Clipper: Online Joint Client Sampling and Power Allocation for Wireless Federated Learning

WEN XU and BEN LIANG, University of Toronto, Canada
GARY BOUDREAU and HAMZA SOKUN, Ericsson, Canada

Communication overhead is a main bottleneck in federated learning (FL) especially in the wireless environment due to the limited data rate and unstable radio channels. The communication challenge necessitates holistic selection of participating clients that accounts for both the computation needs and communication cost, as well as judicious allocation of the limited transmission resource. Meanwhile, the random unpredictable nature of both the training data samples and the communication channels requires an online optimization approach that adapts to the changing system state over time. In this work, we consider a general framework of online joint client sampling and power allocation for wireless FL under time-varying communication channels. We formulate it as a stochastic network optimization problem that admits a Lyapunov-typed solution approach. This leads to per-training-round subproblems with a special bi-convex structure, which we leverage to propose globally optimal solutions, culminating in a meta algorithm that provides strong performance guarantees. We further study three specific FL problems covering multiple scenarios, namely with IID or non-IID data, whether robustness against data drift is required, and with unbiased or biased client sampling. We derive detailed algorithms for each of these problems. Simulation with standard classification tasks demonstrate that the proposed communication-aware algorithms outperform their counterparts under a wide range of learning and communication scenarios.

CCS Concepts: • **Computing methodologies** → **Machine learning**; **Distributed algorithms**; • **Networks**;

Additional Key Words and Phrases: federated learning, client sampling, power allocation, stochastic network optimization

ACM Reference Format:
Wen Xu, Ben Liang, Gary Boudreau, and Hamza Sokun. 2024. Clipper: Online Joint Client Sampling and Power Allocation for Wireless Federated Learning. *ACM Trans. Model. Perform. Eval. Comput. Syst.* XX, X (X 2024), 28 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Distributed machine learning (ML), and in particular federated learning (FL), has become a popular learning paradigm due to the vast amount of available data and continuously increasing computing capabilities of commodity computers and mobile devices [13, 18, 28, 35]. We consider the problem of training an ML model under the client-server system model where multiple resource-constrained clients collaboratively train a single ML model with the assistance of a central server [2, 17, 20]. The local datasets at each client are not allowed to be transmitted and can be heterogeneous, i.e., generated from non-identical data distributions. Furthermore, only the model parameters and some auxiliary control variables are shared during the training. In this setting, the standard learning

Authors' Contact Information: Wen Xu, realwen.xu@mail.utoronto.ca; Ben Liang, liang@ece.utoronto.ca, University of Toronto, Toronto, ON, Canada; Gary Boudreau, gary.boudreau@ericsson.com; Hamza Sokun, hamza.sokun@ericsson.com, Ericsson, Ottawa, Ontario, Canada.

ACM 2376-3647/2024/X-ART
https://doi.org/XXXXXXX.XXXXXXX

objective is to minimize a global loss $f(\boldsymbol{w})$, which is a $\boldsymbol{p}$-weighted sum of the local losses $f_n(\boldsymbol{w})$ that can only be accessed at the corresponding client n, i.e.,

$$\min_{\boldsymbol{w}\in\mathcal{W}} f(\boldsymbol{w}) := \sum_{n=1}^{N} p_n f_n(\boldsymbol{w}), \tag{1}$$

where N is the number of clients, $\mathcal{W}$ is the set of feasible model parameter values, and $\boldsymbol{p} \in \{\boldsymbol{p} \in \mathbb{R}^N : p_n \geq 0, \forall n \in [N], \sum_{n=1}^{N} p_n = 1\}$ is the weight vector, where we define $[N] = \{1, 2, \ldots, N\}$. A typical choice of $\boldsymbol{p}$ is setting $p_n = \frac{D_n}{\sum_{i=1}^{N} D_i}$ for each $n \in [N]$, where D_n is number of data points at client n [20]. Some canonical algorithms to solve (1) in the FL setting are federated stochastic gradient descent (FedSGD) and federated averaging (FedAvg) [20]. However, other algorithms tailored for different system settings have also been considered. For example, enabling arbitrary and unbiased client sampling in [8, 19, 24] and biased client sampling in [9].

Although FedSGD, FedAvg, and many of their variants are easy to implement, the communication overhead remains a major obstacle in realistic settings of FL [13, 18, 35]. A large ML model may take a long time to converge since the server needs to communicate with the clients by sending the model parameters in each training round. This phenomenon is exaggerated when the data and the communication conditions are heterogeneous, which is common for wireless FL in mobile edge computing (MEC). Numerous communication reduction methods have thus been proposed. One line of work is to directly control communication overhead in each round by reducing the size of the model or gradients via compression techniques such as quantization [1] and sparsification [30]. Another line of work is to perform client sampling based on communication constraints such as greedy selection [22] and adaptive sampling [6, 19]. However, client sampling in FL is strongly coupled with communication design. For example, clients that have poor channel condition or are starved of communication resource should have a lower likelihood of being chosen for participation. Therefore, separating the design of client sampling and communication resource allocation will lead to suboptimal learning performance.

The literature on joint optimization of client sampling and communication resource allocation is scarce [7, 10, 24]. In [7] and [10] the client sampling and resource allocation decisions are fixed over the training rounds, which does not account for any system dynamics. However, in wireless FL, the channel conditions between the clients and server are often time varying. With no information on future system states, which includes both the channel condition and the sampled subsets of training data, an online optimization approach is more desirable to adapt to the system dynamics over time. The recent work [24] has considered probabilistic client sampling together with online stochastic network optimization, with time averaged transmission power as constraints to represent the communication overhead. It is based on a simple learning scenario that does not take into account different data ratios among clients or the gradient norms of the clients. It also has no control over the number of sampled clients, which can be important in real-world systems that have limited communication and computation capacity. Furthermore, it ignores the potential need for robustness against data drift [16] or biased client sampling [9] in FL.

In this work, we aim to address all of the technical and practical concerns above, by proposing a comprehensive general framework for online joint client sampling and power allocation in wireless FL. It accounts for the data ratios among clients and the gradient norms, enables control over the number of sample clients, allows consideration for robustness against data drift, and accommodates both unbiased and biased client sampling. As far as we are aware, there is no other work in the literature that considers such general online joint client sampling and power allocation for wireless FL. The main contributions of this work are as follows:

- To overcome the challenge of unpredictable changes in the environment due to stochastic data sampling and random fluctuation of the wireless channels, we adopt an online Lyapunov optimization framework [21] that iteratively tracks the time-varying system state. Even though the Lyapunov optimization technique is well known, the resultant subproblems in each FL training round are non-convex. Nevertheless, even under a general FL formulation where the subproblems can take various forms, we show that there is an efficient solution that yields an optimal solution in each round. This enables our design of a novel meta algorithm, termed CLIPPER (for joint CLIent samPling and PoER allocation), which can be applied to a wide range of FL algorithms and objectives.
- We further investigate three specific scenarios of CLIPPER for different FL algorithms and objectives. (i) CLIPPER-UNBIASED: We consider the standard FL with unbiased client sampling with time-varying communication channels. We provide convergence analysis for both IID and non-IID data and integrate it with CLIPPER. (ii) CLIPPER-ROBUSTFL: We consider the problem of robust FL, where the target data distribution for inference can be different from those of individual clients or of the aggregate training data set. We use a generalization bound in learning to solve for the weight vector $\boldsymbol{p}$, then we utilize our convergence analysis of FL with unbiased client sampling over non-IID data for integration with CLIPPER. (iii) CLIPPER-BIASED: We further demonstrate how CLIPPER can also be applied to biased client sampling in FL.
- We conduct experiment on standard datasets to show that our proposed communication-aware FL algorithms outperform their corresponding communication-agnostic counterparts for all three scenarios.

The rest of this paper is structured as follows. We provide related work of client sampling and resource allocation in FL in Section 2. In Section 3, we present the optimization formulation and solution that culminate in the meta algorithm CLIPPER. We then provide the three case studies on FL with unbiased and biased client sampling in Section 4. We conduct simulation for each use case in Section 5 and conclude the paper in Section 6.

2 RELATED WORK

2.1 Federated Learning and Client Sampling

It has been well recognized that the communication overhead is a main bottleneck in real-world FL deployment due to the large size of models and a large number of training rounds until convergence [13, 18, 35]. One canonical means to reduce communication overhead is to run multiple steps of local model updates before global model aggregation on the server. Many algorithms enjoy this type of periodic model synchronization, including federated averaging (FEDAVG) [20] and local stochastic gradient descent (LOCAL SGD) [27], as well as many variants of these algorithms, e.g., SCAFFOLD [15]. Another line of work to further reduce communication in FL is to send only a compressed version of the gradients or the updated local models via techniques such as quantization [1] and sparsification [30].

Beyond periodic model synchronization and compression techniques, system-aware client selection is a common means to further reduce communication overhead. An early work focusing on client selection formulated an optimization problem whose objective is to maximize the number of selected clients subject to some constraint such as the round time [22]. The authors proposed a greedy client selection strategy that resulted in a model with competitive performance but in a significantly shorter time, compared with full participation of clients. However, the proposed algorithm was purely heuristic without theoretical guarantees or proof of correctness. A tier-based client selection was proposed in [5], which divides clients into tiers based on performance and only

selects clients of the same tier in each round. However, these approaches use a deterministic client selection strategy, which cannot accommodate fluctuation in the communication channels over time.

It is possible to allow probabilistic client selection, for which the term client sampling is commonly used. In fact, in the pioneering work introducing the term *federated learning* [20], the server chooses a fraction of all clients for participation in each round in a uniformly random manner. Client sampling with more general sampling probabilities has also been considered. For example, it was proposed in [8] to set the probabilities by explicitly utilizing the norms of the gradients as substitute for the *importance* of the data at different clients. Taking into account both the *importance* of local data and communication overhead, an optimization to minimize an approximation of the communication time was proposed and solved in closed-form in [36]. A meta algorithm of geometrically increasing client participation to tackle stragglers was proposed in [25], where theoretical analysis was provided to show that it can outperform standard FL for strongly convex objectives.

Other methods have been proposed to further take into account realistic system heterogeneity such as communication conditions and computation capabilities. A new optimization problem to minimize the convergence time of FL, taking into account wireless communications and at the same time optimizing the performance of FL, was formed and solved in [6]. However, though it employed a probabilistic client sampling approach, it required a fixed client to always be connected to the server at each communication round, which is not ideal as the channel condition of that client can fluctuate and even make the client itself a straggler. Minimizing the expected wall-clock time with constraints on convergence and sampling probabilities was considered in [19]. The authors considered both the statistical and system heterogeneity and provided convergence bounds for arbitrary sampling probabilities for smooth and convex objectives. However, their sampling probabilities were fixed in all rounds, and they did not consider client sampling jointly with resource allocation, which our work does. Furthermore, our work is explicitly designed to accommodate time-varying communication channels, while [6] considered a fixed channel for each client and [19] considered fixed communication delay per round for each client.

2.2 Joint Client Sampling and Communication Resource Allocation

As explained in Section 1, for optimal FL performance one should jointly consider client sampling and communication resource allocation. In [7], an optimization problem on joint learning, wireless resource allocation, and client selection was formulated and solved, with consideration for delay requirements, energy consumption, transmit power, and packet error rate. Another optimization problem was proposed in [10] to capture the trade-off between the wall-clock convergence time and energy consumption for wireless FL. Enabling probabilistic client sampling, a novel scheduling policy was proposed to exploit both importance of learning by gradient divergence and channel conditions was proposed in [26], sampling one client per round. However, in all these approaches, they cannot capture the fluctuation in the communication channels over time.

To accommodate time-varying channel and other system conditions, several works proposed to take into account the temporal perspective of client selection and resource allocation in wireless FL [24, 32, 37], utilizing Lyapunov-typed stochastic optimization [21]. In [32], a stochastic optimization problem was formulated with long-term client energy constraints and an algorithm was proposed to solve the joint client selection and bandwidth allocation problem. In [37], an optimization problem was formulated, to minimize the total maximum training delay with constraints of long-term energy consumption and number of selected clients, and it was solved via combinatorial multi-armed bandits. However, both proposed optimization problems in [32] and [37] are mixed-integer nonlinear programming problems, which are hard to solve. To overcome this

difficulty of combinatorial optimization, probabilistic client sampling can be utilized [24]. In [24], joint optimization of the convergence and communication overhead of wireless FL was studied. However, as explained in Section 1, the applicability of [24] is limited by its simple model for FL. In this work, the proposed Clipper framework is a meta algorithm that has general applicability. Depending on the underlying FL algorithm and objective, Clipper can be used for FL with IID or non-IID data, with or without consideration for robustness against data drift during inference, and with unbiased or biased client sampling. A part of this work has appeared in a conference paper [33]. It roughly corresponds to the limited case of Clipper-Unbiased. It does not consider the general meta algorithmic framework, or discuss the cases of robust FL and biased client sampling.

Finally, we note that time-varying channels were also widely considered in FL with over-the-air computation [34]. For example, joint client selection and power control was studied in [11] and joint client selection and uplink beamforming design was studied in [14]. However, these methods are specific to FL with analog transmission and aggregation. They are not applicable to our work.

3 A GENERAL FRAMEWORK OF ONLINE JOINT CLIENT SAMPLING AND POWER ALLOCATION

We present a general algorithmic framework for joint client sampling and power allocation in the online setting, where both the stochastically sampled data and the communication channels of clients are unknown ahead of time. We adopt a stochastic network optimization approach that takes into both the clients' computation towards learning convergence and their communication delay in each round, as well as the communication and power constraints. We then propose a general meta algorithm to solve the stochastic network optimization problem based on an optimal solution to each per-round subproblem.

3.1 Wireless FL System Model

We consider a client-server FL model where a central server coordinates the training of a global ML model with multiple resource-constrained clients, e.g., mobile or IoT devices [2, 17, 20]. All the training data are locally stored at the clients. In each training round, (i) the server broadcasts the current global model and some control variables to a selected subset of clients, (ii) each sampled client performs local model updates utilizing its local data, (iii) each sampled client sends the updated model, potentially also some control variables, back to the server, and (iv) the server aggregates the updated models via weighted averaging and updates the control variables.

It is clear that in each training round, models and some control variables are transmitted between the server and the clients. For communication, we model the uplink transmission rate $r_n^{(t)}$ between client n and the server in the training round t by the Shannon bound, i.e.,

$$r_n^{(t)} = B \log_2 \left(1 + \frac{h_n^{(t)} P_n^{(t)}}{N_0} \right), \tag{2}$$

where B is the bandwidth between the clients and the server, N_0 is the noise power, $h_n^{(t)}$ is the channel power gain of client n, and $P_n^{(t)}$ is the allocated transmission power of client n. We denote the vector of the allocated transmission power of all clients in round t as $\boldsymbol{P}^{(t)} = [P_1^{(t)}, \ldots, P_N^{(t)}]$ and the vector of all channel power gains in round t as $\boldsymbol{h}^{(t)} = [h_1^{(t)}, \ldots, h_N^{(t)}]$. Then the communication time of any client n in training round t is

$$T_{\text{comm},n}^{(t)} = \frac{M}{r_n^{(t)}}, \tag{3}$$

where M is the size of the transmitted model and some potential control variables in bits.

We assume each client n is sampled to participate in the training with probability $q_n^{(t)}$ in round t as an independent Bernoulli trial with a success probability $q_n^{(t)}$. Let $\boldsymbol{q}^{(t)} = [q_1^{(t)}, \ldots, q_N^{(t)}]$. We use $a_n^{(t)}$ to denote the indicator function of the event that client n is sampled in round t. We consider time division multiple access (TDMA), so the expected total communication time by all clients in round t is

$$\begin{aligned}\mathbb{E}[T_{\text{total}}^{(t)}] &= \mathbb{E}_{\boldsymbol{q}^{(t)}}\left[\sum_{n=1}^{N} a_n^{(t)} T_{\text{comm},n}^{(t)}\right] \\ &= \sum_{n=1}^{N} q_n^{(t)} \left(\frac{M}{B\log_2\left(1+\frac{h_n^{(t)}P_n^{(t)}}{N_0}\right)}\right). \end{aligned} \tag{4}$$

For downlink transmission, we assume the server can broadcast the model and the control variables, making the downlink transmission time negligible compared with the uplink transmission time. Furthermore, since the computation time is independent of communication time for sampled clients, we do not need to explicitly consider the time for local computation.

3.2 General Online Optimization Formulation

Our optimization objective considers both some general objectives of learning and the expected per-round communication time. Let $\Phi(\boldsymbol{q}^{(t)})$ be an arbitrary expression that captures some desired property of learning with respect to the client sampling probabilities. For analytical convenience, we assume that $\Phi(\boldsymbol{q}^{(t)})$ is convex in $\boldsymbol{q}^{(t)}$. We will see that this assumption holds in all case studies in Section 4, covering a wide range of FL scenarios. We emphasize here that the loss function of FL does *not* need to be convex, e.g., our work is applicable to FL with non-convex neural networks.

Combining the expected per-round communication time in Section 3.1, we define our overall objective in round t as

$$y_0(t) = \Phi\left(\boldsymbol{q}^{(t)}\right) + \lambda_c \sum_{n=1}^{N} q_n^{(t)} \frac{M}{B\log_2\left(1+\frac{h_n^{(t)}P_n^{(t)}}{N_0}\right)}, \tag{5}$$

where λ_c is a hyperparameter that can be tuned to trade off the learning outcome and communication time. Note that in our objective, B, M, and N_0 are fixed and known constants while $\boldsymbol{q}^{(t)}$ and $\boldsymbol{P}^{(t)}$ are the optimization variables to be determined. The randomness of communication channel is in its power gain, i.e., $h_n^{(t)}$, which is assumed independent among all $n \in [N]$ and all training rounds $t \in \{0, \ldots, T-1\}$. The randomness in learning is captured in $\Phi(\boldsymbol{q}^{(t)})$, and we assume this randomness is independent over t. Thus, the objective $y_0(t)$ is revealed in an online fashion, such that in round t, only the current realization of all random quantities is known and no future information beyond t is revealed.

We consider constraints on the transmission power. The first is that all clients have maximum transmission power $P_{\max}$. The second is that each client n has a long-term average power budget $\bar{P}_n$, which captures the requirement of energy conservation for devices with limited battery capacity. Furthermore, to account for the limited wireless communication capacity, we impose a soft constraint on the number of sampled clients by setting its expected value to m. Thus, we obtain the following stochastic optimization problem:

$$\textbf{P1:} \min_{\{\boldsymbol{q}^{(t)}\},\{\boldsymbol{P}^{(t)}\}} \quad \lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} y_0(t) \tag{6}$$

$$\text{subject to} \quad \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T-1} P_n^{(t)} q_n^{(t)} \le \bar{P}_n, \quad \forall n \in [N], \tag{7}$$

$$0 \le P_n^{(t)} \le P_{\max}, \quad \forall n \in [N], \tag{8}$$

$$\sum_{n=1}^{N} q_n^{(t)} = m, \tag{9}$$

$$0 \le q_n^{(t)} \le 1, \quad \forall n \in [N], \tag{10}$$

where T is the terminal round of FL training, (6) is a time average of $y_0(t)$, (7) is the time average constraint on the expected power, i.e., $P_n^{(t)} q_n^{(t)}$, (8) is the constraints on maximum client power per-round, (9) specifies the expected number of sampled clients, and (10) ensures that $q_n^{(t)}$ is a valid sampling probability.

3.3 Per-round Subproblems and Solutions

We observe that the optimization problem in **P1** is an online optimization problem where the objective is a time-average and the constraints contain both time-average constraints and constraints of action sets. It is easy to check that a stationary randomized solution exists. Therefore, the general min drift-plus-penalty framework of stochastic network optimization [21] is applicable to our problem, and we only need to design a solution to the resultant per-round subproblems.

We first transform the long-term power constraints into queue stability. Let

$$y_n^{(t)} = P_n^{(t)} q_n^{(t)} - \bar{P}_n, \quad \forall n \in [N]. \tag{11}$$

Let $Z_n^{(t)}$ be the backlog of virtual queue n in round t. Define virtual queue update rules

$$Z_n^{(t+1)} = \max\left\{Z_n^{(t)} + y_n^{(t)}, 0\right\}, \quad \forall n \in [N]. \tag{12}$$

Stacking all the queue backlogs at time t into one vector, we obtain a vector $\boldsymbol{\Theta}(t)$. We define the following standard Lyapunov function:

$$L(\boldsymbol{\Theta}(t)) = \frac{1}{2}\|\boldsymbol{\Theta}(t)\|_2^2 = \frac{1}{2}\sum_{n=1}^{N}\left(Z_n^{(t)}\right)^2. \tag{13}$$

Then, the Lyapunov drift is

$$\Delta(\boldsymbol{\Theta}(t)) = \mathbb{E}[L(\boldsymbol{\Theta}(t+1)) - L(\boldsymbol{\Theta}(t))|L(\boldsymbol{\Theta}(t))], \tag{14}$$

and the drift-plus-penalty expression is

$$\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}[y_0(t)|\boldsymbol{\Theta}(t)], \tag{15}$$

where $V \in \mathbb{R}_+$ balances the trade-off between the Lyapunov drift and minimizing the objective function. By [21, Lemma 4.6], we have the following upper bound on the drift-plus-penalty expression:

$$\Delta(\boldsymbol{\Theta}(t)) + V\mathbb{E}\left[y_0(t)|\boldsymbol{\Theta}(t)\right] \le B_0 + V\mathbb{E}[y_0(t)|\boldsymbol{\Theta}(t)] + \sum_{n=1}^{N} Z_n^{(t)}\mathbb{E}\left[y_n^{(t)}|\boldsymbol{\Theta}(t)\right], \tag{16}$$

where B_0 is a positive constant. This leads to the following per-round subproblem:

$$\textbf{P2:} \min_{\boldsymbol{q}^{(t)},\boldsymbol{P}^{(t)}} \quad V y_0(t) + \sum_{n=1}^{N} Z_n^{(t)}\left(P_n^{(t)} q_n^{(t)} - \bar{P}_n\right) \tag{17}$$

$$\text{s. t.} \quad 0 \le P_n^{(t)} \le P_{\max}, \quad \forall n \in [N],$$

$$\sum_{n=1}^{N} q_n^{(t)} = m,$$
$$0 \le q_n^{(t)} \le 1, \quad \forall n \in [N].$$

It is clear that **P2** is not directly decomposable into per-client subproblems, as the constraint on the sum of sampling probabilities in (9) coalesces different clients. However, when $\Phi(\boldsymbol{q}^{(t)})$ is convex in $\boldsymbol{q}^{(t)}$, we have the following special bi-convex structure that induces an efficient solution:

- Given $\boldsymbol{P}^{(t)}$, the objective and constraints are convex in $\boldsymbol{q}^{(t)}$.
- Given $\boldsymbol{q}^{(t)}$, the objective and constraints are convex in $\boldsymbol{P}^{(t)}$. Furthermore, we note that the objective of **P2** can be written as

$$V\left(\Phi(\boldsymbol{q}^{(t)}) + \lambda_c \sum_{n=1}^{N} q_n^{(t)} \frac{M}{B\log_2\left(1+\frac{h_n^{(t)}P_n^{(t)}}{N_0}\right)}\right) + \sum_{n=1}^{N} Z_n^{(t)}\left(P_n^{(t)} q_n^{(t)} - \bar{P}_n\right).$$

  Since $\Phi(\boldsymbol{q}^{(t)})$ is independent of $\boldsymbol{P}^{(t)}$, now the optimization over $\boldsymbol{P}^{(t)}$ can be equivalently decomposed into N subproblems. More importantly, in each of these subproblems, after removing the parts that do not depend on $\boldsymbol{P}^{(t)}$, $q_n^{(t)}$ is a common factor in both terms of the sum, so that its value does not impact the optimization of $P_n^{(t)}$.

Hence, the following two-step approach suffices to compute a globally optimal solution.

Step 1: For each client $n \in [N]$, the subproblem to solve for $P_n^{(t)}$ is

$$\min_{P_n^{(t)}} \quad \frac{MV\lambda_c}{B\log_2\left(1+\frac{h_n^{(t)}P_n^{(t)}}{N_0}\right)} + Z_n^{(t)} P_n^{(t)} \tag{18}$$

$$\text{s. t.} \quad 0 \le P_n^{(t)} \le P_{\max}. \tag{19}$$

Problem (18) is a single-variable convex optimization problem with a convex objective and a box constraint. We now derive a closed-form solution to this problem. For ease of presentation, define $A_1 = \frac{V\lambda_c M \log(2)}{B}$, $A_2 = \frac{h_n^{(t)}}{N_0}$, and $A_3 = Z_n^{(t)}$, where log denotes the natural logarithm. The objective in (18) becomes $\frac{A_1}{\log(1+A_2P_n^{(t)})} + A_3P_n^{(t)}$. Setting its derivative to zero, we have

$$\left(1+A_2P_n^{(t)}\right)\left(\log\left(1+A_2P_n^{(t)}\right)\right)^2 = \frac{A_1A_2}{A_3}. \tag{20}$$

Let

$$x = \frac{\log\left(1+A_2P_n^{(t)}\right)}{2}. \tag{21}$$

Substituting x into (20), we obtain

$$x\exp(x) = \sqrt{\frac{A_1A_2}{4A_3}}, \tag{22}$$

where $\exp(x) = e^x$ is the natural exponential function of x. Therefore

$$x = W_0\left(\sqrt{\frac{A_1A_2}{4A_3}}\right), \tag{23}$$

where W_0 is the principal branch of the Lambert W function. Note that since $\sqrt{\frac{A_1A_2}{4A_3}} \ge 0$, x is non-negative and unique.

Now we can recover $P_n^{(t)}$ from x by inverting (21). If this $P_n^{(t)}$ falls within the range $[0, P_{\max}]$, it is the optimal power. Otherwise, it is easy to derive from the KKT conditions that the optimal power is $P_{\max}$. Summarizing the above, we have

$$P_n^{(t)} = \min\left(\left(\exp\left(2W_0\left(\frac{\sqrt{A}}{2}\right)\right) - 1\right)\frac{N_0}{h_n^{(t)}}, P_{\max}\right), \tag{24}$$

where $A = \frac{V\lambda_c M \log(2) h_n^{(t)}}{BZ_n^{(t)} N_0}$.

Step 2: With $\boldsymbol{P}^{(t)}$ computed from Step 1, the optimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{q}^{(t)}} \quad & V\left(\Phi(\boldsymbol{q}^{(t)}) + \sum_{n=1}^{N} \frac{\lambda_c q_n^{(t)} M}{B\log_2\left(1 + \frac{h_n^{(t)} P_n^{(t)}}{N_0}\right)}\right) \\ & + \sum_{n=1}^{N} P_n^{(t)} q_n^{(t)} Z_n^{(t)} \\ \text{s. t.} \quad & \sum_{n=1}^{N} q_n^{(t)} = m, \\ & 0 \le q_n^{(t)} \le 1, \quad \forall n \in [N]. \end{aligned} \tag{25}$$

This is a convex optimization problem in $\boldsymbol{q}^{(t)}$ as the objective is convex and the constraints are affine. An optimal solution can be found efficiently via standard convex optimization solvers [4].

3.4 Clipper Meta Algorithm

We are now ready to present the proposed CLIPPER meta algorithm. It utilizes some arbitrarily given FL algorithm $\mathcal{A}$, as well as the optimal solutions to the per-round subproblems described in Section 3.3. It outputs a sequence of model parameters $\boldsymbol{w}^{(t)}$ as in any FL algorithm, while taking into consideration our probabilistic client sampling and power allocation requirements.

For each training round t, the server first determines the quantity $\Phi(\boldsymbol{q}^{(t)})$ corresponding to the learning algorithm $\mathcal{A}$. It then estimates the current channel conditions $\boldsymbol{h}^{(t)}$ and solves **P2** to obtain the optimal $\boldsymbol{q}^{(t)}$ and $\boldsymbol{P}^{(t)}$. The server further selects clients $\mathcal{S}^{(t)}$ based on $\boldsymbol{q}^{(t)}$ by running an independent Bernoulli trial for each client n with success probability $q_n^{(t)}$. It then broadcasts $\boldsymbol{w}^{(t)}$ and $P_n^{(t)}$ to each client $n \in \mathcal{S}^{(t)}$. Each client $n \in \mathcal{S}^{(t)}$ performs local model updates based on the FL algorithm $\mathcal{A}$ and sends back the updated model using the allocation power $P_n^{(t)}$. Finally, the server computes an updated global model via the aggregation rule in $\mathcal{A}$ and updates the virtual queues via (12).

The overall procedure of CLIPPER is summarized in Algorithm 1. Note that in this meta algorithm, how the FL algorithm $\mathcal{A}$ operates, e.g., how local model update is performed or how the server aggregates local models, is not specified. We will complete all details of the learning part in each of our three use cases in Section 4.

From our analysis in Section 3.3, the per-round optimization solution can be efficiently obtained with an arbitrary precision of optimality that depends on the numerical convex solver in Step 2. Suppose for each per-round subproblem we achieve an ϵ-optimal solution. From [21, Theorem 4.8],

Algorithm 1: CLIPPER META ALGORITHM

Input: an FL algorithm $\mathcal{A}$.
Output: $\{\boldsymbol{w}^{(t)}\}_{t\in\mathcal{T}}$.
1: Server initializes $\boldsymbol{w}^{(0)}$ and virtual queues $\{Z_n^{(0)}\}_{n=1}^N$.
2: **for** each round $t = 0, 1, \ldots, T-1$ **do**
3: Server obtains the learning related quantity $\Phi(\boldsymbol{q}^{(t)})$.
4: Server estimates $\boldsymbol{h}^{(t)}$.
5: Server calculates $\boldsymbol{q}^{(t)}, \boldsymbol{P}^{(t)}$ from **P2**.
6: Server selects clients $\mathcal{S}^{(t)}$ for $\mathcal{A}$ using $\boldsymbol{q}^{(t)}$.
7: Server broadcasts $\boldsymbol{w}^{(t)}$ and $P_n^{(t)}$ to client $n \in \mathcal{S}^{(t)}$.
8: **for** each client $n \in \mathcal{S}^{(t)}$ **do**
9: update the local model via update rule in $\mathcal{A}$;
10: send the updated model to the server using $P_n^{(t)}$.
11: **end for**
12: Server aggregates the locally updated models via the aggregation rule in $\mathcal{A}$.
13: Server updates the virtual queues via (12).
14: **end for**

our solution provides the following performance guarantee:

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[y_0(t)] \le y_0^{\text{opt}} + \frac{B_0+\epsilon}{V}, \tag{26}$$

where y_0^{opt} is the optimal value of the time-average objective in **P1**. Furthermore, it is guaranteed that the meta algorithm satisfies all the time average constraints when $T \to \infty$.

In each round of CLIPPER, the computational complexity is dominated by the convex optimization solver in Step 2 to achieve a ϵ-optimal solution, which is of complexity $O(\text{poly}(N)\log(1/\epsilon))$, where $\text{poly}(N)$ represents a polynomial of N. There are well-studied numerical methods to solve convex optimization problems in polynomial time, such as the interior point method. Details of their operation and complexity can be found in standard references such as [4]. We emphasize here that our solution requires a numerical solver only in Step 2.

4 CLIPPER CASE STUDIES

We now consider three representative scenarios of FL and show how they can be integrated with CLIPPER. For each use case of CLIPPER, we derive a specific algorithm based on both the meta algorithm and the original FL algorithm denoted by $\mathcal{A}$.

Here, we first formalize the learning setting of FL that is common to all three use cases. We consider distributed supervised learning. Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ be the output space. The space of data points is $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Any data distribution $\mathcal{P}$ is defined over $\mathcal{Z}$. We consider N unknown underlying data distributions $\mathcal{P}_1, \ldots, \mathcal{P}_N$, where each $\mathcal{P}_n$ is the underlying data distribution at client n and can be different from the distribution of the other clients. The available training dataset $\mathcal{D}_n$ at each client n is a finite-size independent sample of its underlying distribution $\mathcal{P}_n$ of size D_n, i.e., $\mathcal{D}_n = \{z_{n,1}, \ldots, z_{n,D_n}\}$.

The hypothesis class $\mathcal{F} \subseteq \{f_{\boldsymbol{w}} \,|\, f_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}\}$ is a subset of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We assume each hypothesis is parameterizable with parameters $\boldsymbol{w} \in \mathcal{W}$. We will use $\boldsymbol{w}$ and $f_{\boldsymbol{w}}$ interchangeably

in the paper. We define the loss function as

$$\ell : \mathcal{F} \times \mathcal{Z} \to \mathbb{R}_+,$$

where $\mathbb{R}_+$ is the set of all non-negative real numbers. The loss function $\ell(f_{\boldsymbol{w}}, z)$ measures the discrepancy between the prediction $f_{\boldsymbol{w}}(x)$ and the actual label y for any $f_{\boldsymbol{w}} \in \mathcal{F}$ and $z = (x, y) \in \mathcal{Z}$. We also define the local loss function

$$f_n(\boldsymbol{w}) = \frac{1}{|\mathcal{D}_n|} \sum_{i \in \mathcal{D}_n} \ell(\boldsymbol{w}, z_i),$$

which represents the local loss for any model $\boldsymbol{w}$ on local dataset $\mathcal{D}_n$. The learning objective is to solve the optimization problem $\min_{\boldsymbol{w} \in \mathcal{W}} f(\boldsymbol{w}) = \sum_{n=1}^{N} p_n f_n(\boldsymbol{w})$ as shown in (1).

In all three of our use cases, we consider the common local model update rule in [20], which is sometimes under the name LOCAL SGD [27]. Each sampled client $n \in \mathcal{S}^{(t)}$ performs model updates using local data via J steps of local stochastic gradient descent (SGD), each step with a mini-batch of uniformly sampled local data points. We let $\boldsymbol{w}_{n,j+1}^{(t)}$ be the local model after j steps of SGD update. For any mini-batch $\mathcal{B}_n \subseteq \mathcal{D}_n$, we define the mini-batch loss

$$f_n(\boldsymbol{w}_{n,j}^{(t)}; \mathcal{B}_n) = \frac{1}{|\mathcal{B}_n|} \sum_{i \in \mathcal{B}_n} \ell(\boldsymbol{w}, z_i).$$

The local update rule of j-th local step is

$$\boldsymbol{w}_{n,j+1}^{(t)} = \boldsymbol{w}_{n,j}^{(t)} - \eta \nabla f_n \left(\boldsymbol{w}_{n,j}^{(t)}; \mathcal{B}_{n,j}^{(t)} \right), \tag{27}$$

where η is the learning rate and $\mathcal{B}_{n,j}^{(t)} \subseteq \mathcal{D}_n$ is the uniformly at random sampled mini-batch in step j of round t. After J steps of local updates, each sampled client sends back $\boldsymbol{w}_{n,J}^{(t)}$ to the server.

Note that the above FL local updating rule is used only for the purpose of illustration in these case studies. CLIPPER can accommodate other updating rules.

4.1 Case One: Federated Learning with Unbiased Sampling

In this case, we consider an FL setting with unbiased aggregation on either IID or non-IID data distributions. We use an unbiased scheme for model aggregation which renders the client sampling also unbiased.

In such unbiased model aggregation, we need to compensate for the sampling probability $q_n^{(t)}$ in the aggregation rule as follows.

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \left(\boldsymbol{w}_{n,J}^{(t)} - \boldsymbol{w}^{(t)} \right), \tag{28}$$

where $a_n^{(t)}$ is an indicator function that evaluates to 1 if client $n \in \mathcal{S}^{(t)}$ and 0 otherwise. Note that $\boldsymbol{w}^{(t+1)}$ is an unbiased estimator of $\sum_{n=1}^{N} p_n^{(t)} \boldsymbol{w}_{n,J}^{(t)}$ since

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{q}^{(t)}} \left[\boldsymbol{w}^{(t+1)} \right] &= \mathbb{E}_{\boldsymbol{q}^{(t)}} \left[\boldsymbol{w}^{(t)} + \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \left(\boldsymbol{w}_{n,J}^{(t)} - \boldsymbol{w}^{(t)} \right) \right] \\
&= \boldsymbol{w}^{(t)} + \sum_{n=1}^{N} p_n \left(\boldsymbol{w}_{n,J}^{(t)} - \boldsymbol{w}^{(t)} \right) \\
&= \sum_{n=1}^{N} p_n \boldsymbol{w}_{n,J}^{(t)},
\end{aligned}$$

where the second equality is by the linearity of expectation and the fact that $\mathbb{E}_{q_n^{(t)}}[a_n^{(t)}] = q_n^{(t)}$ for any n.

We now provide convergence analysis for both the IID and non-IID data. We consider general non-convex loss functions, which allow us to capture popular learning models such as neural networks. The derived convergence bounds will then be used to construct $\Phi(\boldsymbol{q}^{(t)})$.

4.1.1 Convergence Analysis: IID Data. We provide convergence analysis for the IID case, where we assume $\mathcal{P} = \mathcal{P}_n$ for all $n \in [N]$. We make the following commonly used assumptions [3, 24, 29, 33].

Assumption 1 (Lower Boundedness). *The global loss f is lower bounded, i.e., there exists f^* such that*

$$f(\boldsymbol{w}) \geq f^*, \tag{29}$$

holds for all $\boldsymbol{w} \in \mathcal{W}$. f^* is the optimal value of (1).

Assumption 2 (Smoothness). *The global loss f is β-smooth, i.e., there exists a positive β such that*

$$\|\nabla f(\boldsymbol{w}_1) - \nabla f(\boldsymbol{w}_2)\| \leq \beta \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|, \tag{30}$$

holds for all $\boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}$.

Assumption 3 (Unbiased local stochastic gradient). *The stochastic gradient ∇f_n over any mini-batch $\mathcal{B}_n \subseteq \mathcal{D}_n$ is an unbiased estimator of the full gradient for the global loss f, i.e.,*

$$\mathbb{E}[\nabla f_n(\boldsymbol{w}; \mathcal{B}_n)] = \nabla f(\boldsymbol{w}), \tag{31}$$

holds $\forall \boldsymbol{w} \in \mathcal{W}$ and $\forall n \in [N]$.

Assumption 4 (Bounded stochastic gradients). *There exist $G > 0$ such that*

$$\mathbb{E}[\|\nabla f_n(\boldsymbol{w}; \mathcal{B}_n)\|^2] \leq G^2, \tag{32}$$

holds for all $\boldsymbol{w} \in \mathcal{W}$, all $n \in \mathcal{N}$, and all mini-batch $\mathcal{B}_n \subseteq \mathcal{D}_n$.

We obtain the following theorem by modifying the standard convergence analysis techniques for stochastic optimization [3] and unbiased client sampling [24], with further consideration of data sizes [33].

Theorem 1. *Suppose Assumptions 1, 2, 3, and 4 hold, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \leq \frac{2(f(\boldsymbol{w}^{(0)}) - f^*)}{\eta T J} + \frac{\eta^2\beta^2(J-1)(2J-1)G^2}{6} + \frac{\beta\eta J}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\frac{p_n}{q_n^{(t)}}\left(G_n^{(t)}\right)^2, \tag{33}$$

where $\boldsymbol{w}^{(0)}$ is the initial model parameters, $G_n^{(t)} = \mathbb{E}\left[\sqrt{\sum_{j=0}^{J-1}\left\|\nabla f_n(\boldsymbol{w}_{n,j}^{(t)}; \mathcal{B}_{n,j}^{(t)})\right\|^2}\right]$, and f^* is the optimal solution of (1).

Proof. See Appendix A. □

4.1.2 Convergence Analysis: non-IID Data. In the non-IID case, $\mathcal{P}_n$'s can be different among the clients. We require the same lower boundedness assumption in Assumption 1 as the IID case, but Assumptions 2-4 need to be modified as follows.

Assumption 5 (Smoothness). *Each f_n is β-smooth, i.e., there exists a positive β such that*

$$\|\nabla f_n(\boldsymbol{w}_1) - \nabla f_n(\boldsymbol{w}_2)\| \leq \beta \|\boldsymbol{w}_1 - \boldsymbol{w}_2\|,$$

holds $\forall \boldsymbol{w}_1, \boldsymbol{w}_2 \in \mathcal{W}$.

Assumption 6 (Unbiased local stochastic gradient). *The stochastic gradient ∇f_n over any mini-batch $\mathcal{B}_n \subseteq \mathcal{D}_n$ is an unbiased estimator of the full gradient for any local loss f_n, i.e.,*

$$\mathbb{E}[\nabla f_n(\boldsymbol{w}; \mathcal{B}_n)] = \nabla f_n(\boldsymbol{w}), \tag{34}$$

holds $\forall \boldsymbol{w} \in \mathcal{W}$ and $\forall n \in [N]$.

Remark 1. *Unlike the IID case, the local stochastic gradient is no longer an unbiased estimator of the full gradient of the global loss. Instead, it is an unbiased estimator of the full gradient of the corresponding local loss.*

Assumption 7 (Bounded stochastic gradients). *For each $n \in [N]$, there exist $G_n > 0$ such that*

$$\mathbb{E}[\|\nabla f_n(\boldsymbol{w}; \mathcal{B}_n)\|^2] \leq G_n^2,$$

holds for all $\boldsymbol{w} \in \mathcal{W}$ and all mini-batch $\mathcal{B}_n \subseteq \mathcal{D}_n$.

With these assumptions, we are able to derive a convergence bound for any non-convex objectives, which is stated in Theorem 2.

Theorem 2. *Suppose Assumptions 1, 5, 6, and 7 hold, we have*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \leq \frac{2(f(\boldsymbol{w}^{(0)}) - f^*)}{\eta T J} + \frac{\eta^2 \beta^2 (J-1)(2J-1)}{6} \sum_{n=1}^{N} p_n G_n^2 + \frac{\beta \eta J}{T} \sum_{t=0}^{T-1} \sum_{n=1}^{N} \frac{p_n}{q_n^{(t)}} \left(G_n^{(t)}\right)^2, \tag{35}$$

where $\boldsymbol{w}^{(0)}$ is the initial model parameters, $G_n^{(t)} = \mathbb{E}\left[\sqrt{\sum_{j=0}^{J-1} \left\|\nabla f_n(\boldsymbol{w}_{n,j}^{(t)}; \mathcal{B}_{n,j}^{(t)})\right\|^2}\right]$, and f^* is the optimal solution of (1).

Proof. See Appendix B. □

4.1.3 Clipper-Unbiased Algorithm Design. We observe that only the third term of the upper bounds in Theorem 1 and in Theorem 2 is related to the sampling probability $\boldsymbol{q}$. To minimize these upper bounds, without any further assumption on any $\boldsymbol{q}^{(t)}$, we set $q_n^{(t)} = 1$ for all $n \in [N]$ and $t \in \mathcal{T}$, since all $p_n (G_n^{(t)})^2$ are positive as well as β, η, J and T. This is the case where all clients participate in all training rounds. If we impose some constraints on the sampling probability, such as the summation of sampling probabilities in all rounds being given, we can have a closed-form solution as shown in Appendix C, which is similar to the work of optimal client sampling in FL [8]. To further take into account communication as in Section 3, we set

$$\Phi(\boldsymbol{q}^{(t)}) = \sum_{n=1}^{N} \frac{p_n}{q_n^{(t)}} \left(G_n^{(t)}\right)^2. \tag{36}$$

Algorithm 2: CLIPPER-UNBIASED

Input: learning rate η, local epochs L, global rounds T.
Output: $\{\boldsymbol{w}^{(t)}\}_{t=0}^{T-1}$.

Server initializes $\boldsymbol{w}^{(0)}$ and $\{Z_n^{(0)}\}_{n=1}^N$.
for each round $t = 0, 1, \ldots, T-1$ **do**
  Server broadcasts current model $\boldsymbol{w}^{(t)}$ to all clients.
  for each client n **do**
    run J steps of training via (27).
    send $G_n^{(t)}$ to the server.
  end for
  Server estimates $\boldsymbol{h}^{(t)}$.
  Server calculates $\boldsymbol{q}^{(t)}, \boldsymbol{P}^{(t)}$ from **P2**.
  Server selects clients $\mathcal{S}^{(t)}$ based on $\boldsymbol{q}^{(t)}$.
  Server broadcasts $P_n^{(t)}$ to client $n \in \mathcal{S}^{(t)}$.
  for each selected client $n \in \mathcal{S}^{(t)}$ **do**
    send the local model to the server use power $P_n^{(t)}$;
  end for
  Server aggregates the local models via (28).
  Server updates the virtual queues via (12).
end for

Note that $\Phi(\boldsymbol{q}^{(t)})$ is a function of not only $\boldsymbol{q}^{(t)}$ but also $\{G_n^{(t)}\}_{n=1}^N$, which depends on the sampling of previous rounds. However, the dependency is weak in general. When such dependency is negligible, our analysis in Section 3 still applies. The effectiveness of using this form of $\Phi(\boldsymbol{q}^{(t)})$ is also empirically verified in our experiments in Section 5.1.

Combining our meta algorithm in Algorithm 1 and our choice of the FL algorithm in this case, we arrive at a new algorithm, which we term CLIPPER-UNBIASED. Its pseudo-code is given in Algorithm 2. Note that in CLIPPER-UNBIASED, in order to calculate $G_n^{(t)}$, the server should first broadcast the model to all the clients and then the clients send back their estimated $G_n^{(t)}$. This is a scalar, whose transmission overhead is negligible in comparison with the overhead of sending the large global model that the client also needs to send.

4.2 Case Two: Robust Federated Learning

In real-world FL deployment, the data or statistical heterogeneity is another major issue. Due to different user patterns or data collection procedures, it is natural that the data distribution for inference may be different from that of training. Therefore, here we consider a robust FL setting where our objective is to train a global model that generalizes well on an arbitrary reference distribution. We adopt the framework of robust learning from [16] and extend it to FL with unbiased probabilistic client sampling and model aggregation. Thus, we also assume that the server has a small reference dataset, drawn from some arbitrary reference distribution, and each client has its own local dataset from a potentially different source distribution.

4.2.1 Optimization Induced by Generalization Bound. In this case, we consider bounded loss functions, which can be either convex or non-convex, so that we can solve for $\boldsymbol{p}$ that will be integrated into our algorithm of robust FL. Let $\mathcal{P}_r$ and $\mathcal{D}_r$ denote the underlying reference distribution and its finite-sample dataset, respectively. From [16], we have a generalization bound on the excess risk of

Algorithm 3: Clipper for Robust Federated Learning (Clipper-RobustFL)

Input: learning rate η, local epochs L, global rounds T.
Output: $\{\boldsymbol{w}^{(t)}\}_{t\in\mathcal{T}}$.

// Phase (1)
Server broadcasts $\mathcal{D}_r$ to all clients.
for each client $n \in [N]$ **in parallel do**
  Client n estimates $d_{\mathcal{F}}(\mathcal{D}_r, \mathcal{D}_n)$.
  Client n sends $d_{\mathcal{F}}(\mathcal{D}_r, \mathcal{D}_n)$ to the server.
end for
Server solves $\boldsymbol{p}^*$ via **P3**.

// Phase (2)
Server initializes $\boldsymbol{w}^{(0)}$ and $\{Z_n^{(0)}\}_{n=1}^N$.
for each round $t = 0, 1, \ldots, T-1$ **do**
  Server broadcasts current model $\boldsymbol{w}^{(t)}$ to all clients.
  for each client n **do**
    run J steps of training via (27).
    send $G_n^{(t)}$ to the server.
  end for
  Server estimates $\boldsymbol{h}^{(t)}$.
  Server calculates $\boldsymbol{q}^{(t)}, \boldsymbol{P}^{(t)}$ from **P2**.
  Server selects clients $\mathcal{S}^{(t)}$ based on $\boldsymbol{q}^{(t)}$.
  Server broadcasts $P_n^{(t)}$ to client $n \in \mathcal{S}^{(t)}$.
  for each selected client $n \in \mathcal{S}^{(t)}$ **do**
    send the local model to the server use power $P_n^{(t)}$.
  end for
  Server aggregates the local models via (41).
  Server updates the virtual queues via (12).
end for

the optimal solution to problem (1) in terms of the reference distribution via Rademacher complexity of the function class of bound losses induced by the models. By minimizing this generalization upper bound of learning, we formulate the following optimization problem:

$$\textbf{P3:} \min_{\boldsymbol{p}} \quad 2\sum_{n=1}^{N} p_n d_{\mathcal{F}}(\mathcal{D}_n, \mathcal{D}_r) + \lambda_r \sqrt{\sum_{n=1}^{N} \frac{p_n^2}{D_n}} \tag{37}$$

$$\text{s. t.} \quad \sum_{n=1}^{N} p_n = 1, \tag{38}$$

$$0 \le p_n \le 1, \quad \forall n \in [N], \tag{39}$$

where $\lambda_r \ge 0$ is a hyper-parameter and

$$d_{\mathcal{F}}(\mathcal{D}_n, \mathcal{D}_r) = \sup_{f_{\boldsymbol{w}}\in\mathcal{F}} \left(\left| \frac{1}{D_n} \sum_{i\in\mathcal{D}_n} \ell(f_{\boldsymbol{w}}, z_i) - \frac{1}{D_r} \sum_{i\in\mathcal{D}_r} \ell(f_{\boldsymbol{w}}, z_i) \right| \right), \tag{40}$$

which measures the discrepancy between any two datasets $\mathcal{D}_n$ and $\mathcal{D}_r$ with respect to $\mathcal{W}$. The optimization problem **P3** in (37)-(39) is convex as the objective is a convex function of $\boldsymbol{p}$ and the constraints form a convex set of $\boldsymbol{p}$. We can solve it by using any standard convex optimization solver efficiently [4]. Given an optimal solution $\boldsymbol{p}^*$, we can use any suitable distributed algorithm, e.g., FedAvg [20], to solve the resultant FL optimization problem (1).

Thus, the optimization problem **P3** induces a robust FL algorithm $\mathcal{A}$ with arbitrary client sampling. It has two phases as follows.

Phase 1: The server first broadcasts the reference dataset $\mathcal{D}_r$ to all clients. Each client then calculates its corresponding $d_{\mathcal{F}}(\mathcal{D}_r, \mathcal{D}_n)$, and sends it back to the server. The server solves the optimization problem **P3** to get $\boldsymbol{p}^*$.

Phase 2: The $\boldsymbol{p}$-weighted empirical risk is minimized in a distributed manner by T rounds of training. We will use similar design choices as our previous use case: sampling each client n with probability $q_n^{(t)}$ in training round t, updating local models via (27), and aggregating the updated local models via an unbiased aggregation rule:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \sum_{n=1}^{N} \frac{p_n^* a_n^{(t)}}{q_n^{(t)}} \left(\boldsymbol{w}_{n,J}^{(t)} - \boldsymbol{w}^{(t)} \right). \tag{41}$$

4.2.2 Clipper-RobustFL Algorithm Design. We observe that phase 2 of the robust FL above performs FL over non-IID data distributions as in our previous use case. Since local datasets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_N$ and $\mathcal{D}_r$ can be different, we can also apply the convergence analysis for non-IID data distributions in Theorem 2, where p_n is replaced by p_n^*. Still, only the third term of the upper bound is related to the sampling probability $\boldsymbol{q}$. Hence, we set

$$\Phi(\boldsymbol{q}^{(t)}) = \sum_{n=1}^{N} \frac{p_n^*}{q_n^{(t)}} \left(G_n^{(t)} \right)^2. \tag{42}$$

Similar to (36), this form of $\Phi(\boldsymbol{q}^{(t)})$ also depends on $\{G_n^{(t)}\}_{n=1}^N$, but its effectiveness is empirically verified in our experiments in Section 5.2. Thus integrating the Clipper meta algorithm with the robust FL algorithm $\mathcal{A}$ above, we obtain a new algorithm termed Clipper-RobustFL. Its pseudo-code is given in Algorithm 3. Note that in Clipper-RobustFL, we need to estimate $d_{\mathcal{F}}(\mathcal{D}_r, \mathcal{D}_n)$ for all n to solve $\boldsymbol{p}^*$ from **P3**. However, this is performed only once before the training, and each client only transmits one float number to the server.

4.3 Case Three: Biased Client Sampling

In this case, we consider an FL algorithm with biased client sampling. Bias is introduced when we perform model aggregation at the server without compensating for the sampling probability $q_n^{(t)}$ as follows:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \sum_{n \in \mathcal{S}^{(t)}} p_n \left(\boldsymbol{w}_{n,J}^{(t)} - \boldsymbol{w}^{(t)} \right), \tag{43}$$

where $\mathcal{S}^{(t)}$ is the set of sampled clients with potentially different sampling probabilities, and $\boldsymbol{w}_{n,J}^{(t)}$ is the updated model from client n. Clearly, $\boldsymbol{w}^{(t+1)}$ is no longer an unbiased estimator of $\sum_{n=1}^N p_n^{(t)} \boldsymbol{w}_{n,J}^{(t)}$.

Specifically, we consider a state-of-the-art biased client sampling algorithm called Power-of-Choice [9]. The original idea of [9] is to select clients with top m largest local loss of the current

Algorithm 4: CLIPPER-BIASED

Input: learning rate η, local epochs L, global rounds T.
Output: $\{\boldsymbol{w}^{(t)}\}_{t\in\mathcal{T}}$.

Server initializes $\boldsymbol{w}^{(0)}$ and $\{Z_n^{(0)}\}_{n=1}^N$.
for each round $t = 0, 1, \ldots, T-1$ **do**
  Server broadcasts current model $\boldsymbol{w}^{(t)}$ to all clients.
  for each client n **do**
    pick a mini-batch of samples $\mathcal{B}_n^{(t)} \in \mathcal{D}_n$.
    send loss estimation $f_n(\boldsymbol{w}^{(t)}; \mathcal{B}_n^{(t)})$ to the server.
  end for
  Server estimates $\boldsymbol{h}^{(t)}$.
  Server calculates $\boldsymbol{q}^{(t)}, \boldsymbol{P}^{(t)}$ from **P2**.
  Server selects clients $\mathcal{S}^{(t)}$ based on $\boldsymbol{q}^{(t)}$.
  Server broadcasts $P_n^{(t)}$ to client $n \in \mathcal{S}^{(t)}$.
  for each selected client $n \in \mathcal{S}^{(t)}$ **do**
    run J steps of training via (27).
    send the local model to the server use power $P_n^{(t)}$.
  end for
  Server aggregates the local models via (43).
  Server updates the virtual queues via (12).
end for

model in each round t. The problem can be formulated as the following maximization:

$$\mathbf{P4:} \max_{\boldsymbol{q}^{(t)}} \quad \sum_{n=1}^{N} q_n^{(t)} f_n\left(\boldsymbol{w}^{(t)}; \mathcal{B}_n^{(t)}\right) \tag{44}$$

$$\text{s. t.} \quad \sum_{n=1}^{N} q_n^{(t)} = m, \tag{45}$$

$$q_n^{(t)} \in \{0, 1\}, \quad \forall n \in [N]. \tag{46}$$

Note that here $q_n^{(t)}$ only takes values of zero or one, resulting in a mixed integer programming problem, which does not allow the general probabilistic client sampling. Therefore, we relax the integer constraints to allow sampling with a flexible probability. The new optimization problem is as follows:

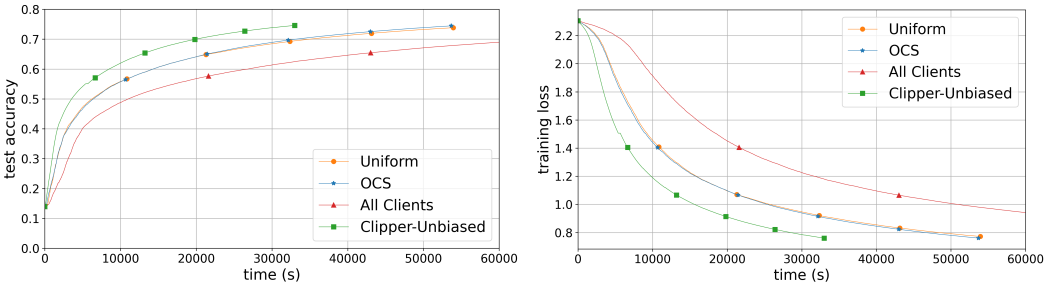
$$\mathbf{P5:} \min_{\boldsymbol{q}^{(t)}} \quad -\sum_{n=1}^{N} q_n^{(t)} f_n\left(\boldsymbol{w}^{(t)}; \mathcal{B}_n^{(t)}\right) \tag{47}$$

$$\text{s. t.} \quad \sum_{n=1}^{N} q_n^{(t)} = m, \tag{48}$$

$$0 \le q_n^{(t)} \le 1, \quad \forall n \in [N], \tag{49}$$

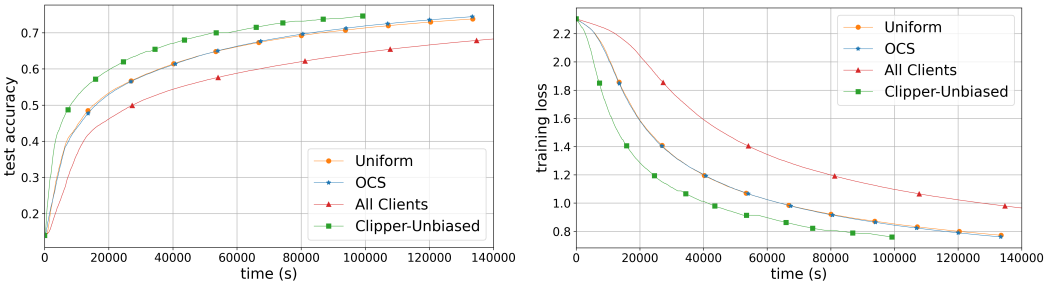
Then we can set $\Phi(\boldsymbol{q}^{(t)})$ to be the objective (47) in **P5**, i.e.,

$$\Phi(\boldsymbol{q}^{(t)}) = -\sum_{n=1}^{N} q_n^{(t)} f_n\left(\boldsymbol{w}^{(t)}; \mathcal{B}_n^{(t)}\right). \tag{50}$$



(a) test accuracy (b) training loss

Fig. 1. Comparison of algorithms on IID data in the homogeneous communication setting.



(a) test accuracy (b) training loss

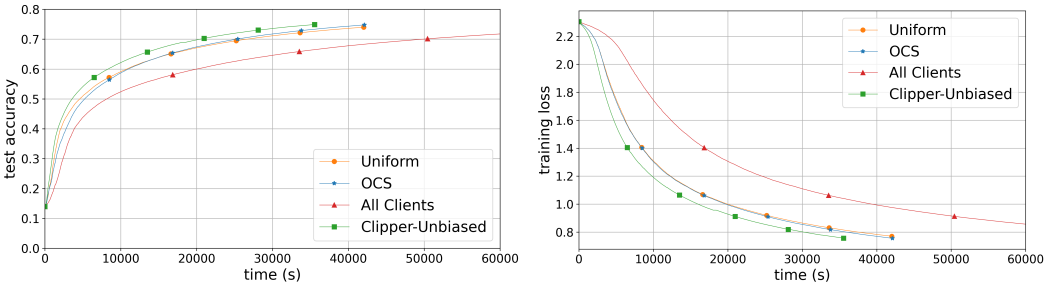
Fig. 2. Comparison of algorithms on IID data in the heterogeneous communication setting.

This allows us to integrate Clipper with biased client sampling to produce a new algorithm termed Clipper-Biased. Its pseudo code is given in Algorithm 4. Here $\Phi(\boldsymbol{q}^{(t)})$ is a function of not only $\boldsymbol{q}^{(t)}$ but also $\boldsymbol{w}^{(t)}$, which weakly depends on the sampling of previous rounds. The effectiveness of this form of $\Phi(\boldsymbol{q}^{(t)})$ is empirically verified in our experiments in Section 5.3. Note that in Clipper-Biased, the clients do not need to estimate and send the norms of the gradients. However, each client still needs to estimate the local loss and transmit it to the server in each training round. Fortunately, such extra communication consists of only a single float number for each client.

5 NUMERICAL EVALUATION

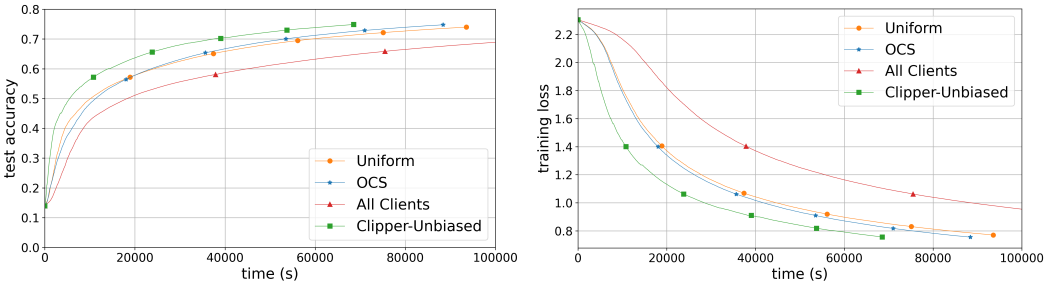
We conduct experiments on classification problems using the Fashion-MNIST dataset [31]. For the learning model, we use a 2-layer MLP, with 300 and 100 neurons in the layers, and ReLU activation functions. We use PyTorch version 2.0.1 [23] as the programming framework.

For communication modeling, we set parameters similar to [24]. We assume bandwidth $B = 22$ MHz and noise power $N_0 = 2 \times 10^{-8}$ W. The communication channels are time-varying. Specifically, we consider an independent Rayleigh fading channel from each client to the server, resulting in $h_n^{(t)}$ following an exponential distribution. Furthermore, we experiment with both homogeneous and heterogeneous communication settings. For the homogeneous setting, the mean of each $h_n^{(t)}$ is fixed at 2×10^{-5}. For the heterogeneous setting, $h_n^{(t)}$ is 2×10^{-5} for half of the clients while $h_n^{(t)}$ is 2×10^{-6} for the other half of the clients, representing the scenario where the distances between



(a) test accuracy (b) training loss

Fig. 3. Comparison of algorithms on non-IID data in the homogeneous communication setting.



(a) test accuracy (b) training loss

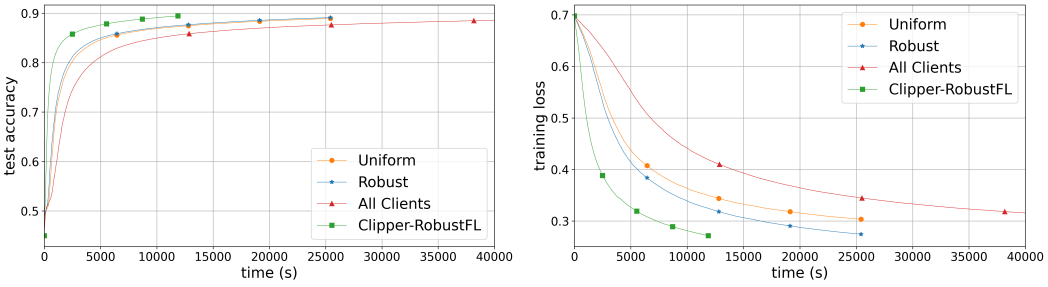
Fig. 4. Comparison of algorithms on non-IID data in the heterogeneous communication setting.

clients and the server can be different. We set the long-term power constraints $\bar{P}_n = 0.01$ W for all n, and maximum power $P_{\max} = 1$ W. We set $V = 1$.

We consider the following benchmarks, which do not explicitly consider communication in client sampling.

- UNIFORM [20]: it is our probablistic client sampling version of FEDAVG [20] with uniform sampling probabilities. This is the main comparison benchmark for CLIPPER-UNBIASED.
- ROBUST [16]: it first solves **P3** to get $\boldsymbol{p}$ and then performs FL with unbiased aggregation and uniform sampling probabilities. This is the main comparison benchmark for CLIPPER-ROBUSTFL.
- POWER-OF-CHOICE [9]: it selects top m clients with highest loss in each round. This is the main comparison benchmark for CLIPPER-BIASED.
- OCS [8]: it finds sampling probabilities via (64), which optimizes the convergence rate with constraints on the expected number of clients.
- ALL CLIENTS: it selects all clients for participation in all rounds. It utilizes all data but neither considers robustness nor communication overhead.

To satisfy the long-term power constraints of these benchmarks that do not consider power allocation, each sampled client transmits with power $\bar{P}_n/q_n^{(t)}$ in round t. Note that we do not numerically compare with [24] here since it does not control the expected number of sampled clients and hence cannot be applied in our system.



(a) test accuracy (b) training loss

Fig. 5. Comparison of algorithms on drifted data distributions in the heterogeneous communication setting.

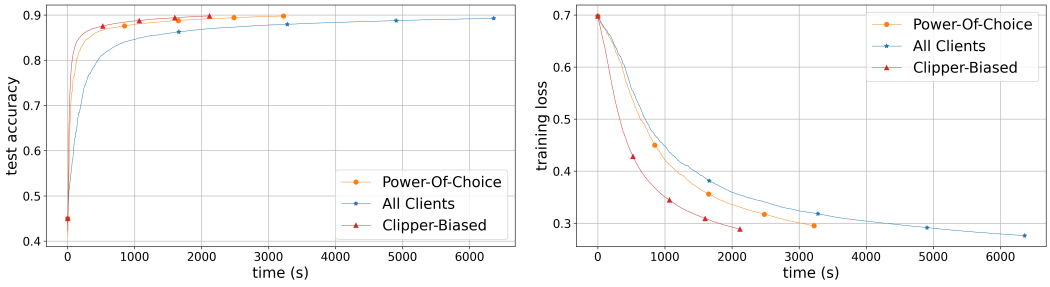
5.1 Clipper-Unbiased

We consider FL over both IID and non-IID data with unbiased aggregation. Each client n contains 5000 data points independently drawn from the whole training dataset uniformly at random for the settings of IID data, while each client n only contains 5000 data points of the nth label for the settings of non-IID data. In each round, we sample on average $m = 5$ clients from a total of $N = 10$ clients. We set the number of communication rounds $T = 50000$, batch size to 1, the learning rate to 0.001, and $\lambda_c = 10$.

The results of Uniform, OCS, All Clients, and Clipper-Unbiased over IID data are shown in Fig. 1 and Fig. 2, respectively for homogeneous and heterogeneous communication settings. For the homogeneous communication setting, we observe that Clipper-Unbiased achieves similar model performance after 50000 rounds compared with OCS, Uniform, and All Clients. However, it saves a significant amount of communication overhead, so that it reaches the same accuracy in much shorter wall-clock time. For the heterogeneous communication setting, Clipper-Unbiased still outperforms all other algorithms. Similar results over non-IID data are observed in Fig. 3 and Fig. 4, respectively for homogeneous and heterogeneous communication settings. For example, as shown in Fig. 3, to reach 74% accuracy in the homogeneous communication setting, Clipper-Unbiased requires only 31613 seconds, which is 24.77% reduction from the 42027 seconds for Uniform, 17.64% reduction from the 38382 seconds for OCS, and more than 47.31% reduction from All Clients, which does not even reach the desired accuracy in 60000 seconds. Similarly, as shown in Fig. 4, to reach 74% accuracy in the heterogeneous communication setting, Clipper-Unbiased takes only 61345 seconds, which is 34.44% reduction from the 93567 seconds for Uniform, 23.91% reduction from the 80623 seconds for OCS, and more than 38.66% reduction from All Clients, which does not reach the desired accuracy in 100000 seconds .

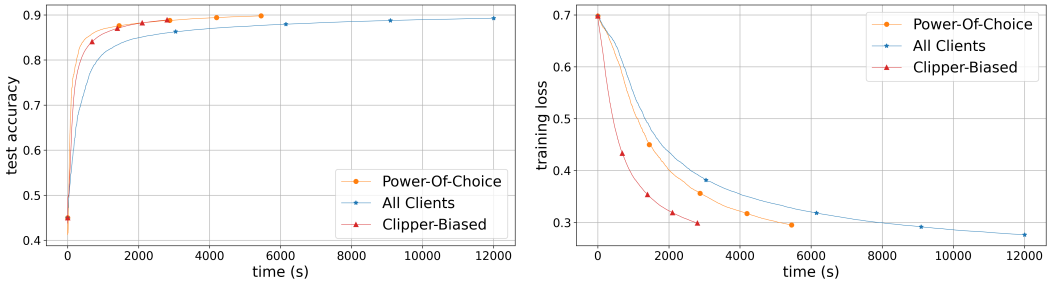
5.2 Clipper-RobustFL

To reduce the complexity in estimating $d_{\mathcal{F}}(\mathcal{D}_n, \mathcal{D}_r)$, we follow the approach in [16] to focus on binary classification problems. We combine the original classes 0 to 4 as the new class 0, and the original classes 5 to 9 as the new class 1. We set $N = 10$ clients, where each client contains 5000 data points independently drawn from the new training dataset uniformly at random. However, unlike the IID or non-IID setting in Section 5.1, we further flip 50% of the labels at a random client to model distribution drift. The reference dataset contains 5000 data points drawn uniformly at random from the remaining training dataset. We set the number of communication rounds $T = 20000$, the batch size to 1, the learning rate to 0.001, $m = 5$, $\lambda_r = 1$, and $\lambda_c = 100$.



(a) test accuracy (b) training loss

Fig. 6. Comparison of algorithms with biased sampling in the homogeneous communication setting.



(a) test accuracy (b) training loss

Fig. 7. Comparison of algorithms with biased sampling in the heterogeneous communication setting.

The results of UNIFORM, ROBUST, CLIPPER-ROBUSTFL, and ALL CLIENTS are shown in Fig. 5 for the heterogeneous communication setting. Here, the test accuracy refers to testing over data drawn from the distribution of the reference dataset. We observe that, in both cases, CLIPPER-ROBUSTFL does not produce the lowest training loss. Yet it reaches the highest accuracy with the least amount of wall-clock time, owning to its proper prediction of the differing data distributions. Specifically, to reach 88% accuracy in the heterogeneous communication setting, CLIPPER-ROBUSTFL takes 5884 seconds, which is a 63.80% reduction from the 16257 seconds for UNIFORM, a 60.10% reduction from the 14748 seconds for ROBUST, and 80.51% from the 30188 seconds for ALL CLIENTS. Results of the homogeneous communication setting are similar and thus omitted here.

5.3 Clipper-Biased

The results of POWER-OF-CHOICE, CLIPPER-BIASED, and ALL CLIENTS are shown in Fig. 6 and Fig. 7, respectively for homogeneous and heterogeneous communication settings. We observe that in both homogeneous and heterogeneous communication settings, as in the previous cases, all tested algorithms are able to reach the highest accuracy, but CLIPPER-BIAS do so with substantially reduced training time. For example, to reach 88% accuracy, it requires only 664 seconds in the homogeneous communication setting, compared with the 1042 seconds for POWER-OF-CHOICE and the 3389 seconds for ALL CLIENTS. In the heterogeneous communication setting, CLIPPER-BIAS requires only 1925 seconds, which is a 69.75% reduction from the 6362 seconds for ALL CLIENTS and is competitive with POWER-OF-CHOICE.

6 CONCLUSION

In this work, we propose a general framework for online optimization in wireless FL that jointly considers client sampling and power allocation with considerations for both learning and communication time. We develop an efficient meta algorithm, termed Clipper, based on optimal solutions to the per-round subproblems resulting from a Lyapunov optimization approach. Then, we discuss three different use cases of wireless FL in detail, demonstrating that Clipper can be integrated with a diverse range of FL algorithms and objectives. Our experiments show that our communication-aware online client sampling and power allocation approach can significantly reduce the wall-clock training time in all use cases. We remark here that while we have presented three representative use cases, the general applicability of Clipper is not limited by these examples.

ACKNOWLEDGEMENTS

This work was supported by Ericsson, the Natural Sciences and Engineering Research Council of Canada (NSERC), and Mitacs.

REFERENCES

[1] Dan Alistarh, Demjan Grubic, Jerry Z. Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, Timon Van Overveldt, David Petrou, Daniel Ramage, and Jason Roselander. 2019. Towards federated learning at scale: System design. In *Proceedings of Machine Learning and Systems (MLSys)*.

[3] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. 2018. Optimization Methods for Large-Scale Machine Learning. *SIAM Rev.* 60, 2 (2018), 223–311.

[4] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization.* Cambridge University Press.

[5] Zheng Chai, Ahsan Ali, Syed Zawad, Stacey Truex, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Heiko Ludwig, Feng Yan, and Yue Cheng. 2020. TiFL: A Tier-Based Federated Learning System. In *Proceedings of International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*.

[6] Mingzhe Chen, H. Vincent Poor, Walid Saad, and Shuguang Cui. 2021. Convergence Time Optimization for Federated Learning Over Wireless Networks. *IEEE Transactions on Wireless Communications* 20, 4 (2021), 2457–2471.

[7] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H. Vincent Poor, and Shuguang Cui. 2021. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications* 20, 1 (2021), 269–283.

[8] Wenlin Chen, Samuel Horváth, and Peter Richtárik. 2022. Optimal Client Sampling for Federated Learning. *Transactions on Machine Learning Research* (2022).

[9] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. 2022. Towards understanding biased client selection in federated learning. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[10] Canh T. Dinh, Nguyen H. Tran, Minh N. H. Nguyen, Choong Seon Hong, Wei Bao, Albert Y. Zomaya, and Vincent Gramoli. 2021. Federated learning over wireless networks: convergence analysis and resource allocation. *IEEE/ACM Transactions on Networking* 29, 1 (2021), 398–409.

[11] Wei Guo, Ran Li, Chuan Huang, Xiaoqi Qin, Kaiming Shen, and Wei Zhang. 2022. Joint Device Selection and Power Control for Wireless Federated Learning. *IEEE Journal on Selected Areas in Communications* 40, 8 (2022), 2395–2410.

[12] Samuel Horváth and Peter Richtárik. 2019. Nonconvex Variance Reduced Optimization with Arbitrary Sampling. In *Proceedings of International Conference on Machine Learning (ICML)*.

[13] Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. Da Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, AdriA Gascan, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Azgaer, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian TramAsr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao.

2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning* 14, 1-2 (2021), 1–210.

[14] Faeze Moradi Kalarde, Min Dong, Ben Liang, Yahia A. Eldemerdash Ahmed, and Ho Ting Cheng. 2024. Beamforming and Device Selection Design in Federated Learning With Over-the-Air Aggregation. *IEEE Open Journal of the Communications Society* 5 (2024), 1710–1723.

[15] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of International Conference on Machine Learning (ICML)*.

[16] Nikola Konstantinov and Christoph Lampert. 2019. Robust learning from untrusted sources. In *Proceedings of International Conference on Machine Learning (ICML)*.

[17] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. 2014. Scaling distributed machine learning with the parameter server. In *Proceedings of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*.

[18] Tian Li, Anit Kumar Sahu, Ameet S. Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine* 37 (2020), 50–60.

[19] Bing Luo, Wenli Xiao, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. 2022. Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*.

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep Networks from decentralized data. In *Proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[21] M. Neely. 2010. *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool. 1–211 pages.

[22] Takayuki Nishio and Ryo Yonetani. 2019. Client selection for federated learning with heterogeneous resources in mobile edge. In *Proceedings of IEEE International Conference on Communications (ICC)*.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

[24] Jake Perazzone, Shiqiang Wang, Mingyue Ji, and Kevin S Chan. 2022. Communication-efficient device scheduling for federated learning using stochastic optimization. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*.

[25] Amirhossein Reisizadeh, Isidoros Tziotis, Hamed Hassani, Aryan Mokhtari, and Ramtin Pedarsani. 2022. Straggler-Resilient Federated Learning: Leveraging the Interplay Between Statistical Accuracy and System Heterogeneity. *IEEE Journal on Selected Areas in Information Theory* 3, 2 (2022), 197–205.

[26] Jinke Ren, Yinghui He, Dingzhu Wen, Guanding Yu, Kaibin Huang, and Dongning Guo. 2020. Scheduling for cellular federated edge learning with importance and channel awareness. *IEEE Transactions on Wireless Communications* 19, 11 (2020), 7690–7703.

[27] Sebastian U Stich. 2019. Local SGD converges fast and communicates little. In *Proceedings of International Conference on Learning Representations (ICLR)*.

[28] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S. Rellermeyer. 2020. A survey on distributed machine learning. *Comput. Surveys* 53, 2 (2020), 1–33.

[29] Jianyu Wang and Gauri Joshi. 2021. Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms. *Journal of Machine Learning Research* 22, 1 (2021), 9709–9758.

[30] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient sparsification for communication-efficient distributed optimization. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. https://github.com/zalandoresearch/fashion-mnist.

[32] Jie Xu and Heqiang Wang. 2020. Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective. *IEEE Transactions on Wireless Communications* 20, 2 (2020), 1188–1200.

[33] Wen Xu, Ben Liang, Gary Boudreau, and Hamza Sokun. 2023. Probabilistic client sampling and power allocation for wireless federated learning. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*.

[34] Kai Yang, Tao Jiang, Yuanming Shi, and Zhi Ding. 2020. Federated Learning via Over-the-Air Computation. *IEEE Transactions on Wireless Communications* 19, 3 (2020), 2022–2035.

[35] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology* 10, 2 (2019), 1–19.

[36] Maojun Zhang, Guangxu Zhu, Shuai Wang, Jiamo Jiang, Qing Liao, Caijun Zhong, and Shuguang Cui. 2022. Communication-efficient federated edge learning via optimal probabilistic device scheduling. *IEEE Transactions on Wireless Communications* 21, 10 (2022), 8536–8551.

[37] Konglin Zhu, Fuchun Zhang, Lei Jiao, Bowei Xue, and Lin Zhang. 2024. Client selection for federated learning using combinatorial multi-armed bandit under long-term energy constraint. *Computer Networks* 250 (2024), 110512.

A PROOF OF THEOREM 1

Proof. For ease of notation, we use $g_n(\boldsymbol{w})$ to represent the stochastic gradient of $f_n(\boldsymbol{w})$. By the unbiased aggregation rule in (28), we have

$$\begin{aligned}\boldsymbol{w}^{(t+1)} &= \boldsymbol{w}^{(t)} + \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} (\boldsymbol{w}_{n,L}^{(t)} - \boldsymbol{w}_{n,0}^{(t)}) \\ &= \boldsymbol{w}^{(t)} - \eta \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)}). \end{aligned} \tag{51}$$

From the β-smoothness of f in Assumption 2 and substituting the update rule in (51), we have

$$\begin{aligned} & f(\boldsymbol{w}^{(t+1)}) - f(\boldsymbol{w}^{(t)}) \\ & \le \nabla f(\boldsymbol{w}^{(t)})^T (\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}) + \frac{\beta}{2} \|\boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)}\|^2 \\ & \le -\nabla f(\boldsymbol{w}^{(t)})^T (\eta \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})) + \frac{\beta\eta^2}{2} \| \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)}) \|^2. \end{aligned} \tag{52}$$

Taking expectation over (52), we have

$$\begin{aligned} & \mathbb{E}[f(\boldsymbol{w}^{(t+1)})] - f(\boldsymbol{w}^{(t)}) \\ & \le -\nabla f(\boldsymbol{w}^{(t)})^T (\mathbb{E}[\eta \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})]) + \frac{\beta\eta^2}{2} \mathbb{E}[\| \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)}) \|^2] \\ & \overset{(a)}{=} -\nabla f(\boldsymbol{w}^{(t)})^T (\eta \sum_{n=1}^{N} p_n \sum_{j=0}^{J-1} \mathbb{E}[\nabla f(\boldsymbol{w}_{n,j}^{(t)})]) + \frac{\beta\eta^2}{2} \mathbb{E}[\| \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)}) \|^2] \\ & = \eta \sum_{j=0}^{J-1} \underbrace{-\mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T (\sum_{n=1}^{N} p_n \nabla f(\boldsymbol{w}_{n,j}^{(t)}))]}_{A} + \underbrace{\frac{\beta\eta^2}{2} \mathbb{E}[\| \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)}) \|^2]}_{B}, \end{aligned}$$

where (a) is by the facts that $a_n^{(t)}$ and $g_n(\cdot)$ are independent, $\mathbb{E}[a_n^{(t)}] = q_n^{(t)}$, and $\mathbb{E}[g_n(\boldsymbol{w}_{n,j}^{(t)})] = \mathbb{E}[\nabla f(\boldsymbol{w}_{n,j}^{(t)})]$ as in Assumption 3.

We now give a bound on A as follows

$$\begin{aligned} & -\mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T (\sum_{n=1}^{N} p_n \nabla f(\boldsymbol{w}_{n,j}^{(t)}))] \\ & = -\mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T (\sum_{n=1}^{N} p_n \nabla f(\boldsymbol{w}_{n,j}^{(t)}) - \nabla f(\boldsymbol{w}^{(t)}) + \nabla f(\boldsymbol{w}^{(t)}))] \end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T(\nabla f(\boldsymbol{w}^{(t)}) - \sum_{n=1}^{N} p_n \nabla f(\boldsymbol{w}_{n,j}^{(t)}))] - \mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T \nabla f(\boldsymbol{w}^{(t)})] \\
&\overset{(a)}{\leq} \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] - \mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] + \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)}) - \sum_{n=1}^{N} p_n \nabla f(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&= \frac{1}{2}\mathbb{E}[\|\sum_{n=1}^{N} p_n (\nabla f(\boldsymbol{w}^{(t)}) - \nabla f(\boldsymbol{w}_{n,j}^{(t)}))\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(b)}{\leq} \frac{1}{2}\sum_{n=1}^{N} p_n \mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)}) - \nabla f(\boldsymbol{w}_{n,j}^{(t)})\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(c)}{\leq} \frac{\beta^2}{2}\sum_{n=1}^{N} p_n \mathbb{E}[\|\boldsymbol{w}^{(t)} - \boldsymbol{w}_{n,j}^{(t)}\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(d)}{\leq} \frac{\beta^2}{2}\sum_{n=1}^{N} p_n \mathbb{E}[\|\sum_{i=0}^{j-1} \eta g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(e)}{\leq} \frac{\eta^2\beta^2 j}{2}\sum_{n=1}^{N} p_n \sum_{i=0}^{j-1} \mathbb{E}[\|g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2],
\end{aligned} \tag{53}
$$

where (a) is by $\boldsymbol{a}^T\boldsymbol{b} \leq (\boldsymbol{a}^2 + \boldsymbol{b}^2)/2$, (b) is by Jensen's inequality, (c) is by the β-smoothness of f in Assumption 2, (d) is by the substituting the local update rule in (27), and (e) is by Jensen's inequality.

We then give a bound on B as follows

$$
\begin{aligned}
&\mathbb{E}[\|\sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&\overset{(a)}{\leq} \sum_{n=1}^{N} p_n \mathbb{E}[\|\frac{a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&\overset{(b)}{=} \sum_{n=1}^{N} p_n \mathbb{E}[|\frac{a_n^{(t)}}{q_n^{(t)}}|^2]\mathbb{E}[\|\sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&\overset{(c)}{=} \sum_{n=1}^{N} \frac{p_n}{q_n^{(t)}} \mathbb{E}[\|\sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&\overset{(d)}{\leq} J\sum_{n=1}^{N} \frac{p_n}{q_n^{(t)}} \sum_{j=0}^{J-1} \mathbb{E}[\|g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2],
\end{aligned} \tag{54}
$$

where (a) is by Jensen's inequality, (b) is by the independence of $a_n^{(t)}$ and $g_n(\cdot)$, (c) is by the fact that $\mathbb{E}[(a_n^{(t)})^2] = q_n^{(t)}$, and (d) is by Jensen's inequality again.

Combining the bounds of A in (53) and B in (54), we obtain

$$
\begin{aligned}
&\mathbb{E}[f(\boldsymbol{w}^{(t+1)})] - f(\boldsymbol{w}^{(t)}) \\
&\leq \frac{\eta^3\beta^2}{2}\sum_{n=1}^{N} p_n \sum_{j=0}^{J-1} j \sum_{i=0}^{j-1} \mathbb{E}[\|g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] - \frac{\eta J}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2]
\end{aligned}
$$

$$+\frac{\beta\eta^2 J}{2}\sum_{n=1}^{N}\frac{p_n}{q_n^{(t)}}\sum_{j=0}^{J-1}\mathbb{E}[\|g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2]. \tag{55}$$

Rearranging terms, summing over t from 0 to $T-1$, taking total expectation, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2]\\
&\le \frac{2(f(\boldsymbol{w}^{(0)})-f^*)}{\eta TJ}+\frac{\eta^2\beta^2}{TJ}\sum_{t=0}^{T-1}\sum_{n=1}^{N}p_n\sum_{j=0}^{J-1}i\sum_{i=0}^{j-1}\mathbb{E}[\|g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2]\\
&\quad+\frac{\beta\eta}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\frac{p_n}{q_n^{(t)}}\sum_{j=0}^{J-1}\mathbb{E}[\|g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2]\\
&\le \frac{2(f(\boldsymbol{w}^{(0)})-f^*)}{\eta TJ}+\frac{\eta^2\beta^2(J-1)(2J-1)G^2}{6}\\
&\quad+\frac{\beta\eta J}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\frac{p_n}{q_n^{(t)}}(G_n^{(t)})^2.
\end{aligned}\tag{56}$$

□

B PROOF OF THEOREM 2

Proof. We first show that the global loss f is also β-smooth under Assumption 5. For all $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ in $\mathcal{W}$, we have

$$\begin{aligned}
\|\nabla f(\boldsymbol{w}_1)-\nabla f(\boldsymbol{w}_2)\| &\overset{(a)}{=} \|\sum_{n=1}^{N}p_n\nabla f_n(\boldsymbol{w}_1)-\sum_{n=1}^{N}p_n\nabla f_n(\boldsymbol{w}_2)\|\\
&\overset{(b)}{\le} \sum_{n=1}^{N}p_n\|\nabla f_n(\boldsymbol{w}_1)-\nabla f_n(\boldsymbol{w}_2)\|\\
&\overset{(c)}{\le} \sum_{n=1}^{N}p_n\beta\|\boldsymbol{w}_1-\boldsymbol{w}_2\|\\
&= \beta\|\boldsymbol{w}_1-\boldsymbol{w}_2\|,
\end{aligned}\tag{57}$$

where (a) is by the definition of global loss f in (1), (b) is by triangle inequality, and (c) is by the β-smoothness of each f_n.

Similarly to the proof of Theorem 1, where we leverage the β-smoothness of f and the update rule in (51), we have

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{w}^{(t+1)})]-f(\boldsymbol{w}^{(t)})\\
&\le -\nabla f(\boldsymbol{w}^{(t)})^T(\mathbb{E}[\eta\sum_{n=1}^{N}\frac{p_n a_n^{(t)}}{q_n^{(t)}}\sum_{j=0}^{J-1}g_n(\boldsymbol{w}_{n,j}^{(t)})])+\frac{\beta\eta^2}{2}\mathbb{E}[\|\sum_{n=1}^{N}\frac{p_n a_n^{(t)}}{q_n^{(t)}}\sum_{j=0}^{J-1}g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2]\\
&\overset{(a)}{=} -\nabla f(\boldsymbol{w}^{(t)})^T(\eta\sum_{n=1}^{N}p_n\sum_{j=0}^{J-1}\mathbb{E}[\nabla f_n(\boldsymbol{w}_{n,j}^{(t)})])+B
\end{aligned}$$

$$= \eta \sum_{j=0}^{J-1} \underbrace{-\mathbb{E}[\nabla f_n(\boldsymbol{w}^{(t)})^T (\sum_{n=1}^{N} p_n \nabla f(\boldsymbol{w}_{n,j}^{(t)}))]}_{C} + B, \tag{58}$$

where (a) is by the facts that $a_n^{(t)}$ and $g_n(\cdot)$ are independent, $\mathbb{E}[a_n^{(t)}] = q_n^{(t)}$, and $\mathbb{E}[g_n(\boldsymbol{w}_{n,j}^{(t)})] = \mathbb{E}[\nabla f_n(\boldsymbol{w}_{n,j}^{(t)})]$; and $B = \mathbb{E}[\| \sum_{n=1}^{N} \frac{p_n a_n^{(t)}}{q_n^{(t)}} \sum_{j=0}^{J-1} g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2]$ has the same definition as in the proof of Theorem 1 in Appendix A.

We now give a bound on C as follows

$$\begin{aligned}
&- \mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T (\sum_{n=1}^{N} p_n \nabla f_n(\boldsymbol{w}_{n,j}^{(t)}))] \\
&= -\mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T (\sum_{n=1}^{N} p_n \nabla f_n(\boldsymbol{w}_{n,j}^{(t)}) - \nabla f(\boldsymbol{w}^{(t)}) + \nabla f(\boldsymbol{w}^{(t)}))] \\
&= \mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T (\nabla f(\boldsymbol{w}^{(t)}) - \sum_{n=1}^{N} p_n \nabla f_n(\boldsymbol{w}_{n,j}^{(t)}))] - \mathbb{E}[\nabla f(\boldsymbol{w}^{(t)})^T \nabla f(\boldsymbol{w}^{(t)})] \\
&\overset{(a)}{\leq} \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] - \mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] + \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)}) - \sum_{n=1}^{N} p_n \nabla f_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&= \frac{1}{2}\mathbb{E}[\| \sum_{n=1}^{N} p_n (\nabla f_n(\boldsymbol{w}^{(t)}) - \nabla f_n(\boldsymbol{w}_{n,j}^{(t)}))\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(b)}{\leq} \frac{1}{2} \sum_{n=1}^{N} p_n \mathbb{E}[\|\nabla f_n(\boldsymbol{w}^{(t)}) - \nabla f_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(c)}{\leq} \frac{\beta^2}{2} \sum_{n=1}^{N} p_n \mathbb{E}[\|\boldsymbol{w}^{(t)} - \boldsymbol{w}_{n,j}^{(t)}\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(d)}{\leq} \frac{\beta^2}{2} \sum_{n=1}^{N} p_n \mathbb{E}[\| \sum_{i=0}^{j-1} \eta g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\overset{(e)}{\leq} \frac{\eta^2 \beta^2 j}{2} \sum_{n=1}^{N} p_n \sum_{i=0}^{j-1} \mathbb{E}[\|g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] - \frac{1}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2],
\end{aligned} \tag{59}$$

where (a) is by $\boldsymbol{a}^T\boldsymbol{b} \leq (\boldsymbol{a}^2 + \boldsymbol{b}^2)/2$, (b) is by Jensen's inequality, (c) is by the β-smoothness of each f_n in Assumption 5, (d) is by the substituting the local update rule in (27), and (e) is by Jensen's inequality.

Combining the bounds of C in (59) and B in (54), we obtain

$$\begin{aligned}
&\mathbb{E}[f(\boldsymbol{w}^{(t+1)})] - f(\boldsymbol{w}^{(t)}) \\
&\leq \frac{\eta^3 \beta^2}{2} \sum_{n=1}^{N} p_n \sum_{j=0}^{J-1} j \sum_{i=0}^{j-1} \mathbb{E}[\|g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] - \frac{\eta J}{2}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\quad + \frac{\beta \eta^2 J}{2} \sum_{n=1}^{N} \frac{p_n}{q_n^{(t)}} \sum_{j=0}^{J-1} \mathbb{E}[\|g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2].
\end{aligned}$$

Rearranging terms, summing over t from 0 to $T-1$, taking total expectation, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(\boldsymbol{w}^{(t)})\|^2] \\
&\leq \frac{2(f(\boldsymbol{w}^{(0)})-f^*)}{\eta TJ} + \frac{\eta^2\beta^2}{TJ}\sum_{t=0}^{T-1}\sum_{n=1}^{N}p_n\sum_{j=0}^{J-1}i\sum_{i=0}^{j-1}\mathbb{E}[\|g_n(\boldsymbol{w}_{n,i}^{(t)})\|^2] + \frac{\beta\eta}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\frac{p_n}{q_n^{(t)}}\sum_{j=0}^{J-1}\mathbb{E}[\|g_n(\boldsymbol{w}_{n,j}^{(t)})\|^2] \\
&\leq \frac{2(f(\boldsymbol{w}^{(0)})-f^*)}{\eta TJ} + \frac{\eta^2\beta^2(J-1)(2J-1)}{6T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}p_nG_n^2 + \frac{\beta\eta J}{T}\sum_{t=0}^{T-1}\sum_{n=1}^{N}\frac{p_n}{q_n^{(t)}}(G_n^{(t)})^2.
\end{aligned}$$

□

C CLOSED FORM SOLUTION WITH CONSTRAINTS ON EXPECTED NUMBER OF SAMPLED CLIENTS

Suppose we want to select $m \in (0, N]$ clients on average in each training round. For simplicity, let $A_n = p_n(G_n^{(t)})^2 > 0$, $x_n = q_n^{(t)}$, and $\boldsymbol{x} = [x_1, \ldots, x_N]$. The general optimization problem can be formulated as

$$\min_{\boldsymbol{x}} \quad \sum_{n=1}^{N}\frac{A_n}{x_n} \tag{60}$$

$$\text{s. t.} \quad \sum_{n=1}^{N}x_n = m \tag{61}$$

$$0 \leq x_n \leq 1, \quad \forall n \in [N]. \tag{62}$$

It is a convex optimization problem and any solution satisfying the KKT conditions is optimal. Closed-form solutions can be derived from the KKT conditions (similar to [12, Lemma 2]). Without loss of generality, we assume $0 < A_1 \leq A_2 \leq \cdots \leq A_N$. Let k be the largest integer such that $0 < m + k - N \leq \frac{\sum_{i=1}^{k}\sqrt{A_i}}{\sqrt{A_k}}$, which always holds for $k = N - m + 1$. Then the closed-form solution of x_n is:

$$x_n = \begin{cases} (m+k-N)\frac{\sqrt{A_n}}{\sum_{i=1}^{k}\sqrt{A_i}}, & \text{if } n \leq k, \\ 1, & \text{if } n > k, \end{cases} \tag{63}$$

In our case, we have

$$q_n^{(t)} = \begin{cases} (m+k-N)\frac{\sqrt{p_n}G_n^{(t)}}{\sum_{i=1}^{k}\sqrt{p_i}G_i^{(t)}}, & \text{if } n \leq k, \\ 1, & \text{if } n > k, \end{cases} \tag{64}$$

where k is the largest integer such that $0 < m + k - N \leq \frac{\sum_{i=1}^{k}\sqrt{p_i}G_i^{(t)}}{\sqrt{p_k}G_k^{(t)}}$ holds. Hence, a subset of clients is assigned with sampling probability 1, and the others are assigned sampling probability proportional to $\sqrt{p_n}G_n^{(t)}$.