A Superquantile Approach to Federated Learning with Heterogeneous Devices

Yassine Laguel
Université Grenoble Alpes
Grenoble, France
Email: yassine.laguel@univ-grenoble-alpes.fr

Krishna Pillutla
University of Washington
Seattle, USA

Jerôme Malick
CNRS
Grenoble, France

Zaid Harchaoui
University of Washington
Seattle, USA

***Abstract*—We present a federated learning framework that allows one to handle heterogeneous client devices that may not conform to the population data distribution. The proposed approach hinges upon a parameterized superquantile-based objective, where the parameter ranges over levels of conformity. We introduce a stochastic optimization algorithm compatible with secure aggregation, which interleaves device filtering steps with federated averaging steps. We conclude with numerical experiments with neural networks on computer vision and natural language processing data.**

I. Introduction

In federated learning [1, 2], a number of client devices with privacy-sensitive data collaboratively learn a machine learning model under the orchestration of a central server, while keeping their data decentralized. This is achieved by pushing the computation to the devices while the server coordinates with the devices for aggregation of model updates. Federated learning has found myriad applications such as smartphone apps [3, 4] and healthcare [5].

A key feature of federated learning is statistical heterogeneity, i.e., client data distributions are *not* identical. Each user has unique characteristics which are reflected in the data they generate. These characteristics are influenced by personal, cultural, and geographical factors. For instance, the varied use of language contributes to data heterogeneity in a next word prediction task.

Vanilla federated learning [1], aims to minimize the prediction loss of a given model on average over a population of devices available for training. While this approach works for users with local data distribution close to the average distribution, it is liable to fail on individuals who do not conform to the population, leading to poor user experience. The goal of this work is to present a framework to improve the experience of these diversely non-conforming users without sacrificing the good experience of conforming users.

In this paper, we introduce the Δ-*FL* framework, summarized in Fig. 1, to handle heterogeneity of client data distributions. The framework relies on a superquantile-based objective parameterized by the conformity level, which is a scalar summary of how closely a device conforms to the population. To optimize the Δ-*FL* objective, we present an algorithm which interleaves device filtering with federated optimization steps. We discuss its special features compared to the standard *FedAvg* and to recent algorithms [6]–[8] about heterogeneity in federated learning. We analyze our algorithm in the convex setting and establish bounds on total communication cost.

We demonstrate the breadth of our framework with numerical experiments using convolutional and recurrent neural networks on tasks including image classification, and sentiment analysis based on public datasets. The simulations demonstrate superior performance of Δ-*FL* over state-of-the-art baselines on the upper quantiles of the error on test devices, while being competitive on the mean error.

Outline. Section II describes the setting and precisely defines conformity. Section III discusses the Δ-*FL* framework, the training objective, and the related approaches. Section IV gives an optimization algorithm for the Δ-*FL* objective and analyzes its convergence. Section V presents numerical experiments of the proposed method. Full proofs and additional details can be found in [9]. The code and the scripts to reproduce numerical results are publicly available at https://github.com/krishnap25/simplicial-fl.

II. Problem Setting

We introduce the notion of conformity in federated learning to measure how conforming are test devices from the population of train devices.

Train Devices and Trend Distribution. Consider N clients devices with respective probability distribution q_k and weights $\alpha_k > 0$. We assume $\sum_{k=1}^N \alpha_k = 1$ and that the data on device k are distributed i.i.d. according to q_k. The loss of a device with distribution q is $F(w; q) := \mathbb{E}_{\xi\sim q}[f(w; \xi)]$, where $f(w; \xi)$ is the loss on the input-output pair ξ under model $w \in \mathbb{R}^d$. The loss on k^{th} training device is then $F_k(w) := F(w; q_k)$.

The standard federated learning objective is the weighted average of the local losses of the training devices:

$$\min_{w\in\mathbb{R}^d} \sum_{k=1}^N \alpha_k F_k(w) + \frac{\lambda}{2}\|w\|_2^2\,, \tag{1}$$

with a regularization parameter λ. We define the *trend distribution* as $p_\alpha = \sum_{i=1}^n \alpha_k q_k$, the intrinsic distribution on which standard federated learning models are trained. Indeed, (1) exactly minimizes $F(w; p_\alpha) + (\lambda/2)\|w\|_2^2$. Our approach will consider test distribution that depart from the trend distribution.

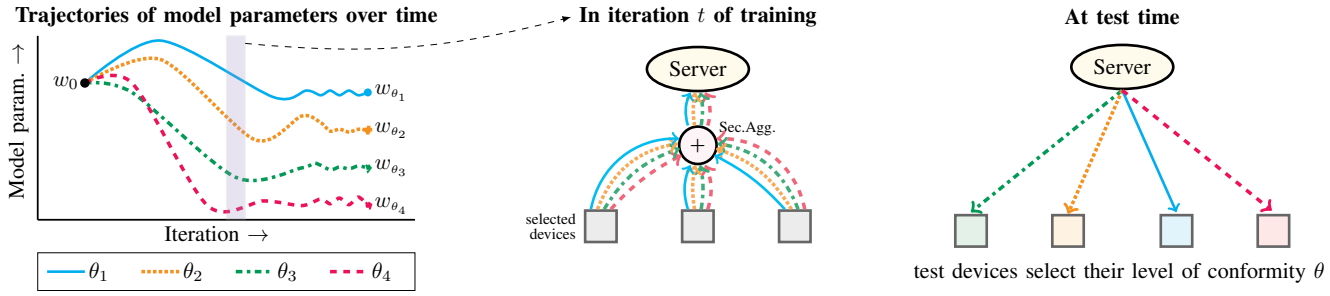

Fig. 1. Schematic summary of the Δ-*FL* framework. **Left**: The server maintains multiple models w_{θ_j}, one for each level of conformity θ_j. **Middle**: During training, selected devices participate in training *each* model w_{θ_j}. Individual updates are securely aggregated to update the server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

Test Devices and Conformity. We consider "test" devices, unseen during training, whose distribution can be written as a mixture of the training distributions. A mixture p_π with weight $\pi \in \Delta^{N-1}$ is $p_\pi := \sum_{k=1}^N \pi_k q_k$. Here, Δ^{N-1} is the probability simplex in $\mathbb{R}^N$. We now define the conformity of mixture with respect to the trend distribution.

Definition 1. The *conformity* $\mathrm{conf}(p_\pi) \in (0,1]$ of a mixture p_π with weight π is defined as $\min_{k\in[N]} \alpha_k/\pi_k$. The conformity of a test device refers to the conformity of its data distribution.

Given a large representative set of training devices, it is reasonable to assume that each test device can be well-approximated by a mixture p_π. Then, the conformity of a device is *a scalar summary of how close it is to the population*. A test device with conformity $\theta \approx 1$ has its local distribution close to the trend distribution. Then, a model trained with the distribution p_α is expected to have a high predictive power for this test device. In contrast, a test device with $\theta \approx 0$ would be vastly different from the distribution p_α, and the predictive power of a model trained on p_α could be poor.

There is a trade-off between fitting to the population and supporting non-conforming test devices. 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 $\mathrm{conf}(p_\pi) \geq \theta$.

III. The Δ-*FL* Framework

The Δ-*FL* framework supplies each test device 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, as laid down in Fig. 1. The local data is not allowed to leave a device due to privacy restrictions; hence, the conformity of a test device cannot be measured. Instead, we allow each test device to tune their conformity.

Superquantiles come into play. To train a model with a given conformity, we aim to do well on *all* mixtures p_π with $\mathrm{conf}(p_\pi) \geq \theta$. This leads us to the minimax objective

$$\min_{w\in\mathbb{R}^d} \left[F_\theta(w) := \max_{\pi\in\mathcal{P}_\theta} F(w; p_\pi) + \frac{\lambda}{2}\|w\|_2^2 \right]$$
$$\text{where} \quad \mathcal{P}_\theta := \left\{ \pi \in \Delta^{N-1} \,:\, \mathrm{conf}(p_\pi) \geq \theta \right\}.$$

Using duality [10] we can show that F_θ can be written as a minimum: $F_\theta(w) = \min_{\eta\in\mathbb{R}} G_\theta(w,\eta)$, where

$$G_\theta(w,\eta) := \eta + \frac{1}{\theta}\sum_{k=1}^N \alpha_k \max\left\{F_k(w) - \eta, 0\right\} + \frac{\lambda}{2}\|w\|_2^2. \quad (2)$$

This reveals that F_θ is a $(1-\theta)$-superquantile; see *e.g.* the overview [11] and the drawing of Fig. 2.

Δ-*FL* Objective. We now introduce the Δ-*FL* objective as the dual to the minimax objective, i.e.,

$$\min_{w\in\mathbb{R}^d, \eta\in\mathbb{R}} G_\theta(w,\eta).$$

In Algorithm 1, we propose to solve this problem by alternating updates of w and η. From standard results on superquantiles [11], the optimal solution of partial minimization in η admits a closed form as the $(1-\theta)$-quantile of the distribution of losses $(F_k(w))_{1\leq k\leq N}$ with weights $(\alpha_k)_{1\leq k\leq N}$. In contrast, the w-step is executed by a standard federated optimization algorithm such as *FedAvg* until a stopping criterion is met. Precisely, given a target suboptimality $\varepsilon_t > 0$, we require that

$$\mathbb{E}\left[G(w_{t+1}, \eta_t) \,|\, w_t\right] - \min_w G(w, \eta_t) \leq \varepsilon_t. \quad (3)$$

Note that existing works about the use of superquantiles always consider centralized settings and often use convex optimization approaches, such as interior point algorithms [11]. The alternating minimization is suitable for our setting, by using a federated optimization algorithm for the w-step.

Putting the method into perspective. Let us discuss further some aspects of the Δ-*FL* framework and Algorithm 1.

(a) *Device Filtering.* Algorithm 1 can be viewed as interleaving device filtering (where devices with $F_k(w) < \eta$ are filtered out) with federated optimization steps. If a device is filtered out, it means that the model fits well the data on that device.

(b) *Computation Cost.* The computation cost for each communication round of the w-step of Algorithm 1 is the same as that of the base federated optimization algorithm used. The η-step requires a pass over the local data on each device.

(c) *Communication Cost.* The total communication cost is dominated by the communication of model parameters. The theoretical bound on communication rounds presented in Section IV exhibits the same dependence on target

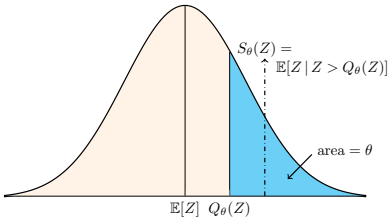

Fig. 2. For a continuous random variable Z, drawing of $(1-\theta)$-quantile $Q_\theta(Z)$ and $(1-\theta)$-superquantile $S_\theta(Z)$, defined as an expectation. In this work, we heavily rely on the variational expression of the superquantile as the minimum of G_θ over η in (2); see [11].

accuracy ε as the base federated optimization algorithm, e.g., *FedAvg*. We also corroborate this observation with numerical experiments in Section V.

(d) *Tuning Conformity*. If using a single global value of conformity for all test users, one should account for all implications of this choice. This depends on the distributions of users. Committing blindly to a single conformity level could fail to balance supporting non-conforming users with fitting the population. On the other hand, measuring the conformity of users requires transfer of user data, a violation of privacy. As such, decisions impacting fairness and privacy are questions of policy rather than of mathematics. Both Δ-*FL* and [8] circumvent this issue by training a family of models for different conformity levels, and allows a test user to tune their conformity.

Related Work on Heterogeneity. Past works have aimed to tackle heterogeneity in federated learning by modifying the objective. *AFL* [6] uses a minimax objective that is effective only with coarse groups of devices; this requires domain knowledge. We note that Δ-*FL* interpolates between *AFL* ($\theta \to 0$) and standard federated learning ($\theta \to 1$).

Like Δ-*FL*, the method q-*FFL* [8] also interpolates between these two extremes but in an different way: it raises losses to the power $1+q$, thus penalizing large losses more, while Δ-*FL* minimizes the average of the largest losses (see Fig. 2). q-*FFL* also maintains multiple global models for different parameters q. However, q does not admit a natural interpretation and its range could be unbounded. In contrast, Δ-*FL*'s conformity parameter $\theta \in (0,1)$ is the conformity level, which has a clear meaning. We provide an experimental comparison of these methods with respect to the client heterogeneity in Section V.

Other works minimize the standard federated learning objective, but aim to improve convergence in the presence of heterogeneity by correcting for potential client drift. Examples include *FedProx* [7] and *Scaffold* [12]. Such algorithms could thus be combined with our framework to reduce the cost of each w-step but that goes beyond the scope of this paper.

IV. Convergence Analysis

We now analyze Algorithm 1 in the convex setting. We assume at each F_k is convex (which is the case for instance for linear models and convex losses). This yields that G_θ is convex, but it is still not smooth, due to the $\max\{\cdot, 0\}$ term. So, we propose to use a smooth surrogate $G_{\theta,\nu}$ and the analogous $F_{\theta,\nu}$ defined for $\nu > 0$ as

$$G_{\theta,\nu}(w,\eta) := \eta + \frac{1}{\theta}\sum_{k=1}^{N} \alpha_k h_\nu\big(F_k(w) - \eta\big) + \frac{\lambda}{2}\|w\|^2,$$
$$F_{\theta,\nu}(w) := \min_{\eta\in\mathbb{R}} G_{\theta,\nu}(w,\eta),$$

where $h_\nu(\rho) = \min_u \big\{\max\{u,0\} + (u-\rho)^2/(2\nu)\big\}$ is the so-called Moreau envelope of $\max\{\cdot\,,0\}$, It provides a smooth and tractable uniform approximation [13]. Our working function is convex and smooth, as formalized in the next lemma.

Algorithm 1 Alternating Minimization for Δ-*FL*

Input: Function $G : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$, $w_0 \in \mathbb{R}^d$, inexactness sequence (ε_t), time horizon t^*

1: **for** $t = 0, 1, \ldots, t^* - 1$ **do**
2: $\quad \eta_t \in \arg\min_{\eta\in\mathbb{R}} G(w_t, \eta)$
3: $\quad w_{t+1} \approx \arg\min_{w\in\mathbb{R}^d} G(w, \eta_t)$ such that (3) holds with ε_t
4: **return** w_{t^*}

Lemma 2. *If F_k is convex, B-Lipshitz and L-smooth, then $G_{\theta,\nu}$ is jointly convex in (w,η) over $\mathbb{R}^d \times \mathbb{R}$ and we have that $w \mapsto \nabla_w G_{\theta,\nu}(w,\eta)$ is L_w-Lipschitz for all $\eta \in \mathbb{R}$ where $L_w := \lambda + (L + B^2/\nu)/\theta$, and, $\eta \mapsto \frac{\partial}{\partial\eta} G_{\theta,\nu}(w,\eta)$ is L_η-Lipschitz for all $w \in \mathbb{R}^d$ where $L_\eta := (\nu\theta)^{-1}$.*

In the next proposition, we use this lemma to establish an upper bound of the outer-loop complexity of Algorithm 1.

Proposition 3. *Under the assumptions of Lemma 2, denote the condition number of $G_{\theta,\nu}(\cdot,\eta)$ by $\kappa = 1 + (L + B^2/\nu)/(\theta\lambda)$. Consider Algorithm 1 with inputs $G_{\theta,\nu}$ and stopping criterion (3) with $\varepsilon_t = \varepsilon_0\, \mathrm{e}^{-t/\kappa}$ for some $\varepsilon_0 > 0$. Then,*

$$\mathbb{E}\left[F_{\theta,\nu}(w_t)\right] - F^\star_{\theta,\nu} \le t\,\mathrm{e}^{-t/\kappa}\left(F_{\theta,\nu}(w_0) - F^\star_{\theta,\nu} + 2\varepsilon_0\right),$$

where appears the minimum value $F^\star_{\theta,\nu} = \min F_{\theta,\nu}$.

Proof. For each iteration t, denote $\psi_t(w) := G_{\theta,\nu}(w,\eta_t)$, and let $\widetilde{w}_t = w_t - \nabla\psi_t(w_t)/L_w$. Lemma 2 ensures that ψ_t is L_w-smooth and λ-strongly convex which in turn yields [14]:

$$\psi_t(w_t) - \psi_t(\widetilde{w}_t) \ge \frac{1}{2L_w}\|\nabla\psi_t(w_t)\|^2 \text{ (smoothness)}$$
$$\psi_t(w_t) - \min\psi_t \le \frac{1}{2\lambda}\|\nabla\psi_t(w_t)\|^2 \text{ (strong convexity)}.$$

Let $\mathcal{F}_t$ denote the σ-algebra generated by w_t. Putting these together with the stopping criterion (3), we get

$$\begin{aligned}\mathbb{E}[\psi_t(w_{t+1})|\mathcal{F}_t] &\le \min\psi_t + \varepsilon_t \;\le\; \psi_t(\widetilde{w}_t) + \varepsilon_t \\ &\le \psi_t(w_t) - \kappa^{-1}\big(\psi_t(w_t) - \min\psi_t\big) + \varepsilon_t\,.\end{aligned}$$

Using now the three following facts $F_{\theta,\nu}(w_{t+1}) \le \psi_t(w_{t+1})$, $F_{\theta,\nu}(w_t) = \psi_t(w_t)$, and $\min_w F_{\theta,\nu}(w) \le \min\psi_t$, we get,

$$\begin{aligned}&\mathbb{E}\left[F_{\theta,\nu}(w_{t+1})|\mathcal{F}_t\right] \\ &\quad\le F_{\theta,\nu}(w_t) - \kappa^{-1}\big(F_{\theta,\nu}(w_t) - \min F_{\theta,\nu}\big) + \varepsilon_t\,.\end{aligned}$$

Rearranging this and taking an expectation over $\mathcal{F}_t$ yields

$$\begin{aligned}&\mathbb{E}\left[F_{\theta,\nu}(w_{t+1})\right] - \min F_{\theta,\nu}\\ &\leq (1-\kappa^{-1})\big(\mathbb{E}\left[F_{\theta,\nu}(w_t)\right] - \min F_{\theta,\nu}\big) + \varepsilon_t\\ &\leq \exp(-\kappa^{-1})\big(\mathbb{E}\left[F_{\theta,\nu}(w_t)\right] - \min F_{\theta,\nu}\big) + \varepsilon_t\,.\end{aligned}$$

Using the shorthand $\Delta_t := \mathbb{E}\left[F_{\theta,\nu}(w_t)\right] - \min F_{\theta,\nu}$ and unrolling the above inequality gives

$$\begin{aligned}\Delta_{t+1} &\leq \exp\left(-\frac{t+1}{\kappa}\right)\Delta_0 + \sum_{\tau=0}^{t}\exp\left(-\frac{t-\tau}{\kappa}\right)\varepsilon_\tau\\ &\leq t\exp\left(-\frac{t}{\kappa}\right)(\Delta_0 + 2\varepsilon_0)\,,\end{aligned}$$

which completes the proof. □

Next, we bound the *total* communication rounds (where a round refers to a secure aggregation of model updates) required by Algorithm 1, taking into account the cost of solving each w-step. We state the result with the w-step solved by *FedAvg* with full device participation (a.k.a. local SGD). The proof uses a result on communication rounds of *FedAvg* from [15].

Proposition 4. *Consider the setting of Proposition 3 with $\alpha_k \equiv 1/N$. Consider using local SGD with τ local steps per communication round to solve w-step of Algorithm 1. Suppose that stochastic gradients (w.r.t. w) of $h_\nu(F_k(w)-\eta)$ have a bounded variance σ_k^2, with $\sigma^2 = \sum_k \alpha_k\sigma_k^2$. Let D bound the diversity of F_k; see [15, Assumption 3.b] for the definition. Then the total rounds T of communication to obtain a point $\hat{w}$ such that $\mathbb{E}[F_{\theta,\nu}(\hat{w})] - F^\star_{\theta,\nu} \leq \varepsilon$ is at most*

$$O\left(\frac{\sigma^2\kappa^2\Delta_0}{N\lambda\tau\varepsilon} + \sqrt{\frac{\sigma^2\kappa^3\Delta_0}{\lambda^2\tau\varepsilon}} + \sqrt{\frac{D^2\kappa^4\Delta_0}{\lambda\varepsilon}} + \kappa^2\right),$$

where $\Delta_0 = (F_{\theta,\nu}(w_0) - F^\star_{\theta,\nu})/\varepsilon_0 + 1$, and big $O(\cdot)$ includes constants and polylog factors.

Proof. For simplicity, C denotes some universal constant which may change from one line to the next. Let $\Delta_0' := \Delta_0 + 2\varepsilon_0 = F_{\theta,\nu}(w_0) - F^\star_{\theta,\nu} + 2\varepsilon_0$. We use the fact that for $\varepsilon \leq 2\Delta_0'/3$,

$$t \geq \kappa\log\left(\frac{2\kappa\Delta_0'}{\varepsilon}\right) + \kappa\log\log\left(\frac{2\kappa\Delta_0'}{\varepsilon}\right)$$

implies $\varepsilon \geq t\exp\left(-t/\kappa\right)\Delta_0'$. Thus, using Proposition 3, the number of outer iterations t^* to get $\mathbb{E}\left[F_{\theta,\nu}(w_{t^*})\right] - F^\star_{\theta,\nu} \leq \varepsilon$ is

$$t^* = \kappa\log\left(\frac{2\kappa\Delta_0'}{\varepsilon}\right) + \kappa\log\log\left(\frac{2\kappa\Delta_0'}{\varepsilon}\right).$$

From [15, Theorem 2], the number n_t of communication rounds to obtain w_{t+1} satisfying the stopping criterion (3) is

$$n_t \leq C\left(\frac{\|\alpha\|_\infty\sigma^2}{\lambda\tau\varepsilon_t} + \sqrt{\frac{\sigma^2}{\lambda^2\tau\varepsilon_t}} + \sqrt{\frac{\kappa D^2}{\lambda\varepsilon_t}} + \kappa\log\left(\frac{\kappa\tau\Delta_t}{\varepsilon_t}\right)\right)$$

with $\Delta_t = F_{\theta,\nu}(w_t) - F^\star_{\theta,\nu} \geq G_{\theta,\nu}(w_t,\eta_t) - \min G_{\theta,\nu}(\cdot,\eta_t)$. The total number of communication rounds is then

$$\begin{aligned}T = \sum_{t=0}^{t^*-1} n_t \leq C\Bigg(&\frac{\sigma^2}{N\lambda\tau\varepsilon_0}\sum_{t=0}^{t^*-1}\exp(t/\kappa)\\ &+\left(\sqrt{\frac{\sigma^2}{\lambda^2\tau\varepsilon_0}} + \sqrt{\frac{\kappa D^2}{\lambda\varepsilon_0}}\right)\sum_{t=0}^{t^*-1}\exp\left(t/(2\kappa)\right)\\ &+\kappa t^*\log\left(\frac{\kappa\tau\Delta_{\max}}{\varepsilon_0}\right) + \sum_{t=0}^{t^*-1} t\Bigg).\end{aligned}$$

For the first term, we use $\exp(\kappa^{-1}) \geq 1+\kappa^{-1}$ to get,

$$\sum_{t=0}^{t^*-1}\exp(t/\kappa) \leq \frac{2\kappa^2\Delta_0'}{\varepsilon}\log\left(\frac{2\kappa\Delta_0'}{\varepsilon}\right).$$

We similarly treat the second term to get

$$\sum_{t=0}^{t^*-1}\exp(t/(2\kappa)) \leq C\sqrt{\frac{\kappa^3}{\varepsilon}\Delta_0'\log\left(\frac{2\kappa\Delta_0'}{\varepsilon}\right)}.$$

The last two terms can be bounded by simply plugging in t^*. Putting these terms together completes the proof. □

Setting $\nu = \theta\varepsilon/2$ we get that T is bounded by

$$O\left(\frac{1}{N\tau\theta^2(\lambda\varepsilon)^3} + \frac{1}{\varepsilon^2\sqrt{\theta^3\lambda^5\tau}} + \frac{1}{\theta^2(\lambda\varepsilon)^{5/2}}\right).$$

V. Numerical Experiments

We now experimentally check the performance of Δ-*FL*. The experiments were implemented in Python using automatic differentiation provided by PyTorch, while the data was preprocessed using LEAF [16].

A. Numerical Setting

We optimize Δ-*FL* with a variant of Algorithm 1 with an iteration budget. In particular, we reduce each w-step to *FedAvg* with a single communication round with a fixed budget of local gradient descent on selected devices The resulting algorithm is given in Algorithm 2.

Datasets, Tasks and Models. We consider two learning tasks.

(a) *Character Recognition*: We use the EMNIST dataset [17], 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 device 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 use a convolutional neural network architecture (ConvNet).

(b) *Sentiment Analysis*: We use the Sent140 dataset [18] where the input x is a tweet and the output $y = \pm 1$ is its sentiment. Each device is a distinct Twitter user. The weight α_k assigned to user k is the number of tweets publised by this user. The neural network model is a LSTM [19] built on the GloVe embeddings. We refer to the latter as "RNN".

We trained on half the devices and tested on the rest, where each training device k is weighted by its number of datapoints.

Algorithm 2 Δ-*FL* with a fixed budget

Input: N devices $\{(q_k, \alpha_k)\}_{k\in[N]}$, number of local updates n_{local}, learning rate sequence (γ_t), devices per round m, initial iterate w_0, conformity level $\theta \in (0, 1)$

Server executes:

for $t = 1, 2, \cdots$ **do**
  Sample devices $S_t \sim \text{Unif}([N])^m$
  Broadcast w_t to each device $k \in S_t$
  Each $k \in S_t$ computes $F_k(w_t)$ and sends to server
  $\eta_t \leftarrow \textit{Quantile}\left(1-\theta, \left(F_k(w_t), \alpha_k\right)\right)_{k\in S_t}\Big)$
  Filter out $S'_t = \{k \in S_t \,:\, F_k(w_t) \geq \eta_t\}$
  for each device $k \in S'_t$ **in parallel do**
    $w_{k,t} \leftarrow \textit{LocalUpdate}(k, w_t)$
  $w_{t+1} \leftarrow \textit{SecureAggregate}\left(\{(w_{k,t}, \alpha_k)\}_{k\in S'_t}\right)$

function *LocalUpdate*(k, w) ▷ Run on device k
  for $i = 1, \ldots, n_{\text{local}}$ **do**
    Update $w \leftarrow w - \gamma_t \nabla f(w; \xi_i)$ using $\xi_i \sim q_k$
  return w

We use the logistic loss for training and misclassification error for evaluation. Each experiment is repeated 5 times with a different sampling of train and test devices.

Methods. We compare Δ-*FL* with the standard objective (1) optimized using *FedAvg*, *FedProx* [7] (which shows more stable convergence than *FedAvg*), and *FedAvg-Sub*, which is *FedAvg* with as many devices per round as Δ-*FL* for the best conformity level. We also compare Δ-*FL* with other heterogeneity-sensitive objectives, namely q-*FFL* [8], *AFL* [6].

Metrics. We track the loss F_k of each training device and the misclassification error on each test device. We summarize these distributions with their mean and the 90^{th} percentile, where the latter measures performance on devices with low conformity.

Hyperparameters. We run *FedAvg* until convergence and fix the corresponding number of iterations as the budget for all other algorithms. We tuned a learning rate schedule using grid search to find the best terminal loss averaged over training devices 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 100 devices per round for training. For all methods, local SGD is run on selected devices for 1 epoch. 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* devices. We use q-*FFL* with $q = 10$ as a proxy for *AFL*, as it was found to converge faster with similar performance across devices [8].

B. Experimental Results

Performances across Iterations. Fig. 3 compares the convergence of Algorithm 1 with *FedAvg*, measured in terms of the number of communication rounds. We see that Δ-*FL* converges on par with *FedAvg*, while using the same hyperparameters such as learning rate in the w-step of Algorithm 1. We noticed also that tuning the communication budget for the w-step of Algorithm 1 (set to 1 for the rest of the experiments as in Algorithm 2) allows minor improvements in the terminal error. Still, the *rate* of convergence is the same.

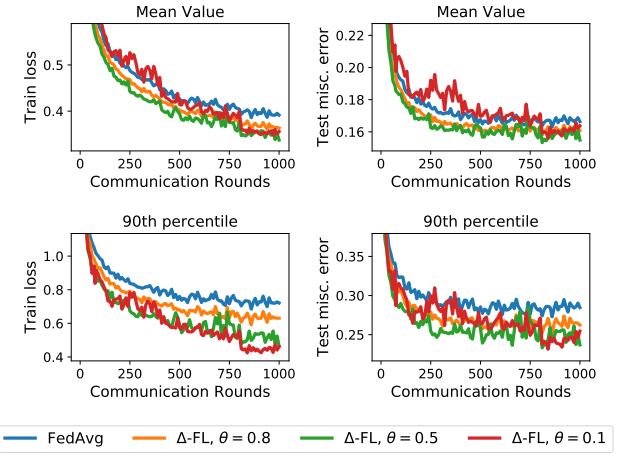

Fig. 3. Performance across iterations of loss on training devices and misclassification error of test devices for EMNIST.

Performances and comparisons. Table I records the mean and 90^{th} percentile of the misclassification error on the test devices. We find that Δ-*FL* improves the 90^{th} percentile test error in three out of four dataset-model pairs, and is within one standard deviation of the best error in the other last one. A 3.3% absolute (12% relative) improvement on EMNIST is most striking. Indeed, Δ-*FL* accounts for test users who do not conform to the population. Δ-*FL* is competitive with the other methods on the mean error, and, perhaps surprisingly, attains the smallest mean error in two cases. In Sent140-RNN, Δ-*FL* with $\theta = 0.1$ displays unstable behavior. This could be due to the objective being unsuitable (*AFL* does poorly too) coupled with η-step filtering out too many devices. Nevertheless, Δ-*FL* is competitive for multiple values of θ.

Performance Across Devices. Fig. 4 and 5 visualize distribution of error on test devices, respectively as a histogram and a scatter plot against the number of datapoints. We find that Δ-*FL* exhibits thinner upper and lower tails of the error in general and a lower variance of the distribution.

Regarding Fig. 5, we observe that improvement over the worst cases is achieved regardless of the local data size of the devices. Indeed, the device filtering step operates a sorting of the loss of the devices which does not prevent small devices from being selected. In contrast, *FedAvg*, by averaging with respect to the weights of the devices, is likely to give more importance to the updates of devices with larger local data size. Secondly, Δ-*FL* appears to reduce the variance of the loss on the train devices. Lastly, note that amongst test devices 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.

VI. Conclusion and Perspectives

We presented Δ-*FL*, a federated learning framework that can handle heterogeneous client devices that do not conform to the

TABLE I
MEAN AND 90^{TH} PERCENTILE OF THE DISTRIBUTION OF MISCLASSIFICATION ERROR (IN %) ON THE TEST DEVICES.

	EMNIST		Sent140	
	Mean	90^{th} Percentile	Mean	90^{th} Percentile
FedAvg	16.64 ± 0.50	28.46 ± 1.07	30.16 ± 0.44	49.67 ± 3.95
FedAvg-Sub	16.23 ± 0.23	27.57 ± 0.81	$\mathbf{29.86} \pm 0.46$	46.94 ± 3.84
FedProx	16.02 ± 0.54	27.01 ± 1.86	30.20 ± 0.48	49.86 ± 4.07
q-FFL (best q)	16.59 ± 0.30	28.02 ± 0.80	29.96 ± 0.56	48.66 ± 4.68
AFL	33.01 ± 0.37	45.08 ± 1.00	37.74 ± 0.65	57.78 ± 1.19
Δ-*FL*, $\theta = 0.8$	16.09 ± 0.40	26.23 ± 1.15	30.31 ± 0.33	$\mathbf{46.46} \pm 4.39$
Δ-*FL*, $\theta = 0.5$	$\mathbf{15.49 \pm 0.30}$	$\mathbf{23.69 \pm 0.94}$	33.59 ± 2.44	50.48 ± 8.24
Δ-*FL*, $\theta = 0.1$	16.37 ± 1.03	25.46 ± 2.77	51.98 ± 11.81	86.45 ± 10.95

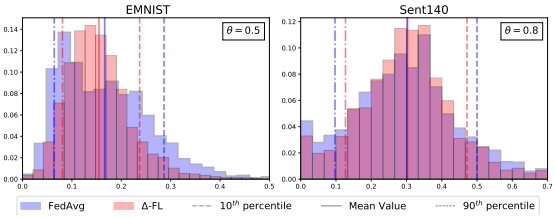

Fig. 4. Histogram of misclassification error on test devices.

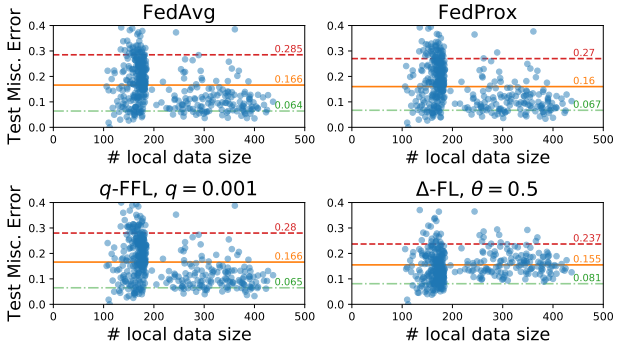

Fig. 5. Scatter plots of misclassification error on test devices against its data size for EMNIST.

population. We provided an optimization algorithm, proved its rate of convergence and demonstrated numerically that Δ-*FL* boosts performance on non-conforming devices.

An interesting venue for future work is the exploration of settings where new training users arrive in an online fashion and the trend distribution is revealed incrementally over time. It is also interesting to consider other divergence to measure the conformity between a test distribution p_π and the trend distribution p_α, for instance, the χ^2-divergence $1/(2N)\sum_{k=1}^{N}(\pi_k/\alpha_k - 1)^2$ or the ℓ_1 norm $\|\pi - \alpha\|_1$.

ACKNOWLEDGMENTS

We would like to thank the authors of [15] 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] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *AISTATS*, 2017, pp. 1273–1282.
[2] P. Kairouz *et al.*, "Advances and Open Problems in Federated Learning," *arXiv Preprint*, 2019.
[3] 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.
[4] 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.
[5] 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*, vol. 99, 2019.
[6] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic Federated Learning," in *ICML*, 2019.
[7] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," in *MLSys*, 2020.
[8] T. Li, M. Sanjabi, and V. Smith, "Fair Resource Allocation in Federated Learning," in *ICLR*, 2020.
[9] Y. Laguel, K. Pillutla, J. Malick, and Z. Harchaoui, "A Superquantile Approach to Federated Learning with Heterogeneous Devices," *arXiv preprint*, 2021.
[10] R. T. Rockafellar and S. Uryasev, "Optimization of Conditional Value-at-Risk," *Journal of Risk*, vol. 2, pp. 21–42, 2000.
[11] R. T. Rockafellar, J. O. Royset, and S. I. Miranda, "Superquantile regression with applications to buffered reliability, uncertainty quantification, and conditional value-at-risk," *European Journal of Operational Research*, vol. 234, no. 1, pp. 140–154, 2014.
[12] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in *ICML*, 2020.
[13] A. Beck and M. Teboulle, "Smoothing and First Order Methods: A Unified Framework," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 557–580, 2012.
[14] G. Lan, *First-order and Stochastic Optimization Methods for Machine Learning*. Springer, 2020.
[15] 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.
[16] 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.
[17] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *arXiv Preprint*, 2017.
[18] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *CS224N Project Report, Stanford*, 2009.
[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.