Fast-Convergent Federated Learning

Hung T. Nguyen, Vikash Sehwag, Seyyedali Hosseinalipour, *Member, IEEE*, Christopher G. Brinton, *Senior Member, IEEE*, Mung Chiang, *Fellow, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*— Federated learning has emerged recently as a promising solution for distributing machine learning tasks through modern networks of mobile devices. Recent studies have obtained lower bounds on the expected decrease in model loss that is achieved through each round of federated learning. However, convergence generally requires a large number of communication rounds, which induces delay in model training and is costly in terms of network resources. In this paper, we propose a fast-convergent federated learning algorithm, called FOLB, which performs intelligent sampling of devices in each round of model training to optimize the expected convergence speed. We first theoretically characterize a lower bound on improvement that can be obtained in each round if devices are selected according to the expected improvement their local models will provide to the current global model. Then, we show that FOLB obtains this bound through uniform sampling by weighting device updates according to their gradient information. FOLB is able to handle both communication and computation heterogeneity of devices by adapting the aggregations according to estimates of device's capabilities of contributing to the updates. We evaluate FOLB in comparison with existing federated learning algorithms and experimentally show its improvement in trained model accuracy, convergence speed, and/or model stability across various machine learning tasks and datasets.**

***Index Terms*— Federated learning, distributed optimization, fast convergence rate.**

I. Introduction

OVER the past decade, the intelligence of devices at the network edge has increased substantially. Today, smartphones, wearables, sensors, and other Internet-connected devices possess significant computation and communication capabilities, especially when considered collectively. This has created interest in migrating computing methodologies from cloud to edge-centric to provide near-real-time results [1].

Manuscript received July 27, 2020; revised September 27, 2020; accepted October 21, 2020. Date of publication November 9, 2020; date of current version December 16, 2020. The work of Hung T. Nguyen and Mung Chiang was supported in part by the Defense Advanced Research Projects Agency (DARPA) under Contract AWD1005371 and Contract AWD1005468. The work of H. Vincent Poor was supported in part by the U.S. National Science Foundation under Grant CCF-1908308. *(Corresponding author: Hung T. Nguyen.)*

Hung T. Nguyen, Vikash Sehwag, and H. Vincent Poor are with the Department of Electrical Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: hn4@princeton.edu; vvikash@princeton.edu; poor@princeton.edu).

Seyyedali Hosseinalipour, Christopher G. Brinton, and Mung Chiang are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: hosseina@purdue.edu; cgb@purdue.edu; chiang@purdue.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2020.3036952.

Digital Object Identifier 10.1109/JSAC.2020.3036952

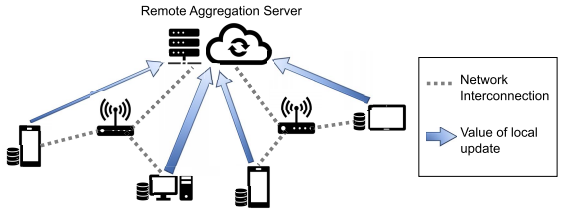


Fig. 1. Different from standard federated learning algorithms which are based on uniform sampling, our proposed methodology improves convergence rates through intelligent sampling that factors in the values of local updates that devices provide.

Most applications of interest today involve machine learning (ML). Federated learning (FL) has emerged recently as a technique for distributing ML model training across edge devices. It allows solving machine learning tasks in a distributing setting comprising a central server and multiple participating "worker" nodes, where the nodes themselves collect the data and never transfer it over the network, which minimizes privacy concerns. At the same time, the federated learning setting introduces challenges of statistical and system heterogeneity that traditional distributed optimization methods [2]–[11] are not designed for and thus may fail to provide convergence guarantees.

One such challenge is the number of devices that must participate in each round of computation. To provide convergence guarantees, recent studies [12]–[15] in distributed learning have to assume full participation of all devices in every round of optimization, which results in excessively high communication costs in edge network settings. On the other hand, [6], [8], [10], [16]–[19] violate the statistical heterogeneity property. In contrast, FL techniques provide flexibility in selecting only a fraction of clients in each round of computations [20]. However, such a selection of devices, which is often done uniformly, naturally causes the convergence rates to be slower.

In this paper, we take into consideration that in each computation round, some clients provide more valuable updates in terms of reducing the overall model loss than others, as illustrated in Figure 1. By taking this into account, we show that the convergence in federated learning can be vastly improved with an appropriate non-uniform device selection method. We first theoretically characterize the overall loss decrease of the *non-uniform* version of the recent state-of-the-art FedProx algorithm [21], where clients in each round are selected based on a target probability distribution. Under such a non-uniform device selection scheme, we obtain a lower bound on the expected decrease in global loss function at every

computation round at the central server. We further improve this bound by incorporating gradient information from each device into the aggregation of local parameter updates and characterize a device selection distribution, named *LB-near-optimal*, which can achieve a near-optimal lower bound over all non-uniform distributions at each round.

Straightforwardly computing such distribution in every round involves a heavy communication step across *all devices* which defeats the purpose of federated learning where the assumption is that only a subset of devices participates in each round. We address this communication challenge with a novel federated learning algorithm, named FOLB, which is based on a simple yet effective re-weighting mechanism of updated parameters received from participating devices in every round. With twice the number of devices selected in baseline federated learning settings, i.e., as in the popular FedAvg and FedProx algorithms, FOLB achieves the near-optimal decrease in global loss as that of the LB-near-optimal device selection distribution, whereas with the same number of devices, FOLB provides a guarantee of global loss decrease close to that of the LB-near-optimal and even better in some cases.

Another challenge in federated learning is device heterogeneity, which affects the computation and communication capabilities across devices. We demonstrate that FOLB can easily adapt to such device heterogeneity by adjusting its re-weighting mechanism of the updated parameters returned from participating devices. Computing the re-weighting coefficients involves presumed constants which are related to the loss function characteristics and solvers used in distributed devices, and more importantly, may not be available beforehand. Even estimating those constants may be difficult and incur considerable computation and communication overhead. Thus, we show a greater flexibility of FOLB that its re-weighting mechanism can group all presumed constants into a *single* hyper-parameter which can be optimized with line search.

A. Outline and Summary of Contributions

Compared to related work (discussed next), in this paper we make the following contributions:

- We provide a theoretical characterization of fast federated learning based on a non-uniform selection of participating devices. In particular, we establish lower bounds on decrease in global loss given a non-uniform device selection from any target distribution, and compare these bounds directly with FedProx. We demonstrate how local gradient information from each devices can be aggregated to improve the lower bound and also compute a near-optimal distribution for device selection (Section III).
- We propose FOLB, a federated learning algorithm which employs an accurate and communication-efficient approximation of a near-optimal distribution of device selection to accelerate convergence (Section IV).
- We show a successful generalization on FOLB in federated learning with computation and communication heterogeneity among participating devices (Section V).
- We perform extensive experiments on synthetic, vision, and language datasets to demonstrate the success of FOLB over FedAvg and FedProx algorithms in terms of model accuracy, training stability, and/or convergence speed (Section VI).

B. Related Work

Distributed optimization has been vastly studied in the literature [2]–[11] which focuses on a datacenter environment model where (i) the distribution of data to different machine is under control, e.g., uniformly at random, and (ii) all the machines are relatively close to one another, e.g., minimal cost of communication. However, those approaches no longer work on the emerging environment of distributed mobile devices due to its peculiar characteristics, including non-i.i.d. and unbalanced data distributions, limited communication, and heterogeneity of computation between devices. Thus, many recent efforts [6], [8], [10], [12]–[24] have been devoted to coping with these new challenges.

Most of the existing works [6], [8], [10], [12]–[19] either assume the full participation of all devices or violate statistical heterogeneity property inherent in our environment. McMahan *et al.* [20] was the first to define federated learning setting in which a learning task is solved by a loose federation of participating devices which are coordinated by a central server and proposed the heuristic FedAvg algorithm. FedAvg runs through multiple rounds of optimization, in each round, it randomly selects a small set of K devices to perform local stochastic gradient descent with respect to their local data. Then, the locally updated model parameters are sent back to the central server where an averaging is taken and regarded as new parameters. It was shown in [20] to perform well in terms of both performance and communication cost. More recently, [25] shows convergence rate of FedAvg when the cost function is strongly convex and smooth. Federated multi-task learning was proposed in [26] that allows slightly different models in different devices and framed the problem in multi-task learning framework. More recent work in [23], [24] propose federated optimizers and algorithms that improve over FedAvg in terms of convergence rate subject to a number of assumptions about the loss functions and non-i.i.d. distributions of data. However, heterogeneity in computation and communication across devices have not been a focus of these models.

Very recently, [21] proposed FedProx with the main difference from FedAvg of adding a proximal term in every local loss function to keep the updated parameters across devices more similar. FedProx follows the same steps as FedAvg, however, it provides convergence rate for both convex and non-convex losses and deals with statistical heterogeneity. FedProx also allows any local optimizer at the local devices. Our work utilizes the idea of adding a proximal term to local loss function, however, our proposed algorithm FOLB takes a unique approach that aims at a near-optimal device selection distribution to maximize the loss decrease at every round of optimization. On the other hand, FedProx and FedAvg select devices uniformly at random in each round.

Other aspects of federated learning have also been studied, such as privacy of user data [27]–[31], fairness in federated learning [32], federated learning over communication systems [33]–[37], and federated learning for edge networks [38]–[40]. We refer the interested reader to comprehensive surveys in [41], [42] and references therein for more details.

II. Preliminaries and Modeling Assumptions

We first formalize federated learning, including the standard system model (Section II-A), learning algorithms (Section II-B), and common theoretical assumptions (Section II-C).

A. System and Learning Model

Consider a network of N devices, indexed $k \in \{1, \ldots, N\}$, where each device possesses its own local (private) dataset D_k. Each data point $d \in D_k$ is assumed to contain a feature vector $\mathbf{x}_d$ and a target variable y_d. The objective of federated learning is to train a machine learning (ML) model of interest over this network, i.e., to learn a mapping $g_{\mathbf{w}} : \mathbf{x}_d \rightarrow \hat{y}_d$ from a given input sample $\mathbf{x}_d$ to a predicted output $\hat{y}_d$ parameterized by a vector $\mathbf{w}$, with each device processing its own data to minimize communication overhead.

For our purposes, an ML model is specified according to its parameter vector $\mathbf{w}$ and loss function $f(\mathbf{w}) = (1/|D|) \sum_{d \in D} l(\mathbf{w}, \mathbf{x}_d, y_d)$ to be minimized. Here, D is the training dataset available, and $l(\mathbf{w}, \mathbf{x}_d, y_d)$ represents the error between $\hat{y}_d$ and y_d (e.g., the squared distance). Thus, we seek to find $\mathbf{w}$ that minimizes $f(\mathbf{w})$ over the data $D = \cup_k D_k$ in the network. In federated learning, this minimization is not performed directly, as each device k only has access to D_k. Defining $F_k(\mathbf{w}) = (1/|D_k|) \sum_{d \in D_k} l_d$ as the local loss function at k over D_k, if we assume that $|D_i| = |D_j| \ \forall i, j$, i.e., each device processes the same amount of data, we can express the optimization as an average over the $F_k(\mathbf{w})$:

$$\min_{\mathbf{w}} f(\mathbf{w}), \text{ where } f(\mathbf{w}) \coloneqq \frac{1}{N} \sum_{k=1}^{N} F_k(\mathbf{w}). \tag{1}$$

More generally, nodes may process different amounts of data, e.g., due to heterogeneous compute capabilities. In such cases, we can replace the factor $1/N$ with $p_k = |D_k|/|D|$ for a weighted average of the $F_k(\mathbf{w})$ [21], [43]. This is the approach we take throughout this paper.

Federated learning algorithms differ in how (1) is solved. In our case, we will assume that a central server is available to orchestrate the learning across the devices. Such a scenario is increasingly common in fog or edge computing systems, where an edge server may be connected to several edge devices, e.g., in a smart factory [43]. We will next introduce the standard algorithms for federated learning in these environments.

B. Standard Federated Learning Algorithms

Federated learning algorithms generally solve (1) in three steps: local learning, aggregation, and synchronization, which are repeated over several rounds [20]. In each round t, the server selects a set $\mathcal{K}_t$ of K devices among the N total to update the current estimate $\mathbf{w}^t$ for the optimal set of parameters $\mathbf{w}^\star$. Each device $k \in \mathcal{K}_t$ selected then updates $\mathbf{w}^t$ based on its local loss $F_k(\mathbf{w})$, producing $\mathbf{w}_k^{t+1}$, and sends this back to the server. The server then aggregates these locally updated parameters according to

$$\mathbf{w}^{t+1} = \frac{1}{K} \sum_{k \in \mathcal{K}_t} \mathbf{w}_k^{t+1}, \tag{2}$$

and synchronizes the devices with this update before beginning the next round.

FedAvg [20] is the standard federated learning algorithm that uses this framework. In FedAvg, the loss $F_k(\mathbf{w})$ is directly minimized during the local update step, using gradient descent techniques. Formally, each device calculates $\mathbf{w}_k^{t+1} = \mathbf{w}^t - \eta \nabla F_k(\mathbf{w}^t)$, where $\nabla F_k(\mathbf{w}^t) = (1/|D_k|) \sum_{d \in D_k} \nabla l(\mathbf{w}^t, \mathbf{x}_d, y_d)$ is the average of the loss gradient over device k's data. It is also possible to use multiple iterations of local updates between global aggregations [15].

More recently, FedProx was introduced [21], which differs from FedAvg in the local update step: instead of minimizing $F_k(\mathbf{w})$ at device k, it minimizes

$$h_k(\mathbf{w}, \mathbf{w}^t) = F_k(\mathbf{w}) + \frac{\mu}{2} \left\| \mathbf{w} - \mathbf{w}^t \right\|^2. \tag{3}$$

The proximal term $\frac{\mu}{2} \left\| \mathbf{w} - \mathbf{w}^t \right\|^2$ added to each local loss function brings two modeling benefits: (i) it restricts the divergence of parameters between devices that will arise due to heterogeneity in their data distributions, and (ii) for appropriate choice of μ, it will turn a non-convex loss function $F_k(\mathbf{w})$ into a convex $h_k(\mathbf{w}, \mathbf{w}^t)$ which is easier to optimize. The approach we develop beginning in Section III will build on FedProx. Note that by setting $\mu = 0$, $h_k(\mathbf{w}, \mathbf{w}^t) = F_k(\mathbf{w})$ and we get back the setting in FedAvg. Thus, our algorithm FOLB naturally applies on FedAvg and our theoretical results still hold if all $F_k(\mathbf{w}), k = 1, \ldots, N$ are strongly convex.

C. ML Model Assumptions

For theoretical analysis of federated learning algorithms, a few standard assumptions are typically made on the ML models (see e.g., [15], [21], [43]). We will employ the following in our analysis:

Assumption 1 (L-Lipschitz Gradient): $F_k(\mathbf{w})$ is L-Lipschitz gradient for each device $k \in \{1, \ldots, N\}$, i.e., $\|\nabla F_k(\mathbf{w}) - \nabla F_k(\mathbf{w}')\| \leq L \|\mathbf{w} - \mathbf{w}'\|$ for any two parameter vectors $\mathbf{w}, \mathbf{w}'$. This also implies (via the triangle inequality) that that the global $f(\mathbf{w})$ is L-Lipschitz gradient.

Assumption 2 (B-Dissimilar Gradients): The gradient of $F_k(\mathbf{w})$ is at most B-dissimilar from $f(\mathbf{w})$ for each k, i.e., $\|\nabla F_k(\mathbf{w})\| \leq B \|\nabla f(\mathbf{w})\|$ for each $\mathbf{w}$.

Assumption 3 (σ-Bounded Hessians): The smallest eigenvalue of the Hessian matrix $\nabla^2 F_k$ is $-\sigma$ for each k, i.e., $\nabla^2 F_k \succeq -\sigma \mathbf{I}$ for the identity matrix $\mathbf{I}$. This implies that $h_k(\mathbf{w}, \mathbf{w}^t)$ in (3) is μ'-strongly convex, where $\mu' = \mu - \sigma$.

Assumption 4 (γ-Inexact Local Solvers): Local updates will yield a γ-inexact solution $\mathbf{w}_k^{t+1}$ of $\min_{\mathbf{w}} h_k(\mathbf{w}, \mathbf{w}^t)$ for every k and t, i.e., $\left\| \nabla h_k(\mathbf{w}_k^{t+1}, \mathbf{w}^t) \right\| \leq \gamma \|\nabla h_k(\mathbf{w}^t, \mathbf{w}^t)\|$. We assume that γ is in the range $[0, 1]$ since $\gamma = 0$ corresponds to solving to optimality, and $\gamma = 1$ happens with the initial

parameters $\mathbf{w}_k^{t+1} = \mathbf{w}_k^t$ and since the function $h(\mathbf{w}, \mathbf{w}^t)$ is convex, the local optimization algorithm at device should reduce the gradient norm, e.g., gradient descent algorithm.

In [15], [43], Assumptions 3&4 are replaced with a stronger assumption that the $F_k(\mathbf{w})$ are convex. This corresponds to the case where $\sigma \leq 0$ in Assumption 3, meaning $\nabla^2 F_k$ is positive semidefinite, and FedAvg can be used to minimize the $F_k(\mathbf{w})$ directly without a proximal term. Similar to [21], the results we derive in this work will more generally hold for non-convex $F_k(\mathbf{w})$, which is true of many ML models today (e.g., neural networks). We also note that FedProx makes a similar assumption to Assumption 4 in deriving its convergence bound [21], i.e., on the precision of the local solvers. In Section V, we will present a technique where each device k estimates its own γ_k based on its local gradient update.

Technical Approach: In the following sections, we first investigate the general non-uniform device selection in federated learning and show that in each round, a device's contribution in reducing the global loss function is bounded by the inner product between its local gradient and the global one. Hence, a near-optimal device selection distribution is introduced, that samples devices according to the inner products between their local and global gradients. Unfortunately, trivial solutions to compute or estimate this distribution are excessively expensive in communication demand. We next introduce FOLB to address this challenge with the core idea of using 2 independent sets of devices, one for estimating the global gradient and another for carrying out local optimization. The locally updated parameters from the second set are then re-weighted by the inner products between their gradients and the estimated global gradient and aggregated to form a new global model. We also analyze the version using a single set of devices and how to handle communication and computation heterogeneity with FOLB.

III. FedNu: Non-Uniform Federated Learning

In this section, we develop our methodology for improving the convergence speed of federated learning. This includes non-uniform device selection in the local update (Section III-A), and inclusion of gradient information in the aggregation (Section III-B). Our theoretical analysis on the expected decrease in loss in each round of learning leads to a selection distribution update that achieves an efficient lower bound (Section III-C).

A. Non-Uniform Device Selection

As discussed in Section II-B, standard federated learning approaches select a set of K devices uniformly at random for local updates in each round. In reality, certain devices will provide better improvements to the global model than others in a round, depending on their local data distributions. If we can estimate the expected decrease in loss each device will provide to the system in a particular round, then the device selections can be made according to those that are expected to provide the most benefit. This will in turn minimize the model convergence time.

Algorithm 1 Federated Learning With Non-Uniform Device Selection

Input : $K, T, \mu, \gamma, \mathbf{w}^0, N, P_k^t \; k = 1, \ldots, N$

for $t = 0, \ldots, T-1$ **do**

- Server samples (with replacement) a multiset S_t of K devices according to $P_k^t, k = 1, \ldots, N$
- Server sends $\mathbf{w}^t$ to all devices $k \in S_t$
- Each device $k \in S_t$ finds a $\mathbf{w}_k^{t+1}$ that is a γ_k^t-inexact minimizer of $\arg\min_{\mathbf{w}} h_k(\mathbf{w}, \mathbf{w}^t)$, as defined in (3)
- Each device $k \in S_t$ sends $\mathbf{w}_k^{t+1}$ back to the server
- Server aggregates the $\mathbf{w}_k^{t+1}$ according to $\mathbf{w}^{t+1} = \frac{1}{K} \sum_{k \in S_t} \mathbf{w}_k^{t+1}$

Formally, we let P_k^t be the probability assigned to device k for selection in round t, where $0 \leq P_k^t \leq 1$ and $\sum_{k=1}^N P_k^t = 1 \; \forall t$. In our federated learning scheme, during round t, the server chooses a multiset S_t of size K by sampling K times from the distribution $P_1^t, \ldots, P_N^t$. Note that this sampling occurs with replacement, i.e., a device may appear in S_t multiple times and K is the cardinality of this multiset. Each unique $k \in S_t$ then performs a local update on the global model estimate $\mathbf{w}^t$ to find a γ-inexact minimizer $\mathbf{w}_k^{t+1}$ of $h_k(\mathbf{w}, \mathbf{w}^t)$ in (3), which the server aggregates to form $\mathbf{w}^{t+1}$. Algorithm 1 summarizes this procedure, assuming averaging for aggregation; if k appears in S_t more than once, this aggregation effectively places a larger weight on $\mathbf{w}_k^{t+1}$.

Given the introduction of P_k^t, we call our methodology FedNu, i.e., non-uniform federated learning. A key aspect will be developing an algorithm for P_k^t estimation in each round. The following theorem gives a lower bound on the expected decrease in loss achieved from round t of Algorithm 1, which will assist in this development:

Theorem 1: With loss functions F_k satisfying Assumptions 1-4, supposing that $\mathbf{w}^t$ is not a stationary solution, in Algorithm 1, the expected decrease in the global loss function satisfies

$$\begin{aligned}\mathbb{E}[f(\mathbf{w}^{t+1})] &\leq f(\mathbf{w}^t) - \frac{1}{K\mu}\mathbb{E}\Big[\sum_{k \in S_t} \langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle\Big] \\ &\quad + B\Big(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\Big)\left\|\nabla f(\mathbf{w}^t)\right\|^2, \quad (4)\end{aligned}$$

where $\mu' = \mu - \sigma > 0$, and the expectation $\mathbb{E}$ is with respect to the choice of K devices following probabilities P_k^t. As a corollary, after T rounds,

$$\begin{aligned}\mathbb{E}[f(\mathbf{w}^T)] &\leq f(\mathbf{w}^0) - \frac{1}{K\mu}\mathbb{E}\Big[\sum_{t=0}^{T-1}\sum_{k \in S_t} \langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle\Big] \\ &\quad + B\Big(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\Big)\sum_{t=0}^{T-1}\left\|\nabla f(\mathbf{w}^t)\right\|^2,\end{aligned}$$

where the expectation is with respect to the random selections of $S_0, S_1, \ldots, S_{T-1}$.

The full proof of Theorem 1 as well as proofs of later theorems/propositions are presented in appendix.

Theorem 1 provides a bound on how rapidly the global loss can be expected to improve in each iteration based on the selection of devices in Algorithm 1. It shows a dependency on parameters L, B, γ, and μ of the ML model. In particular, we see that $\mathbb{E}[f(\mathbf{w}^{t+1})] \propto B^2$, meaning that as the dissimilarity between local and global model gradients grows larger, the bound weakens. Intuitively, B depends on the variance between local data distributions: as the datasets D_k approach being independent and identically distributed (i.i.d.) across k, the gradients will become more similar, and B will approach 1. As they become less i.i.d., however, the gradients will diverge, and B will increase. Hence, Theorem 1 gives quantitative insight into the effect of data heterogeneity on federated learning convergence.

Compared to the bound of FedProx [21], which was shown to work on the particular uniform distribution, our result in Theorem 1 is more general and applicable for any given probability distribution. Moreover, our result offers a new approach to optimize convergence rate through maximizing the inner product term $\mathbb{E}\big[\sum_{k\in S_t}\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle\big]$. The proof of Theorem 1 also takes a different path compared to that of FedProx in [21], which relies on the uniform distribution to first establish intermediate relations of $f(\mathbf{w}^{t+1})$ and $f(\mathbf{w}^t)$ with $f(\bar{\mathbf{w}}^{t+1})$, where $\bar{\mathbf{w}}^{t+1} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{w}_k^{t+1}$, and then connects $f(\mathbf{w}^{t+1})$ with $f(\mathbf{w}^t)$. Our result applies for any distribution and thus required direct proof of the relation between $f(\mathbf{w}^{t+1})$ and $f(\mathbf{w}^t)$ via bounding each of the terms given by the L-Lipschitz continuity of f.

B. Aggregation With Gradient Information

An immediate suggestion from the expectation term in Theorem 1 is that any devices which have a negative inner product $\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle < 0$ between their gradients $\nabla F_k(\mathbf{w}^t)$ and the global gradient $\nabla f(\mathbf{w}^t)$ would actually hurt model performance. This is due to the averaging technique used for model aggregation in Algorithm 1, which is common in federated learning algorithms due to its simplicity [15], [21], [43]. It is consistent with the characteristics of distributed gradient descent [5], [7], where the global gradient (i.e., across the entire dataset) can reduce the overall loss while individual local gradients (i.e., at individual devices) that are not well aligned with the global objective – in this case, those with negative inner product – will not help improve the overall loss.

If we assume the server can estimate when a device's inner product is negative, then we can immediately improve FedNu with an aggregation rule of

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \frac{1}{K}\sum_{k\in S_t}\operatorname{sign}(\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle)(\mathbf{w}_k^{t+1} - \mathbf{w}^t) \tag{5}$$

in Algorithm 1 based on the signum function. This negates local updates from devices in S_t that have $\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle < 0$, and provides a stronger lower-bound than given in Theorem 1:

Proposition 1: With the same assumptions on F_k and $\mathbf{w}^t$ as in Theorem 1, with (5) *used as the aggregation rule in Algorithm 1 (Line 6), the expected decrease in the global loss satisfies*

$$\begin{aligned}\mathbb{E}[f(\mathbf{w}^{t+1})] &\le f(\mathbf{w}^t) - \frac{1}{K\mu}\mathbb{E}\Big[\sum_{k\in S_t}|\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle|\Big] \\ &\quad + B\left(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2. \end{aligned} \tag{6}$$

Proposition 1 is clearly stronger than Theorem 1: by incorporating gradient information, the inner products are replaced with their absolute values, making the expected decrease in loss faster. We will next propose a method for setting the selection probabilities P_k^t to optimize this bound, and then develop algorithms to estimate the inner products.

C. LB*-Near-Optimal Device Selection*

The set S_t of selected devices affects Theorem 1 through the expectation $\mathbb{E}[\sum_{k\in S_t}|\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle|]$. To maximize the convergence speed, we seek to minimize the upper bound on the loss update in each round t, which corresponds to the following optimization problem for choosing S_t:

$$\begin{aligned}&\underset{P_k^t}{\text{maximize}}\ \mathbb{E}\Big[\sum_{k\in S_t}|\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle|\Big] \\ &\text{subject to}\ \sum_k P_k^t = 1, \quad P_k^t \ge 0\ \ \forall k.\end{aligned}$$

This problem is difficult to solve analytically given the sampling relationship between S_t and P_k^t.[1] It is clear, however, that the solution which maximizes this expectation will assign higher probability of being selected to devices with higher inner product $|\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle|$. A natural candidate which satisfies this criterion is $P_k^t \propto |\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle|$. We call this distribution LB-near-optimal, i.e., near-optimal lower-bound, formally defined as follows:

*Definition 1 (*LB*-Near-Optimal Selection Distribution): The selection distribution $P_{\mathsf{lb}_k}^t$ achieving a near-optimal lower-bound on loss decrease in Theorem 1 is called the* LB*-near-optimal selection distribution, and has the form*

$$P_{\mathsf{lb}_k}^t = \frac{|\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle|}{\sum_{k'=1}^{N}|\langle\nabla f(\mathbf{w}^t), \nabla F_{k'}(\mathbf{w}^t)\rangle|}, \tag{7}$$

with the corresponding lower bound of expected loss being

$$\begin{aligned}\mathbb{E}[f(\mathbf{w}^{t+1})] &\le f(\mathbf{w}^t) - \frac{1}{\mu}\sum_{k=1}^{N}|\langle\nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle| P_{\mathsf{lb}_k}^t \\ &\quad + B\left(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2. \end{aligned} \tag{8}$$

1) Comparison to FedProx [21]: Our lower bound in (8) of Definition 1, corresponding to the near-optimal device selection distribution and achieved by our proposed algorithm

[1]Formally, the probability mass function of S_t is formed from K repeated trials of the N-dimensional categorical distribution [44] over $P_1^t, \ldots, P_N^t$.

FOLB in Section IV, is more general than the bound of FedProx in [21], which is restricted to the uniform distribution. Specifically, our bound in (8) is stronger if

$$\frac{1}{\mu}\sum_{k=1}^{N}|\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle|P_{\mathsf{lb}}{}_k^t$$

$$\geq\Big(\frac{1}{\mu}-\frac{B(1+\gamma)\sqrt{2}}{\mu'\sqrt{K}}-\frac{LB^2(1+\gamma)^2}{\mu'^2K}(2\sqrt{2K}+2)\Big)\big\|\nabla f(\mathbf{w}^t)\big\|^2,$$

which holds since

$$\begin{aligned}
&\frac{1}{\mu}\sum_{k=1}^{N}|\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle|P_{\mathsf{lb}}{}_k^t\\
&=\frac{1}{\mu}\frac{\sum_{k=1}^{N}|\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle|^2}{\sum_{k'=1}^{N}|\langle\nabla f(\mathbf{w}^t),\nabla F_{k'}(\mathbf{w}^t)\rangle|}\\
&\geq\frac{1}{\mu}\frac{1}{N}\sum_{k=1}^{N}|\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle| \qquad \text{(Cauchy-Schwarz)}\\
&\geq\frac{1}{\mu}|\frac{1}{N}\sum_{k=1}^{N}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle| \qquad \text{(triangle inequality)}\\
&\geq\frac{1}{\mu}|\langle\nabla f(\mathbf{w}^t),\nabla f(\mathbf{w}^t)\rangle|=\frac{1}{\mu}\big\|\nabla f(\mathbf{w}^t)\big\|^2.
\end{aligned}$$

The last inequality holds due to $f(\mathbf{w}^t)=\frac{1}{N}\sum_{k=1}^{N}F_k(\mathbf{w}^t)$.

2) Convergence Property: Starting from the lower-bound in (8), we can show the convergence rate in the form of the gradient converging to zero when the parameter settings satisfy certain constraints, similarly to [21]. Furthermore, since the bound in (8) is stronger than that of FedProx in [21], the corresponding convergence rate is also faster. Specifically, applying (8) for all $t=0,\ldots,T$ gives us a series of inequalities, and taking the sum of these yields the desired form of gradient convergence (see [21] for more details).

In Definition 1, the expectation term in the bound on $\mathbb{E}[f(\mathbf{w}^{t+1})]$ has been computed in terms of the selection distribution $P_{\mathsf{lb}}{}_k^t$. Unfortunately, the values of $\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle$ needed to compute the $P_{\mathsf{lb}}{}_k^t$ cannot be evaluated at the server at the beginning of round t, since the local and global gradients are not available at the time of device selection. In the rest of this section, and in Section IV, our goal will be to develop a federated learning algorithm that (i) achieves the performance of the distribution in Definition 1, i.e., provides the same loss decrease at every round, and (ii) results in an efficient implementation in a client-server network architecture. We refer to such an algorithm as an LB-near-optimal-efficient federated learning algorithm:

*Definition 2 (*LB*-Near-Optimal-Efficient Federated Learning Algorithm): An iterative federated learning algorithm is called* LB*-near-optimal-efficient if it achieves the near-optimal lower-bound of loss decrease in Definition 1, which corresponds to the near-optimal selection distribution at every round, and does not require communication between devices that is significantly more expensive than standard federated learning.*

D. Naive Algorithms for Fast Convergence

We first present two algorithms that are straightforward modifications of the methods described in this section towards the goal of satisfying Definition 2. We will see that each of these fails to satisfy one criterion in Definition 2, however, motivating our main algorithms in Section IV.

1) Direct Computation of LB*-Near-Optimal Distribution:* The most straightforward approach to achieving LB-near-optimality is enabling computation of the LB-near-optimal distribution $P_{\mathsf{lb}}{}_k^t$ at the beginning of round t and using this to sample devices. This approach requires the server to send $\mathbf{w}^t$ to all N devices, have them compute $\nabla F_k(\mathbf{w}^t)$, and then send it back to the central server. With these values, the server can exactly calculate the LB-near-optimal distribution through (7).

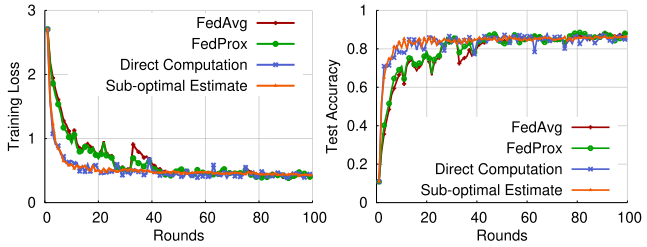
Clearly, this algorithm will obtain the LB-near-optimal distribution, leading to a fast convergence rate (assuming that this initial round of communication does not significantly increase the time of each round t). However, this algorithm requires one iteration of expensive communication between the server and all N devices. The gradient $\nabla F_k(\mathbf{w}^t)$ is the same dimension as $\mathbf{w}^t$, and the purpose of algorithms like FedAvg and FedProx selecting K of N devices is to avoid this excessive communication between a server and edge devices in contemporary network architectures [43].

As an aside, if we were able to afford this extra communication of gradients in each round, then why not just carry out the exact (centralized) gradient descent at the server? Federated learning would still be beneficial in this scenario for two reasons. First, during their local updates, each device usually carries out multiple iterations of gradient descent, saving potentially many more rounds of gradient communication to/from the server [15]. Second, while batch gradient descent converges slowly, federated learning has a flavor of stochastic gradient descent which tends to converge faster [16].

2) Sub-Optimal Estimation of LB*-Near-Optimal Distribution:* A possible workaround for the issue of expensive communication in the first approach is to further upper bound $|\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle| \leq \|\nabla f(\mathbf{w}^t)\|\,\|\nabla F_k(\mathbf{w}^t)\|$ using the Cauchy-Schwartz inequality. Since $\|\nabla f(\mathbf{w}^t)\|$ is the same for all the devices, we can take $P_k^t\propto\|\nabla F_k(\mathbf{w}^t)\|$. Hence, while this approach still requires the server to send out $\mathbf{w}^t$ to all devices for them to compute gradients, each device k only needs to send back a single number, $\|\nabla F_k(\mathbf{w}^t)\|$. This is much less expensive given the fact that edge devices tend to have larger download than upload capacities, typically by an order of magnitude [43].

While this algorithm is closer to the communication efficiency of standard federated learning algorithms, there is no guarantee on how accurately $\|\nabla f(\mathbf{w}^t)\|\,\|\nabla F_k(\mathbf{w}^t)\|$ approximates $|\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle|$, which could result in an inaccurate estimate of $P_{\mathsf{lb}}{}_k^t$. Thus, it may not satisfy the LB-near-optimal criteria of Definition 2.

We demonstrate the better performance when using directly or estimating the LB-near-optimal selection distribution than existing state-of-the-art federated learning algorithms in Fig. 2. Here we run the above two naive algorithms targeting the



(a) Training loss (b) Test accuracy

Fig. 2. Training loss and test accuracy of our motivating idea and state-of-the-art approaches on MNIST dataset ($\mu = 1$, see Sec. VI for details on experimental settings).

LB-near-optimal distribution along with FedAvg and FedProx, and observe significant improvements over both FedAvg and FedProx in terms of convergence speed. Our methods quickly converge after only a few rounds of communication. This motivates our proposed algorithm, FOLB, which also targets the LB-near-optimal distribution, however, removes the communication burden in the naive algorithms.

IV. FOLB: An LB-Near-Optimal-Efficient Federated Learning Algorithm

As discussed in Section III-C, the LB-near-optimal selection distribution given in Definition 1 for maximizing the loss decrease in round t cannot be computed by the server at the beginning of round t, since it involves all local gradients of the current global estimate $\mathbf{w}^t$. The straightforward approximation using Cauchy-Schwartz still requires one iteration of additional communication where the server sends $\mathbf{w}^t$ to all devices, and does not guarantee LB-near-optimality. With the goals of fast convergence and low communication overhead in mind, the challenges we face in developing an LB-near-optimal-efficient federated learning algorithm for FedNu described in Definition 2 are two-fold:

(1) How can we accurately estimate (preferably with performance guarantees) the LB-near-optimal probability distribution without involving all local gradients?
(2) How can we obtain this estimate efficiently, i.e., with minimal communication overhead on top of standard federated learning algorithms?

In this section, we develop a federated learning algorithm called FOLB (Section IV-A) that addresses these challenges. The key idea of FOLB is a novel calibration procedure for aggregating local model updates from devices selected uniformly at random. This calibration weighs the updates received by their estimated importance to the model, which we show matches the performance of Theorem 1 (Section IV-B). We also demonstrate a technique to further optimize the communication demand of FOLB (Section IV-C).

A. Proposed FOLB Algorithm

The FOLB algorithm is summarized in Algorithm 2. At the start of round t, the server selects two multisets S_1^t and S_2^t of devices of size K uniformly at random, and sends $\mathbf{w}^t$ to each $k \in S_1^t$ and $k' \in S_2^t$. Each $k \in S_1^t$ computes its γ_k^t-inexact local update $\mathbf{w}_k^{t+1}$, sending both $\mathbf{w}_k^{t+1}$ and $\nabla F_k(\mathbf{w}^t)$ back to the server. Each $k' \in S_2^t$, by contrast, only computes $\nabla F_{k'}(\mathbf{w}^t)$ and sends this back, for the purpose of calibrating the updates. Then, instead of simple averaging, the server aggregates the received update parameters according to the following rule:

Algorithm 2 FOLB Algorithm for LB-Near-Optimal-Efficient Federated Learning

Input : $K, T, \mu, \gamma, \mathbf{w}^0, N$
for $t = 0, \ldots, T-1$ **do**
  Server selects two multisets S_1^t and S_2^t each of K devices uniformly at random
  Server sends $\mathbf{w}^t$ to all $k \in S_1^t$ and $k' \in S_2^t$
  for *each device* $k \in S_1^t$ **do**
    Device k computes its gradient $\nabla F_k(\mathbf{w}^t)$
    Device k sends $\nabla F_k(\mathbf{w}^t)$ back to the server
    Device k finds a γ_k^t-inexact minimizer of $\arg\min_{\mathbf{w}} h_k(\mathbf{w}, \mathbf{w}^t)$, as defined in (3)
    Device k sends $\mathbf{w}_k^{t+1}$ back to the server
  for *each device* $k' \in S_2^t$ **do**
    Device k' computes its gradient $\nabla F_{k'}(\mathbf{w}^t)$
    Device k' sends $\nabla F_{k'}(\mathbf{w}^t)$ back to the server
  Server computes $\nabla_1 f(\mathbf{w}^t), \nabla_2 f(\mathbf{w}^t)$ according to (10) and aggregates the $\mathbf{w}_k^{t+1}$ (9)

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \sum_{k \in S_1^t} \frac{\langle \nabla F_k(\mathbf{w}^t), \nabla_1 f(\mathbf{w}^t) \rangle}{\sum_{k' \in S_2^t} \langle \nabla F_{k'}(\mathbf{w}^t), \nabla_2 f(\mathbf{w}^t) \rangle} \Delta \mathbf{w}_k^{t+1}, \tag{9}$$

where

$$\nabla_i f(\mathbf{w}^t) = \frac{1}{K} \sum_{k \in S_i^t} \nabla F_k(\mathbf{w}^t), \tag{10}$$

is the gradient of the global loss $f(\mathbf{w}^t)$ estimated from the local losses across devices in $S_i^t, i \in \{1, 2\}$, and $\Delta \mathbf{w}_k^{t+1} = \mathbf{w}_k^{t+1} - \mathbf{w}^t$ is the change that device $k \in S_1^t$ made to $\mathbf{w}^t$ at round t during its local update.

The intuition behind (9) is that the local update of each device $k \in S_1^t$ is weighted by a measure of how correlated its gradient $\nabla F_k(\mathbf{w}^t)$ is with the global gradient $\nabla f(\mathbf{w}^t)$. This correlation is assessed relative to $\nabla_1 f(\mathbf{w}^t)$, which is an unbiased estimate of $\nabla f(\mathbf{w}^t)$ using gradient information obtained from S_1^t. The weights are normalized relative to a second unbiased estimate of total correlation among K devices, obtained over S_2^t.

B. Proof of LB-Near-Optimality

We now prove that FOLB obtains the same lower-bound of loss decrease at every round as the LB-near-optimal selection distribution. In particular, we have the following theorem:

Theorem 2: In Algorithm 2, with the same assumptions on F_k and $\mathbf{w}^t$ as in Theorem 1, the lower-bound achieved on the expected decrease of the global loss in round t matches (8), i.e., the LB-near-optimal selection probability distribution.

The following lemma provides a key insight into how S_1^t and S_2^t can be used to estimate the global gradient when computing the inner products with local gradients, and will help in proving Theorem 2 in Appendix D.

Lemma 1: Let $\nabla_i f(\mathbf{w}^t)$ be defined as in (10). Then,

$$\mathbb{E}\Big[\sum_{k\in S_1^t}\langle \nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle^2\Big] = \frac{K}{N}\sum_{k=1}^{N}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle^2, \tag{11}$$

and

$$\mathbb{E}\Big[\sum_{k'\in S_2^t}\langle \nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle\Big] \leq \frac{K}{N}\sum_{k'=1}^{N}|\langle \nabla f(\mathbf{w}^t), \nabla F_{k'}(\mathbf{w}^t)\rangle|. \tag{12}$$

C. Optimizing FOLB Communication Efficiency

Theorem 2 establishes the LB-near-optimal property of FOLB. Algorithm 2 does, however, call for local updates from $2K$ devices across the two sets S_1^t and S_2^t in each round (and for S_1^t, communication of both the updates and the gradients), whereas standard federated learning algorithms only sample K devices.

To reduce the communication demand further, we can make two practical adjustments to Algorithm 2. First, we can set $S_1^t = S_2^t$ in each round, i.e., only selecting one set of K random devices and using the received gradients both for parameter updates and for normalizing the weights on these updates, dropping the total to K. Second, similar to the technique in Section III-B, rather than discarding updates from devices with $\langle \nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle < 0$, we can aggregate the negatives of their $\Delta\mathbf{w}_k^{t+1}$, thereby leveraging all K. Our modified aggregation rule becomes

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \sum_{k\in S_1^t}\frac{\langle \nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle}{\sum_{k'\in S_1^t}|\langle \nabla F_{k'}(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle|}\Delta\mathbf{w}_k^{t+1}. \tag{13}$$

A key step in the proof of Theorem 2, for (31), relied on the independence between sampling S_1^t and S_2^t. With $S_1^t = S_2^t$, this clearly no longer holds. Instead, we have the following:

Proposition 2: In FOLB, with the same assumptions on F_k and $\mathbf{w}^t$ as in Theorem 1, and (13) used as the aggregation rule in Algorithm 2, the lower-bound on expected decrease in the global objective loss function satisfies

$$\begin{aligned}\mathbb{E}[f(\mathbf{w}^{t+1})] &\leq f(\mathbf{w}^t) - \frac{K}{\mu N}\sum_{k=1}^{N}|\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle| \\ &+ B\left(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2. \end{aligned}\tag{14}$$

Proof: The proof is similar to that of Proposition 2, with the key difference being that Lemma 1 now holds with equality. □

Comparison: In comparing our result in Proposition 2 with that of the LB-near-optimal selection distribution in Definition 1, the new bound is better when $\frac{K}{\mu N}\sum_{k=1}^{N}|\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle| > \frac{1}{\mu}\sum_{k=1}^{N}|\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle| P_{\mathsf{lb}_k^t}$. This is the case when the data distribution across different devices becomes more uniform. To see this, let us consider two extreme cases: (i) under a uniform distribution of data, $P_{\mathsf{lb}_k^t} \approx 1/N$ and the new bound is K times better than the LB-near-optimal bound; (ii) when only one device has data, then the new bound is K/N times worse than the LB-near-optimal bound. In practice, the scenarios closer to case 1 will be much more prevalent than those similar to case 2, and thus most of the time, the new bound tends to be better than the earlier one.

V. Handling Computation and Communication Heterogeneity

A practical consideration of distributed optimization on edge devices is the heterogeneity of computing power and communication between those devices and the central server. In this section, we show how FOLB can be easily adapted to handle heterogeneity by tweaking the aggregation rule slightly.

A. Modeling Heterogeneous Communication and Computation

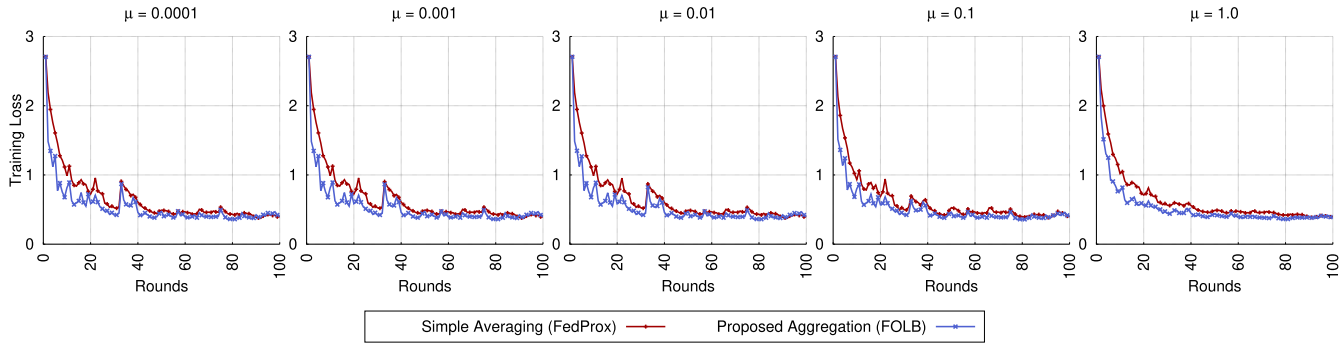
Each device participating in the federated learning process has a different communication delay when communicating with the central server and computation resources reserved for optimization. We model these two aspects as follows:

1) Communication Delay: For each device k, we assume that the time it takes for one round of communication between device k and central server is bounded above by T_k^c. This value T_k^c can be obtained with high confidence by taking the 99th percentile of the distribution used to model the communication delay, e.g. exponential distribution.

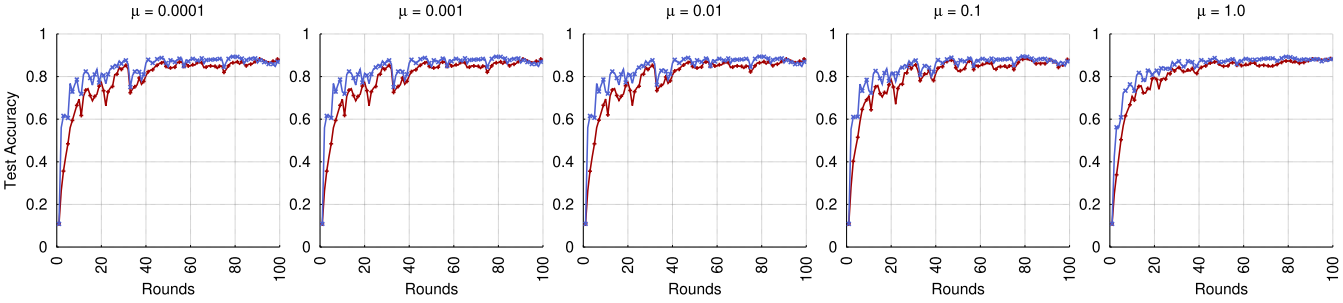
2) Computation Resources: Each device k can only reserve a certain amount of resources to carry out optimization of the local function $h_k(\mathbf{w}; \mathbf{w}^t)$. Thus, we relax our assumption of having an uniform γ-inexact local solver in all devices to allow each device to have particular γ_k-inexact local solver where γ_k can differ at every round of optimization and computed as $\gamma_k = \frac{\left\|\nabla h(\mathbf{w}_k^{t+1}, \mathbf{w}_k^t)\right\|}{\left\|\nabla h(\mathbf{w}_k^t, \mathbf{w}_k^t)\right\|}$. Note that we assume $\gamma_k \in [0, 1]$ as in the case of local solvers being gradient descent algorithm. Hence, let τ is the amount of time for an optimization round dictated by the central server, we allow each selected device k to perform any optimization within $\tau - T_k^c$ time and return the updated parameter $\mathbf{w}_k^{t+1}$ and γ_k back to the central server. This scheme allows great flexibility and practicality since a device can use any amount of resources available and any local optimization algorithm that it has access to at every round.

B. FOLB With Communication and Computation Heterogeneity

We show that FOLB can easily adapt to the inherent heterogeneity nature of communication and computation by adjusting it aggregation scheme to find a near-optimal convergence rate.



(a) Training loss (Lower is better.)



(b) Test accuracy (Higher is better.)

Fig. 3. Effectiveness of our proposed aggregation rule in FOLB compared to simple averaging in FedProx (similarly in FedAvg) across a wide range of proximal parameter μ.

1) New Loss Bound With Heterogeneity Presence: We first prove the following theorem showing the decrease of loss function in non-uniform FedProx with heterogeneity presence:

Theorem 3: With the same assumptions as in Theorem 1 and the presence of communication and computation heterogeneity, suppose that $\mathbf{w}^t$ *is not a stationary solution, in non-uniform FedProx, we have the following expected decrease in the global objective function*:

$$E[f(\mathbf{w}^{t+1})] \leq f(\mathbf{w}^t) - \frac{1}{K\mu} E\left[\sum_{k\in S_t}\left(\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle - B\left(\frac{L}{\mu\mu'}+\frac{1}{\mu}+\frac{3\ LB}{2\ K\mu'^2}\right)\gamma_k \left\|\nabla f(\mathbf{w}^t)\right\|^2\right)\right] + \left(\frac{LB^2}{2\mu'^2}+\frac{LB}{\mu\mu'}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2, \quad (15)$$

where the expectation is with respect to the choice of K *devices following probabilities* $P_k^t, k = 1, \ldots, N$.

2) Implications of Theorem 3: Theorem 3 states that in the presence of communication and computation heterogeneity, the bound of loss decrease at a round depends not only on the inner products between local and global gradients but also on the optimality of the solutions returned by the individual devices. In other words, a device is more beneficial to the global model if the following two conditions hold:

(1) The local gradient $\nabla F_k(\mathbf{w}^t)$ is well aligned with the global gradient $\nabla f(\mathbf{w}^t)$.
(2) It has enough resources to perform optimization to find a decent solution, i.e., small γ_k.

Both of these conditions are intuitive and reflecting the importance of each device during the learning process. Unfortunately, we cannot evaluate any of the two criteria before selecting devices without expensive prior communication and computation. However, we show that FOLB can handle these challenges easily by tweaking the aggregation rule.

3) Near-Optimal Selection Distribution: From Theorem 3, we can obtain a similar optimal selection probability distribution to that of Theorem 1 which focuses on devices with high values of $I_k^t = \langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle - B\left(\frac{L}{\mu\mu'}+\frac{1}{\mu}+\frac{3\ LB}{2\ K\mu'^2}\right)\gamma_k \left\|\nabla f(\mathbf{w}^t)\right\|^2$. In other word, a near-optimal distribution will select device k with probability:

$$P_{\text{lbh}_k}^t = \frac{|I_k^t|}{\sum_{k'=1}^N |I_{k'}^t|} \quad (16)$$

with the loss decrease satisfies:

$$E[f(\mathbf{w}^{t+1})] \leq f(\mathbf{w}^t) - \frac{1}{\mu}\sum_{k=1}^N \left(\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle - B\left(\frac{L}{\mu\mu'}+\frac{1}{\mu}+\frac{3\ LB}{2\ K\mu'^2}\right)\gamma_k \left\|\nabla f(\mathbf{w}^t)\right\|^2\right) P_{\text{lbh}_k}^t + \left(\frac{LB^2}{2\mu'^2}+\frac{LB}{\mu\mu'}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2. \quad (17)$$

FOLB aggregation for communication and computation heterogeneity. FOLB with heterogeneity of communication

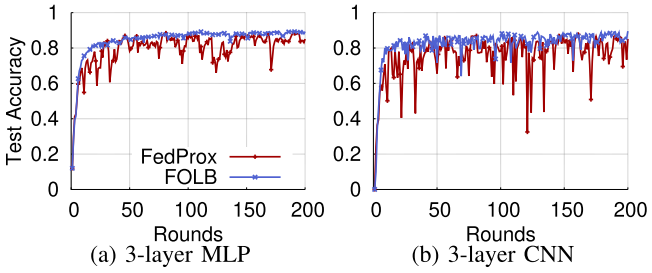


Fig. 4. Performance comparison between FOLB and FedProx considering different neural network models, i.e., CNN and MLP with 3 layers, over the MNIST dataset and $\mu = 0.01$. FOLB results in a more stable model accuracy and outperforms FedProx.

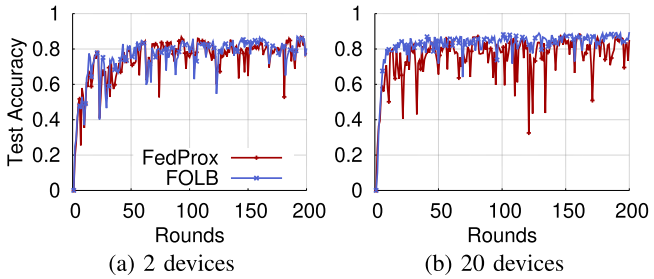


Fig. 5. Performance comparison between FOLB and FedProx considering different number of devices. With increasing the number of devices in each round, FOLB converges faster and stabilizes quicker than FedProx. We use MNIST dataset with a 3-layer CNN and $\mu = 0.01$.

and computation adopts the following aggregation rule:

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \sum_{k \in S_1^t} \frac{I_{1k}^t}{\sum_{k' \in S_1^t} |I_{1k'}^t|} \Delta \mathbf{w}_k^{t+1}, \tag{18}$$

where $I_{1k}^t = \langle \nabla_1\ f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t) \rangle - B \left(\frac{L}{\mu\mu'} + \frac{1}{\mu} + \frac{3\ LB}{2\ K\mu'^2} \right) \gamma_k \left\| \nabla_1\ f(\mathbf{w}^t) \right\|^2$, and $\nabla_1\ f(\mathbf{w}^t)$ is defined in (10).

4) Avoiding Constant Estimations: In the new FOLB that deals with heterogeneity, updating the global parameter according to Equation (18) becomes more complicated compared to Equation (13) due to involving the set of constants B, L, μ' which need to be estimated before hand or on-the-air. Instead of requiring all these constants to be estimated, we propose to use a hyper-parameter $\psi = B \left(\frac{L}{\mu\mu'} + \frac{1}{\mu} + \frac{3\ LB}{2\ K\mu'^2} \right)$ that will be learned though hyper-parameter tuning similarly to μ in FedProx. For tuning ψ, we can use a simple line search with a exponential step size, e.g. $\psi \in \{10^{-1}, 1, 10, 10^2\}$ which is used in our experiments and found to be effective.

VI. Experiments

In this section, we experimentally compare our proposed algorithm with existing state-of-the-art approaches and demonstrate faster convergence across different learning tasks in both synthetic and real datasets. We also confirm the advantage of taking into consideration the individual device optimization capability in the presence of communication and computation heterogeneity, showing our approach more suitable for practical federated learning implementations.

A. Experimental Settings

We first describe our setup of datasets, compared algorithms, testing environment and how statistical and system heterogeneity is simulated. We adopt closely the setup in a very recent work [21] on FedProx and provide details of their setup and the changes we made here for completeness.

1) Dataset: We use the a standard set of datasets used in multiple other works on federated learning [8], [21]. Particularly, we use 10-class *MNIST* [45], 62 class Federated Extended MNIST (*FEMNIST*) [46], and synthetic datasets [8], [21] to study with a multinomial logistic regression model, which extends the binary logistic regression model to multi-class scenarios and uses a different linear predictor function for each class to predict the probability that an observation belongs to that class. The synthetic datasets are generated with Gaussian distributions which are parameterized with a set of control parameters to vary the level of heterogeneity (see [8], [21] for more details). *Synthetic_iid* and *Synthetic_1_1* denote two datasets with no heterogeneity (i.i.d. distribution of data) and high heterogeneity, respectively. For non-convex setting, similarly to [20], [21], we consider a text sentiment analysis task on tweets using *Sent140* [47] dataset and next-character prediction task on the dataset of *The Complete Works of William Shakespeare* [20]. For MNIST, FEMNIST, sent140, and Shakespeare, we consider 1000, 200, 143, 772 devices, respectively. Particularly, for MNIST and FEMNIST datasets, the data is distributed on each device following a power law under the constraint that each device gets images from only two digits. For Sent140, each twitter account corresponds to one device, while in Shakespeare, each speaking role corresponds to one device.

2) Compared Algorithms: We compare FOLB with current state-of-the-art algorithms in the federated learning setting, including the recent FedProx [21] and the original FedAvg [20]. For both FOLB, FedProx and FedAvg, we use $k = 10$ devices in each round of optimization and investigate the effects of K on performance in a later set of experiments. For FedProx, we set $\mu = 1, 1, 1, 0.001,$ and 0.01, for 5 datasets respectively, as suggested in the original paper [21]. For our algorithm FOLB, we apply a similar line search on $\mu \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ and $\psi \in \{10^{-1}, 1, 10, 10^2\}$ when FOLB with heterogeneity consideration is tested. Here, we consider the versions of FOLB that only samples one set of devices in each round of optimization for communication efficiency, i.e., we use the aggregation scheme in (13) and (18). Thus, the communication cost of FOLB, FedAvg and FedProx are the same.

3) Computation and Communication Heterogeneity Simulation: For all algorithms, we simulate the computation and communication heterogeneity by allowing each device to pick a random number between 1 and 20 to be the number of gradient descent steps that the device is able to perform when selected. We initialize the same seed in all the compared algorithms to make sure that these numbers of gradient descents are consistent on all the algorithms. For FedProx and FedAvg, the received parameters from local devices in every round are simply averaged to get the new set of global model parameters.

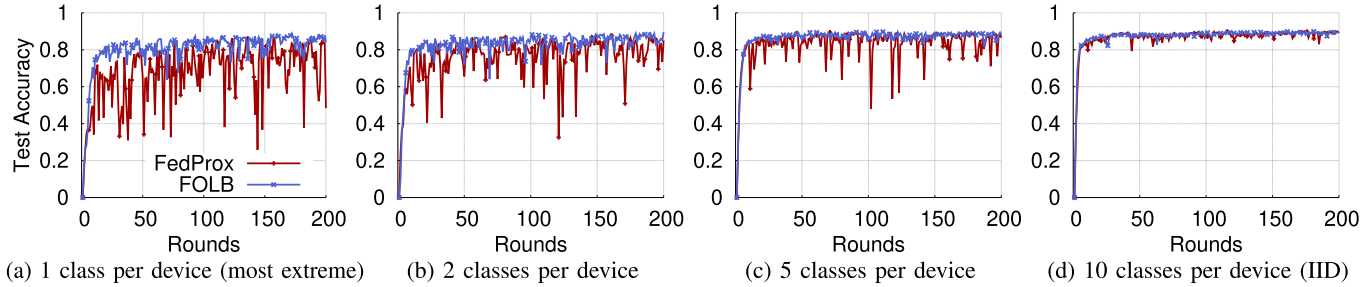


Fig. 6. Testing accuracy with different non-IID settings of the MNIST dataset, i.e., randomly assigning images of only a fixed number of different digits to each device. FOLB performs better than FedProx specially in the most extreme non-IID setting.

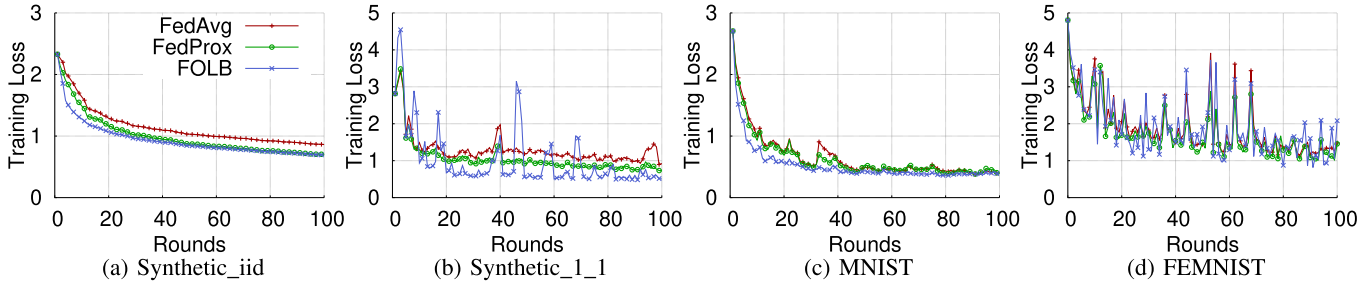


Fig. 7. Training loss of FOLB, FedProx and FedAvg on various datasets using linear model (multinomial logistic regression). FOLB can reach lower loss value than the others.

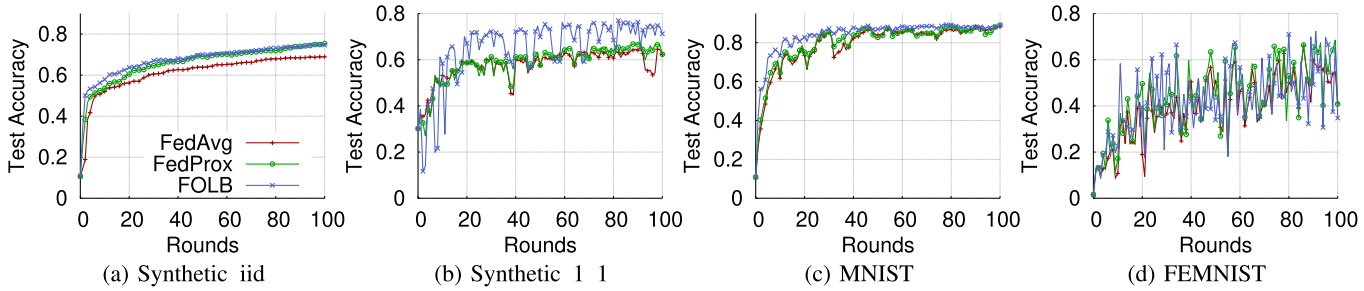


Fig. 8. Testing accuracy of FOLB, FedProx and FedAvg on various datasets using linear model (multinomial logistic regression). FOLB can reach higher level of accuracy than the others.

4) Environment: We performed all experiment on a $8 \times$ 2080Ti GPU cluster using TensorFlow [48] framework. Our codebase is based on the publicly available implementation of FedProx [21] approach.[2] For each dataset, we use stochastic gradient descent (SGD) as a local solver.

B. Experimental Results

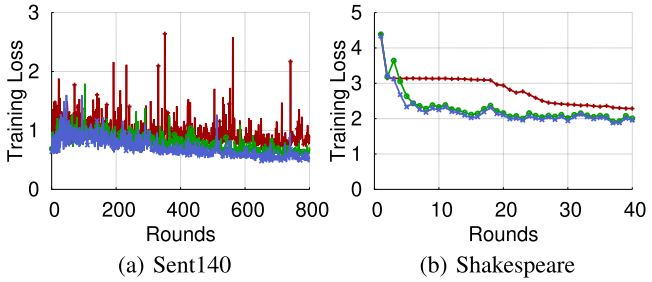
1) Quantifying the Effectiveness of the Proposed Aggregation Rule: We first compare our new aggregation rule with the simple averaging in FedProx (similarly in FedAvg). We vary μ with values from the set $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ in both FedProx and FOLB, and fix $\psi = 0$ in FOLB. The training loss and test accuracy on the first real dataset MNIST are shown in Fig. 3.

From Fig. 3, we observe the better performance of our proposed aggregation rule compared to that of simple averaging in FedProx (and similarly in FedAvg). Specifically, with FOLB, the loss value is always smaller than that of FedProx and its accuracy is higher than that of FedProx at the same time. This is especially significant in early iterations, showing faster convergence rate of FOLB. Our results prove the better effectiveness of our proposed aggregation scheme that principally aims at maximizing a lower-bound of loss decrease in every iteration (4).

Moreover, the better performance of our aggregation rule is more compelling with smaller values of μ. This observation again verifies the critical role of our lower-bound in (4) and our goal of maximizing it. Since maximizing the lower-bound leads to our approach of maximizing $E\Big[\sum_{k\in S_t}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle\Big]$, which is weighted by $\frac{1}{\mu}$ in (4), with smaller μ, the results of maximizing $E\Big[\sum_{k\in S_t}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle\Big]$ have bigger impact in maximizing the lower-bound. This observation of having better actual loss values draws a strong correlation between our lower-bound in (4) and the actual loss decrease and maximizing the lower-bound is sensible.

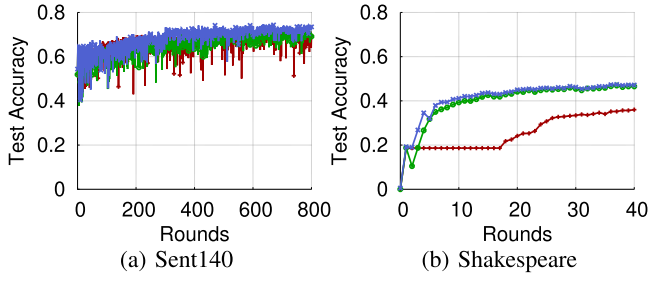
a) Experiments with different neural network models: In earlier experiments, we used a multinomial logistic regression model. Now we compare the performance of FOLB and

[2]https://github.com/litian96/FedProx



(a) Sent140 (b) Shakespeare

Fig. 9. Training loss of FOLB, FedProx and FedAvg on various datasets using non-linear model (LSTM). FOLB can reach lower loss value than the others.



(a) Sent140 (b) Shakespeare

Fig. 10. Testing accuracy of FOLB, FedProx and FedAvg on various datasets using non-linear model (LSTM). FOLB can reach higher level of accuracy than the others.

FedProx when using a Convolutional Neural Network (CNN) or Multi-Layer Perceptron (MLP) with 3 layers each. The results are illustrated in Fig. 4. We find that FOLB converges faster and is much more stable compared to FedProx.

b) Experiments with different number of devices in each round: We present the results with a varying number of devices participating in each round in Fig. 5. As expected, more devices make convergence stable and fast. However, we find this effect significantly better with FOLB compared to FedProx, thanks to our aggregation scheme. With a small number of devices in each round, FOLB is quite similar to FedProx since our aggregation scheme becomes closer to simple averaging.

c) Experiments with different non-IID settings: We simulate various non-IID scenarios on the MNIST dataset by only assigning random images from a fixed number of different digits to each device, i.e., 1, 2, 5, 10. For example, in the most extreme case, each device only has random images from only one digit. The results are demonstrated in Fig. 6 and show a recurring observation that FOLB outperforms FedProx, especially in the extreme cases of non-IID (common in reality).

2) Comparisons on Various Datasets and Models: We compare FOLB with FedProx and FedAvg algorithms. Figs. 7 and 8 present the training loss and test accuracy of all the algorithms on linear model (multinomial logistic regression) and Figs. 9 and 10 report results for non-linear model (LSTM). It is evident that FOLB consistently outperforms FedProx and FedAvg in terms of both reducing loss and improving accuracy. For example, on the Synthetic_1_1 dataset, FOLB is able to reach a low loss value and high accuracy level in only within 20 iterations while the other two methods never reach that level within 100 iterations and seem to converge at much higher loss and lower accuracy. On the other datasets, FOLB reduces loss value (and also increasing accuracy) faster than both FedProx and FedAvg, and can even reach lower loss and higher accuracy level than the other two algorithms.

In Table I, we report the number of optimization rounds that each algorithm needs to perform in order to reach a certain accuracy level (this is chosen based on the maximum accuracy that all three algorithms can reach on each dataset). We see that, usually FOLB only requires half number of rounds taken by FedProx andFedAvg to reach the same

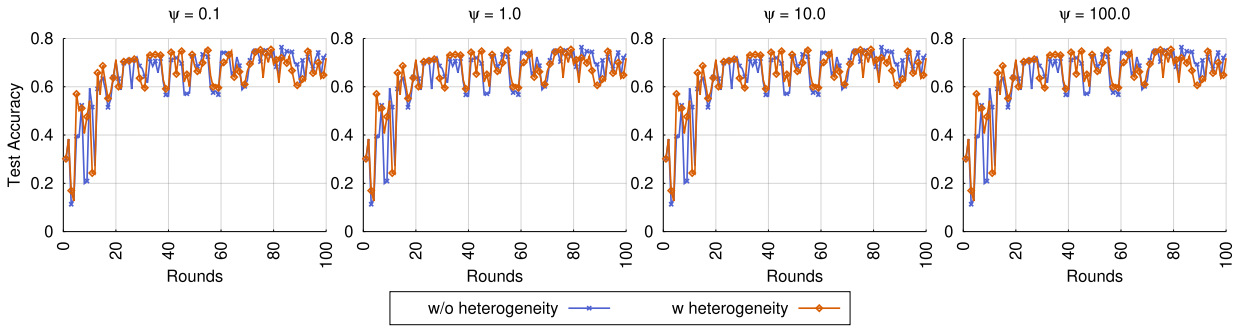
TABLE I

NUMBER OF ROUNDS OF EACH METHOD TO REACH A CERTAIN ACCURACY LEVEL ON EACH DATASET (NOTE THAT ON SHAKESPEARE, FedAvg FAILED TO REACH THE GIVEN ACCURACY WITHIN 40 ROUNDS)

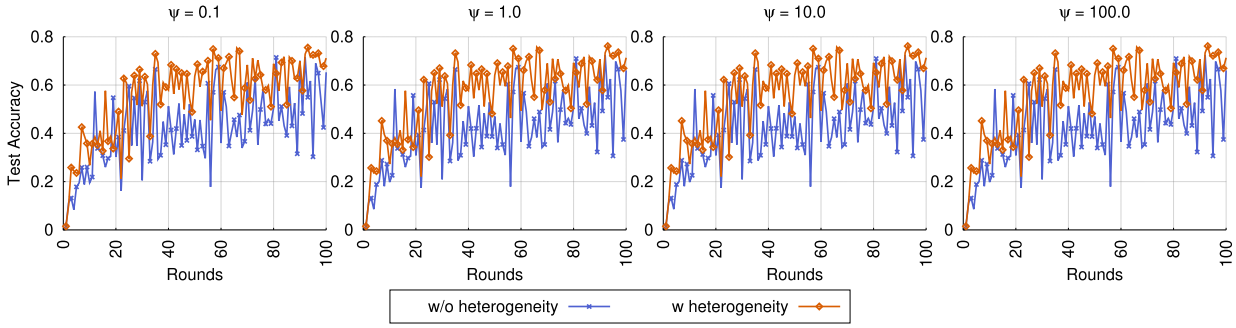
Methods	Accuracy	FOLB	FedProx	FedAvg
Synthetic_iid	70%	**50**	57	113
Synthetic_1_1	70%	**19**	154	177
MNIST	80%	**11**	25	25
FEMNIST	65%	**34**	58	86
Sent140	65%	**31**	132	82
Shakespeare	45%	**20**	25	–

level of accuracy. For example, on Synthetic_1_1 dataset, FOLB only needs 19 rounds while FedProx and FedAvg require 154 and 177 rounds respectively. One exception is on Synthetic_iid where data is independent and identically distributed across different devices, however, FOLB still need fewer rounds than FedProx and FedAvg. Note that due to computation heterogeneity, even on Synthetic_iid, FedAvg performs poorly compared to FedProx and FOLB which directly address heterogeneity. These results again verify the faster convergence rate of FOLB compared to FedProx and FedAvg.

3) FOLB *With and Without Communication and Computation Heterogeneity Consideration:* In this last set of experiments, we compare FOLB with different aggregation rules, i.e., (13) and (18) which are corresponding to before and after taking into account the heterogeneity of communication and computation respectively. Fig. 11 shows the test accuracy of these two variants on Synthetic_1_1 and EMNIST, where the performance of FOLB varies the most (Fig. 8) and with different values of ψ which controls how much heterogeneity contributes in computing aggregation weight of each local update in (18). The results show that by taking into account the inherent heterogeneity, FOLB is more stable than the other variant. In particular, with heterogeneity, FOLB is able to avoid most major drops in accuracy and stays at high accuracy level toward later iterations without any significant fluctuations. On the other hand, the vanilla FOLB can reach high accuracy but fluctuates widely even in later iterations. In addition, from Fig. 11, ψ can take value in a wide range, i.e., $[0.1, 10]$ and still helps stabilize FOLB well.



(a) Synthetic_1_1



(b) FEMNIST

Fig. 11. Accuracy of FOLB with and without heterogeneity consideration. Heterogeneity-aware FOLB avoids major drops of accuracy between iterations and is more robust than vanilla FOLB.

VII. Conclusion

In this work, we have introduced FOLB - a fast-convergent federated learning algorithm, and shown that FOLB theoretically achieves a near-optimal possible lower-bound for the overall loss decrease at every round of communication/optimization. FOLB encloses a novel adaptive aggregation scheme that takes into account both statistical and system heterogeneity inherent in the modern networking environments of massively distributed mobile devices. More importantly, we have shown that across different tasks and datasets, FOLB significantly reduces the number of rounds to reach a certain level of loss value and accuracy.

For future work, a promising direction is to study a device selection methodology that couples decisions across multiple time periods to bring greater performance gains in the long term. This involves deriving new lower-bound that reflects the performance after a number of optimization rounds and taking into account the communication and computation heterogeneity in all those rounds.

Appendix

A. Proof of Theorem 1

Proof: From the L-Lipschitz continuity of f, we have

$$f(\mathbf{w}^{t+1}) \leq f(\mathbf{w}^t) + \langle \nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \rangle + \frac{L}{2} \left\| \mathbf{w}^{t+1} - \mathbf{w}^t \right\|^2 \tag{19}$$

We will separately bound the last two terms on the right-hand side of the above inequality:

- *Bounding* $\left\| \mathbf{w}^{t+1} - \mathbf{w}^t \right\|$: Let $\hat{\mathbf{w}}_k^{t+1} = \arg\min_{\mathbf{w}} h_k(\mathbf{w}, \mathbf{w}^t)$. Due to the μ'-strong convexity of $h_k(\mathbf{w}, \mathbf{w}^t)$ and the γ-inexact local solver assumption for $\mathbf{w}_k^{t+1}$, we have:

$$\begin{aligned} \left\| \hat{\mathbf{w}}_k^{t+1} - \mathbf{w}_k^{t+1} \right\| &\leq \frac{1}{\mu'} \left\| \nabla h(\hat{\mathbf{w}}_k^{t+1}, \mathbf{w}^t) - \nabla h(\mathbf{w}_k^{t+1}, \mathbf{w}^t) \right\| \\ &\leq \frac{\gamma}{\mu'} \left\| \nabla F_k(\mathbf{w}^t) \right\|, \end{aligned} \tag{20}$$

and, similarly,

$$\left\| \hat{\mathbf{w}}_k^{t+1} - \mathbf{w}^t \right\| \leq \frac{1}{\mu'} \left\| \nabla F_k(\mathbf{w}^t) \right\|. \tag{21}$$

Hence, by the triangle inequality and B-dissimilarity of ∇F_k, we obtain:

$$\left\| \mathbf{w}_k^{t+1} - \mathbf{w}^t \right\| \leq \frac{1+\gamma}{\mu'} \left\| \nabla F_k(\mathbf{w}^t) \right\| \leq \frac{B(1+\gamma)}{\mu'} \left\| \nabla f(\mathbf{w}^t) \right\|. \tag{22}$$

Now, noting $\mathbf{w}^{t+1} = \frac{1}{K} \sum_{k \in S_t} \mathbf{w}_k^{t+1}$, we can write

$$\begin{aligned} \left\| \mathbf{w}^{t+1} - \mathbf{w}^t \right\|^2 &\leq \left(\frac{1}{K} \sum_{k \in S_t} \left\| \mathbf{w}_k^{t+1} - \mathbf{w}^t \right\| \right)^2 \\ &\leq \frac{B^2(1+\gamma)^2}{\mu'^2} \left\| \nabla f(\mathbf{w}^t) \right\|^2, \end{aligned} \tag{23}$$

where the first inequality follows from the Cauchy-Schwarz's inequality, and the second follows from applying (22) to each k in the sum.

- *Bounding* $\langle \nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \rangle$: By definition of the aggregation step for $\mathbf{w}^{t+1}$, we can write

$$\langle \nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \rangle = \frac{1}{K} \sum_{k \in S_t} \langle \nabla f(\mathbf{w}^t), \mathbf{w}_k^{t+1} - \mathbf{w}^t \rangle. \tag{24}$$

For each term in the sum, we can express

$$\mathbf{w}_k^{t+1} - \mathbf{w}^t = -\frac{1}{\mu}\nabla F_k(\mathbf{w}^t) + \frac{1}{\mu}(\nabla F_k(\mathbf{w}^t) - \nabla F_k(\mathbf{w}_k^{t+1})) + \frac{1}{\mu}(\nabla F_k(\mathbf{w}_k^{t+1}) + \mu(\mathbf{w}_k^{t+1} - \mathbf{w}^t)).$$

Thus,

$$\begin{aligned}
&\langle \nabla f(\mathbf{w}^t), \mathbf{w}_k^{t+1} - \mathbf{w}^t \rangle \\
&= -\frac{1}{\mu}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle \\
&\quad + \frac{1}{\mu}\langle \nabla f(\mathbf{w}^t),\ \nabla F_k(\mathbf{w}^t) - \nabla F_k(\mathbf{w}_k^{t+1})\rangle \\
&\quad + \frac{1}{\mu}\langle \nabla f(\mathbf{w}^t),\ \nabla F_k(\mathbf{w}_k^{t+1}) + \mu(\mathbf{w}_k^{t+1} - \mathbf{w}^t)\rangle \\
&\leq -\frac{1}{\mu}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle \\
&\quad + \frac{1}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\| \left\|\nabla F_k(\mathbf{w}^t) - \nabla F_k(\mathbf{w}_k^{t+1})\right\| \\
&\quad + \frac{1}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\| \left\|\nabla F_k(\mathbf{w}_k^{t+1}) + \mu(\mathbf{w}_k^{t+1} - \mathbf{w}^t)\right\|,
\end{aligned}$$

where the inequality follows again from Cauchy-Schwarz. Noting that $\left\|\nabla F_k(\mathbf{w}^t) - \nabla F_k(\mathbf{w}_k^{t+1})\right\| \leq L\left\|\mathbf{w}_k^{t+1} - \mathbf{w}^t\right\|$ by Assumption 1, and that $\nabla F_k(\mathbf{w}_k^{t+1}) + \mu(\mathbf{w}_k^{t+1} - \mathbf{w}^t) = \nabla h(\mathbf{w}_k^{t+1}, \mathbf{w}^t)$ by definition, we have

$$\begin{aligned}
&\langle \nabla f(\mathbf{w}^t), \mathbf{w}_k^{t+1} - \mathbf{w}^t \rangle \\
&\leq -\frac{1}{\mu}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle \\
&\quad + \frac{LB(1+\gamma)}{\mu\mu'}\left\|\nabla f(\mathbf{w}^t)\right\|^2 + \frac{B\gamma}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\|^2, \quad (25)
\end{aligned}$$

where we have applied (22) to the middle term, and Assumptions 4&2 to the last term on the right hand side. Combining this with (24), we have

$$\begin{aligned}
&\langle \nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \rangle \\
&= \frac{1}{K}\sum_{k\in S_t}\langle \nabla f(\mathbf{w}^t), \mathbf{w}_k^{t+1} - \mathbf{w}^t\rangle \\
&\leq -\frac{1}{K\mu}\sum_{k\in S_t}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle \\
&\quad + \frac{B}{\mu}\Big(\frac{L(\gamma+1)}{\mu'} + \gamma\Big)\left\|\nabla f(\mathbf{w}^t)\right\|^2. \quad (26)
\end{aligned}$$

Substituting (23) and (26) into (19) and taking the expectation, we obtain

$$\begin{aligned}
&\mathbb{E}[f(\mathbf{w}^{t+1})] \\
&\leq f(\mathbf{w}^t) - \frac{1}{K\mu}\mathbb{E}\left[\sum_{k\in S_t}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle\right] \\
&\quad + B\left(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2,
\end{aligned}$$

where the first and last terms on the right hand side are not written in expectation as they do not depend on the selection of devices in round t. □

B. Proof of Proposition 1

Proof: The key difference from Theorem 1's proof is in the decomposition of $\langle \nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t\rangle$ in (26). In this case, we write

$$\begin{aligned}
&\langle \nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t \rangle \\
&= \frac{1}{K}\Big[\sum_{k\in S_t^+}\langle \nabla f(\mathbf{w}^t), \mathbf{w}_k^{t+1} - \mathbf{w}^t\rangle \\
&\quad + \sum_{k\in S_t^-}\langle \nabla f(\mathbf{w}^t), \mathbf{w}^t - \mathbf{w}_k^{t+1}\rangle\Big], \quad (27)
\end{aligned}$$

where $S_t^+ = \{k \in S_t : \langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle \geq 0\}$ and $S_t^- = \{k \in S_t : \langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle < 0\}$. For $k \in S_t^+$, the derivation follows (25). On the other hand, for $k \in S_t^-$,

$$\begin{aligned}
&\langle \nabla f(\mathbf{w}^t), \mathbf{w}^t - \mathbf{w}_k^{t+1} \rangle \\
&\leq \frac{1}{\mu}\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle \\
&\quad - \frac{1}{\mu}\langle \nabla f(\mathbf{w}^t),\ \nabla F_k(\mathbf{w}^t) - \nabla F_k(\mathbf{w}_k^{t+1})\rangle \\
&\quad - \frac{1}{\mu}\langle \nabla f(\mathbf{w}^t),\ \nabla F_k(\mathbf{w}_k^{t+1}) + \mu(\mathbf{w}_k^{t+1} - \mathbf{w}^t)\rangle \\
&\leq -\frac{1}{\mu}|\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle| \\
&\quad + \frac{1}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\| \left\|\nabla F_k(\mathbf{w}^t) - \nabla F_k(\mathbf{w}_k^{t+1})\right\| \\
&\quad + \frac{1}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\| \left\|\nabla F_k(\mathbf{w}_k^{t+1}) + \mu(\mathbf{w}_k^{t+1} - \mathbf{w}^t)\right\| \\
&\leq -\frac{1}{\mu}|\langle \nabla f(\mathbf{w}^t), \nabla F_k(\mathbf{w}^t)\rangle| \\
&\quad + \frac{LB(1+\gamma)}{\mu\mu'}\left\|\nabla f(\mathbf{w}^t)\right\|^2 + \frac{B\gamma}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\|^2.
\end{aligned}$$

Substituting these expressions in (27) gives the result. □

C. Proof of Lemma 1

We sequentially prove the two statements in the following:

Proof of Eq. (11): We expand $\sum_{k\in S_1^t}\langle \nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle^2$ as follows:

$$\begin{aligned}
&\sum_{k\in S_1^t}\langle \nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle^2 \\
&= \frac{1}{K^2}\sum_{k\in S_1^t}\Big(\sum_{k'\in S_1^t}\langle \nabla F_k(\mathbf{w}^t), \nabla F_{k'}(\mathbf{w}^t)\rangle\Big)^2 \\
&= \frac{1}{K^2}\sum_{k,k',k''\in S_1^t}\langle \nabla F_k(\mathbf{w}^t), \nabla F_{k'}(\mathbf{w}^t)\rangle\langle \nabla F_k(\mathbf{w}^t), \nabla F_{k''}(\mathbf{w}^t)\rangle
\end{aligned}$$

Since $|S_1^t| = K$, the summation in the last equality has K^3 terms. Across all possible multisets S_1^t, there are N^3 possible combinations of k, k', k''. Since device selection in Algorithm 2 occurs uniformly at random, each combination

k, k', k'' has the same probability of appearing in the summation. Therefore, we can write the expectation as a summation over all combinations of three devices from $[N] = \{1, \ldots, N\}$, and simplify the result as follows:

$$\mathbb{E}\Big[\sum_{k\in S_1^t}\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle^2\Big]$$
$$= \frac{K^3}{K^2\ N^3}\sum_{k,k',k''}\langle\nabla F_k(\mathbf{w}^t), \nabla F_{k'}(\mathbf{w}^t)\rangle\langle\nabla F_k(\mathbf{w}^t), \nabla F_{k''}(\mathbf{w}^t)\rangle$$
$$= \frac{K}{N^3}\sum_{k\in[N]}\Big(\sum_{k'\in[N]}\langle\nabla F_k(\mathbf{w}^t), \nabla F_{k'}(\mathbf{w}^t)\rangle\Big)^2$$
$$= \frac{K}{N}\sum_{k\in[N]}\Big(\langle\nabla F_k(\mathbf{w}^t),\ \frac{1}{N}\sum_{k'\in[N]}\nabla F_{k'}(\mathbf{w}^t)\rangle\Big)^2$$
$$= \frac{K}{N}\sum_{k\in[N]}\Big(\langle\nabla F_k(\mathbf{w}^t), \nabla f(\mathbf{w}^t)\rangle\Big)^2,$$

where the last step follows from the definition of $\nabla f(\mathbf{w}^t) = \frac{1}{N}\sum_{k\in[N]}\nabla F_k(\mathbf{w}^t)$. □

Proof of Eq. (12): By definition of $\nabla_2\ f(\mathbf{w}^t)$, we have

$$\sum_{k'\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle = \frac{1}{K}\sum_{k',k''\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla F_{k''}(\mathbf{w}^t)\rangle.$$

Then, similar to the proof of Eq. (11), we can write the expectation as a summation over all possible combinations of device pairs, and simplify:

$$\mathbb{E}\Big[\sum_{k'\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle\Big]$$
$$= \frac{K^2}{KN^2}\sum_{k',k''\in[N]}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla F_{k''}(\mathbf{w}^t)\rangle$$
$$= \frac{K}{N}\sum_{k'\in[N]}\langle\nabla F_{k'}(\mathbf{w}^t),\ \frac{1}{N}\sum_{k''\in[N]}\nabla F_{k''}(\mathbf{w}^t)\rangle$$
$$= \frac{K}{N}\sum_{k'\in[N]}\langle\nabla F_{k'}(\mathbf{w}^t), f(\mathbf{w}^t)\rangle$$
$$\leq \frac{K}{N}\sum_{k'\in[N]}|\langle\nabla F_{k'}(\mathbf{w}^t), f(\mathbf{w}^t)\rangle|.$$

That complete the proof. □

D. Proof of Theorem 2

Proof: As in Theorem 1, we begin with the L-Lipschitz inequality for $f(\mathbf{w}^{t+1})$ given in (19), and bound the last two terms on the right-hand side:

• *Bounding* $\|\mathbf{w}^{t+1} - \mathbf{w}^t\|$: In (9), define

$$\hat{P}_k^t = \frac{\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle}{\sum_{k'\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle}, \tag{28}$$

i.e., an approximation of the LB-near-optimal selection probability in (7). Following the procedure for this bound in Theorem 1, for the update rule (9) of FOLB, we can write

$$\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 \leq \Big(\sum_{k\in S_1^t}\hat{P}_k^t\|\mathbf{w}_k^{t+1} - \mathbf{w}^t\|\Big)^2 \leq \Big(\sum_{k\in S_1^t}\hat{P}_k^t\Big)^2\frac{B^2(1+\gamma)^2}{\mu'^2}\|\nabla f(\mathbf{w}^t)\|^2. \tag{29}$$

• *Bounding* $\langle\nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t\rangle$: Similar to the procedure for this bound in Theorem 1, we can write

$$\langle\nabla f(\mathbf{w}^t), \mathbf{w}^{t+1} - \mathbf{w}^t\rangle = \sum_{k\in S_1^t}\hat{P}_k^t\langle\nabla f(\mathbf{w}^t), \mathbf{w}_k^{t+1} - \mathbf{w}^t\rangle \leq -\frac{1}{\mu}\sum_{k\in S_1^t}\hat{P}_k^t\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle + \sum_{k\in S_1^t}\hat{P}_k^t\frac{B}{\mu}\Big(\frac{L(\gamma+1)}{\mu'} + \gamma\Big)\|\nabla f(\mathbf{w}^t)\|^2, \tag{30}$$

where the equality follows from the FOLB aggregation, and the inequality follows from (25).

Now, substituting (29) and (30) into (19), we have

$$f(\mathbf{w}^{t+1}) \leq f(\mathbf{w}^t) - \frac{1}{\mu}\sum_{k\in S_1^t}\hat{P}_k^t\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle + \sum_{k\in S_1^t}\hat{P}_k^t\frac{B}{\mu}\Big(\frac{L(\gamma+1)}{\mu'} + \gamma\Big)\|\nabla f(\mathbf{w}^t)\|^2 + \Big(\sum_{k\in S_1^t}\hat{P}_k^t\Big)^2\frac{B^2(1+\gamma)^2}{\mu'^2}\|\nabla f(\mathbf{w}^t)\|^2.$$

Note that, with random selection of S_1^t and S_2^t, we can define two random variables $\sum_{k\in S_1^t}\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle$ and $\sum_{k'\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle$ which follow the same distribution and $E\big[\sum_{k\in S_1^t}\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle\big] = E\big[\sum_{k'\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle\big]$. Taking expectation with respect to the uniformly random selection of devices in the two sets S_1^t and S_2^t, and using Taylor's expansion give us

$$E[f(\mathbf{w}^{t+1})] \leq f(\mathbf{w}^t) - \frac{1}{\mu}E\Big[\sum_{k\in S_1^t}\hat{P}_k^t\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle\Big] + B\Big(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\Big)\|\nabla f(\mathbf{w}^t)\|^2.$$

Since S_1^t and S_2^t are independent sets of random devices, the above inequality is equivalent to

$$\mathbb{E}[f(\mathbf{w}^{t+1})] \leq f(\mathbf{w}^t) - \frac{1}{\mu}\frac{\mathbb{E}\Big[\sum_{k\in S_1^t}\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle\langle\nabla F_k(\mathbf{w}^t), \nabla_1\ f(\mathbf{w}^t)\rangle\Big]}{\mathbb{E}\Big[\sum_{k'\in S_2^t}\langle\nabla F_{k'}(\mathbf{w}^t), \nabla_2\ f(\mathbf{w}^t)\rangle\Big]} + B\Big(\frac{L(\gamma+1)}{\mu\mu'} + \frac{\gamma}{\mu} + \frac{BL(1+\gamma)^2}{2\mu'^2}\Big)\|\nabla f(\mathbf{w}^t)\|^2. \tag{31}$$

In the term with expectations, we can apply Eq. (11) and (12) from Lemma 1 to the numerator and

denominator, respectively, giving

$$\begin{aligned}&\mathbb{E}[f(\mathbf{w}^{t+1})]\\&\leq f(\mathbf{w}^t)-\frac{1}{\mu}\frac{\sum_{k\in[N]}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle^2}{\sum_{k'\in[N]}|\langle\nabla f(\mathbf{w}^t),\nabla F_{k'}(\mathbf{w}^t)\rangle|}\\&\quad+B\Big(\frac{L(\gamma+1)}{\mu\mu'}+\frac{\gamma}{\mu}+\frac{BL(1+\gamma)^2}{2\mu'^2}\Big)\left\|\nabla f(\mathbf{w}^t)\right\|^2,\end{aligned}\tag{32}$$

which is equivalent to (8). □

E. Proof of Theorem 3

Proof: From the L-Lipschitz continuity of f, we have

$$\begin{aligned}f(\mathbf{w}^{t+1})\leq f(\mathbf{w}^t)+\langle\nabla f(\mathbf{w}^t),\mathbf{w}^{t+1}-\mathbf{w}^t\rangle\\+\frac{L}{2}\left\|\mathbf{w}^{t+1}-\mathbf{w}^t\right\|^2.\end{aligned}\tag{33}$$

We will bound the last two terms in the right-hand side of the above inequality as follows:

• *Bound* $\left\|\mathbf{w}^{t+1}-\mathbf{w}^t\right\|$: Similar to the proof of Theorem 1, we derive the following bound:

$$\begin{aligned}\left\|\mathbf{w}^{t+1}-\mathbf{w}^t\right\|^2&\leq\Big(\frac{1}{K}\sum_{k\in S_t}\left\|\mathbf{w}_k^{t+1}-\mathbf{w}^t\right\|\Big)^2\\&\leq\frac{B^2}{K^2\mu'^2}\Big(\sum_{k\in S^t}(1+\gamma_k)\Big)^2\left\|\nabla f(\mathbf{w}^t)\right\|^2.\end{aligned}\tag{34}$$

• *Bound* $\langle\nabla f(\mathbf{w}^t),\mathbf{w}^{t+1}-\mathbf{w}^t\rangle$: Following the similar steps in the proof of Theorem 1, we obtain the following:

$$\begin{aligned}\langle\nabla f(\mathbf{w}^t),\mathbf{w}_k^{t+1}-\mathbf{w}^t\rangle&\leq-\frac{1}{\mu}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle\\&\quad+\frac{1}{\mu}\langle\nabla f(\mathbf{w}^t),(\nabla F_k(\mathbf{w}^t)-\nabla F_k(\mathbf{w}_k^{t+1}))\rangle\\&\quad+\frac{1}{\mu}\langle\nabla f(\mathbf{w}^t),(\nabla F_k(\mathbf{w}_k^{t+1})+\mu(\mathbf{w}_k^{t+1}-\mathbf{w}^t))\rangle\\&\leq-\frac{1}{\mu}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle\\&\quad+\frac{1}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\|\left\|\nabla F_k(\mathbf{w}^t)-\nabla F_k(\mathbf{w}_k^{t+1})\right\|\\&\quad+\frac{1}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\|\left\|(\nabla F_k(\mathbf{w}_k^{t+1})+\mu(\mathbf{w}_k^{t+1}-\mathbf{w}^t))\right\|\\&\leq-\frac{1}{\mu}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle+\frac{LB(1+\gamma_k)}{\mu\mu'}\left\|\nabla f(\mathbf{w}^t)\right\|^2\\&\quad+\frac{B\gamma_k}{\mu}\left\|\nabla f(\mathbf{w}^t)\right\|^2,\\&\leq-\frac{1}{\mu}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle+\frac{LB}{\mu\mu'}\left\|\nabla f(\mathbf{w}^t)\right\|^2\\&\quad+\frac{B}{\mu}\left(\frac{L}{\mu'}+1\right)\gamma_k\left\|\nabla f(\mathbf{w}^t)\right\|^2,\end{aligned}\tag{35}$$

and, consequently,

$$\begin{aligned}&\langle\nabla f(\mathbf{w}^t),\mathbf{w}^{t+1}-\mathbf{w}^t\rangle\\&=\frac{1}{K}\sum_{k\in S_t}\langle\nabla f(\mathbf{w}^t),\mathbf{w}_k^{t+1}-\mathbf{w}^t\rangle\\&\leq-\frac{1}{K\mu}\sum_{k\in S_t}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle\\&\quad+\frac{B}{\mu}\sum_{k\in S^t}\left(\frac{L}{\mu'}+1\right)\gamma_k\left\|\nabla f(\mathbf{w}^t)\right\|^2+\frac{LB}{\mu\mu'}\left\|\nabla f(\mathbf{w}^t)\right\|^2.\end{aligned}\tag{36}$$

Combine (33), (34) and (36), we obtain:

$$\begin{aligned}f(\mathbf{w}^{t+1})&\leq f(\mathbf{w}^t)-\frac{1}{K\mu}\sum_{k\in S_t}\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle\\&\quad+B\sum_{k\in S_1^t}\left(\frac{L}{\mu\mu'}+\frac{1}{\mu}+\frac{LB}{K\mu'^2}\right)\gamma_k\left\|\nabla f(\mathbf{w}^t)\right\|^2\\&\quad+\frac{LB^2}{2\ K^2\mu'^2}\left(\sum_{k\in S^t}\gamma_k\right)^2\left\|\nabla f(\mathbf{w}^t)\right\|^2\\&\quad+\left(\frac{LB^2}{2\mu'^2}+\frac{LB}{\mu\mu'}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2.\end{aligned}\tag{37}$$

Thus,

$$\begin{aligned}E[f(\mathbf{w}^{t+1})]&\leq f(\mathbf{w}^t)-\frac{1}{K\mu}E\Bigg[\sum_{k\in S_t}\Bigg(\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle\\&\quad-B\left(\frac{L}{\mu\mu'}+\frac{1}{\mu}+\frac{LB}{K\mu'^2}\right)\gamma_k\left\|\nabla f(\mathbf{w}^t)\right\|^2\\&\quad-\frac{LB^2}{2\ K^2\mu'^2}\sum_{k'\in S^t}\gamma_{k'}\gamma_k\left\|\nabla f(\mathbf{w}^t)\right\|^2\Bigg)\Bigg]\\&\quad+\left(\frac{LB^2}{2\mu'^2}+\frac{LB}{\mu\mu'}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2\\&\leq-\frac{1}{K\mu}E\Bigg[\sum_{k\in S_t}\Bigg(\langle\nabla f(\mathbf{w}^t),\nabla F_k(\mathbf{w}^t)\rangle\\&\quad-B\left(\frac{L}{\mu\mu'}+\frac{1}{\mu}+\frac{3\ LB}{2\ K\mu'^2}\right)\gamma_k\left\|\nabla f(\mathbf{w}^t)\right\|^2\Bigg)\Bigg]\\&\quad+\left(\frac{LB^2}{2\mu'^2}+\frac{LB}{\mu\mu'}\right)\left\|\nabla f(\mathbf{w}^t)\right\|^2.\end{aligned}\tag{38}$$

This completes the proof. □

REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[2] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[3] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1223–1231.

[4] O. Dekel, R. Gilad-Bachrach, O. Shamir, and L. Xiao, "Optimal distributed online prediction using mini-batches," *J. Mach. Learn. Res.*, vol. 13, pp. 165–202, Jan. 2012.

[5] M. Li, "Scaling distributed machine learning with the parameter server," in *Proc. Int. Conf. Big Data Sci. Comput.*, 2014, pp. 19–27.

[6] S. J. Reddi, J. Konený, P. Richtárik, B. Póczós, and A. Smola, "AIDE: Fast and communication efficient distributed optimization," 2016, *arXiv:1608.06879*. [Online]. Available: http://arxiv.org/abs/1608.06879

[7] P. Richtárik and M. Takáč, "Distributed coordinate descent method for learning with big data," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2657–2681, 2016.

[8] A. Ghosh, R. K. Maity, A. Mazumdar, and K. Ramchandran, "Communication efficient distributed approximate Newton method," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2020, pp. 1000–1008.

[9] V. Smith, S. Forte, C. Ma, M. Takac, M. I. Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," 2016, *arXiv:1611.02189*. [Online]. Available: http://arxiv.org/abs/1611.02189

[10] S. Zhang, A. E. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 685–693.

[11] Y. Zhang, J. C. Duchi, and M. J. Wainwright, "Communication-efficient algorithms for statistical optimization," in *Proc. IEEE 51st IEEE Conf. Decision Control (CDC)*, Dec. 2012, pp. 1502–1510.

[12] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," 2018, *arXiv:1807.06629*. [Online]. Available: http://arxiv.org/abs/1807.06629

[13] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 2525–2536.

[14] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," 2019, *arXiv:1905.03817*. [Online]. Available: http://arxiv.org/abs/1905.03817

[15] S. Wang *et al.*, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, Jun. 2019.

[16] T. Lin, S. U. Stich, K. Kshitij Patel, and M. Jaggi, "Don't use large mini-batches, use local SGD," 2018, *arXiv:1808.07217*. [Online]. Available: http://arxiv.org/abs/1808.07217

[17] S. U. Stich, "Local SGD converges fast and communicates little," 2018, *arXiv:1805.09767*. [Online]. Available: http://arxiv.org/abs/1805.09767

[18] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms," 2018, *arXiv:1808.07576*. [Online]. Available: http://arxiv.org/abs/1808.07576

[19] B. E. Woodworth, J. Wang, A. Smith, B. McMahan, and N. Srebro, "Graph oracle models, lower bounds, and gaps for parallel stochastic optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2018, pp. 8496–8506.

[20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[21] T. Li, A. Kumar Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2018, *arXiv:1812.06127*. [Online]. Available: http://arxiv.org/abs/1812.06127

[22] C. T. Dinh *et al.*, "Federated learning over wireless networks: Convergence analysis and resource allocation," 2019, *arXiv:1910.13067*. [Online]. Available: http://arxiv.org/abs/1910.13067

[23] S. Reddi *et al.*, "Adaptive federated optimization," 2020, *arXiv:2003.00295*. [Online]. Available: http://arxiv.org/abs/2003.00295

[24] S. Praneeth Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. Theertha Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," 2019, *arXiv:1910.06378*. [Online]. Available: http://arxiv.org/abs/1910.06378

[25] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," 2019, *arXiv:1907.02189*. [Online]. Available: http://arxiv.org/abs/1907.02189

[26] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[27] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1175–1191.

[28] A. Bhowmick, J. Duchi, J. Freudiger, G. Kapoor, and R. Rogers, "Protection against reconstruction and its applications in private federated learning," 2018, *arXiv:1812.00984*. [Online]. Available: http://arxiv.org/abs/1812.00984

[29] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "CpSGD: Communication-efficient and differentially-private distributed SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2018, pp. 7564–7575.

[30] B. Ghazi, R. Pagh, and A. Velingker, "Scalable and differentially private distributed aggregation in the shuffled model," 2019, *arXiv:1906.08320*. [Online]. Available: http://arxiv.org/abs/1906.08320

[31] Z. Liu, T. Li, V. Smith, and V. Sekar, "Enhancing the privacy of federated learning with sketching," 2019, *arXiv:1911.01812*. [Online]. Available: http://arxiv.org/abs/1911.01812

[32] T. Li, M. Sanjabi, A. Beirami, and V. Smith, "Fair resource allocation in federated learning," 2019, *arXiv:1905.10497*. [Online]. Available: http://arxiv.org/abs/1905.10497

[33] Y.-S. Jeon, M. Mohammadi Amiri, J. Li, and H. Vincent Poor, "A compressive sensing approach for federated learning over massive MIMO communication systems," 2020, *arXiv:2003.08059*. [Online]. Available: http://arxiv.org/abs/2003.08059

[34] M. Mohammadi Amiri, D. Gunduz, S. R. Kulkarni, and H. Vincent Poor, "Convergence of update aware device scheduling for federated learning at the wireless edge," 2020, *arXiv:2001.10402*. [Online]. Available: http://arxiv.org/abs/2001.10402

[35] M. Chen, Z. Yang, W. Saad, C. Yin, H. Vincent Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," 2019, *arXiv:1909.07972*. [Online]. Available: http://arxiv.org/abs/1909.07972

[36] M. Chen, H. Vincent Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," 2020, *arXiv:2001.07845*. [Online]. Available: http://arxiv.org/abs/2001.07845

[37] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "Federated learning with quantization constraints," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8851–8855.

[38] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[39] L. U. Khan *et al.*, "Federated learning for edge networks: Resource optimization and incentive mechanism," 2019, *arXiv:1911.05642*. [Online]. Available: http://arxiv.org/abs/1911.05642

[40] S. Hosseinalipour, C. G. Brinton, V. Aggarwal, H. Dai, and M. Chiang, "From federated to fog learning: Distributed machine learning over heterogeneous wireless networks," 2020, *arXiv:2006.03594*. [Online]. Available: http://arxiv.org/abs/2006.03594

[41] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2019, *arXiv:1912.04977*. [Online]. Available: http://arxiv.org/abs/1912.04977

[42] T. Li, A. Kumar Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," 2019, *arXiv:1908.07873*. [Online]. Available: http://arxiv.org/abs/1908.07873

[43] Y. Tu, Y. Ruan, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 2509–2518.

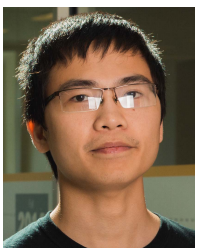
[44] C. M. Bishop, *Pattern Recognition and Machine Learning*. Cham, Switzerland: Springer, 2006.

[45] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[46] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: Extending MNIST to handwritten letters," in *Proc. Int. Joint Conf. Neural Netw.*, May 2017, pp. 2921–2926.

[47] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *CS224N Project Rep.*, vol. 1, no. 12, p. 15, 2009.

[48] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Symp. Operating Syst. Des. Implement.*, 2016, pp. 265–283.



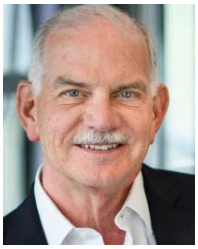
Hung T. Nguyen received the Ph.D. degree in computer science from Virginia Commonwealth University in 2018. He spent a year at Carnegie Mellon University as a Post-Doctoral Fellow before joining Princeton University. He is currently a Post-Doctoral Research Associate with Princeton University working with Prof. Mung Chiang and Prof. H. Vincent Poor. His research interests include distributed optimization, machine learning, and approximation algorithms for graph problems. He received multiple best paper awards from conferences and recognition from universities where he worked.

Vikash Sehwag is currently pursuing the Ph.D. degree in electrical engineering with Princeton University. His primary research interests include embedding the *security by design* principle in machine learning where he aims to improve both performance and robustness of machine learning, simultaneously. His research interests include unsupervised machine learning, open-world machine learning, federated learning, and design of compact and efficient neural networks. He was a recipient of Qualcomm Innovation Fellowship 2019.

Seyyedali Hosseinalipour (Member, IEEE) received the B.S. degree from the Amirkabir University of Technology (Tehran Polytechnic) in 2015 and the Ph.D. degree from NC State University in 2020, in electrical engineering. He received the ECE Doctoral Scholar of the year award at NC State. He is currently a Post-Doctoral Researcher with Purdue University. His research main interests include analysis of modern wireless networks and communication systems.

Christopher G. Brinton (Senior Member, IEEE) received the master's and Ph.D. degrees in electrical engineering from Princeton University in 2013 and 2016, respectively. He is currently an Assistant Professor of ECE with Purdue University. His research interests include the intersection of data science and network optimization, particularly in distributed machine learning. Since joining Purdue in 2019, he has won the Purdue Seed for Success Award, the Purdue ECE Outstanding Faculty Mentor Award, and the 2020 Ruth and Joel Spira Outstanding Teacher Award. His research won the Bede Liu Best Dissertation Award from Princeton University. He is a co-founder of Zoomi Inc., and a coauthor of the book *The Power of Networks: Six Principles that Connect our Lives*.

Mung Chiang (Fellow, IEEE) was the Arthur LeGrand Doty Professor of electrical engineering with Princeton University, where he also served as the Director for the Keller Center for Innovations in Engineering Education and the inaugural Chairman for the Princeton Entrepreneurship Council. He founded the Princeton EDGE Laboratory in 2009, which bridges the theory-practice gap in edge networking research by spanning from proofs to prototypes. He also co-founded a few startup companies in mobile data, IoT, and AI, and co-founded the global nonprofit Open Fog Consortium. He is currently the John A. Edwardson Dean of the College of Engineering and the Roscoe H. George Professor of Electrical and Computer Engineering with Purdue University. His research on networking received the 2013 Alan T. Waterman Award, the highest honor to U.S. young scientists and engineers. His textbook *Networked Life*, popular science book *The Power of Networks*, and online courses reached more than 400,000 students since 2012. In 2019, he was named to the steering committee of the newly expanded Industrial Internet Consortium (IIC).



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977.

From 1977 until 1990, he was on the faculty of the University of Illinois at Urbana–Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor of Electrical Engineering. From 2006 to 2016, he served as the Dean for the Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests include the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the recent book *Multiple Access Techniques for 5G Wireless Networks and Beyond* (Springer, 2019). He is a member of the National Academy of Engineering and the National Academy of Sciences, and a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal and a D.Eng. *honoris causa* from the University of Waterloo awarded in 2019.