client $k \in S$ in parallel **do**
    Initialize $w_{k,0}^{(t)} = w^{(t)}$
    for $j = 0, \cdots, \tau - 1$ **do**
      $w_{k,j+1}^{(t)} = (1 - \gamma\lambda) w_{k,j}^{(t)} - \gamma \nabla F_k(w_{k,j}^{(t)})$
    end for
  end for
  $w^{(t+1)} = \sum_{k \in S} \pi_k^{(t)} w_{k,\tau}^{(t)}$
end for
return w_T

$\hat{\pi} \in \Delta^{N-1}$ considered in Algorithm 1 (assuming that $Q^{(t)}$ is the exact quantile of $\{F_k(w^{(t)}) : k \in S\}$) are given by a hard-thresholding based on whether $F_k(w)$ is larger than its $(1-\theta)$-quantile:

$$\tilde{\pi}_k = \mathbb{I}\big(F_k(w) \geq Q_\theta(Z(w))\big)\,, \quad \text{and,} \quad \hat{\pi}_k = \frac{\tilde{\pi}_k}{\sum_{k'=1}^N \tilde{\pi}_{k'}}. \tag{5}$$

The objective defined by these weights is $\hat{F}_\theta(w) = \sum_{k=1}^N \hat{\pi}_k F_k(w) + (\lambda/2)\|w\|^2$. The next proposition shows that $\hat{F}_\theta(w) = F_\theta(w)$ under certain conditions, or is a close approximation, in general.

Proposition 2. *Assume $F_1(w) < \cdots < F_N(w)$ and let $k^\star = \lceil \theta N \rceil$. Then, we have,*
(a) $\pi^\star = \arg\max_{\pi \in \mathcal{P}_\theta} \sum_{k=1}^N \pi_k F_k(w)$ is unique,
(b) $Q_\theta(Z(w)) = F_{k^\star}(w)$,
(c) if θN is an integer, then $\hat{\pi} = \pi^\star$ so that $\hat{F}_\theta(w) = F_\theta(w)$, and,
(d) if θN is not an integer, then

$$0 \leq F_\theta(w) - \hat{F}_\theta(w) \leq \frac{B}{\theta N}\,.$$

Proof. We assume w.l.o.g. that $\lambda = 0$. We apply the property that the superquantile is a tail mean (cf. Figure 2) for discrete random variables [67, Proposition 8] to get

$$F_\theta(w) = \frac{1}{\theta N} \sum_{k=k^\star+1}^N F_k(w) + \left(1 - \frac{\lfloor \theta N \rfloor}{\theta N}\right) F_{k^\star}(w)\,.$$

Comparing with (2), this gives a closed-form expression for $\pi^\star$, which is unique because $F_{k^\star-1}(w) < F_{k^\star}(w) < F_{k^\star+1}(w)$. For (b), note that $Q_\theta(Z(w)) = \inf\{\eta \in \mathbb{R} : \mathbb{P}(Z(w) > \eta) \leq \theta\}$ equals $F_{k^\star}(w)$ by definition of $k^\star$. Therefore, if $A^\star = \theta$, $\pi^\star$ coincides exactly with $\hat{\pi}$. When $A^\star \neq \theta$, we have

$$\hat{F}_\theta(w) = \frac{1}{N - k^\star + 1} \sum_{k=k^\star}^N F_k(w)\,.$$

The bound on $\hat{F}_\theta(w) - F_\theta(w)$ follows from elementary manipulations together with $0 \leq F_k(w) \leq B$. □

Proposition 2 shows that when θm is an integer, Algorithm 3 is identical to Algorithm 1 where line 5 exactly computes the quantile of the per-client losses. We record another consequence of Proposition 2, namely, that the reweighting $\pi^{(t)}$ is sparse.

Remark 1. *Proposition 2 shows that Δ-FL's reweighting $\pi^{(t)}$ (line 3 of Algorithm 3) is sparse. That is, $\pi_k^{(t)}$ is non-zero only for exactly $\lceil \theta m \rceil$ clients with the largest losses.*

Bias due to Partial Participation. Note that we define the objective (2) as the maximum over all distributions in $\mathcal{P}_\theta$, but Algorithm 3 only maximizes weights over a set S of m clients in each round (line 3). Therefore, the updates performed by Algorithm 3 are not unbiased. In particular, Algorithm 3 minimizes the objective:

$$\overline{F}_\theta(w) := \mathbb{E}_{S\sim U_m}\left[F_{\theta,S}(w)\right], \quad \text{where } F_{\theta,S}(w) = \max_{\pi\in\mathcal{P}_{\theta,S}} \sum_{k\in S} \pi_k F_k(w) + \frac{\lambda}{2}\|w\|^2$$

is the analogue of (2) defined on a sample $S \subset [N]$ of clients, and U_m is the uniform distribution over subsets of $[N]$ of size m. Fortunately, the bias introduced by Algorithm 3 can be bounded as [48, Prop. 1]

$$\sup_{w\in\mathbb{R}^d} \left|\overline{F}_\theta(w) - F_\theta(w)\right| \le \frac{B}{\sqrt{\theta m}}\,. \tag{6}$$

Our general strategy will be to study the convergence (near-stationarity or near-optimality) in terms of the objective $\overline{F}_\theta$ which Algorithm 3 actually minimizes, and then translate that a convergence result on the original objective F_θ using the bias bound (6).

Convergence: Nonconvex Case. We start with the convergence analysis in the nonconvex case with no regularization (i.e., $\lambda = 0$). Since $\overline{F}_\theta$ is nonsmooth and nonconvex, we state the convergence guarantee in terms of the Moreau envelope of $\overline{F}_\theta$ [32] following the idea of [22, 18]. Given a parameter $\mu > 0$, we define the Moreau envelope of $\overline{F}_\theta$ as

$$\overline{\Phi}_\theta^\mu(w) = \inf_{z\in\mathbb{R}^d} \left\{\overline{F}_\theta(z) + \frac{\mu}{2}\|w - z\|^2\right\}. \tag{7}$$

The Moreau envelope satisfies a number of remarkable properties for $\mu > L$ [22, Lemma 4.3]. It is well-defined, and the infimum on the right hand side admits a unique minimizer, called the proximal point of w, and denoted $\operatorname{prox}_{\overline{F}_\theta/\mu}(w)$. Second, the Moreau envelope is continuously differentiable with $\nabla\overline{\Phi}_\theta^\mu(w) = \mu(w - \operatorname{prox}_{\overline{F}_\theta/\mu}(w))$. Finally, the stationary points of $\overline{\Phi}_\theta^\mu$ and $\overline{F}_\theta$ coincide. Interestingly, the bound $\left\|\nabla\overline{\Phi}_\theta^\mu(w)\right\| \le \varepsilon$ directly implies a near-stationarity on F_θ in the following variational sense: the proximal point $z = \operatorname{prox}_{\overline{F}_\theta/\mu}(w)$ satisfies [22, Sec. 4.1]:

(a) z is close to w; that is, $\|z - w\| \le \varepsilon/\mu$,
(b) z is nearly stationary on $\overline{F}_\theta$; that is $\operatorname{dist}\left(0, \partial\overline{F}_\theta(z)\right) \le \varepsilon$, where $\partial\overline{F}_\theta$ refers to the regular subdifferential, and,
(c) $\overline{F}_\theta$ is uniformly close to F_θ as per (6).

Thus, we state the convergence guarantee of our algorithm in the nonsmooth nonconvex case in terms of the Moreau envelope $\overline{\Phi}_\theta^\mu$ (although it never appeared in the algorithm).

Theorem 3. *Let the number of rounds T be fixed and set $\mu = 2L$. Denote $\Delta F_0 = F_\theta(w^{(0)}) - \inf \overline{F}_\theta$. Let $\hat{w}$ be sampled uniformly at random from the sequence $\left(w^{(0)}, \cdots, w^{(T-1)}\right)$ produced by Algorithm 3 with an appropriately tuned learning rate. Then, we have,*

$$\mathbb{E}\left\|\nabla\overline{\Phi}_\theta^\mu(\hat{w})\right\|^2 \le \sqrt{\frac{\Delta_0 L G^2}{T}} + (1-\tau^{-1})^{1/3}\left(\frac{\Delta_0 L G}{T}\right)^{2/3} + \frac{\Delta_0 L}{T}\,.$$

Proof. Let $z^{(t)} = \operatorname{prox}_{\overline{F}_\theta/\mu}(w^{(t)})$ be the proximal point of $w^{(t)}$. We expand out the recursion $w^{(t+1)} = w^{(t)} -$

$\gamma \sum_{k\in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j}^{(t)})$ to get

$$\begin{aligned}
\overline{\Phi}_\theta^\mu(w^{(t+1)}) &\le \overline{F}_\theta(z^{(t)}) + \frac{\mu}{2}\left\|z^{(t)} - w^{(t+1)}\right\|^2 \\
&= \overline{F}_\theta(z^{(t)}) + \frac{\mu}{2}\left\|z^{(t)} - w^{(t)}\right\|^2 + \mu\gamma\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla F_k(w_{k,j}^{(t)})\right\rangle \\
&\qquad + \frac{\mu\gamma^2}{2}\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla F_k(w_{k,j}^{(t)})\right\|^2 \\
&= \overline{\Phi}_\theta^\mu(w^{(t)}) + \mathcal{T}_1 + \mathcal{T}_2\,.
\end{aligned}$$

The term $\mathcal{T}_1$ which carries a $O(\gamma)$-coefficient controls the rate of convergence while $\mathcal{T}_2$ carries a $O(\gamma^2)$-coefficient and is a noise term. The latter can be controlled by making the learning rate small. We can handle the first term $\mathcal{T}_1$ by leveraging a property of $\overline{F}_\theta$ known as *weak convexity*, meaning that adding a quadratic makes it convex. In particular, $\overline{F}_\theta + (L/2)\|\cdot\|^2$ is convex, so that

$$\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S}\pi_k^{(t)}\nabla F_k(w^{(t)})\right\rangle \le F_{\theta,S}(z^{(t)}) - F_{\theta,S}(w^{(t)}) + \frac{L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2.$$

Next, we take an expectation with respect to the sampling S of clients (i.e., conditioned on $\mathcal{F}^{(t)} = \sigma(w^{(t)})$, the σ-algebra generated by $w^{(t)}$). Since $z^{(t)}$ is independent of S (i.e., $z^{(t)}$ is $\mathcal{F}^{(t)}$-measurable), we get $\overline{F}_\theta$ on the right hand side. Next, we use that $z^{(t)}$ minimizes the strongly convex right hand side of (7) to get

$$\mathbb{E}_t\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S}\pi_k^{(t)}\nabla F_k(w^{(t)})\right\rangle \le -L\left\|z^{(t)} - w^{(t)}\right\|^2 = -\frac{1}{4L}\left\|\nabla\overline{\Phi}_\theta^\mu(w^{(t)})\right\|^2.$$

This gives us a bound on A in terms of $\left\|\nabla\overline{\Phi}_\theta^\mu(w^{(t)})\right\|^2$. A standard argument to handle the noise term B and telescoping the resulting inequality over $t = 0, \cdots, T-1$ completes the proof. The full details are given in Appendix A.1. □

Convergence: Convex Case. We consider the convergence of function values in the case where each F_k is convex. Owing to the non-smoothness of F_θ and $\overline{F}_\theta$, we consider the following smoothed version of the objective in (2) and the corresponding modification to Algorithm 3. First, define the Kullback-Leibler (KL) divergence between $\pi \in \Delta^{|S|-1}$ and the uniform distribution $(1/|S|, \cdots, 1/|S|)$ over $S \subset [N]$ as

$$D_S(\pi) = \sum_{k\in S}\pi_k \log(\pi_k\,|S|)\,.$$

We simply write $D(\pi)$ when $S = [N]$. Inspired by [58, 2, 20], we define the smooth counterpart to (2) as

$$F_\theta^\nu(w) = \max_{\pi\in\mathcal{P}_\theta}\left\{\sum_{k=1}^N \pi_k F_k(w) - \nu D(\pi)\right\} + \frac{\lambda}{2}\|w\|^2\,, \tag{8}$$

where $\nu > 0$ is a fixed smoothing parameter. We have that $|F_\theta^\nu(w) - F_\theta(w)| \le 2\nu\log N$. Finally, we modify line 3 of Algorithm 3 to handle F_θ^ν rather than F_θ as

$$\pi^{(t)} = \arg\max_{\pi\in\mathcal{P}_{\theta,S}}\left\{\sum_{k\in S}\pi_k F_k(w^{(t)}) - \nu D_S(\pi)\right\}. \tag{9}$$

Theorem 4. *Suppose each function F_k is convex and $0 < \lambda < L$. Define a condition number $\kappa = (L+\lambda)/\lambda$ and fix a time horizon $T \geq \sqrt{2\kappa^3}$. Consider the sequence $(w^{(t)})_{t=0}^T$ of iterates produced by the Algorithm 3 with line 3 replaced by (9). Define the averaged iterate*

$$\overline{w}^{(t)} = \frac{\sum_{i=0}^t \beta_i w^{(i)}}{\sum_{i=0}^t \beta_i}, \quad \textit{where} \quad \beta_i = \left(1 - \frac{\gamma\lambda\tau}{2}\right)^{-(1+i)},$$

and $w^\star = \arg\min_{w\in\mathbb{R}^d} F_\theta(w)$. Then, with appropriate tuning of the learning rate γ and smoothing parameter ν, we have the bound

$$\mathbb{E}F_\theta(\overline{w}^{(T)}) - F_\theta(w^\star) \leq \lambda\|w^{(0)} - w^\star\|^2 \exp\left(-\frac{T}{\sqrt{2\kappa^3}}\right) + \frac{G^2}{\lambda T} + \frac{G^2\kappa^2}{\lambda T^2} + \frac{B}{\sqrt{\theta m}},$$

where we hide absolute constants and factors polylogarithmic in the problem parameters T, G, λ, κ.

Remark 2 (About the Rate)**.** *As soon as $T \gtrsim \kappa^{3/2}$ (ignoring constants and polylog factors), we achieve the optimal rate of $1/(\lambda T)$ rate of strongly convex stochastic optimization up to the bias $B/\sqrt{\theta m}$.*

Further, the bias $B/\sqrt{\theta m}$ due to partial participation can be controlled by choosing the cohort size m large enough. In the experiments of Section 7, we obtain meaningful numerical results when m is around 50 or 100 and θ around $1/2$, indicating that the worst-case bound (6) can be pessimistic.

Proof Sketch of Theorem 4. We start with some additional notation. We absorb the regularization into the client losses to define $\tilde{F}_k(w) = F_k(w) + (\lambda/2)\|w\|^2$. Now, consider the smoothed counterpart of (2) on a subset $S \subset [N]$ with a smoothing parameter $\nu > 0$ as

$$F^\nu_{\theta,S}(w) = \max_{\pi\in\mathcal{P}_{\theta,S}} \left\{\sum_{k\in S} \pi_k \tilde{F}_k(w) - \nu D_S(\pi)\right\}.$$

It follows from the properties of smoothing [58, 2] and composition rules that $F^\nu_{\theta,S}$ is L'-Lipschitz, where L' is as defined the statement of the theorem. Finally, let $\mathcal{F}_t$ denote the sigma algebra generated by $w^{(t)}$ and let $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot|\mathcal{F}_t]$.

We start the proof with the decomposition

$$\|w^{(t+1)} - w\|^2 = \|w^{(t)} - w\|^2 - 2\gamma \sum_{k\in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \langle \nabla\tilde{F}_k(w_{k,j}^{(t)}, w^{(t)} - w\rangle + \gamma^2 \left\|\sum_{k\in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla\tilde{F}_k(w_{k,j}^{(t)})\right\|^2,$$

where w is arbitrary. We bound the inner product term using the λ-strong convexity and L-smoothness of $\tilde{F}_k$. We bound the third term by using the variance bound of [48, Prop. 2] as

$$\mathbb{E}_{S\sim U_m}\left\|\sum_{k\in S} \pi_k^{(t)} \nabla F_k(w^{(t)}) - \nabla\overline{F}^\nu_\theta(w^{(t)})\right\|^2 \leq \frac{8G^2}{\theta m},$$

where U_m is the uniform distribution over subsets $S \subset [N]$ of size m, and $\overline{F}^\nu_\theta(w) := \mathbb{E}_{S\sim U_m} F^\nu_{\theta,S}(w)$ as the expectation of $F^\nu_{\theta,S}$ over random subsets $S \sim U_m$. Putting these together and taking $w = \overline{w}^\star := \arg\min \overline{F}^\nu_\theta$ gives the inequality,

$$\begin{aligned}\overline{F}^\nu_\theta(w^{(t)}) - \overline{F}^\nu_\theta(\overline{w}^\star) \leq{}& \frac{16G^2\gamma\tau}{\theta n} + \frac{9(L+\lambda)^2}{\lambda\tau} d^{(t)} \\ &+ \frac{1}{\gamma\tau}\left(1 - \frac{\lambda\gamma\tau}{2}\right)\|w^{(t)} - \overline{w}^\star\|^2 - \frac{1}{\gamma\tau}\mathbb{E}_t\|w^{(t+1)} - \overline{w}^\star\|^2,\end{aligned} \tag{10}$$

where $d^{(t)}$ is the client drift term, defined as

$$d^{(t)} := \mathbb{E}_{S\sim U_m}\left[\sum_{k\in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \|w_{k,j}^{(t)} - w^{(t)}\|^2 \,\middle|\, \mathcal{F}_t\right].$$

Note that $d^{(t)} = 0$ if $\tau = 1$, so $d^{(t)}$ is the bias from making multiple local gradient steps on each client. This term can be bounded in terms of the left-hand side again using smoothness and the variance bound as

$$d^{(t)} \leq c\,\gamma^2\tau^2(\tau-1)\left(G^2 + L'\big(\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\big)\right).$$

The final missing piece is a bound which allows us to translate statements about convergence of $\overline{F}_\theta^\nu$ in terms of convergence of F_θ. We achieve this using the bias bound of (6) together with approximation error of smoothing. Summing (10) with weights as required by $\overline{w}^{(T)}$ with Jensen's inequality, and using (i) the bound on the client drift $d^{(t)}$, (ii) the bound on the bias, (iii) optimizing the choice of the learning rate and smoothing coefficient give the final statement of the theorem. The details are provided in Appendix A.2. □

5.2 Privacy Analysis

We now analyze the privacy and utility of Algorithm 2.

First, recall the definition of zero-concentrated differential privacy [9]: a randomized algorithm $\mathcal{A}$ satisfies $(1/2)\varepsilon^2$-concentrated differential privacy if the Rényi α-divergence $D_\alpha(\mathcal{A}(X)\|\mathcal{A}(X')) \leq \alpha\varepsilon^2/2$ for all $\alpha \in (0,\infty)$ and all sequences X, X' of inputs which differ by the addition or removal of the data of one client.

Intuitively, the addition or removal of the data contributed by one client should not change the output distribution of the randomized algorithm by much, as measured by the Rényi divergence. A smaller value of ε implies a stronger privacy guarantee. This notion of differential privacy can be translated back-and-forth with the usual one, cf. [11].

Error Criterion. We approximate the $(1-\theta)$-quantile of $\{\ell_1, \cdots, \ell_N\}$ by the quantile of a histogram $h = (h_1, \cdots, h_n)$ of client losses with individual entries $h_j = \sum_{k=1}^N \mathbb{I}(l_{j-1} \leq \ell_k < l_j)$, where the bin edges $0 = l_0 < l_1 < \cdots < l_n = B$ are given. The bin edge l_j corresponding to index $j \in [n]$ approximates the $(1-\theta)$-quantile well if $h_1 + \cdots + h_j \approx (1-\theta)(h_1 + \cdots + h_n)$. We measure this error of approximation by the difference between the two sides. Formally, we define the error $R_\theta(h, j)$ of approximating the $(1-\theta)$-quantile of histogram $h \in \mathbb{R}^n$ with index $j \in [n]$ by

$$R_\theta(h, j) = \left|\frac{\sum_{i=1}^j h_i}{\sum_{i=1}^n h_i} - (1-\theta)\right|. \tag{11}$$

We define the best achievable error $R_\theta^*(h)$ for estimating the $(1-\theta)$-quantile of histogram h and the best approximating index $j^*(h)$ as

$$R_\theta^*(h) = \min_{j\in[n]} R_\theta(h,j)\,, \quad \text{and} \quad j_\theta^*(h) = \operatorname*{arg\,min}_{j\in[n]} R_\theta(h,j)\,, \tag{12}$$

where we assume ties are broken in an arbitrary but deterministic manner. Lastly, we define the *quantile error* $\Delta_\theta(h, \hat{h})$ of estimating the quantile of h from that of $\hat{h}$ as

$$\Delta_\theta(\hat{h}, h) = R_\theta\big(h, j_\theta^*(\hat{h})\big)\,.$$

Essentially, if the index $j_\theta^*(\hat{h})$ computed from the estimate $\hat{h}$ corresponds to the $(1-\theta')$-quantile of h, the quantile error satisfies $\Delta_\theta(h, \hat{h}) = |\theta - \theta'|$.

Privacy and Utility Analysis. We now analysis the differential privacy bound of Algorithm 2 as well as the error in the quantile computation.

Theorem 5. *Fix a $\delta > 0$. Suppose that $\sigma \geq 1/2$ and $c > 0$ are given, and and the modular arithmetic is performed on base $M \geq 2 + 2cN + 2N\sqrt{2\sigma^2\log(4Nn/\delta)}$. Then we have the following with probability at least $1-\delta$:*
(a) Algorithm 2 satisfies $(1/2)\varepsilon^2$-concentrated DP with

$$\varepsilon = \min\left\{\sqrt{\frac{c^2}{N\sigma^2} + \frac{\psi n}{2}}, \frac{c}{\sqrt{N}\sigma} + \psi\sqrt{n}\right\},$$

where $\psi = 10\sum_{i=1}^{N-1}\exp\big(-2\pi^2\sigma^2 i/(i+1)\big) \leq 10(N-1)\exp(-2\pi^2\sigma^2)$.

(b) The quantile error of histogram $\hat{h}$ returned by Algorithm 2 is at most

$$\Delta_\theta(\hat{h}, h) \le R^*_\theta(\hat{h}) \left(1 + \sqrt{\frac{2\sigma^2 n}{c^2 N} \log \frac{4}{\delta}}\right) + (2-\theta)\sqrt{\frac{2\sigma^2 n}{c^2 N} \log \frac{4}{\delta}},$$

where $R^*_\theta(\hat{h})$ is the error in the estimation of $(1-\theta)$-quantile of histogram $\hat{h}$.

Let us interpret the result. The effective noise scale is σ/c. Since the dominant term of the privacy error is $\varepsilon \approx c/(\sigma\sqrt{N})$, we choose $\sigma/c \approx (\varepsilon\sqrt{N})^{-1}$, so that the algorithm satisfies $(1/2)\varepsilon^2$-concentrated DP. The role of c is to avoid degeneracy of the discrete Gaussian as $\sigma \to 0$. In particular, the theorem requires $\sigma \ge 1/2$. The error $\Delta_\theta(\hat{h}, h)$ is (ignoring constants and log factors)

$$\Delta_\theta(\hat{h}, h) \lesssim R^*_\theta(\hat{h}) \left(1 + \frac{\sqrt{n}}{\varepsilon N}\right) + (1+\rho)\frac{\sqrt{n}}{\varepsilon N}.$$

If we take $\sigma = O(1)$ and $c = O(\varepsilon\sqrt{N})$, we require $M \gtrsim N^{3/2}$, ignoring constants and log factors.

Proof. Define the event

$$E_{\mathrm{mod}} = \bigcap_{k=1}^{N} \bigcap_{j=1}^{n} \left\{ -\frac{M-2}{2N} \le c x_{k,j} + \xi_{k,j} \le \frac{M-2}{2N} \right\}. \tag{13}$$

Note that under E_{mod}, no modular wraparound occurs in the algorithm, i.e., $\tilde{x}_k = c x_k + \xi_k$ and $\hat{h} = \sum_{k=1}^N \frac{\tilde{x}_k}{c} = \sum_{k=1}^n \left(x_k + \frac{\xi_k}{c}\right)$. We assume that E_{mod} holds throughout.

The analysis of the privacy follows from the sensitivity of the sum query. Namely, let $X = (x_1, \cdots, x_N)$ be a sequence and define $A(X) = \sum_{k=1}^N c x_k$ as the (rescaled) sum query. In our case, each x_i is a canonical basis vector since it is a local histogram constructed from a single scalar. Algorithm 2 adds discrete Gaussian noise to the sum query to make it differentially private. That is, we get the randomized algorithm $\mathcal{A}(X) = A(X) + \sum_{k=1}^N \xi_k$. It was shown in [36, Corollary 12] that $\mathcal{A}(X)$ is approximately distributed as $\mathcal{N}_{\mathbb{Z}}(A(X), N\sigma^2)$, so a desired privacy guarantee follows from that of the discrete Gaussian mechanism [11]. In particular, for two sequences X and X' differing by the addition or removal of a single basis vector x', we have that

$$D_\alpha(\mathcal{A}(X) \| \mathcal{A}(X')) \approx D_\alpha(\mathcal{N}_{\mathbb{Z}}(A(X), N\sigma^2) \| \mathcal{N}_{\mathbb{Z}}(A(X'), N\sigma^2)) = \frac{\alpha c^2}{2N\sigma^2}.$$

A rigorous analysis of the error, following the recipe of [36] leads to the first part of the theorem; the details can be found in Appendix B.

For the second part, we analyze the quantile error. Define $\hat{N} = \sum_{j=1}^n \hat{h}_j$, as the analogue to $N = \sum_{j=1}^n h_j$ and shorthand $\rho = 1 - \theta$. We bound the quantile error as

$$\begin{aligned}
\Delta_\theta\left(\hat{h}, h\right) &= \left| \frac{1}{N} \sum_{j=1}^{j^*_\theta(\hat{h})} h_j - \rho \right| \\
&\le \frac{1}{N} \left| \sum_{j=1}^{j^*_\theta(\hat{h})} h_j - \hat{h}_j \right| + \frac{1}{N} \left| \sum_{j=1}^{j^*_\theta(\hat{h})} \hat{h}_j - \hat{N}\rho \right| + \frac{\rho}{n} \left| \hat{N} - N \right| \\
&\le \max_{i \in [n]} \frac{1}{cN} \left| \sum_{j=1}^{i} \sum_{k=1}^{N} \xi_{k,j} \right| + \left(1 + \frac{|\hat{N} - N|}{N}\right) R^*_\theta(\hat{h}) + \frac{\rho}{N} \left| \hat{N} - N \right|.
\end{aligned}$$

Let us define an event E_{sum} under which we can bound each of the terms to get the desired bound:

$$E_{\mathrm{sum}} = \left\{ \max_{j \in [n]} \left| \sum_{i=1}^{j} \sum_{k=1}^{N} \xi_{k,i} \right| \le \sqrt{2\sigma^2 N n \log(4/\delta)} \right\}. \tag{14}$$

Finally, it remains to bound the probability of E_{mod} and E_{sum}. This can be achieved using standard concentration arguments, which we defer to Appendix B. □

6 Discussion

We discuss connections of Δ-FL to risk measures and fair resource allocation.

Connection to Risk Measures. The framework of risk measures in economics and finance formalizes the notion of minimizing the worst-case cost over a set of distributions [26, 68, 27]. The superquantile $\mathbb{S}_\theta(\cdot)$ is a special case of a risk measure. The Δ-FL framework, which minimizes the superquantile of the per-client losses (Property 1), can be extended to other risk measures $\mathbb{M}$ by minimizing the objective

$$F_M(w) := \mathbb{M}(Z(w)) + \frac{\lambda}{2}\|w\|^2\,,$$

where $Z(w)$ is a discrete random variable which takes value $F_k(w)$ with probability α_k for $k \in [N]$. Another example of a risk measure is the *entropic risk measure*, which is defined as $\mathbb{M}^\nu_{\text{ent}}(Z) = \mathbb{E}[\exp(\nu Z)]/\nu$ where $\nu \in \mathbb{R}_+$ is a parameter. The analogue of Δ-FL with the entropic risk minimizes

$$F^\nu_{\text{ent}}(w) = \frac{1}{\nu}\log\left(\sum_{k=1}^N \alpha_k \exp\left(\nu F_k(w)\right)\right) + \frac{\lambda}{2}\|w\|^2\,.$$

This objective $F^\nu_{\text{ent}}(w)$ coincides with the one studied recently in [52] under the name Tilted-ERM subsequent to the first presentation of this work [44]. Finally, we note that F^ν_{ent} is also related to the smoothed objective F^ν_θ from (8) as the limit

$$F^\nu_{\text{ent}}(w) = \lim_{\theta\to 0} F^\nu_\theta(w)\,,$$

where we extend the definition $D(\pi) = \sum_{k=1}^N \pi_k \log(\pi_k/\alpha_k)$ as the KL divergence between π and α for unequal α_k's.

Maximin Strategy for Resource Allocation. We would like to point out an interesting analogy between distributional robustness and proportional fairness. The superquantile-based objective in Eq. (2) is a maximin-type objective that is reminiscent of maximin objectives used in load balancing and network scheduling [41, 75, 60].

We can draw an analogy between the two worlds, federated learning and resource allocation resp., by identifying errors to rates and clients to users. The maximin fair strategy to resource allocation seeks to treat all users as fairly as possible by making their rates as large and as equal as possible, so that no rate can be increased without sacrificing other rates that are smaller or equal [60].

Our superquantile-based Δ-FL framework builds off the maximin decision theoretic foundation to frame an objective that we *optimize* with respect to *parameters* of models, and this, iteratively, over multiple rounds of client-server communication, while preserving privacy of each client.

This compositional nature of our problem, where we optimize a composition (in the mathematical sense) of a maximin-type objective and a loss function and model predictions is a difference with resource allocation in communication networks. Further explorations of the analogy are left for future work.

Model Family and Conformity Levels θ. Using a single global value of the conformity level θ for all clients could fail to balance supporting clients with low conformity with fitting the population. On the other hand, measuring the conformity of clients requires transfer of user data, a violation of privacy. To circumvent this issue, we use a similar idea to the one of [51] where a family of models is trained simultaneously for various levels, and each test client can then tune its conformity.

7 Experiments

In this section, we demonstrate the effectiveness of Δ-FL in handling heterogeneity in federated learning. Our experiments were implemented in Python using automatic differentiation provided by PyTorch while the data was

Table 1: Dataset description and statistics.

Task	Dataset	#Classes	Devices	#Data per client	
				Median	Max
Image Recognition	EMNIST	62	1730	179	447
Sentiment Analysis	Sent140	2	877	69	549

preprocessed using LEAF [10]. The code to reproduce our experiments can be found online.[4] We start by describing the datasets, tasks and models in Section 7.1. We present numerical comparisons to several recent works – we list them in Section 7.2 and present the experimental results in Section 7.3. Finally, we demonstrate that Δ-FL provides the most favorable tradeoff between average error and the error on nonconforming clients in Section 7.4. Full details regarding the experiments as well as additional results are provided in the supplementary material.

7.1 Datasets, Tasks and Models

We consider two learning tasks. The dataset and task statistics are summarized in Table 1.

(a) *Character Recognition*: We use the EMNIST dataset [15], where the input x is a 28×28 grayscale image of a handwritten character and the output y is its label (0-9, a-z, A-Z). Each client is a writer of the character x. The weight α_k assigned to author k is the number of characters written by this author. We train both a linear model and a convolutional neural network architecture (ConvNet). The ConvNet consists in two 5×5 convolutional layers with max-pooling followed by one fully connected layer. Outputs are vectors of scores with respect to each of the 62 classes. The multinomial logistic loss is used to train both models.

(b) *Sentiment Analysis*: We use the Sent140 dataset [29] where the input x is a tweet and the output $y = \pm 1$ is its sentiment. Each client is a distinct Twitter user. The weight α_k assigned to user k is the number of tweets publised by this user. We train both a logistic regression and a Long-Short Term Memory neural network architecture (LSTM). The LSTM is built on the GloVe embeddings of the words of the tweet [33]. The hidden dimension of the LSTM is same as the embedding dimension, i.e., 50. We refer to the latter as "RNN". The loss used to train both models is the binary logistic loss.

7.2 Algorithms and Hyperparameters

We list here the recent works we perform numerical comparisons with and discuss their hyperparameters.

Algorithms. As discussed in Section 3, a federated learning method is characterized by the objective function as well as the federated optimization algorithm. We consider two methods optimizing the vanilla FL objective: FedAvg [54] and FedProx [50]. The latter augments FedAvg with a proximal term for more stable optimization. We compare to one more variant of FedAvg. Note that Δ-FL the weight $\pi^{(t)}$ (see line 3 of Algorithm 3) is sparse, i.e., it is non-zero for only some of the m selected clients, cf. Proposition 2. This is equivalent to a fewer number of effective clients per round, which is θm on average. We use as baseline FedAvg with θm clients per round, where m is the number of clients per round in Δ-FL; we call it FedAvg-Sub.

We also consider two heterogeneity-aware objectives: Tilted-ERM [52], which is the analogue of Δ-FL with the entropic risk measure (cf. Section 6) and AFL [57], whose objective is obtained as the limit $\lim_{\theta \to 0} F_\theta(w)$ of the Δ-FL objective. We also consider q-FFL [51], which raises the per-client loss F_k to the $(q+1)^{\text{th}}$ power, for some $q > 0$. We optimize q-FFL and Tilted-ERM with the federated optimization algorithms proposed in their respective papers. We use q-FFL with $q = 10$ in place of AFL, as it found to have more stable convergence with similar performance.

Hyperparameters. We fix the number of clients per round to be $m = 100$ for each dataset-model pair with the exception of Sent140-RNN, for which we use $m = 50$. We fixed an iteration budget for each dataset during which

[4] https://github.com/krishnap25/simplicial-fl

Table 2: **90^{th} percentile** of the distribution of misclassification error (in $\%$) on the test devices. Each entry is the mean over five random seeds while the standard deviation is reported in the subscript. The boldfaced/highlighted entries denote the smallest value for each dataset-model pair.

	EMNIST		**Sent140**	
	Linear	**ConvNet**	**Linear**	**RNN**
FedAvg	$49.66_{0.67}$	$28.46_{1.07}$	$46.83_{0.54}$	$49.67_{3.95}$
FedAvg-Sub	$50.28_{0.77}$	$27.57_{0.81}$	$46.60_{0.38}$	$46.94_{3.84}$
FedProx	$49.15_{0.74}$	$27.01_{1.86}$	$46.83_{0.54}$	$49.86_{4.07}$
q-FFL	$49.90_{0.58}$	$28.02_{0.80}$	$\mathbf{46.39}_{0.40}$	$48.66_{4.68}$
Tilted-ERM	$48.59_{0.62}$	$25.46_{1.49}$	$46.69_{0.49}$	$46.54_{3.27}$
AFL	$51.62_{0.28}$	$45.08_{1.00}$	$47.52_{0.32}$	$57.78_{1.19}$
Δ-FL, $\theta = 0.8$	$49.10_{0.24}$	$26.23_{1.15}$	$46.44_{0.38}$	$\mathbf{46.46}_{4.39}$
Δ-FL, $\theta = 0.5$	$\mathbf{48.44_{0.38}}$	$\mathbf{23.69_{0.94}}$	$46.64_{0.41}$	$50.48_{8.24}$
Δ-FL, $\theta = 0.1$	$50.34_{0.95}$	$25.46_{2.77}$	$51.39_{1.07}$	$86.45_{10.95}$

FedAvg converged. We tuned a learning rate schedule using grid search to find the smallest terminal loss averaged over training clients for FedAvg. The same iteration budget and learning rate schedule were used for *all* other methods including Δ-FL. Each method, except FedAvg-Sub, selected m clients per round for training, as specified earlier. The regularization parameter λ, and the proximal weight of *FedProx* were tuned to minimize the 90^{th} percentile of the misclassification error on a held-out subset of training clients. We run q-FFL for $q \in \{10^{-3}, 10^{-2}, \ldots, 10\}$ and report q with the smallest 90^{th} percentile of misclassification error on *test* clients. We run Tilted-ERM with a temperature parameter $\nu \in \{0.1, 0.5, 1, 5, 10, 50, 100, 200\}$ and also report ν with the smallest 90^{th} percentile of misclassification error on *test* clients. We optimize Δ-FL with Algorithm 3 for conformity $\theta \in \{0.8, 0.5, 0.1\}$.

7.3 Experimental Results

We measure in Table 2 the 90^{th} percentile of the misclassification error across the test clients, as a measure of the right tail of the per-client performance. We also measure in Table 3 the mean error, which measures the average test performance. Our main findings are summarized below.

Δ-FL consistently achieves the smallest 90^{th} percentile error. Δ-FL achieves a 3.3% absolute (12% relative) improvement over any vanilla FL objective on EMNIST-ConvNet. Among the heterogeneity aware objectives, Δ-FL achieves 1.8% improvement over the next best objective, which is Tilted-ERM. We note that q-FFL marginally outperforms Δ-FL on Sent140-Linear, but the difference 0.05% is much smaller than the standard deviation across runs.

Δ-FL is competitive at multiple values of θ. For EMNIST-ConvNet, Δ-FL with $\theta \in \{0.5, 0.8\}$ is better in 90^{th} percentile error than *all* other methods we compare to, and Δ-FL with $\theta = 0.1$ is tied with $Tilted - ERM$, the next best method. We also empirically confirm that Δ-FL interpolates between FedAvg ($\theta \to 1$) and AFL ($\theta \to 0$).

Δ-FL works best for larger values of conformity levels. We observe that Δ-FL with $\theta = 0.1$ is unstable for Sent140-RNN. This is consistent with Theorem 4, which requires m to be much larger than $1/\theta$ (cf. Remark 2). Indeed, this can be explained by Δ-FL's sparse re-weighting, which only gives non-zero weights to $\theta m = 5$ clients on average in each round (cf. Remark 1).

Δ-FL is yet competitive in terms of average error. Perhaps surprisingly, Δ-FL actually gets the best test error performance on EMNIST-ConvNet and Sent140-Linear. This suggests that the average test distribution is shifted relative to the average training distribution p_α. In the other cases, we find that the reduction in mean error is small

Table 3: **Mean** of the distribution of misclassification error (in %) on the test devices. Each entry is the mean over five random seeds while the standard deviation is reported in the subscript. The boldfaced/highlighted entries denote the smallest value for each dataset-model pair.

	EMNIST		Sent140	
	Linear	**ConvNet**	**Linear**	**RNN**
FedAvg	$34.38_{0.38}$	$16.64_{0.50}$	$34.75_{0.31}$	$30.16_{0.44}$
FedAvg-Sub	$34.51_{0.47}$	$16.23_{0.23}$	$34.47_{0.03}$	$\mathbf{29.86}_{0.46}$
FedProx	$\mathbf{33.82}_{\mathbf{0.30}}$	$16.02_{0.54}$	$34.74_{0.31}$	$30.20_{0.48}$
q-FFL	$34.34_{0.33}$	$16.59_{0.30}$	$34.48_{0.06}$	$29.96_{0.56}$
Tilted-ERM	$34.02_{0.30}$	$15.68_{0.38}$	$34.70_{0.31}$	$30.04_{0.25}$
AFL	$39.33_{0.27}$	$33.01_{0.37}$	$35.98_{0.08}$	$37.74_{0.65}$
Δ-FL, $\theta = 0.8$	$34.49_{0.26}$	$16.09_{0.40}$	$\mathbf{34.41}_{0.22}$	$30.31_{0.33}$
Δ-FL, $\theta = 0.5$	$35.02_{0.20}$	$\mathbf{15.49}_{\mathbf{0.30}}$	$35.29_{0.25}$	$33.59_{2.44}$
Δ-FL, $\theta = 0.1$	$38.33_{0.48}$	$16.37_{1.03}$	$37.79_{0.89}$	$51.98_{11.81}$

relative to the gains in the 90^{th} percentile error compared to Vanilla FL methods.

Minimizing superquantile loss over all clients performs better than minimizing worst error over all clients. Specifically, AFL which aims to minimize the worst error among all clients, as well as other objectives which approximate it (Δ-FL with $\theta \to 0$, q-FFL with $q \to \infty$, Tilted-ERM with $\nu \to 0$) tend to achieve poor performance. We find that AFL achieves the highest error both in terms of 90^{th} percentile and the mean. Δ-FL offers a more nuanced and more effective approach via the constraint set $\text{conf}(p_\pi) \geq \theta$ than the straight pessimistic approach minimizing the worst error among all clients.

7.4 Exploring the Trade-off Between Average and Tail Error

We visualize in Figures 3 and 4 the distribution of test errors to explore the trade-off various methods provide between the average error and the error on nonconforming clients.

Δ-FL yields improved prediction on non-conforming clients. This can be observed from the histogram of Δ-FL in Figure 3, which exhibits thinner tails than FedAvg or Tilted-ERM. We see that the vanilla FL objective of FedAvg sacrifices performance on the nonconforming clients. Tilted-ERM does improve over FedAvg in this regard, but Δ-FL has a thinner right tail than Tilted-ERM, showing a better handling of heterogeneity.

Δ-FL yields improved prediction on data-poor clients. We observe in Figure 4 that Tilted-ERM and q-FFL mainly improve the performance on data-rich clients, that is clients with lots of data. On the other hand, Δ-FL gives a greater reduction in misclassification error on data-poor clients, that is clients with little data (< 200 examples per client).

8 Conclusion

We present the Δ-FL framework that operates with heterogeneous clients while still guaranteeing a minimal level of predictive performance to each individual client. We model the similarity between client data distributions using the conformity, which is a scalar summary of how closely a client conforms to the population. Δ-FL relies on a superquantile-based objective, parameterized by the conformity, to minimize the tail statistics of the prediction errors on the client data distributions. We present a federated optimization algorithm compatible with secure aggregation, which interleaves client reweighting steps with federated averaging steps. We derive finite time convergence guarantees that cover both convex and non-convex settings. Experimental results on federated learning benchmarks demonstrate

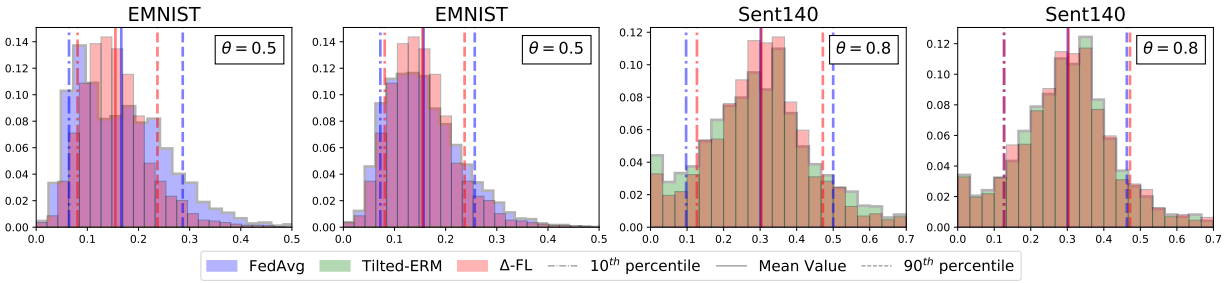

Figure 3: Histogram of misclassification error on test clients for the EMNIST-ConvNet and Sent140-RNN.

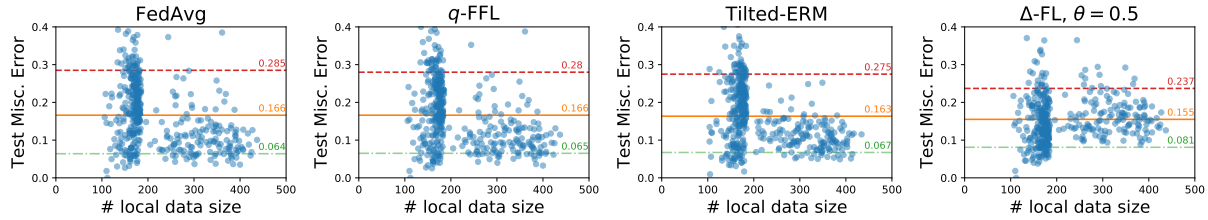

Figure 4: Scatter plots of misclassification error on test clients against its data size for the EMNIST-ConvNet.

superior performance of Δ-FL over state-of-the-art baselines on the upper quantiles of the error on test clients, with particular improvements on data-poor clients, while being competitive on the mean error.

Acknowledgements

The authors would like to thank Zachary Garrett, Peter Kairouz, Jakub Konečný, Brendan McMahan, Sewoong Oh, Krzysztof Ostrowski, Keith Rush, and Lun Wang for fruitful discussions. We acknowledge support from NSF DMS 2023166, DMS 1839371, CCF 2019844, the CIFAR program "Learning in Machines and Brains", faculty research awards, and a JP Morgan PhD fellowship. This work has been partially supported by MIAI – Grenoble Alpes, (ANR-19-P3IA-0003).

References

[1] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent Measures of Risk. *Mathematical finance*, 9(3):203–228, 1999.

[2] A. Beck and M. Teboulle. Smoothing and First Order Methods: A Unified Framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

[3] A. Ben-Tal and M. Teboulle. An Old-New Concept of Convex Risk Measures: The Optimized Certainty Equivalent. *Mathematical Finance*, 17(3):449–476, 2007.

[4] A. Ben-Tal, D. den Hertog, A. D. Waegenaere, B. Melenberg, and G. Rennen. Robust Solutions of Optimization Problems Affected by Uncertain Probabilities. *Management Science*, 59(2):341–357, 2013.

[5] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical Secure Aggregation for Privacy-Preserving Machine Learning. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[6] K. A. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, B. McMahan, T. V. Overveldt, D. Petrou, D. Ramage, and J. Roselander. Towards Federated Learning at Scale: System Design. In *Proceedings of Machine Learning and Systems 2019, MLSys 2019*, 2019.

[7] L. Bottou, F. E. Curtis, and J. Nocedal. Optimization Methods for Large-Scale Machine Learning. *Siam Review*, 60(2):223–311, 2018.

[8] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi. Federated learning of predictive models from federated Electronic Health Records. *Int. J. Medical Informatics*, 112:59–67, 2018.

[9] M. Bun and T. Steinke. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In M. Hirt and A. D. Smith, editors, *Theory of Cryptography Conference*, volume 9985, pages 635–658, 2016.

[10] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konečný, H. B. McMahan, V. Smith, and A. Talwalkar. LEAF: A benchmark for federated settings. *arXiv Preprint*, 2018.

[11] C. L. Canonne, G. Kamath, and T. Steinke. The Discrete Gaussian for Differential Privacy. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[12] A. Cassel, S. Mannor, and A. Zeevi. A General Approach to Multi-Armed Bandits Under Risk Criteria. In *Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1295–1306, 2018.

[13] Y. Chow, A. Tamar, S. Mannor, and M. Pavone. Risk-Sensitive and Robust Decision-Making: a CVaR Optimization Approach. In *Advances in Neural Information Processing Systems 28*, pages 1522–1530, 2015.

[14] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-Constrained Reinforcement Learning with Percentile Risk Criteria. *J. Mach. Learn. Res.*, 18:167:1–167:51, 2017.

[15] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. EMNIST: an extension of MNIST to handwritten letters. *arXiv Preprint*, 2017.

[16] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai. Exploiting Shared Representations for Personalized Federated Learning. In *International Conference on Machine Learning*, volume 139, pages 2089–2099, 2021.

[17] S. Curi, K. Y. Levy, S. Jegelka, and A. Krause. Adaptive Sampling for Stochastic Risk-Averse Learning. In *Neural Information Processing Systems*, 2020.

[18] D. Davis and D. Drusvyatskiy. Stochastic Model-Based Minimization of Weakly Convex Functions. *SIAM Journal on Optimization*, 29(1):207–239, 2019.

[19] Y. Deng, M. M. Kamani, and M. Mahdavi. Distributionally Robust Federated Averaging. In *Neural Information Processing Systems*, 2020.

[20] O. Devolder, F. Glineur, and Y. E. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Math. Program.*, 146(1-2):37–75, 2014.

[21] A. Dieuleveut and K. K. Patel. Communication Trade-offs for Local-SGD with Large Step Size. In *Advances in Neural Information Processing Systems*, pages 13579–13590, 2019.

[22] D. Drusvyatskiy and C. Paquette. Efficiency of minimizing compositions of convex functions and smooth maps. *Mathematical Programming*, 178(1):503–558, 2019.

[23] J. C. Duchi and H. Namkoong. Variance-based Regularization with Convex Objectives. *Journal of Machine Learning Research*, 20(68):1–55, 2019.

[24] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[25] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith. Calibrating Noise to Sensitivity in Private Data Analysis. *J. Priv. Confidentiality*, 7(3):17–51, 2016.

[26] H. Föllmer and A. Schied. Convex measures of risk and trading constraints. *Finance Stochastics*, 6, 2002. doi: 10.1007/s007800200072.

[27] H. Föllmer and A. Schied. *Stochastic Finance: An Introduction in Discrete Time*. 2016. doi: 10.1515/9783110463453.

[28] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor. Federated Learning: A Signal Processing Perspective. *arXiv Preprint*, 2021.

[29] A. Go, R. Bhayani, and L. Huang. Twitter Sentiment Classification using Distant Supervision. *CS224N Project Report, Stanford*, 2009.

[30] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe. Local SGD with Periodic Averaging: Tighter Analysis and Adaptive Synchronization. In *Advances in Neural Information Processing Systems*, pages 11080–11092, 2019.

[31] A. Hard, K. Rao, R. Mathews, S. Ramaswamy, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated Learning for Mobile Keyboard Prediction. *arXiv Preprint*, 2018.

[32] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I: Fundamentals*. Grundlehren der mathematischen Wissenschaften. 1996. ISBN 9783540568506.

[33] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural computation*, 9(8):1735–1780, 1997.

[34] L. Huang, A. L. Shea, H. Qian, A. Masurkar, H. Deng, and D. Liu. Patient Clustering Improves Efficiency of Federated Machine Learning to Predict Mortality and Hospital stay time using Distributed Electronic Medical Records. *Journal of Biomedical Informatics*, 99, 2019.

[35] D. Jhunjhunwala, A. Gadhikar, G. Joshi, and Y. C. Eldar. Adaptive Quantization of Model Updates for Communication-Efficient Federated Learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3110–3114, 2021.

[36] P. Kairouz, Z. Liu, and T. Steinke. The Distributed Discrete Gaussian Mechanism for Federated Learning with Secure Aggregation. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 5201–5212. PMLR, 2021.

[37] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. A. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, H. Qi, D. Ramage, R. Raskar, M. Raykova, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.*, 14(1-2):1–210, 2021.

[38] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh. SCAFFOLD: stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143, 2020.

[39] A. Khaled, K. Mishchenko, and P. Richtárik. Tighter Theory for Local SGD on Identical and Heterogeneous Data. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[40] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich. A Unified Theory of Decentralized SGD with Changing Topology and Local Updates. In *ICML*, 2020.

[41] W. Kubiak. *Proportional Optimization and Fairness*. International Series in Operations Research & Management Science. Springer US, 2008.

[42] D. Kuhn, P. M. Esfahani, V. A. Nguyen, and S. Shafieezadeh-Abadeh. Wasserstein Distributionally Robust Optimization: Theory and Applications in Machine Learning. In *Operations Research & Management Science in the Age of Analytics*, pages 130–166. 2019.

[43] Y. Laguel, J. Malick, and Z. Harchaoui. First-Order Optimization for Superquantile-Based Supervised Learning. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6, 2020.

[44] Y. Laguel, K. Pillutla, J. Malick, and Z. Harchaoui. Device Heterogeneity in Federated Learning: A Superquantile Approach. *arXiv preprint*, 2020.

[45] Y. Laguel, K. Pillutla, J. Malick, and Z. Harchaoui. A Superquantile Approach to Federated Learning with Heterogeneous Devices. In *IEEE CISS*, 2021.

[46] J. Lee and M. Raginsky. Minimax statistical learning with Wasserstein distances. In *Advances in Neural Information Processing Systems*, pages 2687–2696, 2018.

[47] D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-Scale Methods for Distributionally Robust Optimization. In *Neural Information Processing Systems*, 2020.

[48] D. Levy, Y. Carmon, J. C. Duchi, and A. Sidford. Large-Scale Methods for Distributionally Robust Optimization. In *Advances in Neural Information Processing Systems*, 2020.

[49] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[50] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated Optimization in Heterogeneous Networks. In *MLSys*. 2020.

[51] T. Li, M. Sanjabi, and V. Smith. Fair Resource Allocation in Federated Learning. In *International Conference on Learning Representations*, 2020.

[52] T. Li, A. Beirami, M. Sanjabi, and V. Smith. Tilted Empirical Risk Minimization. In *International Conference on Learning Representations*, 2021.

[53] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the Convergence of FedAvg on Non-IID Data. In *ICLR*, 2020.

[54] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS*, pages 1273–1282, 2017.

[55] J. Mills, J. Hu, and G. Min. Communication-Efficient Federated Learning for Wireless Edge Intelligence in IoT. *IEEE Internet Things J.*, 7(7):5986–5994, 2020.

[56] M. Mohammadi Amiri and D. Gündüz. Machine Learning at the Wireless Edge: Distributed Stochastic Gradient Descent Over-the-Air. *IEEE Transactions on Signal Processing*, 68:2155–2169, 2020.

[57] M. Mohri, G. Sivek, and A. T. Suresh. Agnostic Federated Learning. In *ICML*, 2019.

[58] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[59] E. Nurminskii. The quasigradient method for the solving of the nonlinear programming problems. *Cybernetics*, 9 (1):145–150, 1973.

[60] A. Pantelidou and A. Ephremides. *Scheduling in Wireless Networks*. Foundations and trends in networking. Now Publishers, 2011.

[61] M. Paulik, M. Seigel, H. Mason, D. Telaar, J. Kluivers, R. C. van Dalen, C. W. Lau, L. Carlson, F. Granqvist, C. Vandevelde, S. Agarwal, J. Freudiger, A. Byde, A. Bhowmick, G. Kapoor, S. Beaumont, Á. Cahill, D. Hughes, O. Javidbakht, F. Dong, R. Rishi, and S. Hung. Federated Evaluation and Tuning for On-Device Personalization: System Design & Applications. *arXiv Preprint*, 2021.

[62] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing*, pages 1532–1543, 2014.

[63] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konečný, S. Kumar, and H. B. McMahan. Adaptive Federated Optimization. In *International Conference on Learning Representations*, 2021.

[64] A. Reisizadeh, F. Farnia, R. Pedarsani, and A. Jadbabaie. Robust Federated Learning: The Case of Affine Distribution Shifts. In *Neural Information Processing Systems*, 2020.

[65] A. Rezaei, A. Liu, O. Memarrast, and B. D. Ziebart. Robust Fairness Under Covariate Shift. In *AAAI Conference on Artificial Intelligence*, pages 9419–9427, 2021.

[66] R. T. Rockafellar and S. Uryasev. Optimization of Conditional Value-at-Risk. *Journal of Risk*, 2:21–42, 2000.

[67] R. T. Rockafellar and S. Uryasev. Conditional Value-at-Risk for General Loss Distributions. *Journal of banking & finance*, 26(7):1443–1471, 2002.

[68] R. T. Rockafellar and S. Uryasev. The Fundamental Risk Quadrangle in Risk Management, Optimization and Statistical Estimation. *Surveys in Operations Research and Management Science*, 18(1-2):33–53, 2013.

[69] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis*, volume 317. 2009.

[70] R. T. Rockafellar, S. Uryasev, and M. Zabarankin. Risk tuning with generalized linear regression. *Mathematics of Operations Research*, 33(3):712–729, 2008.

[71] A. Sani, A. Lazaric, and R. Munos. Risk-Aversion in Multi-armed Bandits. In *Advances in Neural Information Processing Systems 25s*, pages 3284–3292, 2012.

[72] F. Sattler, K.-R. Müller, and W. Samek. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2020.

[73] T. Sery, N. Shlezinger, K. Cohen, and Y. C. Eldar. Over-the-Air Federated Learning From Heterogeneous Data. *IEEE Transactions on Signal Processing*, 69:3796–3811, 2021.

[74] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui. UVeQFed: Universal Vector Quantization for Federated Learning. *IEEE Trans. Signal Process.*, 69:500–514, 2021.

[75] S. Stanczak, M. Wiczanowski, and H. Boche. *Fundamentals of Resource Allocation in Wireless Networks: Theory and Algorithms*. Foundations in Signal Processing, Communications and Networking. Springer Berlin Heidelberg, 2009.

[76] S. U. Stich. Local SGD Converges Fast and Communicates Little. In *International Conference on Learning Representations*, 2019.

[77] A. Tamar, Y. Chow, M. Ghavamzadeh, and S. Mannor. Policy Gradient for Coherent Risk Measures. In *Advances in Neural Information Processing Systems 28*, pages 1468–1476, 2015.

[78] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. In *Neural Information Processing Systems*, 2020.

[79] J. Wang, Z. Charles, Z. Xu, G. Joshi, H. B. McMahan, M. Al-Shedivat, G. Andrew, S. Avestimehr, K. Daly, D. Data, et al. A Field Guide to Federated Optimization. *arXiv Preprint*, 2021.

[80] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE J. Sel. Areas Commun.*, 37(6):1205–1221, 2019. doi: 10.1109/JSAC.2019.2904348. URL https://doi.org/10.1109/JSAC.2019.2904348.

[81] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor. Federated Learning With Differential Privacy: Algorithms and Performance Analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[82] R. C. Williamson and A. K. Menon. Fairness Risk Measures. In *International Conference on Machine Learning*, 2019.

[83] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage, and F. Beaufays. Applied Federated Learning: Improving Google Keyboard Query Suggestions. *arXiv Preprint*, 2018.

[84] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni. Bayesian Nonparametric Federated Learning of Neural Networks. In *International Conference on Machine Learning*, pages 7252–7261, 2019.

[85] F. Zhou and G. Cong. On the Convergence Properties of a K-step Averaging Stochastic Gradient Descent Algorithm for Nonconvex Optimization. In *International Joint Conference on Artifical Intelligence*, pages 3219–3227, 07 2018.

Federated Learning with Heterogeneous Devices: A Superquantile Optimization Approach

Supplementary Material

Table of Contents

A Convergence Analysis

Below, we restate and prove Theorem 3 as Theorem 6 in Appendix A.1 and Theorem 4 as Theorem 7 in Appendix A.2,

A.1 Convergence Analysis: Non-convex Case

We review some definitions of subdifferentials and weak convexity before we get to the main theorem.

Nonconvex Subdifferentials. We start by recalling the definition of subgradients for nonsmooth functions (in finite dimension), following the terminology of [69]. Consider a function $\psi\colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ and a point $\bar{w}$ such that $\psi(\bar{w}) < +\infty$. The regular (or Fréchet) subdifferential of ψ at $\bar{w}$ is defined by

$$\partial\psi(\bar{w}) = \left\{ s \in \mathbb{R}^d : \ \psi(w) \geq \psi(\bar{w}) + \langle s, w - \bar{w}) \rangle + o(\|w - \bar{w}\|) \right\}.$$

The regular subdifferential thus corresponds to the set of gradients of smooth functions that are below ψ and coincide with it at $\bar{w}$. These notions generalize (sub)gradients of both smooth functions and convex functions: it reduces to the singleton $\{\nabla\psi(\bar{w})\}$ when ψ is smooth and to the standard subdifferential from convex analysis when ψ is convex.

Weak Convexity. We recall the notion of weak convexity, which is one way of characterizing functions which are "close" to convex. A function $\psi : \mathbb{R}^d \to \mathbb{R}$ is said to be η-weakly convex if the function $w \mapsto \psi(w) + (\eta/2)\|w\|^2$ is convex [59]. The class of weakly convex functions includes all convex functions (with $\eta = 0$) and all L-smooth functions (with $\eta = L$).

Weak convexity also admits an equivalent first-order condition: for any $w, z \in \mathbb{R}^d$ and $s \in \partial\psi(w)$, we have,

$$\psi(z) \geq \psi(w) + \langle s, z - w \rangle - \frac{\eta}{2}\|z - w\|^2 . \tag{15}$$

Weak convexity will feature in our developments in two ways:

- In our case, both F_θ as well as $F_{\theta,S}$ are L-weakly convex, since each can be written as the maximum of a family of L-smooth functions [22, Lemma 4.2].
- The prox operator for weakly convex function is well-defined. Let ψ be a η-weakly convex function. Its proximal or prox operator, with parameter $\mu > 0$ is defined as

$$\mathrm{prox}_{\psi/\mu}(w) = \arg\min_z \left\{ \psi(z) + \frac{\mu}{2}\|w - z\|^2 \right\} .$$

  It is well-defined (i.e., the argmin exists and is unique) for $\mu > \eta$, since the function inside the argmin is $(\mu - \eta)$-strongly convex.

In nonsmooth and nonconvex optimization of weakly convex functions, we are interested in finding stationary points w.r.t. the regular subdifferential, i.e., points w satisfying $0 \in \partial\psi(w)$. A natural measure of near-stationarity is, therefore,

$$\mathrm{dist}(0, \partial\psi(w)) = \inf_{s \in \partial\psi(w)} \|s\| .$$

Moreau Envelope. Given a parameter $\mu > 0$, we define the Moreau envelope of $\overline{F}_\theta$ as

$$\overline{\Phi}^\mu_\theta(w) = \inf_z \left\{ \overline{F}_\theta(z) + \frac{\mu}{2}\|w - z\|^2 \right\} .$$

The Moreau envelope is well-defined since $\overline{F}_\theta$ is bounded from below by our assumptions. We will use two standard properties of the Moreau envelope:

- Since $\overline{F}_{\theta,S}$ is L-weakly convex, we have that its Moreau envelope $\overline{\Phi}^\mu_\theta(w)$ is continuously differentiable for $\mu > L$ with

$$\nabla\overline{\Phi}^\mu_\theta(w) = \mu \left(w - \mathrm{prox}_{\overline{F}_\theta/\mu}(w) \right) .$$

- The stationary points of $\overline{\Phi}^\mu_\theta$ and $\overline{F}_\theta$ coincide and $\inf \overline{\Phi}^\mu_\theta = \inf \overline{F}_\theta$ for $\mu > L$.

- We have for all $\mu > 0$ that $\overline{\Phi}_\theta^\mu(w) \leq \overline{F}_\theta(w)$.

Notation. Let $S = S^{(t)}$ denote the random set of clients selected in round t of Algorithm 3. We define

$$\tilde{\nabla} F_{\theta,S}(w^{(t)}) = \sum_{k \in S} \pi_k^{(t)} \nabla F_k(w^{(t)}) \,, \tag{16}$$

where $\pi_k^{(t)} \in \arg\max_{\pi \in \mathcal{P}_{\theta,S}} \sum_{k \in S} \pi_k F_k(w^{(t)})$ is selected as in line 3 of Algorithm 3. A key consequence of the chain rule [69, Thm. 10.6] is

$$\tilde{\nabla} F_{\theta,S}(w^{(t)}) \in \partial F_{\theta,S}(w^{(t)}) \,. \tag{17}$$

Convergence Analysis. We now state and prove the convergence result in the nonconvex case.

Theorem 6. *Fix and the number of rounds T, fix $\mu = 2L$ and set the learning rate*

$$\gamma = \left\{ \frac{1}{4\tau L}, \frac{1}{\tau\sqrt{T}} \sqrt{\frac{\Delta F_0}{LG^2}}, \frac{1}{\tau T^{1/3}} \left(\frac{\Delta F_0}{32 L G^2 (1-\tau^{-1})} \right)^{1/3} \right\} ,$$

where we denote $\Delta F_0 = \overline{\Phi}_\theta^\mu(w^{(0)}) - \inf \overline{\Phi}_\theta^\mu \leq \overline{F}_\theta(w^{(0)}) - \inf \overline{F}_\theta$. Let $\hat{w}$ be sampled uniformly at random from $\{w^{(0)}, \cdots, w^{(T-1)}\}$. Ignoring absolute constants, we have the bound,

$$\mathbb{E} \left\| \nabla \overline{\Phi}_\theta^\mu(\hat{w}) \right\|^2 \leq \sqrt{\frac{\Delta F_0 L G^2}{T}} + \left(\frac{\Delta F_0 L G (1-\tau^{-1})^{1/2}}{T} \right)^{2/3} + \frac{\Delta F_0 L}{T} \,.$$

Proof. We start with some notation. Throughout, we denote $z^{(t)}$ as the proximal point of $w^{(t)}$:

$$z^{(t)} = \operatorname{prox}_{\overline{F}_\theta/\mu}(w^{(t)}) = \arg\min_z \left\{ \overline{F}_\theta(z) + \frac{\mu}{2} \left\| z - w^{(t)} \right\|^2 \right\} .$$

Let $\mathcal{F}^{(t)}$ denote the sigma algebra generated by $w^{(t)}$ and define $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}^{(t)}]$. By definition, we have that $z^{(t)}$ is also $\mathcal{F}^{(t)}$-measurable.

We use the update $w^{(t+1)} = w^{(t)} - \gamma \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j}^{(t)})$ to get

$$\begin{aligned}
&\overline{\Phi}_\theta^\mu(w^{(t+1)}) = \min_z \left\{ \overline{F}_\theta(z) + \frac{\mu}{2} \left\| z - w^{(t+1)} \right\|^2 \right\} \\
&\leq \overline{F}_\theta(z^{(t)}) + \frac{\mu}{2} \left\| z^{(t)} - w^{(t+1)} \right\|^2 \\
&= \overline{F}_\theta(z^{(t)}) + \frac{\mu}{2} \left\| z^{(t)} - w^{(t)} \right\|^2 + \mu\gamma \left\langle z^{(t)} - w^{(t)}, \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j}^{(t)}) \right\rangle + \frac{\mu\gamma^2}{2} \left\| \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j})^{(t)}) \right\|^2 \\
&= \overline{\Phi}_\theta^\mu(w^{(t)}) + \underbrace{\mu\gamma \left\langle z^{(t)} - w^{(t)}, \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j}^{(t)}) \right\rangle}_{=:\mathcal{T}_1} + \underbrace{\frac{\mu\gamma^2}{2} \left\| \sum_{k \in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j}^{(t)}) \right\|^2}_{=:\mathcal{T}_2} .
\end{aligned} \tag{18}$$

We handle both $\mathcal{T}_1$ and $\mathcal{T}_2$ separately. We start with $\mathcal{T}_1$ by defining

$$C_j := \left\langle z^{(t)} - w^{(t)}, \sum_{k \in S} \pi_k^{(t)} \left(\nabla F_k(w_{k,j}^{(t)}) - \nabla F_k(w^{(t)}) \right) \right\rangle .$$

We use the weak convexity of $F_{\theta,S}$, in particular (15), to bound

$$\begin{aligned}\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S} \pi_k^{(t)} \nabla F_k(w_{k,j}^{(t)}) \right\rangle &= \left\langle z^{(t)} - w^{(t)}, \sum_{k\in S} \pi_k^{(t)} \nabla F_k(w^{(t)}) \right\rangle + C_j \\ &\overset{(16)}{=} \left\langle z^{(t)} - w^{(t)}, \tilde{\nabla} F_{\theta,S}(w^{(t)}) \right\rangle + C_j \\ &\overset{(15),(17)}{\leq} F_{\theta,S}(z^{(t)}) - F_{\theta,S}(w^{(t)}) + \frac{L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2 + C_j\,.\end{aligned}$$

Taking an expectation conditioned on $\mathcal{F}^{(t)}$ and noting that $z^{(t)}$ is $\mathcal{F}^{(t)}$-measurable (so the expectation is only over the randomness in S), we get

$$\begin{aligned}&\mathbb{E}_t\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S} \pi_k^{(t)} \nabla F_k(w_{k,j}^{(t)}) \right\rangle \\ &\leq \left(\overline{F}_\theta(z^{(t)}) + \frac{\mu}{2}\left\|z^{(t)} - w^{(t)}\right\|^2\right) + \overline{F}_\theta(w^{(t)}) - \frac{\mu - L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2 + \mathbb{E}_t[C_j]\,.\end{aligned}$$

Note that the function

$$h(z) := \overline{F}_\theta(z) + \frac{\mu}{2}\left\|z - w^{(t)}\right\|^2$$

is $(\mu - L)$-strongly convex and $z^{(t)}$ is its minimizer. This gives,

$$h(w^{(t)}) - h(z^{(t)}) \geq \frac{\mu - L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2.$$

This implies,

$$\mathbb{E}_t\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S} \pi_k^{(t)} \nabla F_k(w_{k,j}^{(t)}) \right\rangle \leq -(\mu - L)\left\|z^{(t)} - w^{(t)}\right\|^2 + \mathbb{E}_t[C_j]\,.$$

Next, we bound the term C_j as follows:

$$\begin{aligned}|C_j| &= \left|\left\langle z^{(t)} - w^{(t)}, \sum_{k\in S} \pi_k^{(t)} \left(\nabla F_k(w_{k,j}^{(t)}) - \nabla F_k(w^{(t)})\right)\right\rangle\right| \\ &\leq \frac{\mu - L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2 + \frac{1}{2(\mu - L)}\left\|\sum_{k\in S} \pi_k^{(t)} \left(\nabla F_k(w_{k,j}^{(t)}) - \nabla F_k(w^{(t)})\right)\right\|^2 \\ &\leq \frac{\mu - L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2 + \frac{1}{2(\mu - L)}\sum_{k\in S} \pi_k^{(t)}\left\|\left(\nabla F_k(w_{k,j}^{(t)}) - \nabla F_k(w^{(t)})\right)\right\|^2 \\ &\leq \frac{\mu - L}{2}\left\|z^{(t)} - w^{(t)}\right\|^2 + \frac{L^2}{2(\mu - L)}\sum_{k\in S} \pi_k^{(t)}\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2.\end{aligned}$$

Together with the previous inequality and the equality $\nabla\overline{\Phi}_\theta^\mu(w^{(t)}) = \mu(w^{(t)} - z^{(t)})$, we get a bound on $\mathcal{T}_1$ as

$$\mathbb{E}_t[\mathcal{T}_1] \leq -\frac{\gamma\tau(\mu - L)}{2\mu}\left\|\nabla\overline{\Phi}_\theta^\mu(w^{(t)})\right\|^2 + \frac{\mu\gamma L^2}{2(\mu - L)} d^{(t)}\,, \tag{19}$$

where $d^{(t)} = \sum_{k\in S}\sum_{j=0}^{\tau-1} \pi_k^{(t)}\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2$ is the client drift. Next, we bound $\mathcal{T}_2$ as

$$\begin{aligned}\mathcal{T}_2 = \frac{\mu\gamma^2}{2}\left\|\sum_{k\in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1} \nabla F_k(w_{k,j}^{(t)})\right\|^2 &\leq \frac{\mu\gamma^2\tau}{2}\sum_{k\in S} \pi_k^{(t)} \sum_{j=0}^{\tau-1}\left\|\nabla F_k(w_{k,j}^{(t)})\right\|^2 \\ &\leq \frac{\mu\gamma^2\tau^2 G^2}{2}\,,\end{aligned} \tag{20}$$

where we used Jensen's inequality and $\left\|\nabla F_k(w_{k,j}^{(t)})\right\|^2 \leq G^2$ since F_k is G-Lipschitz.

Next, we plug in (19) and (20) into (18) and invoke Proposition 12 to bound the client drift $d^{(t)}$ to get

$$\mathbb{E}_t\left[\overline{\Phi}_\theta^\mu(w^{(t+1)})\right] \leq \overline{\Phi}_\theta^\mu(w^{(t)}) - \frac{\gamma\tau(\mu-L)}{2}\left\|\nabla\overline{\Phi}_\theta^\mu(w^{(t)})\right\|^2 + \frac{\mu\gamma^2\tau^2G^2}{2}\left(1+\frac{8L^2\gamma}{\mu-L}(\tau-1)\right).$$

Finally, taking an unconditional expectation, summing this up over $t=0$ to $T-1$ and rearranging gives us the bound

$$\mathbb{E}\left\|\nabla\overline{\Phi}_\theta^\mu(\hat{w})\right\|^2 \leq \frac{2\Delta F_0}{\gamma\tau T} + 2\gamma\tau LG^2\left(1+8L\gamma(\tau-1)\right),$$

where we plugged in $\mu = 2L$. Plugging in the choice of γ (cf. Lemma 15) completes the proof. □

A.2 Convergence Analysis: Strongly Convex Case

The main result is the following.

Theorem 7 (Convergence rate, Strongly Convex Case)**.** *Suppose that each F_k is convex and the regularization parameter satisfies $0 < \lambda < L$. Define notation $\kappa = (L+\lambda)/\lambda$, $w^\star = \arg\min_w F_\theta(w)$ and $\Delta_0 = \|w^{(0)} - w^\star\|^2$. Assume also that the number of rounds is $T \geq \sqrt{2\kappa^3}$. Fix a smoothing parameter*

$$\nu = \begin{cases} \frac{G^2}{\lambda\kappa}, & \text{if } T \leq \sqrt{2\kappa^3}\log\left(1\vee\frac{CT^2}{\kappa^2}\right), \\ \frac{2G^2\kappa^2}{\lambda T^2}\log\left(1\vee\frac{CT^2}{\kappa^2}\right), & \text{if } \sqrt{2\kappa^3}\log\left(1\vee\frac{CT^2}{\kappa^2}\right) \leq T \leq \kappa^2\log\left(1\vee\frac{CT^2}{\kappa^2}\right), \\ \frac{2G^2}{\lambda T}\log\left(1\vee CT\right), & \text{else}, \end{cases}$$

where $C = \lambda^2\Delta_0/(2G^2\log m)$, and a learning rate

$$\gamma = \min\left\{\frac{\sqrt{\lambda}}{18\tau(L+\lambda)\sqrt{L'}}, \frac{1}{4\tau L'}, \frac{1}{\lambda\tau T}\log\left(1\vee\frac{\lambda^2\Delta_0\theta m}{G^2}T\right), \frac{1}{\lambda\tau T}\log^2\left(1\vee\frac{\lambda^2\Delta_0T^2}{G^2\kappa^2(1-\tau^{-1})}T\right)\right\},$$

where $L' = L+\lambda+G^2/\nu$. Consider the sequence $(w^{(t)})_{t=0}^T$ produced by Algorithm 3 run with smoothing parameter ν and learning rate γ, and the corresponding averaged iterate

$$\overline{w}^{(T)} := \frac{\sum_{t=0}^T w^{(t)}\left(1-\frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}}{\sum_{r=0}^T\left(1-\frac{\lambda\gamma\tau}{2}\right)^{-(1+r)}}.$$

Then, ignoring absolute constants, we have the bound,

$$\begin{aligned}\mathbb{E}\left[F_\theta(\overline{w}^{(T)}) - F_\theta(w^\star)\right] \leq{}& \lambda\|w^{(0)}-w^\star\|^2\exp\left(-T/\sqrt{2\kappa^3}\right) + \frac{B}{\sqrt{\theta m}} \\ &+ \frac{G^2}{\lambda T}\left(\frac{1}{\theta m}+\log m\right)\log\left(1\vee\frac{\lambda^2\Delta_0T}{G^2}\right) \\ &+ \frac{G^2\kappa^2}{\lambda T^2}\left(1-\tau^{-1}+\log m\right)\log^2\left(1\vee\frac{\lambda^2\Delta_0T^2}{G^2\kappa^2}\right).\end{aligned}$$

We give the proof in a sequence of results. We start with some notation.

Notation. We define the client drift as

$$d^{(t)} := \mathbb{E}_{S\sim U_m}\left[\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\|w_{k,j}^{(t)} - w^{(t)}\|^2 \,\middle|\, \mathcal{F}_t\right]. \tag{21}$$

We define the averaged superquantile as

$$\overline{F}_\theta^\nu(w) = \mathbb{E}_{S\sim U_m}[F_{\theta,S}^\nu(w)]\,, \tag{22}$$

where U_m is the uniform distribution over subsets of $[N]$ of size m. Finally, let $\overline{w}^\star = \arg\min_w \overline{F}_\theta^\nu(w)$.

Effect of One Round. The crux of the proof of Theorem 7 is the following statement.

Proposition 8. *Consider the setting of Theorem 7. Let $(w^{(t)})_{t\geq 0}$ the sequence of global models generated by Algorithm 3. For any $t \geq 0$, we have:*

$$\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star) \leq \frac{1}{\gamma\tau}\left(1 - \frac{\lambda\gamma\tau}{2}\right)\|w^{(t)} - \overline{w}^\star\|^2 - \frac{1}{\gamma\tau}\|w^{(t+1)} - \overline{w}^\star\|^2 + \frac{16\tau G^2\gamma}{\theta m} + \frac{9\,(L+\lambda)^2}{\tau\lambda}d^{(t)}\,,$$

where $d^{(t)}$ denotes the client drift, defined in (21).

Proof. We denote $\mathbb{E}_t[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_t]$. We expand the update $w^{(t+1)} = w^{(t)} - \gamma\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla\tilde{F}_k(w_{k,j}^{(t)})$ to get

$$\mathbb{E}_t\|w^{(t+1)} - \overline{w}^\star\|^2 = \|w^{(t)} - \overline{w}^\star\|^2 \underbrace{- 2\gamma\mathbb{E}_t\left[\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\left\langle\nabla\tilde{F}_k(w_{k,j}^{(t)}), w^{(t)} - \overline{w}^\star\right\rangle\right]}_{=:A}$$
$$\underbrace{+\,\gamma^2\mathbb{E}_t\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla\tilde{F}_k(w_{k,j}^{(t)})\right\|^2}_{=:B}\,.$$

Let us first bound A. We use λ-strong convexity (cf. (28)) of the functions $\tilde{F}_k$ to get

$$\begin{aligned}\left\langle\sum_{k\in S}\pi_k^{(t)}\nabla\tilde{F}_k(w_{k,j}^{(t)}),\ w^{(t)} - \overline{w}^\star\right\rangle &= \left\langle\sum_{k\in S}\pi_k^{(t)}\nabla\tilde{F}_k(w^{(t)}), w^{(t)} - \overline{w}^\star\right\rangle \\ &\quad + \left\langle\sum_{k\in S}\pi_k^{(t)}\left(\nabla\tilde{F}_k(w_{k,j}^{(t)}) - \nabla\tilde{F}_k(w^{(t)})\right), w^{(t)} - \overline{w}^\star\right\rangle \\ &\geq F_{\theta,S}^\nu(w^{(t)}) - F_{\theta,S}^\nu(\overline{w}^\star) + \frac{\lambda}{2}\|w^{(t)} - \overline{w}^\star\|^2 \\ &\quad - \left|\left\langle\sum_{k\in S}\pi_k^{(t)}\left(\nabla\tilde{F}_k(w_{k,j}^{(t)}) - \nabla\tilde{F}_k(w^{(t)})\right), w^{(t)} - \overline{w}^\star\right\rangle\right|\,.\end{aligned}$$

Next, using successively the triangle inequality, the Cauchy-Schwartz inequality and $(L+\lambda)$-smoothness of the $\tilde{F}_k$ yields:

$$\begin{aligned}\left|\left\langle\sum_{k\in S}\pi_k^{(t)}\left(\nabla\tilde{F}_k(w_{k,j}^{(t)}) - \nabla\tilde{F}_k(w^{(t)})\right), w^{(t)} - \overline{w}^\star\right\rangle\right| &\leq \sum_{k\in S}\pi_k^{(t)}\left|\left\langle\nabla\tilde{F}_k(w_{k,j}^{(t)}) - \nabla\tilde{F}_k(w^{(t)}), w^{(t)} - \overline{w}^\star\right\rangle\right| \\ &\leq \sum_{k\in S}\pi_k^{(t)}\left\|\nabla\tilde{F}_k(w_{k,j}^{(t)}) - \nabla\tilde{F}_k(w^{(t)})\right\|\left\|w^{(t)} - \overline{w}^\star\right\| \\ &\leq \sum_{k\in S}\pi_k^{(t)}(L+\lambda)\left\|w_{k,j}^{(t)} - w^{(t)}\right\|\left\|w^{(t)} - \overline{w}^\star\right\|\,.\end{aligned}$$

Finally, using $2|ab| \le a^2/c^2 + c^2 b^2$ and the convexity of $t \mapsto t^2$,

$$\begin{aligned}
&\left| \left\langle \sum_{k \in S} \pi_k^{(t)} \left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)}) \right), w^{(t)} - \overline{w}^\star \right\rangle \right| \\
&\le \frac{4}{\lambda} \left(\sum_{k \in S} \pi_k^{(t)} (L+\lambda) \|w_{k,j}^{(t)} - w^{(t)}\| \right)^2 + \frac{\lambda}{4} \|w^{(t)} - \overline{w}^\star\|^2 \\
&\le \frac{\lambda}{4} \|w^{(t)} - \overline{w}^\star\|^2 + \frac{4(L+\lambda)^2}{\lambda} \sum_{k \in S} \pi_k^{(t)} \|w_{k,j}^{(t)} - w^{(t)}\|^2 .
\end{aligned}$$

Overall, we bound A as

$$\begin{aligned}
A &\ge 2\gamma \, \mathbb{E}_t \left[\sum_{j=0}^{\tau-1} \left(F_{\theta,S}^{\nu}(w^{(t)}) - F_{\theta,S}^{\nu}(\overline{w}^\star) + \frac{\lambda}{4} \|w^{(t)} - \overline{w}^\star\|^2 - \frac{4(L+\lambda)^2}{\lambda} \sum_{k \in S} \pi_k^{(t)} \|w_{k,j}^{(t)} - w^{(t)}\|^2 \right) \right] \\
&\ge 2\gamma\tau \left(\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star) \right) + \frac{\lambda\gamma\tau}{2} \|w^{(t)} - \overline{w}^\star\|^2 - \frac{8\gamma(L+\lambda)^2}{\lambda} d^{(t)} ,
\end{aligned}$$

where we use the definition of $d^{(t)}$ from (21). We bound B using Proposition 13. Putting these together, we get,

$$\begin{aligned}
\mathbb{E}_t \|w^{(t+1)} - \overline{w}^\star\|^2 \le \left(1 - \frac{\lambda\gamma\tau}{2}\right) \|w^{(t)} - \overline{w}^\star\|^2 - (2\gamma\tau - 4\gamma^2\tau^2 L')(\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star)) \\
+ \frac{16\tau^2 G^2 \gamma^2}{\theta m} + 2\left(\gamma^2 \tau (L+\lambda)^2 + 4\gamma \frac{(L+\lambda)^2}{\lambda} \right) d^{(t)} .
\end{aligned}$$

With $\gamma \le (4\tau L')^{-1}$ we have $2\gamma\tau - 4\gamma^2\tau^2 L' \ge \gamma\tau$. Likewise, the same condition on γ also implies $2(\gamma(L+\lambda)^2 + 4(L+\lambda)^2/(\tau\lambda)) \le 9(L+\lambda)^2/(\tau\lambda)$. Rearranging completes the proof. □

Proof of Theorem 7. We are now ready to prove the theorem.

Proof of Theorem 7. Plugging in the client drift bound of Proposition 12 into the bound of Proposition 8 and rearranging, we get

$$\begin{aligned}
&\left(1 - \frac{18L'(L+\lambda)^2 \tau^2 \gamma^2 e^2}{\lambda}\right) \left(\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star) \right) \le \\
&\frac{1}{\gamma\tau} \left(1 - \frac{\lambda\gamma\tau}{2}\right) \|w^{(t)} - \overline{w}^\star\|^2 - \frac{1}{\gamma\tau} \mathbb{E}_t \|w^{(t+1)} - \overline{w}^\star\|^2 + \frac{16\tau G^2 \gamma}{\theta m} + \frac{9G^2 (L+\lambda)^2 \tau^2 \gamma^2 e^2}{\lambda} \left(4 + \frac{8}{\theta m}\right) .
\end{aligned}$$

Since $36e^2 \le 18^2$ for $\gamma \le \sqrt{\lambda}(18\tau(L+\lambda)\sqrt{L'})^{-1}$, we have $18L'(L+\lambda)^2 \tau^2 \gamma^2 e^2/\lambda \le \frac{1}{2}$ which implies:

$$\begin{aligned}
\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star) \le \frac{2}{\gamma\tau} \left(1 - \frac{\lambda\gamma\tau}{2}\right) \|w^{(t)} - \overline{w}^\star\|^2 - \frac{2}{\gamma\tau} \mathbb{E}_t \|w^{(t+1)} - \overline{w}^\star\|^2 \\
+ \underbrace{\frac{32\tau G^2 \gamma}{\theta m} + \frac{18G^2 (L+\lambda)^2 \tau^2 \gamma^2 e^2}{\lambda} \left(4 + \frac{8}{\theta m}\right)}_{=: \mathcal{T}_1} .
\end{aligned}$$

Next, we use convexity to get

$$\mathbb{E}\left[\overline{F}_\theta^{\nu}(\overline{w}^{(T)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star) \right] \le \frac{1}{\sum_{t=0}^{T} \left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}} \sum_{t=0}^{T} \left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)} \mathbb{E}\left[\overline{F}_\theta^{\nu}(w^{(t)}) - \overline{F}_\theta^{\nu}(\overline{w}^\star) \right]$$

so that telescoping the sum yields

$$\mathbb{E}\left[\overline{F}_\theta^\nu(\overline{w}^{(T)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right] \leq \frac{2\left\|w^{(0)} - \overline{w}^\star\right\|^2}{\gamma\tau\sum_{t=0}^T \left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)}} + \mathcal{T}_1 \ .$$

Now, we can lower bound the denominator with

$$\sum_{t=0}^T \left(1 - \frac{\lambda\gamma\tau}{2}\right)^{-(1+t)} \geq \frac{1}{\gamma\tau\lambda} e^{T\gamma\tau\lambda} \,,$$

to get the bound

$$\mathbb{E}\left[\overline{F}_\theta^\nu(\overline{w}^{(T)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right] \leq 2\lambda e^{-T\gamma\tau\lambda}\|w^{(0)} - \overline{w}^\star\|^2 + \mathcal{T}_1 \ . \tag{23}$$

It remains to translate the results on $\overline{F}_\theta^\nu$ into F_θ. For the left hand side, we use the bias bound of Property 9. For the right hand side, we use the λ-strong convexity of F_θ and Property 9 we have:

$$\begin{aligned}
\|w^{(0)} - \overline{w}^\star\|^2 &\leq 2\|w^{(0)} - w^\star\|^2 + 2\|\overline{w}^\star - w^\star\|^2 \\
&\leq 2\|w^{(0)} - w^\star\|^2 + \frac{4}{\lambda}\left(F_\theta(\overline{w}^\star) - F_\theta(w^\star)\right) \\
&\leq 2\|w^{(0)} - w^\star\|^2 + \frac{4}{\lambda}\left(F_\theta(\overline{w}^\star) - \overline{F}_\theta^\nu(\overline{w}^\star) + \overline{F}_\theta^\nu(\overline{w}^\star) - \overline{F}_\theta^\nu(w^\star) + \overline{F}_\theta^\nu(w^\star) - F_\theta(w^\star)\right) \\
&\leq 2\|w^{(0)} - w^\star\|^2 + \frac{4}{\lambda}\left(\frac{2B}{\sqrt{\theta m}} + 4\,\nu\log m\right) \quad ,
\end{aligned}$$

since $\overline{F}_\theta^\nu(\overline{w}^\star) - \overline{F}_\theta^\nu(w^\star) \leq 0$. Plugging this into (23) gives us the bound

$$\begin{aligned}
\mathbb{E}\left[F_\theta(\overline{w}^{(T)}) - F_\theta(w^\star)\right] \leq\ & 4\lambda\|w^{(0)} - w^\star\|^2 e^{-T\gamma\tau\lambda} + \frac{32\tau G^2\gamma}{\theta m} + \\
& \frac{18G^2\left(L+\lambda\right)^2\tau^2\gamma^2 e^2}{\lambda}\left(4 + \frac{8}{\theta m}\right) + \left(\frac{2B}{\sqrt{\theta m}} + 2\,\nu\log m\right)\left(1 + 8e^{-T\gamma\tau\lambda}\right) \ .
\end{aligned}$$

Hyperparameter Optimization. To complete the proof from here, it remains to optimize the learning rate γ and the smoothing parameter ν. We invoke Lemma 14 to optimize for γ. Ignoring absolute constants, this gives us the bound

$$\begin{aligned}
\mathbb{E}\left[F_\theta(\overline{w}^{(T)})\right] - F_\theta(w^\star) \leq\ & \lambda\Delta_0\exp(-\lambda\tau\Gamma T) + \frac{G^2}{\theta m\lambda T}\log\left(1 \vee \frac{\lambda^2\Delta_0 T\theta m}{G^2}\right) \\
& + \frac{G^2\kappa^2}{\lambda T^2}(1-\tau^{-1})\log^2\left(1 \vee \frac{\lambda^2\Delta_0 T^2}{G^2\kappa^2}\right) + \frac{B}{\sqrt{\theta m}} + \nu\log m \,,
\end{aligned}$$

where we take

$$\Gamma = \min\left\{\frac{\sqrt{\lambda}}{18\tau(L+\lambda)\sqrt{L'}}, \frac{1}{4\tau L'}, \right\}$$

Next, we set ν. The two terms that depend on ν are

$$\begin{aligned}
\lambda\Delta_0\exp(-\lambda\tau\Gamma T) + \nu\log m &= \lambda\Delta_0\exp\left(-\frac{T}{\left(\kappa + \frac{G^2}{\lambda\nu}\right) \vee \kappa\sqrt{\kappa + \frac{G^2}{\lambda\nu}}}\right) + \nu\log m \\
&\leq \lambda\Delta_0\max\left\{\exp\left(-\frac{\lambda\nu T}{G^2}\right), \exp\left(-\frac{T}{\kappa}\sqrt{\frac{\lambda\nu}{2G^2}}\right)\right\} + \nu\log m \,.
\end{aligned}$$

Assume now that $\nu \leq 2G^2/(\lambda\kappa^2)$, so the first term in the max is active. The conditions of Lemma 14 are met since $T \geq 2\kappa$ is assumed; that gives us the choice

$$\nu = \frac{G^2}{\lambda\kappa} \wedge \frac{2G^2}{\lambda T} \log\left(1 \vee \frac{\lambda^2 \Delta_0 T}{2G^2 \log m}\right),$$

so that the error is bounded by

$$\lambda\Delta_0 \exp\left(-\frac{T}{2\kappa}\right) + \frac{2G^2}{\lambda T} \log(m) \log\left(1 \vee \frac{\lambda^2 \Delta_0 T}{2G^2 \log m}\right).$$

Likewise, if $\nu > 2G^2/(\lambda\kappa^2)$, the second term inside the max is active. The conditions of Lemma 14 are met since $T \geq \sqrt{2\kappa^3}$ is assumed. That gives us the choice

$$\nu \leq \frac{G^2}{\lambda\kappa} \wedge \frac{2G^2\kappa^2}{\lambda T^2} \log^2\left(1 \vee \frac{\lambda^2 \Delta_0 T^2}{2G^2\kappa^2 \log m}\right),$$

so that the error is bounded by

$$\lambda\Delta_0 \exp\left(-\frac{T}{\sqrt{2\kappa^3}}\right) + \frac{2G^2\kappa^2}{\lambda T^2} \log(m) \log^2\left(1 \vee \frac{\lambda^2 \Delta_0 T^2}{2G^2\kappa^2 \log m}\right).$$

Plugging in these choices completes the proof. □

A.3 Intermediate Results

We present some prerequisites and some intermediate results which are required in the convergence proofs of both the convex and nonconvex cases.

Note that for any $S \subset [N]$ of size m, the partial superquantile is differentiable at w with :

$$\nabla F^{\nu}_{\theta,S}(w) = \sum_{k \in S} \pi^{\star}_k \nabla \tilde{F}_k(w) \tag{24}$$

where $\pi^{\star}$ denotes solution to the maximization

$$F^{\nu}_{\theta,S}(w) = \max_{\pi \in \mathcal{P}_{\theta,S}} \sum_{k \in S} \pi_k \tilde{F}_k(w) - \nu\, D_S(\pi)$$

Bias and variance of the partial superquantile. We use the partial superquantile defined on a subset $S \subset [N]$ to approximate the full superquantile. We start with the quality of this approximation.

Property 9. *Let U_m denote the uniform distribution over all subsets of $[N]$ of size m. For any $w \in \mathbb{R}^d$, we have*

$$\left|\overline{F}^{\nu}_{\theta}(w) - F_{\theta}(w)\right| \leq \frac{B}{\sqrt{\theta m}} + 2\,\nu \log m\,,$$
$$\mathbb{E}_{S \sim U_m} \left\|\nabla F^{\nu}_{\theta,S}(w) - \nabla \overline{F}^{\nu}_{\theta}(w)\right\|^2 \leq \frac{8G^2}{\theta m}\,.$$

Smoothing and smoothness constants. The following result is standard [2, Theorem 4.1, Lemma 4.2].

Property 10. *For every $\nu > 0$, we have that $F^{\nu}_{\theta,S}$ and $\overline{F}^{\nu}_{\theta,S}$ are L'-smooth with $L' = L + \lambda + \frac{G^2}{\nu}$.*

Bounding Gradient Dissimilarity. Bounding of the variance of gradient estimators is a key assumption in the analysis of stochastic gradients methods (see e.g. the textbook [7]). In the centralized setting, when a stochastic objective

$\mathbb{E}_\xi[f(w,\xi)]$, it is standard to assume for a given estimator g_w of $\nabla_w \mathbb{E}\left[f(w,\xi\right]$ that there exists some constants $M_1, M_2 > 0$ such that for all $w \in \mathbb{R}^{\mathrm{d}}$,

$$\left\|\mathbb{E}\left[g_w\right]\right\|^2 \leq M_1 \quad \text{or} \quad \left\|\mathbb{E}\left[g_w\right]\right\|^2 \leq M_1 + M_2 \left\|\nabla_w \mathbb{E}\left[f(w,\xi\right]\right\|^2 .$$

In the federated setting, the use of a subset $S \subset [N]$ of clients in each round induces noise on the estimation of the average gradient over the whole network. Thus, such assumption translates into a *bound on the gradient dissimilarity* among the clients [38, 80]:

$$\frac{1}{N}\sum_{k\in[N]} \left\|\nabla \tilde{F}_k(w)\right\|^2 \leq M_1 + M_2 \left\|\frac{1}{N}\sum_{k\in[N]} \nabla \tilde{F}_k(w)\right\|^2 .$$

In this work, we also consider the minimization of the global loss F_θ^ν by a stochastic algorithm based on a partial participation of the clients in the network, with the additional difficulties that we only have access to a biased estimator $\overline{F}_\theta^\nu$ of the loss F_θ^ν and its gradient. In particular, the adaptive reweighting of the clients selected at each round does not permit the direct use of such assumption. We show instead in the next lemma that the variance of stochastic gradient estimator can also be bounded, thanks to the Lipschitz assumption.

Proposition 11 (Gradient Dissimilarity)**.** *Consider the quantities $\pi^{(t)}, w^{(t)}$ from Algorithm 3. We have,*

$$\mathbb{E}\left[\sum_{k\in S} \pi_k^{(t)} \left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2 \middle| \mathcal{F}_t\right] \leq \left(4 + \frac{8}{\theta m}\right) G^2 + \left\|\nabla \overline{F}_\theta^\nu(w^{(t)})\right\|^2 .$$

Proof. We drop the superscript t throughout this proof. By centering the second moment (cf. (27)), we have:

$$\begin{aligned}\sum_{k\in S} \pi_k \left\|\nabla \tilde{F}_j(w)\right\|^2 &= \sum_{k\in S} \pi_k \left\|\nabla \tilde{F}_k(w) - \nabla F_{\theta,S}^\nu(w)\right\|^2 + \left\|\nabla F_{\theta,S}^\nu(w)\right\|^2 \\ &= \sum_{k\in S} \pi_k \left\|(\nabla F_k(w) - \sum_{i\in S} \pi_i \nabla F_i(w))\right\|^2 + \left\|\nabla F_{\theta,S}^\nu(w)\right\|^2 .\end{aligned}$$

Now since the weights π_k sum to one, we may use the convexity of $\|\cdot\|^2$ to get:

$$\sum_{k\in S} \pi_k \left\|\nabla \tilde{F}_j(w)\right\|^2 \leq \sum_{k,i\in S} \pi_k \pi_i \left\|\nabla F_i(w) - \nabla F_k(w)\right\|^2 + \left\|\nabla F_{\theta,S}^\nu(w)\right\|^2 .$$

The squared triangle inequality (cf. (26)) together with the Lipschitz assumption on the functions F_k yields:

$$\begin{aligned}\sum_{k\in S} \pi_k \left\|\nabla \tilde{F}_k(w)\right\|^2 &\leq 2 \sum_{k,i\in S} \pi_k \pi_i \left(\left\|\nabla F_k(w)\right\|^2 + \left\|\nabla F_i(w)\right\|^2\right) \left\|\nabla F_{\theta,S}^\nu(w)\right\|^2 \\ &\leq 4\, G^2 + \left\|\nabla F_{\theta,S}^\nu(w)\right\|^2 .\end{aligned}$$

Thus, taking an expectation over $S \sim U_m$ gives

$$\mathbb{E}\left[\sum_{k\in S} \pi_k \left\|\nabla \tilde{F}_j(w)\right\|^2 \middle| \mathcal{F}_t\right] \leq 4\, G^2 + \mathbb{E}_{S\sim U_m}\left[\left\|\nabla F_{\theta,S}^\nu(w)\right\|^2\right] .$$

By centering (cf. (27)), we get,

$$\mathbb{E}\left[\sum_{k\in S} \pi_k \left\|\nabla \tilde{F}_k(w)\right\|^2 \middle| \mathcal{F}_t\right] \leq 4\, G^2 + \left\|\nabla \overline{F}_\theta^\nu(w)\right\|^2 + \mathbb{E}\left[\left\|\nabla F_{\theta,S}^\nu(w) - \nabla \overline{F}_\theta^\nu(w)\right\|^2 \middle| \mathcal{F}_t\right] . \tag{25}$$

Finally, substituting the variance bound from Lemma 9 into (25) yields the stated result. □

Bounding the Client Drift. During federated learning, each client takes multiple local steps. This causes the resulting update to be a biased estimator of a descent direction for the global objective. This phenomenon has been referred to as "client drift" [53, 38]. Current proof techniques rely on treating this as a "noise" term which is to be controlled. In the context of this work, the reweighting by $\pi^{(t)}$ requires us to adapt this typical definition of client drift to our setting. In particular, recall that we define the client drift $d^{(t)}$ in outer iteration t of the algorithm as

$$d^{(t)} := \mathbb{E}_{S\sim U_m}\left[\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\|w_{k,j}^{(t)}-w^{(t)}\|^2\,\middle|\,\mathcal{F}_t\right].$$

Proposition 12 (Client Drift)**.** *If $\gamma \leq \frac{1}{4\tau(L+\lambda)}$, we have the following bounds for any $t \geq 0$:*

$$d^{(t)} \leq \tau^2(\tau-1)\gamma^2 e^2\left(\left(4+\frac{8}{\theta m}\right)G^2 + \|\nabla\overline{F}_\theta^\nu(w^{(t)})\|^2\right) \text{ and,}$$
$$d^{(t)} \leq \tau^2(\tau-1)\gamma^2 e^2\left(\left(4+\frac{8}{\theta m}\right)G^2 + 2L'\left(\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right)\right).$$

Furthermore, if $\lambda = 0$, we have the bound

$$d^{(t)} \leq 8\tau^2(\tau-1)\gamma^2G^2\,.$$

The last bound also works without smoothing, i.e., $\nu = 0$.

Proof. If $\tau = 1$, there is nothing to prove as both sides of the inequality are 0. We assume now that $\tau > 1$. Let us first fix $S \subset [N]$ of size $|S| = m$. For any $k \in S$ and $j \in \{1,\dots,\tau-1\}$, by the squared triangle inequality (cf. (26)), we have:

$$\begin{aligned}\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2 &= \left\|w_{k,j-1}^{(r)} - \gamma\nabla\tilde{F}_k(w_{k,j-1}^{(t)}) - w^{(t)}\right\|^2\\ &\leq \left(1+\frac{1}{\tau-1}\right)\left\|w_{k,j-1}^{(t)} - w^{(t)}\right\|^2 + \tau\gamma^2\left\|\nabla\tilde{F}_k(w_{k,j-1}^{(t)})\right\|^2.\end{aligned}$$

The squared triangle inequality (cf. (26)) together with the smoothness of the local losses gives:

$$\begin{aligned}\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2 &\leq \left(1+\frac{1}{\tau-1}\right)\left\|w_{k,j-1}^{(t)} - w^{(t)}\right\|^2 + 2\tau\gamma^2\left(\left\|\nabla\tilde{F}_k(w_{k,j-1}^{(t)}) - \nabla\tilde{F}_k(w^{(t)})\right\|^2 + \left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\right)\\ &\leq \left(1+\frac{1}{\tau-1}\right)\left\|w_{k,j-1}^{(t)} - w^{(t)}\right\|^2 + 2\tau\gamma^2(L+\lambda)^2\left\|w_{k,j-1}^{(t)} - w^{(t)}\right\|^2 + 2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2.\end{aligned}$$

Hence, for $\gamma \leq \frac{1}{4\tau(L+\lambda)}$, we get:

$$\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2 \leq \left(1+\frac{2}{\tau-1}\right)\left\|w_{k,j-1}^{(t)} - w^{(t)}\right\|^2 + 2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2.$$

Unrolling this recursion yields for any $j \leq \tau-1$

$$\begin{aligned}\left\|w_{k,j}^{(t)} - w^{(t)}\right\|^2 &\leq \sum_{i=0}^{j-1}\left(1+\frac{2}{\tau-1}\right)^i\left(2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\right)\\ &\leq \frac{\tau-1}{2}\left(1+\frac{2}{\tau-1}\right)^j\left(2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\right)\\ &\leq \frac{\tau-1}{2}\left(1+\frac{2}{\tau-1}\right)^{\tau-1}\left(2\tau\gamma^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2\right)\\ &\leq \tau(\tau-1)\gamma^2e^2\left\|\nabla\tilde{F}_k(w^{(t)})\right\|^2,\end{aligned}$$

where we use $(1+1/x)^x \leq e$ for any $x > 0$. If $\lambda = 0$ we have that $\left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2 = \left\|\nabla F_k(w^{(t)})\right\|^2 \leq G^2$ since F_k is G-Lipschitz; this gives us the final bound in the statement. When $\lambda \neq 0$, this does not hold. In this case, we apply Lemma 11 to get

$$\begin{aligned} d^{(t)} &\leq \tau^2(\tau-1)\gamma^2 e^2\, \mathbb{E}_{S\sim U_m}\left[\sum_{k\in S}\pi_k^{(t)}\left\|\nabla \tilde{F}_k(w^{(t)})\right\|^2 \,\middle|\, \mathcal{F}^{(t)}\right] \\ &\leq \tau^2(\tau-1)\gamma^2 e^2\left(\left(4+\frac{8}{\theta m}\right)G^2 + \left\|\nabla \overline{F}_\theta^\nu(w^{(t)})\right\|^2\right). \end{aligned}$$

This gives the first bound. The second bound follows from the first by smoothness (cf. (29)). □

Bound on the Norm of Each Update. We bound the expected squared norm of each update $w^{(t+1)} - w^{(t)}$, which has the closed form expression:

$$w^{(t+1)} - w^{(t)} = -\gamma \sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla \tilde{F}_k(w_{k,j}^{(t)}).$$

Proposition 13. *We have the bounds,*

$$\begin{aligned} &\gamma^2\mathbb{E}\left[\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla \tilde{F}_k(w_{k,j}^{(t)})\right\|^2 \,\middle|\, \mathcal{F}_t\right] \\ &\leq 2\gamma^2\tau\,(L+\lambda)^2\; d^{(t)} + \frac{16\tau^2\gamma^2G^2}{\theta m} + 2\tau^2\gamma^2\left\|\nabla \overline{F}_\theta^\nu(w^{(t)})\right\|^2 \\ &\leq 2\gamma^2\tau\,(L+\lambda)^2\; d^{(t)} + \frac{16\tau^2\gamma^2G^2}{\theta m} + 4\tau^2\gamma^2L'\left(\overline{F}_\theta^\nu(w^{(t)}) - \overline{F}_\theta^\nu(\overline{w}^\star)\right), \end{aligned}$$

where $d^{(t)}$ is the client drift term defined in (21).

Proof. Using the the squared triangle inequality (cf. Eq. (26)) together with the gradient formula (24), we get:

$$\begin{aligned} &\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla \tilde{F}_k(w_{k,j}^{(t)})\right\|^2 \\ &\leq 2\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\left(\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)})\right)\right\|^2 + 2\left\|\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\nabla \tilde{F}_k(w^{(t)})\right\|^2 \\ &\leq 2\tau\sum_{k\in S}\pi_k^{(t)}\sum_{j=0}^{\tau-1}\left\|\nabla \tilde{F}_k(w_{k,j}^{(t)}) - \nabla \tilde{F}_k(w^{(t)})\right\|^2 + 2\tau^2\left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2. \end{aligned}$$

For the first term, we invoke $(L+\lambda)$-smoothness of $\tilde{F}_k$ and take an expectation to get $2\tau(L+\lambda)^2 d^{(t)}$. For the second term, we use centering (cf. Eq. (27)) followed by the variance bound of Lemma 9 to get:

$$\begin{aligned} \mathbb{E}_t\left[\left\|\nabla F_{\theta,S}^\nu(w^{(t)})\right\|^2\right] &= \mathbb{E}_t\left[\left\|\sum_{k\in S}\pi_k^{(t)}\nabla \tilde{F}_k(w^{(t)}) - \nabla \overline{F}_\theta^\nu(w^{(t)})\right\|^2\right] + \left\|\nabla \overline{F}_\theta^\nu(w^{(t)})\right\|^2 \\ &\leq \frac{8G^2}{\theta m} + \left\|\nabla \overline{F}_\theta^\nu(w^{(t)})\right\|^2. \end{aligned}$$

This gives the first bound. The second bound follows from the first by smoothness (cf. Eq. (29)). □

A.4 Useful Inequalities and Technical Results

We recall a few standard inequalities:

- Squared Triangle inequality: For any $x, y \in \mathbb{R}^d$ and $\alpha > 0$ we have:

$$\|x+y\|^2 \leq (1+\alpha)\|x\|^2 + \left(1+\frac{1}{\alpha}\right)\|y\|^2 . \tag{26}$$

- Centering the second moment: For any $\mathbb{R}^d$-valued random vector X such that $\mathbb{E}\|X\|^2 < \infty$,

$$\mathbb{E}\|\mathrm{X}\|^2 = \mathbb{E}\|\mathrm{X} - \mathbb{E}\,[\mathrm{X}]\|^2 + \|\mathbb{E}\,[\mathrm{X}]\|^2 \tag{27}$$

- Strong convexity: Let $F : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ be μ-strongly convex. Then for any $x, y \in \mathbb{R}^{\mathrm{d}}$, we have:

$$\langle \nabla F(x), x-y\rangle \geq F(x) - F(y) + \frac{\mu}{2}\|x-y\|^2 \tag{28}$$

- Smoothness: Let $F : \mathbb{R}^{\mathrm{d}} \to \mathbb{R}$ be L-smooth and let $F^\star$ be the minimum value of F (assuming it exists). Then for any $x \in \mathbb{R}^{\mathrm{d}}$, we have:

$$\|\nabla F(x)\|^2 \leq 2L\,(F(x) - F^\star) \tag{29}$$

Lemma 14. *Consider the map $\varphi : (0, \Gamma] \to \mathbb{R}_+$ given by*

$$\varphi(\gamma) = A\exp(-\gamma T) + B\gamma + C\gamma^2 ,$$

where $\Gamma, A, B, C > 0$ are given. If $T > 1/\Gamma$, then, we have,

$$\varphi(\gamma^\star) \leq A\exp(-\Gamma T) + \frac{B}{T} + \frac{B}{T}\log\left(1 \vee \frac{AT}{B}\right) + \frac{C}{T^2} + \frac{C}{T^2}\log^2\left(1 \vee \frac{AT^2}{C}\right) ,$$

where $\gamma^\star$ is given by

$$\gamma^\star = \min\left\{\Gamma, \frac{1}{T}\log\left(1 \vee \frac{AT}{B}\right), \frac{1}{T}\log\left(1 \vee \frac{AT^2}{C}\right)\right\} .$$

Proof. Define $\gamma_1 = T^{-1}\log(1 \vee AT/B)$ and $\gamma_2 = T^{-1}\log(1 \vee AT^2/C)$. If $\gamma^\star = \Gamma$, we have that $\Gamma \leq \gamma_1$ and $\Gamma \leq \gamma_2$ so that

$$\varphi(\gamma^\star) = A\exp(-\Gamma T) + B\Gamma + C\Gamma^2 \leq A\exp(-\Gamma T) + B\gamma_1 + C\gamma_2^2 .$$

Now suppose that $\gamma^\star = \gamma_1$ so that $\gamma_1 \leq \gamma_2$. Then, we have,

$$\varphi(\gamma^\star) = A\exp(-\gamma_1 T) + B\gamma_1 + C\gamma_1^2 \leq \frac{B}{T} + \frac{B}{T}\log(1 \vee AT/B) + C\gamma_2^2 .$$

The third case is identical to the second. □

The proof of the next lemma is elementary and is omitted.

Lemma 15. *Consider the map $\varphi : (0, \Gamma] \to \mathbb{R}_+$ given by*

$$\varphi(\gamma) = \frac{A}{\gamma T} + B\gamma + C\gamma^2 ,$$

where $\Gamma, A, B, C > 0$ are given. Then, we have,

$$\varphi(\gamma^\star) \leq \frac{A}{\Gamma T} + 2\left(\frac{AB}{T}\right)^{1/2} + 2\left(\frac{AC}{T}\right)^{2/3} ,$$

where $\gamma^\star$ is given by

$$\gamma^\star = \min\left\{\Gamma, \sqrt{\frac{A}{BT}}, C^{1/3}\left(\frac{A}{T}\right)^{1/3}\right\} .$$

B Privacy Analysis

B.1 Preliminaries

The discrete Gaussian mechanism was introduced in [11] as an extension of the Gaussian mechanism to integer data. A random variable ξ is said to satisfy the discrete Gaussian distribution with mean μ and variance proxy σ^2 if

$$\mathbb{P}(\xi = i) = C \, \exp\left(-\frac{(i-\mu)^2}{2\sigma^2}\right) \quad \text{for all } i \in \mathbb{Z}\,,$$

where C is an appropriate normalizing constant. We denote it by $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$. We need the following property of the discrete Gaussian.

Property 16. *Let ξ be distributed according to $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$. Then, $\mathbb{E}[\xi] = \mu$. Furthermore, if $\mu = 0$, then ξ is sub-Gaussian with variance proxy σ^2, i.e., $\mathbb{E}[\exp(\lambda\xi)] \le \exp(\lambda^2\sigma^2/2)$ for all $\lambda > 0$.*

B.2 Proof of Privacy and Utility of Quantile Computation

Proof of Theorem 5. We start by defining and controlling the probabilities of some events. Throughout, let $\delta > 0$ be fixed. Define the event

$$E_{\text{mod}} = \bigcap_{k=1}^{N} \bigcap_{j=1}^{n} \left\{ -\frac{M-2}{2N} \le \gamma x_{k,j} + \xi_{k,j} \le \frac{M-2}{2N} \right\}. \tag{30}$$

Note that under E_{mod}, no modular wraparound occurs in the algorithm, i.e., $\tilde{x}_k = \gamma x_k + \xi_k$ and

$$\hat{h} = \sum_{k=1}^{N} \frac{\tilde{x}_k}{\gamma} = \sum_{k=1}^{N} \left(x_k + \frac{\xi_k}{\gamma} \right).$$

We assume that E_{mod} holds throughout.

Privacy Analysis. We start by establishing the sensitivity of the sum query over x_k's as 1. Define the input space $\mathcal{X}$ to be the canonical basis vectors in $\mathbb{R}^n$, i.e., the set of all vectors in $\{0,1\}^n$ with only one 1, and let $\mathcal{X}^* = \cup_{N=1}^{\infty}$ denote the set of all sequences of elements of $\mathcal{X}$. We consider the rescaled sum query $A((x_1, \cdots, x_N)) = \sum_{k=1}^{N} \gamma x_N$. The L_2 sensitivity $S(A)$ of this query A is supremum over all $X \in \mathcal{X}^*$ and X' which is obtained by concatenating x' to X:

$$S(A) = \sup_{X, X'} \|A(X) - A(X')\|_2 = \sup_{x' \in \mathcal{X}} \gamma \, \|x'\|_2 = \gamma\,.$$

We invoke the privacy bound of sums of discrete Gaussians (Lemma 19) to claim that an algorithm $\mathcal{A}$ returning $A(x) + \sum_{k=1}^{N} \xi_k$ satisfies $(1/2)\varepsilon^2$-concentrated DP where ε is as in the theorem statement. The fact that the quantile and all further functions of it remains private follows from the post-processing property of DP (also known as the data-processing inequality).

Utility Analysis. Define $\hat{N} = \sum_{j=1}^{n} \hat{h}_j$, as the analogue to $N = \sum_{j=1}^{n} h_j$. Below, we use shorthand $\rho = 1 - \theta$. We bound the quantile error as

$$\begin{aligned}
\Delta_\theta\left(\hat{h}, h\right) = R_\theta\left(h, j_\theta^*(\hat{h})\right) &= \left| \frac{1}{N} \sum_{j=1}^{j_\theta^*(\hat{h})} h_j - \rho \right| \\
&\le \frac{1}{N} \left| \sum_{j=1}^{j_\theta^*(\hat{h})} h_j - \hat{h}_j \right| + \frac{1}{N} \left| \sum_{j=1}^{j_\theta^*(\hat{h})} \hat{h}_j - \hat{N}\rho \right| + \frac{\rho}{n} \left| \hat{N} - N \right| \\
&\le \max_{j' \in [n]} \frac{1}{\gamma N} \left| \sum_{j=1}^{j'} \sum_{k=1}^{N} \xi_{k,j} \right| + \left(1 + \frac{|\hat{N} - N|}{N} \right) R_\theta^*(\hat{h}) + \frac{\rho}{N} \left| \hat{N} - N \right|.
\end{aligned}$$

Let us define an event E_{sum} under which the first term and last terms are bounded:

$$E_{\text{sum}} = \left\{ \max_{j \in [n]} \left| \sum_{j'=1}^{j} \sum_{k=1}^{N} \xi_{k,j'} \right| \leq \sqrt{2\sigma^2 Nn \log(4/\delta)} \right\}. \tag{31}$$

Under E_{sum}, we also have

$$|N - \hat{N}| = \frac{1}{\gamma} \left| \sum_{j=1}^{n} \sum_{k=1}^{N} \xi_{k,j} \right| \leq \sqrt{\frac{2\sigma^2 Nn}{\gamma} \log \frac{4}{\delta}}.$$

Plugging this back into $\Delta_\rho(\hat{h}, h)$ gives us the desired bound, provided E_{sum} holds.

Bounding the Failure Probability. The algorithm fails when at least one of E_{mod} or E_{sum} fail to hold. We have from Claim 17 that $\mathbb{P}(E_{\text{mod}}) \geq 1 - \delta/2$ and from Claim 18 that $\mathbb{P}(E_{\text{sum}}) \geq 1 - \delta/2$. With a union bound, we get that $\mathbb{P}(E_{\text{sum}} \cap E_{\text{prod}}) \geq 1 - \delta$, i.e., the algorithm succeeds with probability at least $1 - \delta$. □

We state and prove bounds on probabilities of the events $E_{\text{mod}}, E_{\text{sum}}$ defined above.

Claim 17. *If $M \geq 2 + 2\gamma N + 2N\sqrt{2\sigma^2 \log(4Nn/\delta)}$, then $\mathbb{P}(E_{\text{mod}}) \geq 1 - \delta/2$.*

Proof. Each discrete Gaussian random variable $\xi_{k,j}$ is centered and sub-Gaussian with variance proxy σ^2 (cf. Property 16). A Cramér-Chernoff bound (cf. Lemma 20) gives us the exponential tail bound

$$\mathbb{P}\left(|\xi_{k,j}| > \sqrt{2\sigma^2 \log(4Nn/\delta)} \right) \leq \frac{\delta}{2Nn}.$$

Using a union bound for $k \in [N], j \in [n]$ and $x_{k,j} \in \{0, 1\}$ completes the proof. □

Claim 18. *We have that $\mathbb{P}(E_{\text{sum}}) \geq 1 - \delta/2$.*

Proof. Each discrete Gaussian random variable $\xi_{k,j}$ is centered and sub-Gaussian with variance proxy σ^2, i.e., $\mathbb{E}[\xi_{k,j}] = 0$ and $\mathbb{E}[\exp(\lambda \xi_{k,j})] \leq \exp(\lambda^2 \sigma^2/2)$ for all $\lambda \in \mathbb{R}$ (cf. Property 16). Therefore, $\zeta_j := \sum_{k=1}^{N} \xi_{k,j}$ is centered and sub-Gaussian with variance proxy $n\sigma^2$, since $\mathbb{E}[\zeta_j] = 0$, and

$$\mathbb{E}[\exp(\lambda \zeta_j)] = \prod_{k=1}^{N} \mathbb{E}[\exp(\lambda \xi_{k,j})] \leq \exp(\lambda^2 \sigma^2 N/2)$$

by independence. We get a bound on the partial sums from Lemma 21; this involves constructing a martingale $(\sum_{j'=1}^{j} \zeta_{j'})_{j=1}^{n}$ and applying the maximal inequality. The bound we get is

$$\mathbb{P}\left(\max_{j \in [n]} \left| \sum_{j'=1}^{j} \zeta_{j'} \right| > t \right) \leq \exp\left(-\frac{t^2}{2\sigma^2 Nn} \right).$$

Plugging in $t = \sqrt{2\sigma^2 Nn \log(2/\delta)}$ completes the proof. □

B.3 Useful Results

The distributed discrete Gaussian mechanism gets privacy guarantees by adding a sum of discrete Gaussian random variables. We give a bound in its privacy. The following lemma is due to [36].

Lemma 19 (Privacy of Sum of Discrete Gaussians)**.** *Fix $\sigma \geq 1/2$. Let M be a deterministic algorithm with ℓ_2-sensitivity S. Define a randomized algorithm $\mathcal{A}$, which when given an input x, samples $\xi_1, \cdots, \xi_n \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2 I_d)$ and returns $A(x) + \sum_{i=1}^{n} \xi_i$. Then, $\mathcal{A}$ satisfies $\varepsilon^2/2$-concentrated DP with*

$$\varepsilon = \min\left\{ \sqrt{\frac{S}{n\sigma^2} + \frac{\psi d}{2}}, \frac{S}{\sqrt{n}\sigma} + \psi\sqrt{d} \right\},$$

where $\psi = 10 \sum_{k=1}^{n-1} \exp\big(-2\pi^2\sigma^2 k/(k+1)\big) \leq 10(n-1)\exp(-2\pi^2\sigma^2)$.

Next, we record two standard concentration results.

Lemma 20 (Cramér-Chernoff)**.** *Let ξ be a real-valued and centered sub-Gaussian random variable with variance proxy σ^2, i.e., $\mathbb{E}[\xi] = 0$ and $\mathbb{E}[\exp(\lambda\xi)] \leq \exp(\lambda^2\sigma^2/2)$ for all $\lambda > 0$. Then, we have for any $t > 0$,*

$$\mathbb{P}(|\xi| > t) \leq 2\exp\left(-\frac{t^2}{2\sigma^2}\right).$$

Lemma 21 (Maximal Inequality)**.** *Let $\xi_1, \xi_2, \cdots$ be i.i.d. centered sub-Gaussian random variables with variance proxy σ^2, i.e., $\mathbb{E}[\xi_j] = 0$ and $\mathbb{E}[\exp(\lambda\xi_j)] \leq \exp(\lambda^2\sigma^2/2)$ for all $\lambda \in \mathbb{R}$ and $j = 1, 2, \cdots$. Then, it holds for any $t > 0$ and integer $n \geq 1$ that*

$$\mathbb{P}\left(\max_{k\in[n]}\left|\sum_{j=1}^{k}\xi_t\right| > t\right) \leq 2\exp\left(-\frac{t^2}{2\sigma^2 n}\right).$$

C Numerical Experiments: Complete Results

We conduct our experiments on two datasets from computer vision and natural language processing. These datasets contain a natural, non-iid split of data which is reflective of data heterogeneity encountered in federated learning. In this section, we describe in details the experimental setup and the results. Here is its outline:

- Appendix C.1 describes the datasets and tasks.
- Appendix C.2 presents the algorithm and the hyperparameters used.
- Appendix C.3 details the evaluation methodology.
- Appendix C.4 gives the experimental comparison of Δ-FL to baselines.

Since each client has a finite number of datapoints in the examples below, we let its probability distribution π_k to be the empirical distribution over the available examples, and the weight α_k to be proportional to the number of datapoints available on the client.

C.1 Datasets and Tasks

We use the two following datasets, described in detail below. The data was preprocessed using LEAF [10].

EMNIST for handwritten-letter recognition.

Dataset. EMNIST [15] is a character recognition dataset. This dataset contains images of handwritten digits or letters, labeled with their identification (a-z,A-Z, 0-9). The images are grey-scaled pictures of $28 \times 28 = 784$ pixels.

Train and Test Devices. Each image is also annotated with the "writer" of the image, i.e., the human subject who hand-wrote the digit/letter during the data collection process. Each client corresponds to one writer. From this set of clients, we discard all clients containing less than 100 images. The remaining clients were partitioned into two groups — 1730 training and 1730 testing clients. For each experiment we subsampled 865 training and 865 testing clients for computational tractability, where the sampled clients vary based on the random seed of each experiment.

Model. We consider the following models for this task.

- **Linear Model**: We use a linear softmax regression model. In this case each F_k is convex. We train parameters $w \in \mathbb{R}^{62\times 784}$. Given an input image $x \in \mathbb{R}^{784}$, the score of each class $c \in [62]$ is the dot product $\langle w_c, x\rangle$. The probability p_c assigned to each class is then computed as a softmax: $p_c = \exp\langle w_c, x\rangle / \sum_{c'} \exp\langle w_{c'}, x\rangle$. The prediction for a given image is then the class with the highest probability.

- **ConvNet**: We also consider a convolutional neural network with two convolutional layers with max-pooling and one fully connected layer (F.C) of which outputs a vector in $\mathbb{R}^{62}$. The outputs of the ConvNet are scores with respect to each class. They are also used with a softmax operation to compute probabilities.

The loss used to train both models is the multinomial logistic loss $L(p, y) = -\log p_y$ where p denotes the vector of probabilities computed by the model and p_y denotes its y^{th} component. In the convex case we add a quadratic regularization term of the form $(\lambda/2)\|w\|_2^2$.

Sent140 for Sentiment Analysis.

Dataset. Sent140 [29] is a text dataset of 1,600,498 tweets produced by 660,120 Twitter accounts. Each tweet is represented by a character string with emojis redacted. Each tweet is labeled with a binary sentiment reaction (i.e., positive or negative), which is inferred based on the emojis in the original tweet.

Train and Test Devices. Each client represents a twitter account and contains only tweets published by this account. From this set of clients we discarded all clients containing less that 50 tweets, and split the 877 remaining clients rest of clients into a train set and a test set of sizes 438 and 439 respectively. This split was held fixed for all experiments. Each word in the tweet is encoded by its 50-dimensional GloVe embedding [62].

Model. We consider the following models.

- **Linear Model**: We consider a l_2-regularized linear logistic regression model where the parameter vector w is of dimension 50. In this case, each F_k is convex. We summarize each tweet by the average of the GloVe embeddings of the words of the tweet.
- **RNN**: The nonconvex model is a Long Short Term Memory (LSTM) model [33] built on the GloVe embeddings of the words of the tweet. The hidden dimension of the LSTM is same as the embedding dimension, i.e., 50. We refer to it as "RNN".

The loss function is the binary logistic loss.

C.2 Algorithms and Hyperparameters

Algorithm and Baselines.
The proposed Δ-FL is run for three values of $\theta \in \{0.8, 0.5, 0.1\}$. We compare it to the following baselines:

- FedAvg [54]: It is the de facto standard for the vanilla federated learning objective.
- FedAvg, θ: We also consider FedAvg with a random client subselection step: local updates are run on a fraction of the initial number of clients randomly selected per round. For each dataset, we try three values of, corresponding to the average number of clients selected by Δ-FL for the three values of θ used. In the main paper, we report as FedAvg-Sub the performance of FedAvg, θ with $\theta \in \{0.8, 0.5, 0.1\}$ which gives the best performance on Δ-FL (i.e., lowest 90^{th} percentile of test misclassification error). Here we report numbers for all values of θ considered.
- FedProx [50]: It augments FedAvg with a proximal term but still minimizes the vanilla federated learning objective.
- q-FFL [51]: It raises the per-client losses to the power $(1 + q)$, where $q \geq 0$ is a parameter, in order to focus on clients with higher loss. We run q-FFL for values of q in $\{10^j, j \in \{-3, \cdots, 1\}\}$.
- AFL [57]: It aims to minimize the worst per-client loss. We implement it as an asymptotic version of q-FFL, using a large value of q, as this was found to yield better convergence with comparable performance [51]. In the experiments we take $q = 10.0$.

The experiments are conducted on the datasets described in Appendix C.1.

Hyperparameters.

Rounds. We measure the progress of each algorithm by the number of calls to secure aggregation routine for weight vectors, i.e., the number of communication rounds.

For the experiments, we choose the number of communication rounds depending on the convergence of the optimization for FedAvg. For the EMNIST dataset, we run the algorithm for 3000 communication rounds with the linear model and 1000 for the ConvNet. For the Sent140 dataset, we run the 1000 communication rounds for the linear model and 600 for the RNN.

Devices per Round. We choose the same number of clients per round for each method, with the exception of $FedAvg, \theta$. All clients are assumed to be available and selections are made uniformly at random. In particular, we select 100 clients per round for all experiments with the exception of Sent140 RNN for which we used 50 clients per round.

Local Updates and Minibatch Size. Each selected client locally runs 1 epoch of mini-batch stochastic gradient descent locally. We used the default mini-batch of 10 for all experiments [54], except for 16 for EMNIST ConvNet. This is because the latter experiments were run using on a GPU, as we describe in the section on the hardware.

Learning rate scheme. We now describe the learning rate γ_t used during *LocalUpdate*. For the linear model we used a constant fixed learning rate $\gamma_t \equiv \gamma_0$, while for the neural network models, we using a step decay scheme of the learning rate $\gamma_t = \gamma_0 c^{-\lfloor t/t_0 \rfloor}$ for some where γ_0 and $0 < c \leq 1$ are tuned. We tuned the learning rates only for the baseline FedAvg and used the same learning rate for the other baselines and Δ-FL at all values of θ.

For the neural network models, we fixed t_0 so that the learning rate was decayed once or twice during the fixed time horizon T. In particular, we used $t_0 = 400$ for EMNIST ConvNet (where $T = 1000$) and $t_0 = 200$ for Sent140 RNN (where $T = 600$). We tuned c from the set $\{2^{-3}, 2^{-2}, 2^{-1}, 1\}$, while the choice of the range of γ_0 depended on the dataset-model pair. The tuning criterion we used was the mean of the loss distribution over the training clients (with client k weighted by α_k) at the end of the time horizon. That is, we chose the γ_0, c which gave the best terminal training loss.

Tuning of the regularization parameter. The regularization parameter λ for linear models was tuned with cross validation from the set $\{10^{-k} : k \in \{3, \cdots, 8\}\}$. This was performed as described below.

For each dataset, we held out half the training clients as validation clients. Then, for different values of the regularization parameter, we trained a model with the (smaller subset of) training clients and evaluate its performance on the validation clients. We selected the value of the regularization parameter as the one which gave the smallest 90^{th} percentile of the misclassification error on the validation clients.

Baselines Parameters. We tune the proximal parameter of FedProx with cross validation. The procedure we followed is identical to the procedure we described above for the regularization parameter λ. The set of parameters tested is $\{10^{-j}, j \in \{0, \ldots, 3\}\}$. We did not attempt to tune the parameter q of q-FFL and report the performance of all values of q which we tried.

Hyperparameters of Δ-FL. We optimize Δ-FL via Algorithm 3 with a fixed number of local steps, corresponding to one epoch. For simplicity, we calculate the quantile exactly, assuming client losses are available to the server.

C.3 Evaluation Strategy and Other Details

Evaluation metrics. We record the loss of each training client and the misclassification error of each testing client, as measured on its local data.

The evaluation metrics noted in Section C.4 are the following : the weighted mean of the loss distribution over the training clients, the (unweighted) mean misclassification error over the testing clients, the weighted τ-percentile of the loss over the training client and the (unweighted) τ-percentile of the misclassification error over the testing clients for values of τ among $\{20, 50, 60, 80, 90, 95\}$. We also present the 90^{th} and 95^{th} superquantile of the test misclassification error (i.e., average misclassification error of the worst 10% and 5% of the clients respectively), as well as the average test misclassification error of the best 10% clients. The weight α_k used for training client k was set proportional to the

Table 4: Metrics for the test misclassification error for EMNIST (Linear Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	34.38 ± 0.38	18.39 ± 0.33	21.54 ± 0.35	32.61 ± 0.39	49.65 ± 0.67
FedAvg $\theta = 0.8$	34.20 ± 0.45	18.25 ± 0.22	$\mathbf{21.37} \pm 0.26$	32.10 ± 0.34	49.92 ± 1.16
FedAvg $\theta = 0.5$	34.51 ± 0.47	18.21 ± 0.30	21.40 ± 0.36	32.36 ± 0.59	50.28 ± 0.77
FedAvg $\theta = 0.1$	34.60 ± 0.46	18.58 ± 0.31	21.71 ± 0.37	32.54 ± 0.37	50.33 ± 1.28
FedProx	$\mathbf{33.82} \pm 0.30$	18.25 ± 0.23	21.37 ± 0.35	$\mathbf{31.75} \pm 0.20$	49.15 ± 0.74
q-FFL (Best $q = 1.0$)	34.71 ± 0.27	19.34 ± 0.30	22.33 ± 0.41	32.80 ± 0.23	49.90 ± 0.58
Tilted-ERM (Best $t = 1.0$)	34.15 ± 0.25	10.78 ± 0.30	22.43 ± 0.29	32.36 ± 0.23	48.59 ± 0.62
AFL	39.32 ± 0.27	25.42 ± 0.27	28.64 ± 0.43	38.16 ± 0.34	51.62 ± 0.28
Δ-FL $\theta = 0.8$	34.48 ± 0.26	19.16 ± 0.32	22.24 ± 0.32	32.85 ± 0.31	49.10 ± 0.24
Δ-FL $\theta = 0.5$	35.01 ± 0.20	20.46 ± 0.34	23.64 ± 0.22	33.83 ± 0.34	$\mathbf{48.44} \pm 0.38$
Δ-FL $\theta = 0.1$	38.32 ± 0.48	23.86 ± 0.59	27.27 ± 0.64	37.52 ± 0.67	50.34 ± 0.95

Table 5: Metrics for the test misclassification error for EMNIST (ConvNet Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	16.63 ± 0.50	$\mathbf{4.94} \pm 0.14$	$\mathbf{6.43} \pm 0.24$	15.34 ± 0.37	28.46 ± 1.07
FedAvg $\theta = 0.8$	15.95 ± 0.42	5.25 ± 0.19	6.86 ± 0.38	14.84 ± 0.24	26.82 ± 1.28
FedAvg $\theta = 0.5$	16.22 ± 0.23	5.06 ± 0.17	6.47 ± 0.28	15.05 ± 0.25	27.56 ± 0.81
FedAvg $\theta = 0.1$	15.97 ± 0.43	5.40 ± 0.42	7.10 ± 0.64	14.76 ± 0.20	26.35 ± 2.08
FedProx	16.01 ± 0.54	5.16 ± 0.32	6.68 ± 0.44	14.88 ± 0.29	27.01 ± 1.86
q-FFL (Best $q = 0.001$)	16.58 ± 0.30	5.05 ± 0.21	6.53 ± 0.20	15.40 ± 0.43	28.02 ± 0.80
Tilted-ERM (Best $t = 1.0$)	15.69 ± 0.38	7.31 ± 0.68	7.26 ± 0.51	$\mathbf{14.66} \pm 0.16$	25.46 ± 1.49
AFL	33.00 ± 0.37	20.38 ± 0.23	22.92 ± 0.23	31.58 ± 0.27	45.07 ± 1.00
Δ-FL $\theta = 0.8$	16.08 ± 0.40	5.60 ± 0.14	7.31 ± 0.29	14.85 ± 0.48	26.23 ± 1.15
Δ-FL $\theta = 0.5$	$\mathbf{15.48} \pm 0.30$	6.13 ± 0.15	8.08 ± 0.16	14.73 ± 0.22	$\mathbf{23.69 \pm 0.94}$
Δ-FL $\theta = 0.1$	16.37 ± 1.03	6.61 ± 0.42	8.28 ± 0.65	15.49 ± 1.03	25.45 ± 2.77

number of datapoints on the client.

Evaluation times. We evaluate the model during training process for once every l communication rounds. The value of l used was $l = 50$ for EMNIST linear model, $l = 10$ for EMNIST ConvNet, $l = 20$ for Sent140 linear model and $l = 25$ for Sent140 RNN.

Hardware. We run each experiment as a simulation as a single process. The linear models were trained on m5.8xlarge AWS instances, each with an Intel Xeon Platinum 8000 series processor with 128 GB of memory running at most 3.1 GHz. The neural network experiments were trained on workstation with an Intel i9 processor with 128 GB of memory at 1.2 GHz, and two Nvidia Titan Xp GPUs. The Sent140 RNN experiments were run on a CPU while the other neural network experiments were run using GPUs.

Software Packages. Our implementation is based on NumPy using the Python language. In the neural network experiments, we use PyTorch to implement the *LocalUpdate* procedure, i.e., the model itself and the automatic differentiation routines provided by PyTorch to make SGD updates.

Randomness. Since several sampling routines appear in the procedures such as the selection of clients or the local stochastic gradient, we carry our experiments with five different seeds and plot the average metric value over these seeds. Each simulation is run on a single process. Where appropriate, we report one standard deviation from the mean.

Table 6: Metrics for the test misclassification error for Sent140 (Linear Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	34.74 ± 0.31	12.16 ± 0.15	21.89 ± 0.24	34.81 ± 0.38	46.83 ± 0.54
FedAvg $\theta = 0.8$	34.47 ± 0.03	12.08 ± 0.16	21.69 ± 0.26	34.62 ± 0.17	46.59 ± 0.38
FedAvg $\theta = 0.5$	34.46 ± 0.07	12.11 ± 0.24	$\mathbf{21.55} \pm 0.51$	$\mathbf{34.48} \pm 0.20$	47.00 ± 0.40
FedAvg $\theta = 0.1$	34.79 ± 0.32	11.97 ± 0.37	22.08 ± 0.75	34.93 ± 0.35	46.69 ± 0.84
FedProx	34.74 ± 0.31	12.16 ± 0.15	21.89 ± 0.24	34.82 ± 0.39	46.83 ± 0.54
q-FFL (Best $q = 1.0$)	34.48 ± 0.06	11.96 ± 0.14	21.61 ± 0.24	34.57 ± 0.16	$\mathbf{46.38} \pm 0.40$
Tilted-ERM (Best $t = 1.0$)	34.71 ± 0.31	12.00 ± 0.14	21.83 ± 0.34	34.91 ± 0.39	46.70 ± 0.50
AFL	35.97 ± 0.08	11.83 ± 0.09	23.58 ± 0.28	36.09 ± 0.17	47.51 ± 0.32
Δ-FL $\theta = 0.8$	$\mathbf{34.41} \pm 0.22$	12.17 ± 0.11	21.77 ± 0.34	34.64 ± 0.25	46.44 ± 0.38
Δ-FL $\theta = 0.5$	35.28 ± 0.25	$\mathbf{11.68} \pm 0.40$	23.03 ± 0.38	35.55 ± 0.53	46.64 ± 0.41
Δ-FL $\theta = 0.1$	37.78 ± 0.89	12.86 ± 0.52	23.93 ± 0.99	37.80 ± 1.30	51.38 ± 1.07

Table 7: Metrics for the test misclassification error for Sent140 (RNN Model).

Method	Mean	Standard Deviation	10^{th} Percentile	Median	90^{th} Percentile
FedAvg	30.16 ± 0.44	4.36 ± 1.26	10.06 ± 2.06	29.51 ± 0.33	49.66 ± 3.95 1
FedAvg $\theta = 0.8$	$\mathbf{29.85} \pm 0.46$	5.39 ± 1.32	11.90 ± 2.27	29.57 ± 0.31	46.93 ± 3.84 1
FedAvg $\theta = 0.5$	31.06 ± 1.01	$\mathbf{4.33} \pm 2.73$	$\mathbf{9.69} \pm 4.89$	30.14 ± 0.71	53.10 ± 7.22 1
FedAvg $\theta = 0.1$	31.96 ± 1.47	4.82 ± 2.09	11.65 ± 4.83	31.55 ± 1.13	52.87 ± 8.41 1
FedProx	30.20 ± 0.48	4.35 ± 1.23	10.37 ± 2.08	$\mathbf{29.51} \pm 0.32$	49.85 ± 4.07
q-FFL (Best $q = 0.01$)	29.99 ± 0.56	4.90 ± 1.66	10.98 ± 2.88	29.56 ± 0.39	48.65 ± 4.68
Tilted-ERM (Best $t = 1.0$)	30.13 ± 0.49	14.17 ± 2.10	13.18 ± 3.33	29.96 ± 0.84	46.54 ± 3.27
AFL	37.74 ± 0.65	9.90 ± 1.46	18.19 ± 1.99	36.95 ± 1.03	57.78 ± 1.19
Δ-FL $\theta = 0.8$	30.30 ± 0.33	6.75 ± 2.68	13.05 ± 3.87	29.92 ± 0.38	$\mathbf{46.46} \pm 4.39$
Δ-FL $\theta = 0.5$	33.58 ± 2.44	8.74 ± 3.98	16.77 ± 6.62	33.28 ± 2.27	50.47 ± 8.24
Δ-FL $\theta = 0.1$	51.97 ± 11.81	9.11 ± 5.47	16.67 ± 9.15	52.44 ± 13.21	86.44 ± 10.95

C.4 Experimental Results

We now present the experimental results of the paper.

- We present different metrics on the distribution of test misclassification error over the clients, comparing Δ-FL to baselines.
- We study the convergence of Algorithm 3 for Δ-FL over the course of the optimization, and compare it with FedAvg.
- We plot the histograms of the distribution of losses over train clients as well as the test misclassification errors over test clients at the end of the training process.
- We present in the form of scatter plots the training loss and test misclassification error across clients achieved at the end of training, versus the number of local data points on the client.
- We present the number of clients having a loss greater than the quantile at each communication round for Δ-FL. This gives the effective number of clients selected in each round, cf. Proposition 2 and Remark 1.

Comparison to Baselines. We now present a detailed comparison of various statistics of the test misclassification error distribution for different methods in Table 4- For each column the smallest mean over five random runs is highlighted in

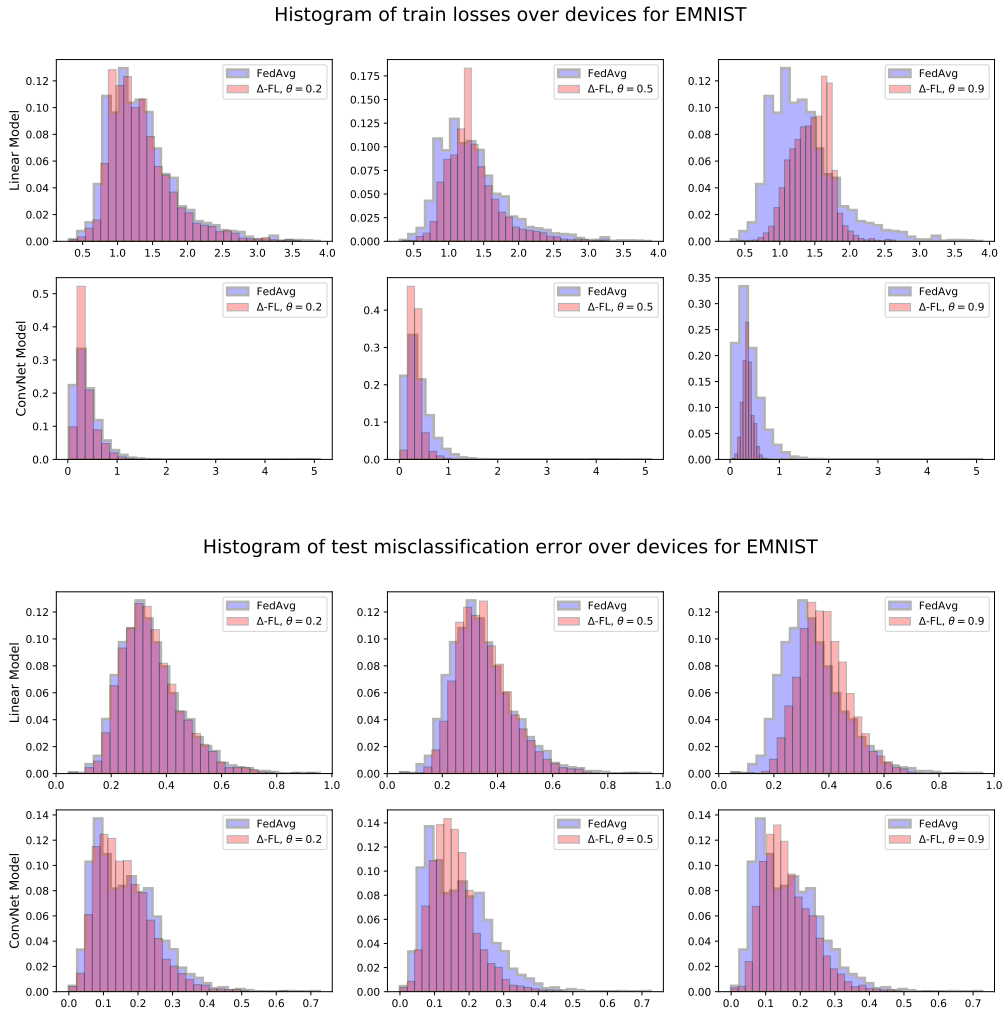

Figure 5: Histogram of loss distribution over training clients and misclassification error distribution over testing clients for EMNIST. The identification of the model (linear or ConvNet) is given on the y-axis of the histograms.

bold. Further, if no other method is within one standard deviation of this method, the entire entry (i.e., mean $\pm$ std) is highlighted in bold.

Histograms of Loss and Test Misc. Error over Devices. Here, we plot the histograms of the loss distribution over training clients and the misclassification error distribution over testing clients. We report the losses and errors obtained at the end of the training process. Each metric is averaged per client over 5 runs of the random seed. Figure 5 shows the histograms for EMNIST, while Figure 6 shows the histograms for Sent140 dataset. for Sent140. We note that Δ-FL tends to exhibit thinner upper tails at at multiple values of θ and a lower variance of the distribution in most of the cases. This is also confirmed by the figures in table 4 to 7. This shows the benefit of using Δ-FL over vanilla FedAvg.

Performance compared to local data size. Next, we plot the loss on training clients versus the amount of local data on the client and the misclassification error on the test clients versus the amount of local data on the client. See Figure 7 for EMNIST and Figure 8 for Sent140.

Observe firstly that improvement over the worst cases is achieved regardless of the local data size of the clients. Indeed, the client re-weighting step operates a sorting of the loss of the clients which does not prevent small clients

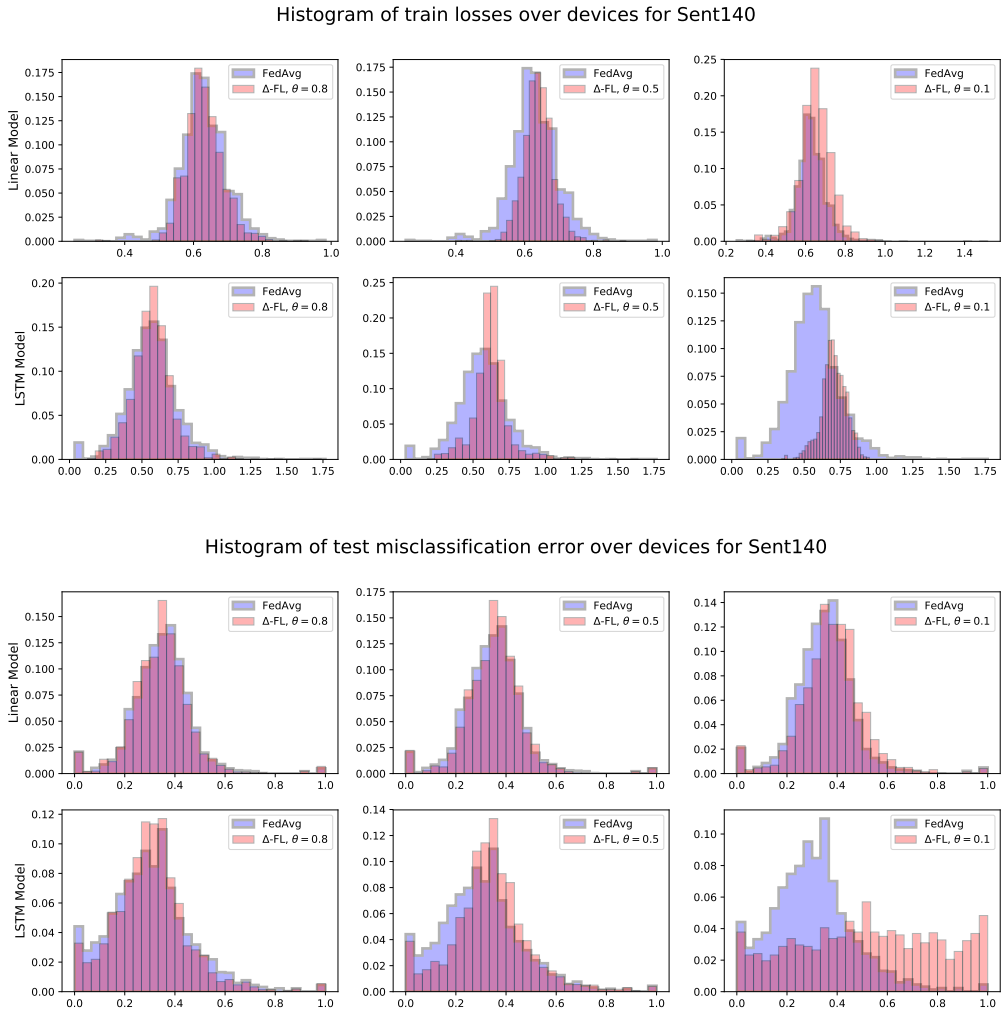

Figure 6: Histogram of loss distribution over training clients and misclassification error distribution over testing clients for Sent140. The identification of the model (linear or RNN) is given on the y-axis of the histograms.

from being selected. In contrary, FedAvg, by averaging with respect to the weights of the clients is likely to put more the accent on the clients with larger local data size. Secondly, Δ-FL appears to reduce the variance of of the loss on the train clients. Lastly, note that amongst test clients with a small number of data points (e.g., < 200 for EMNIST or < 100 for Sent140), Δ-FL reduces the variance of the misclassification error. Both effects are more pronounced on the neural network models.

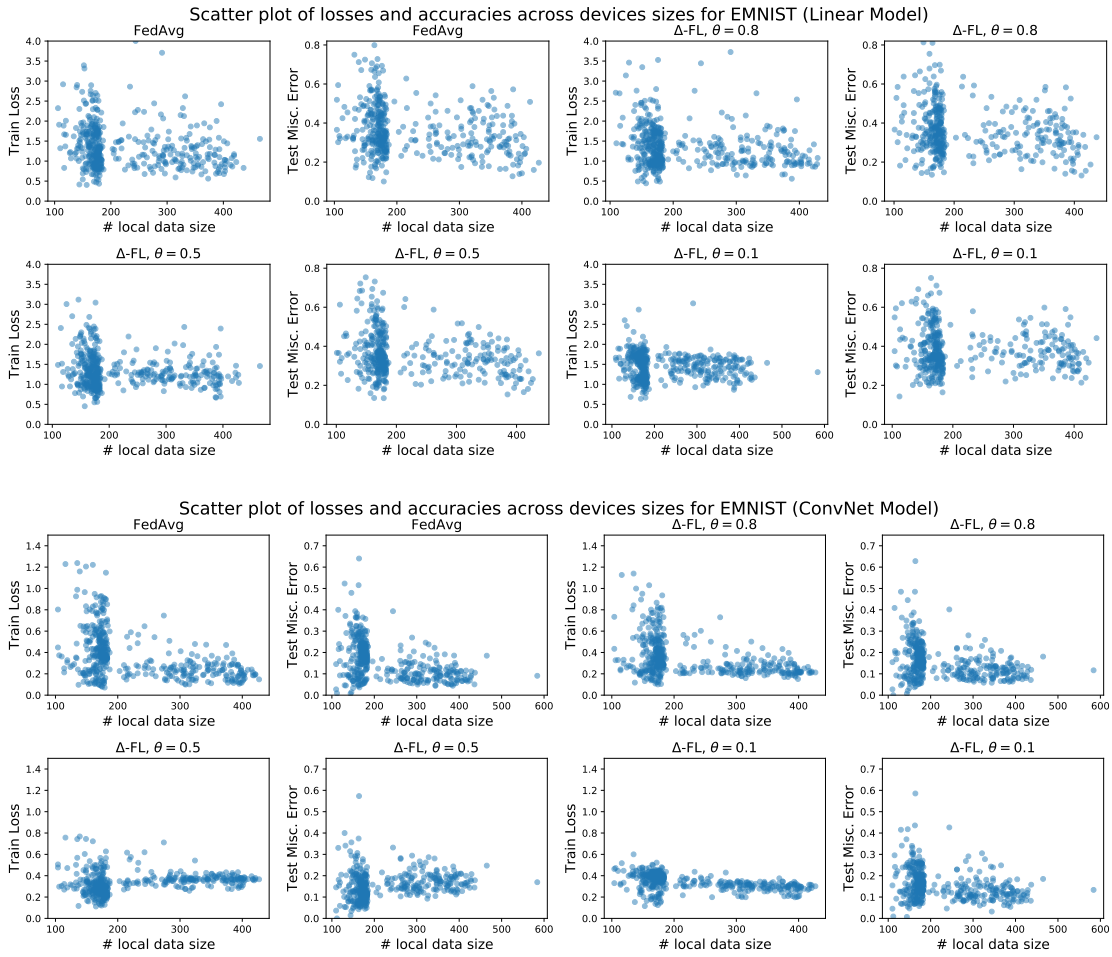

Figure 7: Scatter plot of (a) loss on training client vs. amount of local data, and (b) misclassification error on testing client vs. amount of local data for EMNIST.

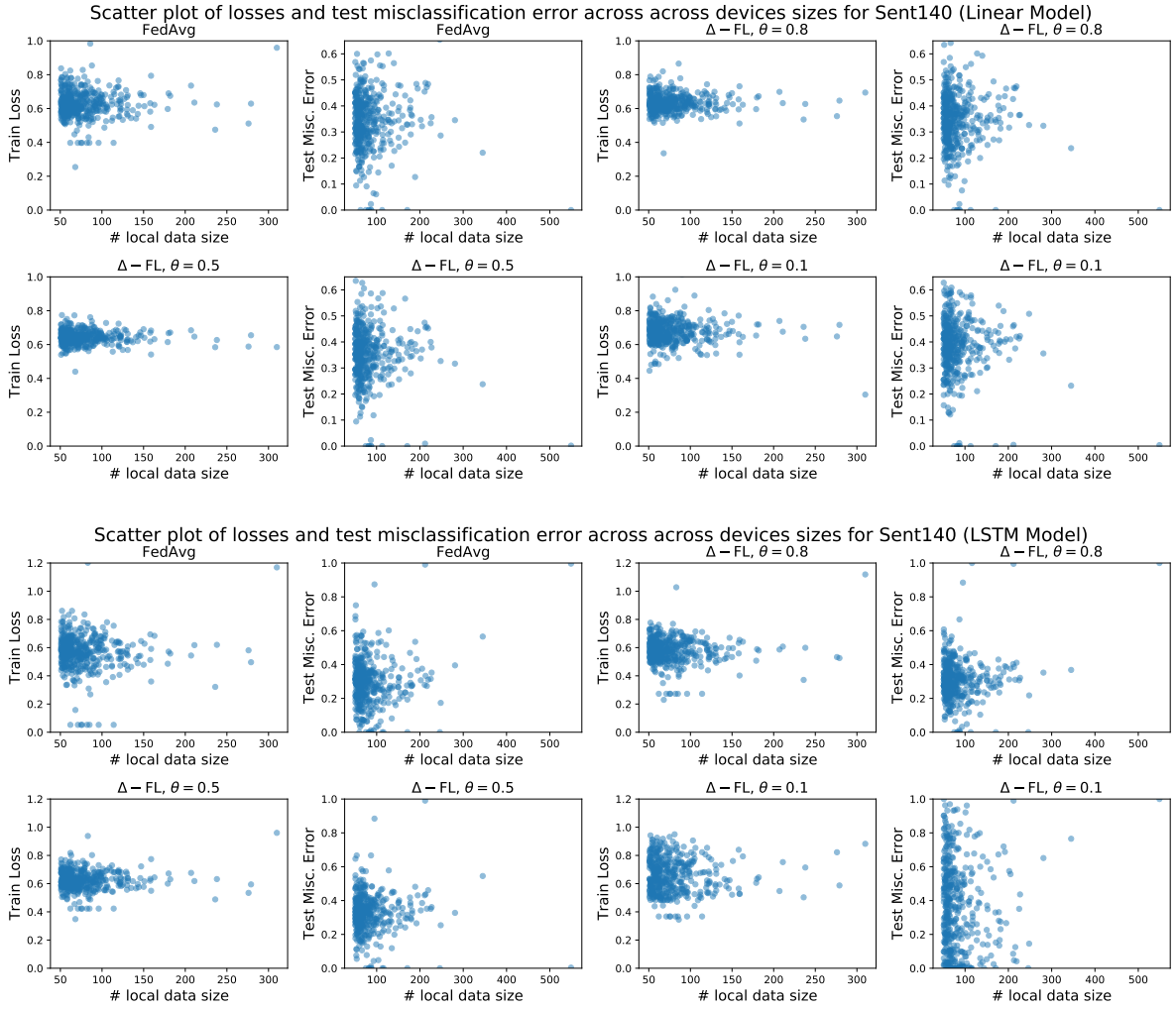

Figure 8: Scatter plot of (a) loss on training client vs. amount of local data, and (b) misclassification error on testing client vs. amount of local data for Sent140.