# Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling

Bing Luo*†§, Wenli Xiao†*, Shiqiang Wang‡, Jianwei Huang†*, Leandros Tassiulas§
*Shenzhen Institute of Artificial Intelligence and Robotics for Society, China
†School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China
‡IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
§Department of Electrical Engineering and Institute for Network Science, Yale University, USA
Email: {luobing, wenlixiao, jianweihuang}@cuhk.edu.cn, shiqiang.wang@ieee.org, leandros.tassiulas@yale.edu

***Abstract*—Federated learning (FL) algorithms usually sample a fraction of clients in each round (partial participation) when the number of participants is large and the server's communication bandwidth is limited. Recent works on the convergence analysis of FL have focused on unbiased client sampling, e.g., sampling uniformly at random, which suffers from slow wall-clock time for convergence due to high degrees of system heterogeneity and statistical heterogeneity. This paper aims to design an adaptive client sampling algorithm that tackles both system and statistical heterogeneity to minimize the wall-clock convergence time. We obtain a new tractable convergence bound for FL algorithms with arbitrary client sampling probabilities. Based on the bound, we analytically establish the relationship between the total learning time and sampling probabilities, which results in a non-convex optimization problem for training time minimization. We design an efficient algorithm for learning the unknown parameters in the convergence bound and develop a low-complexity algorithm to approximately solve the non-convex problem. Experimental results from both hardware prototype and simulation demonstrate that our proposed sampling scheme significantly reduces the convergence time compared to several baseline sampling schemes. Notably, our scheme in hardware prototype spends 73% less time than the uniform sampling baseline for reaching the same target loss.**

## I. Introduction

Federated learning (FL) enables many clients[1] to collaboratively train a model under the coordination of a central server while keeping the training data decentralized and private (e.g., [1]–[3]). Compared to traditional distributed machine learning techniques, FL has two unique features (e.g., [4]–[9]), as shown in Fig. 1. First, clients are massively distributed and with diverse and low communication rates (known as *system heterogeneity*), where stragglers can slow down the physical training time.[2] Second, the training data are distributed in a non-i.i.d. and unbalanced fashion across the clients (known as *statistical heterogeneity*), which negatively affects the convergence behavior.

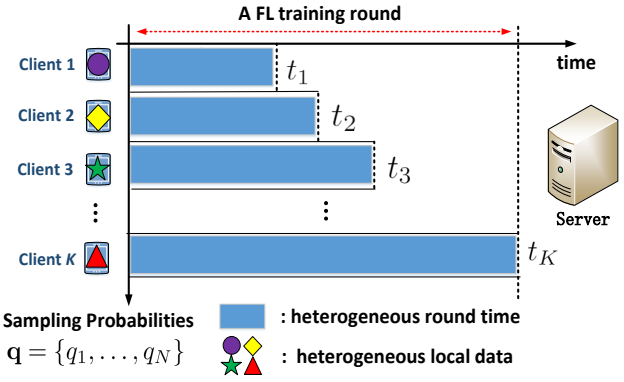



Fig. 1. An FL training round with system and statistical heterogeneity, with $K$ out of $N$ clients sampled according to probability $\mathbf{q} = \{q_1, \ldots, q_N\}$.

Due to limited communication bandwidth and across geographically dispersed devices, FL algorithms (e.g., the de facto FedAvg algorithm in [3]) usually perform multiple local iterations on *a fraction of randomly sampled clients (known as partial participation)* and then aggregates their resulting local model updates via the central server periodically [3]–[6]. Recent works have provided theoretical convergence analysis that demonstrates the effectiveness of FL with partial participation in various non-i.i.d. settings [15]–[19].

However, these prior works [15]–[19] have focused on sampling schemes that select clients uniformly at random or proportional to the clients' data sizes, which often suffer from slow convergence with respect to wall-clock (physical) time[3] due to high degrees of the system and statistical heterogeneity. This is because the total FL time depends on *both the number of training rounds* for reaching the target precision and *the physical time in each round* [20]. Although uniform sampling guarantees that the aggregated model update in each round is unbiased towards that with full client participation, the aggregated model may have a high variance due to data heterogeneity, thus, *requiring more training rounds to converge to a target precision*. Moreover, considering clients' heterogeneous communication delay, uniform sampling also suffers from the straggling effect, as the probability of sampling a straggler

The work of Bing Luo, Wenli Xiao, and Jianwei Huang was supported by the Shenzhen Science and Technology Program (JCYJ20210324120011032), Shenzhen Institute of Artificial Intelligence and Robotics for Society, and the Presidential Fund from the Chinese University of Hong Kong, Shenzhen. The work of Leandros Tassiulas was supported by the AI Institute for Edge Computing Leveraging Next Generation Networks (Athena) under Grant NSF CNS-2112562 and Grant NRL N00173-21-1-G006. (Corresponding author: Jianwei Huang.)

[1]We use "device" and "client" interchangeably in this paper.

[2]As suggested [10]–[14], we consider mainstream synchronized FL in this paper due to its composability with other techniques (such as secure aggregation protocols and differential privacy).

[3]We use wall-clock time to distinguish from the number of training rounds.

within the sampled subset in each round can be relatively high,[4] thus *yielding a long per-round time*.

One effective way of speeding up the convergence with respect to the number of training rounds is to choose clients according to some sampling distribution where "important" clients have high probabilities [21]–[24]. For example, recent works adopted importance sampling approaches based on clients' statistical property [25]–[28]. However, their sampling schemes did not account for the heterogeneous physical time in each round, especially under straggling circumstances. Another line of works aims to minimize the learning time via optimizing client selection and scheduling based on their heterogeneous system resources [20], [29]–[41]. However, their optimization schemes did not consider how client selection schemes influence the convergence behavior due to data heterogeneity and thus may negatively affect the total learning time.

In a nutshell, the fundamental limitation of existing works is the *lack of joint consideration of the impact of the inherent system heterogeneity and statistical heterogeneity on client sampling*. In other words, clients with valuable data may have poor communication capabilities, whereas those who communicate fast may have low-quality data. This motivates us to study the following key question.

**Key Question:** *How to design an optimal client sampling scheme that tackles both system and statistical heterogeneity to achieve fast convergence with respect to wall-clock time?*

The challenge of this question is threefold: (1) It is difficult to obtain an analytical FL convergence result for arbitrary client sampling probabilities. (2) The total learning time minimization problem can be complex and non-convex due to the straggling effect. (3) The optimal client sampling solution contains unknown parameters from the convergence result, which we can only estimate during the learning process (known as the *chicken-and-egg problem*).

In light of the above discussion, we state the main results and key contributions of this paper as follows:

- *Optimal Client Sampling for Heterogeneous FL:* We study how to design the optimal client sampling strategy to minimize FL wall-clock time with convergence guarantees. To the best of our knowledge, this is the first work that aims to optimize client sampling probabilities to address both system and statistical heterogeneity.
- *Convergence Bound for Arbitrary Sampling:* Using an adaptive client sampling and model aggregation design, we obtain a new tractable convergence upper bound for FL algorithms with arbitrary client sampling probabilities. This enables us to establish the analytical relationship between the total learning time and client sampling probabilities and formulate a non-convex training time minimization problem.
- *Optimization Algorithm and Sampling Principle:* We propose a low-cost substitute sampling approach to learn the convergence-related unknown parameters and develop an efficient algorithm to approximately solve the non-convex problem with low computational complexity. Our solution characterizes the impact of communication time (system heterogeneity) and data quantity and quality (statistical heterogeneity) on the optimal client sampling design.
- *Simulation and Prototype Experimentation:* We evaluate the performance of our proposed algorithms through both a simulated environment and a hardware prototype. Experimental results demonstrate that for both convex and non-convex learning models, our proposed sampling scheme significantly reduces the convergence time compared to several baseline sampling schemes. For example, with our hardware prototype and the EMNIST dataset, our sampling scheme spends 73% less time than baseline uniform sampling for reaching the same target loss.

[4]For example, suppose there are 100 clients with only 5 stragglers, then the probability of sampling at least a straggler for uniformly sampling 10 clients in each round is more than 40%.

## II. Related Work

Active client sampling and selection play a crucial role in addressing the statistical and system heterogeneity challenges in cross-device FL. In the existing literature, the research efforts in speeding up the training process mainly focus on two aspects: importance sampling and resource-aware optimization-based approaches.

define The goal of importance sampling is to reduce the variance in traditional optimization algorithms based on stochastic gradient descent (SGD), where SGD draws data samples uniformly at random during the learning process (e.g., [21]–[24]). Recent works have adopted this idea in FL systems to improve communication efficiency via designing client sampling strategy. Specifically, clients with "important" data would have higher probabilities to be sampled in each round. For example, existing works use clients' local gradient information (e.g., [25]–[27]) or local losses (e.g., [28]) to measure the importance of clients' data. However, these schemes did not consider the speed of error convergence with respect to *wall-clock time*, especially the straggling effect due to heterogeneous transmission delays.

Another line of works aims to minimize wall-clock time via resource-aware optimization-based approaches, such as CPU frequency allocation (e.g., [29]), and communication bandwidth allocation (e.g., [30], [31]), straggler-aware client scheduling (e.g., [20], [32]–[37]), parameters control (e.g., [36]–[39]), and task offloading (e.g., [40], [41]). While these papers provided some novel insights, their optimization approaches did not consider how client sampling affects the total wall-clock time and thus are orthogonal to our work.

Unlike all the above-mentioned works, our work focuses on how to design the optimal client sampling strategy that tackles both system and statistical heterogeneity to minimize the wall-clock time with convergence guarantees. In addition, most existing works on FL are based on computer simulations. In contrast, we implement our algorithm in an actual hardware prototype with resource-constrained devices, which allows us to capture real system operations.

The organization of the rest of the paper is as follows. Section III introduces the system model and problem formulation. Section IV presents our new error-convergence bound with arbitrary client sampling. Section V gives the optimal client sampling algorithm and solution insights. Section VI provides the simulation and prototype experimental results. We conclude this paper in Section VII.

## III. Preliminaries and System Model

We start by summarizing the basics of FL and its de facto algorithm FedAvg with unbiased client sampling. Then, we introduce the proposed adaptive client sampling for statistical and system heterogeneity based on FedAvg. Finally, we present our formulated optimization problem.

### A. Federated Learning (FL)

Consider a federated learning system involving a set of $\mathcal{N} = 1, \ldots, N$ clients, coordinated by a central server. Each client $i$ has $n_i$ local training data samples $(\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,n_i})$, and the total number of training data across $N$ devices is $n_{\text{tot}} := \sum_{i=1}^{N} n_i$. Further, define $f(\cdot,\cdot)$ as a loss function where $f(\mathbf{w}; \mathbf{x}_{i,j})$ indicates how the machine learning model parameter $\mathbf{w}$ performs on the input data sample $\mathbf{x}_{i,j}$. Thus, the local loss function of client $i$ can be defined as

$$F_i(\mathbf{w}) := \frac{1}{n_i} \sum\nolimits_{j=1}^{n_i} f(\mathbf{w}; \mathbf{x}_{i,j}). \tag{1}$$

Denote $p_i = \frac{n_i}{n_{\text{tot}}}$ as the weight of the $i$-th device such that $\sum_{i=1}^{N} p_i = 1$. Then, by denoting $F(\mathbf{w})$ as the global loss function, the goal of FL is to solve the following optimization problem [1]:

$$\min_{\mathbf{w}} F(\mathbf{w}) := \sum\nolimits_{i=1}^{N} p_i F_i(\mathbf{w}). \tag{2}$$

The most popular and de facto optimization algorithm to solve (2) is FedAvg [3]. Here, denoting $r$ as the index of an FL *round*, we describe one round (e.g., the $r$-th) of the FedAvg algorithm as follows:

1) The server *uniformly at random samples* a subset of $K$ clients (i.e., $K := |\mathcal{K}^r|$ with $\mathcal{K}^r \subseteq \mathcal{N}$) and broadcasts the latest model $\mathbf{w}^r$ to the selected clients.
2) Each sampled client $i$ chooses $\mathbf{w}_i^{r,0} = \mathbf{w}^r$, and runs $E$ steps[5] of local SGD on (1) to compute an updated model $\mathbf{w}_i^{r,E}$. Then, the sampled client lets $\mathbf{w}_i^{r+1} = \mathbf{w}_i^{r,E}$ and send it back to the server.
3) The server *aggregates* (with weight $p_i$) the clients' updated model and computes a new global model $\mathbf{w}^{\tau+1}$.

The above process repeats for many rounds until the global loss converges.

Recent works have demonstrated the effectiveness of FedAvg with theoretical convergence guarantees in various settings [15]–[19]. However, these works assume that the server samples clients either uniformly at random or proportional to data size, which may slow down the wall-clock time for convergence due to the straggling effect and non-i.i.d. data [20]. Thus, a careful client sampling design should tackle both system and statistical heterogeneity for fast convergence.

[5] $E$ is originally defined as epochs of SGD in [3]. In this paper, we denote $E$ as the number of local iterations for theoretical analysis.

### B. System Model of FL with Client Sampling $\mathbf{q}$

We aim to sample clients according to a probability distribution $\mathbf{q} = \{q_i, \forall i \in \mathcal{N}\}$, where $0 < q_i < 1$ and $\sum_{i=1}^{N} q_i = 1$. Through optimizing $\mathbf{q}$, we want to address system and statistical heterogeneity so as to minimize the wall-clock time for convergence. We describe the system model as follows.

*1) Sampling Model:* Following recent works [6], [15]–[19], we assume that the server establishes the sampled client set $\mathcal{K}(\mathbf{q})^r$ by sampling $K$ times *with replacement* from the total $N$ clients, where $\mathcal{K}(\mathbf{q})^r$ is a *multiset* in which a client may appear more than once. The aggregation weight of each client $i$ is multiplied by the number of times it appears in $\mathcal{K}(\mathbf{q})^r$.

*2) Statistical Heterogeneity Model:* We consider the standard FL setting where the training data are distributed in an unbalanced and non-i.i.d. fashion among clients.

*3) System Heterogeneity Model:* Following the same setup of [9] and [20], we denote $t_i$ as the round time of client $i$, which includes both local model computation time and global communication time. For simplicity, we assume that $t_i$ remains the same across different rounds for each client $i$, while for different clients $i$ and $j$, $t_i$ and $t_j$ can be different. The extension to time-varying $t_i$ is left for future work. Without loss of generality, as illustrated in Fig. 1, we sort all $N$ clients in the ascending order $\{t_i\}$, such that

$$t_1 \leq t_2 \leq \ldots \leq t_i \leq \ldots \leq t_N. \tag{3}$$

*4) Total Wall-clock Time Model:* We consider the mainstream synchronized FL model where each sampled client performs multiple (e.g., $E$) steps of local SGD before sending back their model updates to the server (e.g., [3], [4], [15]–[19]). For synchronous FL, the per-round time is limited by the slowest client (known as straggler). Thus, the per-round time $T^{(r)}(\mathbf{q})$ of the entire FL process is

$$T^{(r)}(\mathbf{q}) := \max_{i \in \mathcal{K}(\mathbf{q})^{(r)}} \{t_i\}. \tag{4}$$

Therefore, the total learning time $T_{\text{tot}}(\mathbf{q}, R)$ after $R$ rounds is

$$T_{\text{tot}}(\mathbf{q}, R) = \sum\nolimits_{r=1}^{R} T^{(r)}(\mathbf{q}) = \sum\nolimits_{r=1}^{R} \max_{i \in \mathcal{K}(\mathbf{q})^r} \{t_i\}. \tag{5}$$

### C. Problem Formulation

Our goal is to minimize the expected total learning time $\mathbb{E}[T_{\text{tot}}(\mathbf{q}, R)]$, while ensuring that the expected global loss $\mathbb{E}[F(\mathbf{w}^R(\mathbf{q}))]$ converges to the minimum value $F^*$ with an $\epsilon$ precision, with $\mathbf{w}^R(\mathbf{q})$ being the aggregated global model after $R$ rounds with client sampling probabilities $\mathbf{q}$. This translates into the following problem:

$$\begin{aligned} \textbf{P1:} \quad \min_{\mathbf{q}, R} \quad & \mathbb{E}[T_{\text{tot}}(\mathbf{q}, R)] \\ \text{s.t.} \quad & \mathbb{E}[F(\mathbf{w}^R(\mathbf{q}))] - F^* \leq \epsilon, \\ & \textstyle\sum_{i=1}^{N} q_i = 1, \\ & q_i > 0, \forall i \in \mathcal{N},\ R \in \mathbb{Z}^+. \end{aligned} \tag{6}$$

The expectation in $\mathbb{E}[T_{\text{tot}}(\mathbf{q}, R)]$ and $\mathbb{E}[F(\mathbf{w}^R(\mathbf{q}))]$ in (6) is due to the randomness in client sampling $\mathbf{q}$ and local SGD. Solving Problem **P1**, however, is challenging in two aspects:

1) It is generally impossible to find out how $\mathbf{q}$ and $R$ affect the final model $\mathbf{w}^R(\mathbf{q})$ and the corresponding loss function $\mathbb{E}[F\left(\mathbf{w}^R(\mathbf{q})\right)]$ before actually training the model. Hence, we need to obtain an analytical expression with respect to $\mathbf{q}$ and $R$ to predict how they affect $\mathbf{w}^R(\mathbf{q})$ and $\mathbb{E}[F\left(\mathbf{w}^R(\mathbf{q})\right)]$.
2) The objective $\mathbb{E}[T_{\text{tot}}(\mathbf{q}, R)]$ is complicated to optimize due to the straggling effect in (5), which can result in a non-convex optimization problem even for simplest cases as we will show later.

In Section IV and Section V, we address these two challenges, respectively, and propose approximate algorithms to find an approximate solution to Problem **P1** efficiently.

## IV. Convergence Bound for Arbitrary Sampling

In this section, we address the first challenge by deriving a new tractable convergence bound for arbitrary client sampling probabilities.

### A. Machine Learning Model Assumptions

To ensure a tractable convergence analysis, we first state several assumptions on the local objective functions $F_i(\mathbf{w})$.

**Assumption 1.** *L-smooth: For each client $i \in \mathcal{N}$, $F_i$ is L-smooth, i.e., $\|\nabla f(\mathbf{v}) - \nabla f(\mathbf{w})\| \leq L\|\mathbf{v}-\mathbf{w}\|$ for all $\mathbf{v}$ and $\mathbf{w}$.*

**Assumption 2.** *Strongly-convex: For each client $i \in \mathcal{N}$, $F_i$ is $\mu$-strongly convex, i.e., $F_i(\mathbf{v}) \geq F_i(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_i(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$ for all $\mathbf{v}$ and $\mathbf{w}$.*

**Assumption 3.** *Bounded local variance: For each device $i \in \mathcal{N}$, the variance of its stochastic gradient is bounded: $\mathbb{E}\left\|\nabla F_i\left(\mathbf{w}_i, \xi_i\right) - \nabla F_i\left(\mathbf{w}_i\right)\right\|^2 \leq \sigma_i^2$.*

**Assumption 4.** *Bounded local gradient: For each client $i \in \mathcal{N}$, the expected squared norm of stochastic gradients is bounded: $\mathbb{E}\left\|\nabla F_i\left(\mathbf{w}_i, \xi_i\right)\right\|^2 \leq G_i^2$.*

Assumptions 1–3 are common in many existing studies of convex FL problems, such as $\ell_2$-norm regularized linear regression, logistic regression (e.g., [7], [18], [19], [25], [28], [42]). Nevertheless, the experimental results to be presented in Section VI show that our approach also works well for *non-convex* loss functions. Assumption 4, however, is a less restricted version of the assumption made in [7], [18], [19], [25], [28], [42], where those studies have assumed that $G_i$ is uniformly bounded by a universal $G$. Instead, we allow each client $i$ to have a unique $G_i$, which yields our optimal client sampling design as we will show later.

### B. Aggregation with Arbitrary Client Sampling Probabilities

This section shows how to aggregate clients' model updates under sampling probabilities $\mathbf{q}$, such that the aggregated global model is unbiased compared to that with full client participation, which leads to our convergence result.

We first define the *virtual weighted aggregated model with full client participation* in round $r$ as

$$\overline{\mathbf{w}}^{r+1} := \sum\nolimits_{i=1}^{N} p_i \mathbf{w}_i^{r+1}. \tag{7}$$

With this, we can derive the following result.

**Algorithm 1:** FL with Arbitrary Client Sampling

**Input:** Sampling probabilities $\mathbf{q} = \{q_1, \ldots, q_N\}$, $K$, $E$, precision $\epsilon$, initial model $\mathbf{w_0}$
**Output:** Final model parameter $\mathbf{w}^R$

1 **for** $r \leftarrow 0, 1, 2, ..., R$ **do**
2 *Server randomly samples a subset of clients $\mathcal{K}(\mathbf{q})^r$ according to $\mathbf{q}$, and sends current global model $\mathbf{w}^r$ to the selected clients;* // **Sampling**
3 Each sampled client $i$ lets $\mathbf{w}_i^{r,0} \leftarrow \mathbf{w}^r$, and performs $\mathbf{w}_i^{r,j+1} \leftarrow \mathbf{w}_i^{r,j} - \eta^r \nabla F_k\left(\mathbf{w}_i^{r,j}, \xi_i^{r,j}\right), j = 0, 1, \ldots, E\text{–}1$, and lets $\mathbf{w}_i^{r+1} \leftarrow \mathbf{w}_i^{r,E}$ ; // Computation
4 Each sampled client $i$ sends back updated model $\mathbf{w}_i^{r+1}$ to the server ; // Communication
5 *Server computes a new global model parameter as* $\mathbf{w}^{r+1} \leftarrow \mathbf{w}^r + \sum_{i \in \mathcal{K}(\mathbf{q})^r} \frac{p_i}{K q_i}\left(\mathbf{w}_i^{r+1} - \mathbf{w}^r\right)$ ; // **Aggregation**

**Lemma 1.** ***(Adaptive Client Sampling and Model Aggregation)*** *When clients $\mathcal{K}(\mathbf{q})^r$ are sampled with probability $\mathbf{q} = \{q_1, \ldots q_N\}$ and their local updates are aggregated as $\mathbf{w}^{r+1} \leftarrow \mathbf{w}^r + \sum_{i \in \mathcal{K}(\mathbf{q})^r} \frac{p_i}{K q_i}\left(\mathbf{w}_i^{r+1} - \mathbf{w}^r\right)$, we have*

$$\mathbb{E}_{\mathcal{K}(\mathbf{q})^r}[\mathbf{w}^{r+1}] = \overline{\mathbf{w}}^{r+1}. \tag{8}$$

*Proof Sketch.* The basic idea is to take expectation over the aggregated global model of the sampled clients $\mathcal{K}(\mathbf{q})^r$, and with some mathematical derivations, we have (8). □

**Remark**: The key insight of our sampling and aggregation is that since we sample different clients with different probabilities (e.g., $q_i$ for client $i$), *we need to inversely re-weight their updated model in the aggregation step (e.g., $\frac{1}{q_i}$ for client $i$), such that the aggregated model is still unbiased towards that with full client participation*. We summarize how the server performs client sampling and model aggregation in Algorithm 1, where the main differences compared to the de facto FedAvg in [3] are the *Sampling* (Line 2) and *Aggregation* (Line 5) procedures. Notably, Algorithm 1 recovers FedAvg algorithm with uniform sampling when letting $q_i = \frac{1}{N}$, or with weighted sampling when letting $q_i = p_i$ in [18].

### C. Main Convergence Result for Arbitrary Client Sampling

Based on Lemma 1, we present the main convergence result for arbitrary client sampling in Theorem 1.

**Theorem 1.** ***(Convergence Upper Bound)*** *Let Assumptions 1 to 4 hold, $\gamma = \max\{\frac{8L}{\mu}, E\}$, and decaying learning rate $\eta_r = \frac{2}{\mu(\gamma + r)}$. For given client sampling probabilities $\mathbf{q} = \{q_1, \ldots, q_N\}$ and the corresponding aggregation described in Lemma 1, the optimality gap after $R$ rounds satisfies*

$$\mathbb{E}[F\left(\mathbf{w}^R(\mathbf{q})\right)] - F^* \leq \frac{1}{R}\left(\alpha \sum\nolimits_{i=1}^{N} \frac{p_i^2 G_i^2}{q_i} + \beta\right), \tag{9}$$

*where $\alpha = \frac{8LE}{\mu^2 K}$ and $\beta = \frac{2L}{\mu^2 E} B + \frac{12L^2}{\mu^2 E}\Gamma + \frac{4L^2}{\mu E}\left\|\mathbf{w}_0 - \mathbf{w}^*\right\|^2$, with $B = \sum_{i=1}^{N} p_i^2 \sigma_i^2 + 8\sum_{i=1}^{N} p_i G_i^2 E^2$ and $\Gamma = F^* - \sum_{i=1}^{N} p_i F_i^*$.*

*Proof Sketch.* First, following the similar proof of convergence under full client participation in [18], [42], we show that $\mathbb{E}[F\left(\overline{\mathbf{w}}^R\right)] - F^* \leq \frac{\beta}{R}$, where $\mathbb{E}[F\left(\overline{\mathbf{w}}^R\right)]$ is the expected global loss after $R$ rounds with full participation, and $\beta$ is the same as in (9). Then, for client sampling probabilities $\mathbf{q}$,

as we have shown that the expected aggregated global model $\mathbb{E}_{\mathcal{K}(\mathbf{q})^r}[\mathbf{w}^{r+1}]$ is unbiased compared to full participation $\overline{\mathbf{w}}^{r+1}$ in Lemma 1, we can show that the expected difference of the two (sampling variance) is bounded as follows:

$$\mathbb{E}_{\mathcal{K}(\mathbf{q})^r}\left\|\mathbf{w}^{r+1}-\overline{\mathbf{w}}^{r+1}\right\|^2 \leq \frac{4}{K}\sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i}\left(\eta^r E\right)^2. \quad (10)$$

After that we use induction to obtain a non-recursive bound on $\mathbb{E}_{\mathcal{K}(\mathbf{q})^r}\left\|\mathbf{w}^R-\mathbf{w}^*\right\|^2$, which is converted to a bound on $\mathbb{E}[F\left(\mathbf{w}^R(\mathbf{q})\right)]-F^*$ using $L$-smoothness. Finally, we show that the main difference of the contraction bound compared to full client participation is the sampling variance in (10), which yields the additional term of $\alpha\sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i}$ in (9). □

Our convergence bound in (9) characterizes the relationship between client sampling probabilities $\mathbf{q}$ and the number of rounds $R$ for reaching the target precision ($\mathbb{E}[F\left(\overline{\mathbf{w}}^R\right)]-F^*\leq\epsilon$). Notably, our bound generalizes the convergence results in [18], where clients are uniformly sampled ($q_i=\frac{1}{N}$) or weighted sampled ($q_i=p_i$). Moreover, our convergence bound also motivates the optimal client sampling design for homogeneous systems, i.e., all clients with the same communication time $t_0$, as follows.

**Corollary 1.** *For FL with homogeneous communication time i.e., $t_i=t_0$, for all $i\in\mathcal{N}$, the optimal client sampling probabilities $\mathbf{q}$ for Problem **P1** is*

$$q_i^*=p_iG_i\Big/\sum_{j=1}^{N}p_jG_j\ . \quad (11)$$

*Proof.* When clients have the same communication time, the round time $T^{(r)}(\mathbf{q})$ in (4) is fixed as $t_0$, and thus *minimizing the total learning time $\mathbb{E}[T_{\text{tot}}(\mathbf{q},R)]$ in Problem **P1** is equivalent to minimizing the total number of communication rounds* for reaching the target precision $\epsilon$. Hence, by letting $\mathbb{E}[F\left(\mathbf{w}(\mathbf{q},R)\right)]-F^*=\epsilon$, and by moving $R$ in (9) from the right hand side to the left hand side of inequality, we have

$$R\leq\frac{1}{\epsilon}\left(\alpha\sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i}+\beta\right). \quad (12)$$

Thus, for a target precision $\epsilon$, computing the optimal sampling $\mathbf{q}$ for minimizing the upper bound of $R$ is equivalent to solving

$$\begin{aligned}\min_{\mathbf{q}}\quad & \sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i}\\ \text{s.t.}\quad & \sum_{i=1}^{N}q_i=1,\ q_i>0,\forall i\in\mathcal{N}.\end{aligned} \quad (13)$$

The problem in (13) can be easily solved with the Lagrange multiplier method in closed form as shown in (11). □

In the following, we show how to leverage the derived convergence bound in Theorem 1 to design the optimal client sampling for the general heterogeneous system of Problem **P1**.

## V. Optimal Adaptive Client Sampling Algorithm

In this section, we first obtain the analytical expression of the expected total learning time $\mathbb{E}[T_{\text{tot}}]$ with sampling probabilities $\mathbf{q}$ and training round $R$. Then, we formulate an approximate problem of the original Problem **P1** based on the convergence upper bound in Theorem 1. Finally, we develop an efficient algorithm to solve the new problem with insightful sampling principles.

### A. Analytical Expression for $\mathbb{E}[T_{\text{tot}}(\mathbf{q})]$

**Theorem 2.** *The expected total learning time $\mathbb{E}[T_{\text{tot}}(\mathbf{q},R)]$ is*

$$\mathbb{E}[T_{\text{tot}}(\mathbf{q},R)]=\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i}q_j\right)^K-\left(\sum_{j=1}^{i-1}q_j\right)^K\right]t_iR. \quad (14)$$

*Proof Sketch.* The idea is to show that with sampling probabilities $\mathbf{q}$, the expected per-round time $\mathbb{E}[T^{(r)}(\mathbf{q})]$ in (4) is

$$\mathbb{E}[T^{(r)}(\mathbf{q})]=\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i}q_j\right)^K-\left(\sum_{j=1}^{i-1}q_j\right)^K\right]t_i. \quad (15)$$

We first show that the probability of client $i$ being the slowest one (e.g., straggler) amongst the $K$ sampled clients in each round is $(\sum_{j=1}^{i}q_j)^K-(\sum_{j=1}^{i-1}q_j)^K$. Since we sample devices according to $\mathbf{q}$, taking the expectation of all $N$ clients over time $q_i$ gives (15), and for $R$ rounds we have (14). □

### B. Approximate Optimization Problem for Problem *P1*

Based on Theorem 2, and by letting the analytical convergence upper bound in (9) satisfy the convergence constraint,[6] the original Problem **P1** can be approximated as

$$\begin{aligned}\textbf{P2:}\ \min_{\mathbf{q},R}\ & \sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i}q_j\right)^K-\left(\sum_{j=1}^{i-1}q_j\right)^K\right]t_iR\\ \text{s.t.}\quad & \frac{1}{R}\left(\alpha\sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i}+\beta\right)\leq\epsilon,\\ & \sum_{i=1}^{N}q_i=1,\\ & q_i>0,\forall i\in\mathcal{N},\ R\in\mathbb{Z}^+.\end{aligned} \quad (16)$$

Combining with (9), we can see that Problem **P2** is more constrained than Problem **P1**, i.e., any feasible solution of Problem **P2** is also feasible for Problem **P1**.

We further relax $R$ as a continuous variable to theoretically analyze Problem **P2**. For this relaxed problem, suppose ($\mathbf{q}^*$, $R^*$) is the optimal solution, then we must have

$$\frac{1}{R^*}\left(\alpha\sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i^*}+\beta\right)=\epsilon. \quad (17)$$

This is because if (17) holds with an inequality, we can always find an $R'<R^*$ that satisfies (17) with equality, but the solution ($q^*$, $R'$) can further reduce the objective function value. Therefore, for the optimal $R$, (17) always holds, and we can obtain $R$ from (17) and substitute it into the objective of Problem **P2**. Then, the objective of Problem **P2** is

$$\left(\sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i}q_j\right)^K-\left(\sum_{j=1}^{i-1}q_j\right)^K\right]t_i\right)\left(\alpha\sum_{i=1}^{N}\frac{p_i^2G_i^2}{q_i}+\beta\right), \quad (18)$$

which[7] is only associated with client sampling probabilities $\mathbf{q}$.

The objective function (18), however, is still difficult to optimize because the sampling probabilities $\mathbf{q}$ is in a polynomial sum with an order $K$. For analytical tractability, we define an approximation of $\mathbb{E}[T^{(r)}(\mathbf{q})]$ as

$$\tilde{\mathbb{E}}[T^{(r)}(\mathbf{q})]:=\sum_{i=1}^{N}q_it_i. \quad (19)$$

The approximation $\tilde{\mathbb{E}}[T^{(r)}(\mathbf{q})]$ is exactly the same as $\mathbb{E}[T^{(r)}(\mathbf{q})]$ in the following two cases.

[6]Optimization using upper bound as an approximation has also been adopted in [29], [30], [38], [39].

[7]For ease of analysis, we omit $\epsilon$ as it is a constant multiplied by the entire objective function.

*Case 1*: For homogeneous $t_i$ ($t_i = t_0, \forall i \in \mathcal{N}$), we have

$$\begin{aligned}\mathbb{E}[T^{(r)}(\mathbf{q})] &= \sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i} q_j\right)^K - \left(\sum_{j=1}^{i-1} q_j\right)^K\right] t_0 \\ &= \left[\left(\sum_{j=1}^{N} q_j\right)^K - 0\right] t_0 = \left[1^K - 0^K\right] t_0 \\ &= \sum_{i=1}^{N} q_i t_i = \tilde{\mathbb{E}}[T^{(r)}(\mathbf{q})].\end{aligned} \tag{20}$$

*Case 2*: For heterogeneous $t_i$ with $K = 1$, we have

$$\begin{aligned}\mathbb{E}[T^{(r)}(\mathbf{q})] &= \sum_{i=1}^{N}\left[\left(\sum_{j=1}^{i} q_j\right)^K - \left(\sum_{j=1}^{i-1} q_j\right)^K\right] t_i \\ &= \sum_{i=1}^{N} q_i t_i = \tilde{\mathbb{E}}[T^{(r)}(\mathbf{q})].\end{aligned} \tag{21}$$

For the general case, we can consider $\tilde{\mathbb{E}}[T^{(r)}(\mathbf{q})]$ as an approximation to $\mathbb{E}[T^{(r)}(\mathbf{q})]$. Using this approximation, Problem **P2** can be expressed as

$$\begin{aligned}\textbf{P3:}\ \min_{\mathbf{q}}\quad & \left(\sum_{i=1}^{N} q_i t_i\right)\left(\alpha \sum_{i=1}^{N} \frac{p_i^2 G_i^2}{q_i} + \beta\right) \\ \text{s.t.}\quad & \sum_{i=1}^{N} q_i = 1,\ q_i > 0, \forall i \in \{1, \dots N\}.\end{aligned} \tag{22}$$

**Remark:** The objective function of Problem **P3** is in a more straightforward form compared to Problem **P2**. However, to solve for the optimal sampling probabilities $\mathbf{q}$, we need to know the value of the parameters in (22), e.g., $G_i$, $\alpha$, and $\beta$.[8]

In the following, we solve Problem **P3** as an approximation of the original Problem **P1**. Our empirical results in Section VI demonstrate that the solution obtained from solving Problem **P3** achieves superior total wall-clock time performances compared to baseline client sampling schemes.

### C. Solving Problem *P3*

Problem **P3** is challenging to solve because we can only obtain the unknown parameters $G_i$, $\alpha$ and $\beta$ during the training process of FL. In this subsection, we first show how to estimate these unknown parameters. Then, we develop an efficient algorithm to solve Problem **P3**. We summarize the overall algorithm in Algorithm 2. Finally, we identify some insightful solution properties.

*1) **Estimation of Parameters** $G_i$ **and** $\frac{\alpha}{\beta}$:* We first show how to estimate $\frac{\alpha}{\beta}$ via a substitute sampling scheme.[9] Then, we show that we can indirectly acquire the knowledge of $G_i$ during the estimation process of $\frac{\alpha}{\beta}$.

The basic idea is to utilize the derived convergence upper bound in (9) to approximately solve $\frac{\alpha}{\beta}$ as a single variable, via performing Algorithm 1 with two baseline sampling schemes: uniform sampling $\mathbf{q_1}$ with $q_i = \frac{1}{N}$ and weighted sampling $\mathbf{q_2}$ with $q_i = p_i$, respectively.

Note that we only let sampling schemes $\mathbf{q_1}$ and $\mathbf{q_2}$ run until a pre-defined loss $F_s$ is reached (instead of running all the way until they converge to the precision $\epsilon$), because our goal is to find and run with the optimal sampling scheme $\mathbf{q}^*$ so that we can achieve the target precision with the minimum wall-clock time.

[8]We assume that clients' heterogeneous time $t_i$ and their dataset size $p_i$ can be measured offline.

[9]We only need to estimate the value of $\frac{\alpha}{\beta}$ instead of $\alpha$ and $\beta$ each, because we can divide parameter $\alpha$ on the objective of Problem **P3** without affecting the optimal sapling solution.

**Algorithm 2:** Approximate Optimal Client Sampling for FL with System and Statistical Heterogeneity

**Input:** $N$, $K$, $E$, $t_i$, $p_i$, $\mathbf{w}_0$, loss $F_s$, precision $\epsilon_0$
**Output:** Approximation of $\mathbf{q}^*$

1 **for** $s \leftarrow 1, 2, \dots, S$ **do**
2  Server runs Algorithm 1 with uniform sampling $\mathbf{q_1}$ and weighted sampling $\mathbf{q_2}$, respectively;
3  The sampled clients send back their local gradient norm information along with their updated models;
4  Server updates all clients' $G_i$ based on the received gradient norms;
5  Server records $R_{\mathbf{q_1},s}$ and $R_{\mathbf{q_2},s}$ when reaching $F_s$;
6 Calculate average $\frac{\alpha}{\beta}$ using (24);
7 **for** $M(\epsilon_0) \leftarrow t_1, t_1 + \epsilon_0, t_1 + 2\epsilon_0 \dots, t_N$ **do**
8  Substitute $M(\epsilon_0)$, $\frac{\beta}{\alpha}$, $N$, $t_i$, $p_i$, $G_i$ into **P4**;
9  Solve **P4** via CVX, and obtain $\mathbf{q}^*(M(\epsilon_0))$
10 **return** $\mathbf{q}^*(M^*(\epsilon_0)) \leftarrow \arg\min_{M(\epsilon_0)} \mathbf{q}^*(M(\epsilon_0))$

Specifically, suppose $R_{\mathbf{q_1},s}$ and $R_{\mathbf{q_2},s}$ are the number of rounds for reaching the pre-defined loss $F_s$ for schemes $\mathbf{q_1}$ and $\mathbf{q_2}$, respectively. Considering that the training loss decreases quickly at the beginning and slowly when approaching convergence, the values of $R_{\mathbf{q_1}}$ and $R_{\mathbf{q_2}}$ are normally very small compared to the required number of rounds for reaching the target loss (with precision $\epsilon$). According to (9), we have

$$\begin{cases}(F_s - F^*)\, R_{\mathbf{q_1},s} \approx \alpha N \sum_{i=1}^{N} p_i^2 G_i^2 + \beta, \\ (F_s - F^*)\, R_{\mathbf{q_2},s} \approx \alpha \sum_{i=1}^{N} p_i G_i^2 + \beta.\end{cases} \tag{23}$$

Based on (23), we have

$$\frac{R_{\mathbf{q_1},s}}{R_{\mathbf{q_2},s}} \approx \frac{\alpha N \sum_{i=1}^{N} p_i^2 G_i^2 + \beta}{\alpha \sum_{i=1}^{N} p_i G_i^2 + \beta}. \tag{24}$$

Then, we can obtain $\frac{\alpha}{\beta}$ from (24) once we know the value of $G_i$. Notably, we can estimate $G_i$ during the procedure of estimating $\frac{\alpha}{\beta}$. The idea is to let the sampled clients send back the norm of their local SGD along with their returned local model updates, and then the server updates $G_i$ with the received norm values. This approach does not add much communication overhead, since we only need to additionally transmit the value of the gradient norm (e.g., only a few bits for quantization) instead of the full gradient information. In addition, instead of retraining the model using the calculated $\mathbf{q}^*$ from the initial parameter $\mathbf{w_0}$, we can continue to train the global model after the estimation process, to avoid repeated training and reduce the overall training time.

In practice, due to the sampling variance, we may set several different $F_s$ to obtain an averaged estimation of $\frac{\alpha}{\beta}$. The overall estimation process corresponds to Lines 1–6 of Algorithm 2.

*2) **Optimization Algorithm for** $\mathbf{q}^*$:* We first identify the property of Problem **P3** and then show how to compute $\mathbf{q}^*$.

**Theorem 3.** *Problem **P3** is non-convex.*

*Proof Sketch.* The idea is to show that the Hessian of the objective function in Problem **P3** is not positive semi-definite. For example, for $N = 2$ case, we have $\frac{\partial^2 \tilde{\mathbb{E}}[T_{\text{tot}}]}{\partial^2 q_1} = \frac{2\alpha q_2 t_2 p_1^2 G_1^2}{q_1^3} > 0$, whereas $\frac{\partial^2 \tilde{\mathbb{E}}[T_{\text{tot}}]}{\partial^2 q_1}\frac{\partial^2 \tilde{\mathbb{E}}[T_{\text{tot}}]}{\partial^2 q_2} - \left(\frac{\partial^2 \tilde{\mathbb{E}}[T_{\text{tot}}]}{\partial q_1 \partial q_2}\right)^2 = -\alpha^2\left(\frac{t_1 p_2^2 G_2^2}{q_2^2} - \frac{t_2 p_1^2 G_1^2}{q_1^2}\right)^2 \leq 0$. □

To solve Problem **P3**, we define a new control variable

$$M := \sum_{i=1}^{N} q_i t_i, \tag{25}$$

where $t_1 \leq M \leq t_N$. Then, we rewrite Problem **P3** as

$$\begin{aligned} \textbf{P4:} \quad & \min_{\mathbf{q},M} \quad g(\mathbf{q}, M) := M \cdot \left(\alpha \textstyle\sum_{i=1}^{N} \frac{p_i^2 G_i^2}{q_i} + \beta\right) \\ & \text{s.t.} \quad \textstyle\sum_{i=1}^{N} q_i = 1, \\ & \quad\;\; \textstyle\sum_{i=1}^{N} q_i t_i = M, \\ & \quad\;\; q_i > 0, \forall i \in \mathcal{N}. \end{aligned} \tag{26}$$

For any fixed feasible value of $M \in [t_1, t_N]$, Problem **P4** is convex with $\mathbf{q}$, because the objective function is strictly convex and the constraints are linear.

We will solve Problem **P4** in two steps. First, for any fixed $M$, we will solve for the optimal $\mathbf{q}^*(M)$ in Problem **P4**, via a convex optimization tool, e.g., CVX [43]. This allows us to write the objective function of Problem P4 as $g(\mathbf{q}^*(M), M)$. Then we will further solve the problem by using a linear search method with a fixed step-size $\epsilon_0$ over the interval $[t_1, t_N]$, where we use the optimal $M^*(\epsilon_0)$ and the corresponding $\mathbf{q}^*(M^*(\epsilon_0))$ in the search domain to approximate the optimal $M^*$ and $\mathbf{q}^*$ in Problem **P4**. This optimization process corresponds to Lines 7–10 of Algorithm 2.

**Remark**: Our optimization algorithm is efficient in the sense that the linear search domain $[t_1, t_N]$ is independent of the scale of the problem, e.g., number of $N$.

*3) **Property of Optimal Client Sampling**:* Next we show some interesting properties of the optimal sampling strategy.

**Theorem 4.** *Suppose $\mathbf{q}^*$ is the optimal solution of Problem **P3**. For two different clients $i$ and $j$, if $t_i \leq t_j$ and $p_i G_i \geq p_j G_j$, then $q_i^* \geq q_j^*$.*

*Proof Sketch.* The idea is to show by contradiction that if $q_i^* < q_j^*$, we can simply let $q_i' = q_j^*, q_j' = q_i^*$ such that $q_i' > q_j'$ and achieve a smaller $\mathbb{E}[T_{\text{tot}}(q_i', q_j')]$ than $\mathbb{E}[T_{\text{tot}}(q_i^*, q_j^*)]$. □

Theorem 4 shows that the optimal client sampling strategy should allocate higher probabilities to those who have smaller $t_i$ and larger product value of $p_i G_i$, which characterizes the impact and interplay between system heterogeneity and statistical heterogeneity. Although it may be infeasible to derive an analytical relationship regarding the exact impact of $t_i$ and $p_i G_i$ on $\mathbf{q}^*$ due to non-convexity of Problem **P3**, we show by Corollary 2 that we can obtain the closed-form solution of $\mathbf{q}^*$ with $t_i$ and $p_i G_i$ when $\frac{\beta}{\alpha} \to 0$.

**Corollary 2.** *When $\frac{\beta}{\alpha} \to 0$, the global optimal solution of Problem **P3** is*

$$q_i^* = \frac{p_i G_i}{\sqrt{t_i}} \Big/ \sum_{j=1}^{N} \frac{p_j G_j}{\sqrt{t_j}}. \tag{27}$$

*Proof.* If $\frac{\beta}{\alpha} \to 0$, because $\sum_{i=1}^{N} q_i t_i$ is bounded between $[t_1, t_N]$, we have $\left(\sum_{i=1}^{N} q_i t_i\right) \frac{\beta}{\alpha} \to 0$. Then, the objective of Problem **P3** can be written as

$$\min_{\mathbf{q}} \left(\textstyle\sum_{i=1}^{N} q_i t_i\right) \left(\textstyle\sum_{i=1}^{N} \frac{p_i^2 G_i^2}{q_i}\right). \tag{28}$$

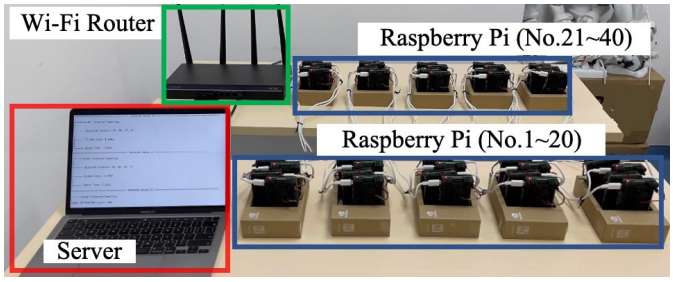



Fig. 2. Hardware prototype with the laptop serving as the central server and 40 Raspberry Pis serving as clients. During the FL experiments, we place the router 5 meters away from all the devices.

By Cauchy-Schwarz inequality, we have

$$\left(\sum_{i=1}^{N} \left(\sqrt{q_i t_i}\right)^2\right) \left(\sum_{i=1}^{N} \left(\frac{p_i G_i}{\sqrt{q_i}}\right)^2\right) \geq \left(\sum_{i=1}^{N} \sqrt{t_i} \cdot p_i G_i\right)^2. \tag{29}$$

Hence, the minimum of (28) is $\left(\sum_{i=1}^{N} \sqrt{t_i} \cdot p_i G_i\right)^2$, which is independent of $\mathbf{q}$. The equality of (29) holds if and only if $\sqrt{q_i t_i} = c \cdot \frac{p_i G_i}{\sqrt{q_i}}$ (for an arbitrary scalar $c$). Noting $\sum_{i=1}^{N} q_i = 1$ yields (27) and concludes the proof. □

Though Corollary 2 is valid only for the special case of $\frac{\beta}{\alpha} \to 0$, the global optimal sampling solution in (27) characterizes an analytical interplay between the system heterogeneity ($t_i$) and statistical heterogeneity ($p_i G_i$). Particularly, when $t_i = t_0$, for each $i \in \mathcal{N}$, $\mathbf{q}^*$ in (27) recovers the optimal sampling solution in Corollary 1 for homogeneous systems.

## VI. Experimental Evaluation

In this section, we empirically evaluate the performance of our proposed client sampling scheme (Algorithm 2) and compare it with four other benchmarks in each round: 1) *full participation*, 2) *uniform sampling*, 3) *weighted sampling*, and 4) *statistical sampling* where we sample clients according to Corollary 1. Benchmarks 1–3 are widely adopted for convergence guarantees in [15]–[19]. The fourth baseline is an offline variant of the proposed schemes in [25], [26].[10]

In the following, we first present the evaluation setup and then show the experimental results.

### A. Experimental Setup

*1) Platforms:* We conduct experiments both on a networked hardware prototype system and in a simulated environment.[11] As illustrated in Fig. 2, our prototype system consists of $N = 40$ Raspberry Pis serving as clients and a laptop computer acting as the central server. All devices are interconnected via an enterprise-grade Wi-Fi router. We develop a TCP-based socket interface for the communication between the server and clients with bandwidth control. In the simulated system, we simulate $N = 100$ virtual devices and a virtual central server.

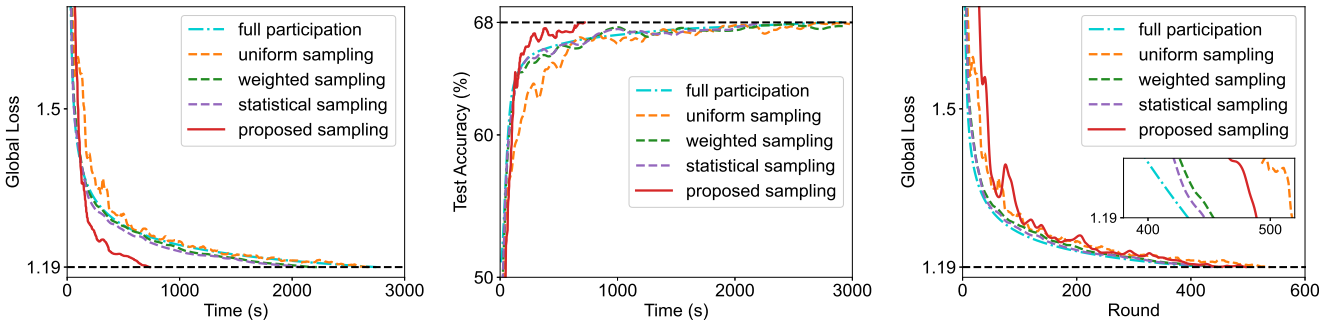
[10]The client sampling in [25], [26] is weighted by the norm of the local stochastic gradient in each round, which frequently requires the knowledge of stochastic gradient from all clients to calculate the sampling probabilities.

[11]The prototype implementation allows us to capture real system operation time, and the simulation system allows us to simulate large-scale FL environments with manipulative parameters.

TABLE I
PERFORMANCES OF WALL-CLOCK TIME FOR REACHING TARGET LOSS FOR DIFFERENT SAMPLING SCHEMES

| Setup \ Sampling scheme | **proposed sampling** | statistical sampling | weighted sampling | uniform sampling | full participation |
|---|---|---|---|---|---|
| **Prototype Setup** (EMNIST dataset) | **733.2 s** | 2095.0 s (**2.9**×) | 2221.7 s (**3.0**×) | 2691.5 s (**3.7**×)† | 2748.4 s (**3.7**×) |
| **Simulation Setup 1** (Synthetic dataset) | **445.5 s** | 952.4 s (**2.1**×) | 940.2 s (**2.1**×) | 933.8 s (**2.1**×) | 1526.8 s (**3.4**×) |
| **Simulation Setup 2** (MNIST dataset) | **245.5 s** | 373.8 s (**1.5**×) | 542.9 s (**2.2**×) | NA | 898.1 s (**3.7**×) |

† "3.7×" represents the wall-clock time ratio of uniform sampling over proposed sampling for reaching the target loss, which is equivalent to proposed sampling takes 73% less time than uniform sampling.




(a) Loss with wall-clock time (b) Accuracy with wall-clock time (c) Loss with number of rounds

Fig. 3. Performance of **Prototype Setup** with logistic regression model, EMNIST dataset, uniform communication time, and target loss 1.19.

*2) Datasets and Models:* We evaluate our results on two real datasets and a synthetic dataset. For the real dataset, we adopted the widely used MNIST dataset and EMNIST dataset [6]. For the synthetic dataset, we follow a similar setup to that in [18], which generates 60-dimensional random vectors as input data. We adopt both the *convex* multinomial logistic regression model and the *non-convex* convolutional neural network (CNN) model with LeNet-5 architecture [44].

*3) Implementation:* we consider three experimental setups.

- **Prototype Setup**: We conduct the first experiment on the prototype system using logistic regression and the EMNIST dataset. To generate heterogeneous data partition, similar to [6], we randomly subsample $33,036$ lower case character samples from the EMNIST dataset and distribute among $N=40$ edge devices in an *unbalanced* (i.e., different devices have different numbers of data samples, following a power-law distribution) and *non-i.i.d.* fashion (i.e., each device has a randomly chosen number of classes, ranging from 1 to 10).[12]
- **Simulation Setup 1**: We conduct the second experiment in the simulated system using logistic regression and the Synthetic dataset. To simulate a heterogeneous setting, we use the non-i.i.d. $Synthetic\ (1,1)$ setting. We generate $20,509$ data samples and distribute them among $N=100$ clients in an *unbalanced* power-law distribution.
- **Simulation Setup 2**: We conduct the third experiment in the simulated system using CNN and the MNIST dataset, where we randomly subsample $15,129$ data samples from MNIST and distribute them among $N=100$ clients in an *unbalanced* (following the power-law distribution) and *non-i.i.d.* (i.e., each device has 1–6 classes) fashion.

*4) Training Parameters:* For all experiments, we initialize our model with $\mathbf{w}_0=\mathbf{0}$ and use an SGD batch size of $b=24$. We use an initial learning rate of $\eta_0=0.1$ with a decay rate of $\frac{\eta_0}{1+r}$, where $r$ is the communication round index. We adopt the similar FedAvg settings as in [4], [18], [25], [32], where we sample 10% of all clients in each round, i.e., $K=4$ for Prototype Setup and $K=10$ for Simulation Setups, with each client performing $E=50$ local iterations.[13]

*5) Heterogeneous System Parameters:* For the Prototype Setup, to enable a heterogeneous communication time, we control clients' communication bandwidth and generate a uniform distribution $t_i \sim \mathcal{U}(0.187, 7.159)$ seconds, with a mean of $3.648$ seconds and the standard deviation of $2.071$ seconds. For the simulation system, we generate the client transmission delays with an exponential distribution, i.e., $t_i \sim \exp(1)$ seconds, with both mean and standard deviation as 1 second.

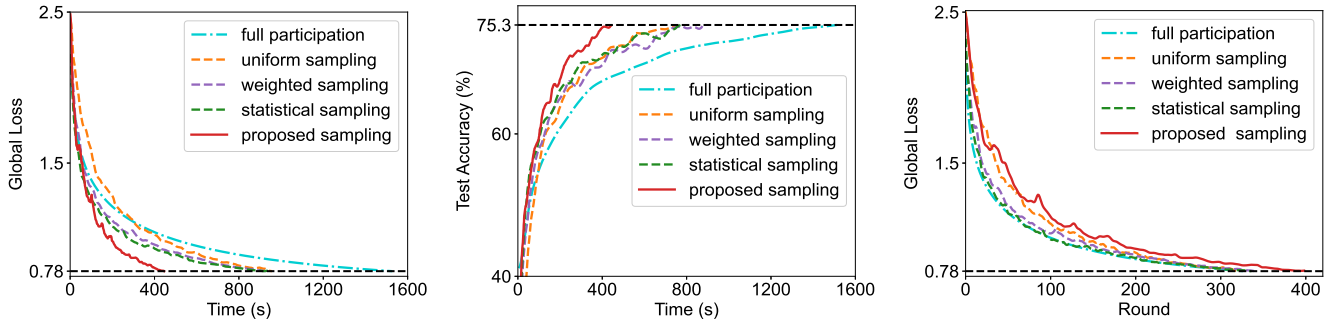
### B. Performance Results

We evaluate the wall-clock time performances of both the global training loss and test accuracy on the aggregated model in each round for all sampling schemes. We average each experiment over 50 independent runs. For a fair comparison, we use the same random seed to compare sampling schemes in a single run and vary random seeds across different runs.

Fig. 3–5 show the results of Prototype Setup, Simulation Setup 1, and Simulation Setup 2, respectively. We summarize the key observations as follows.

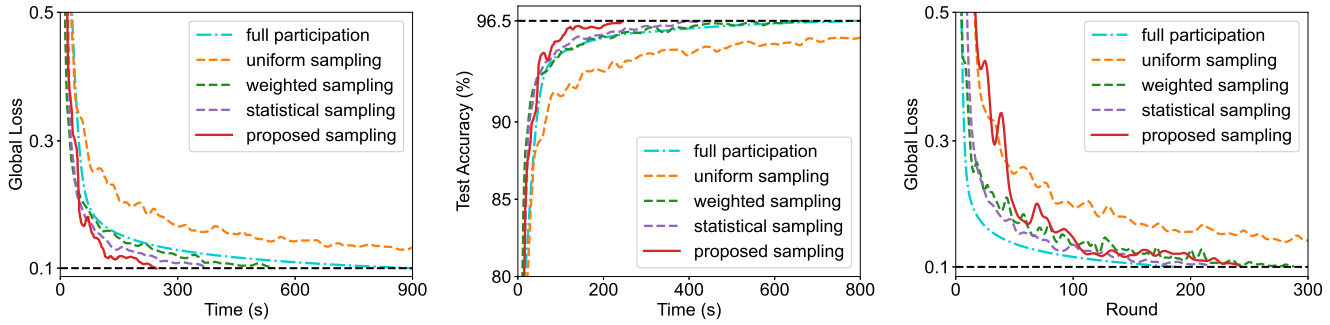
[12]The number of samples and the number of classes are randomly matched, such that clients with more data samples may not have more classes.

[13]We also conduct experiments both on Prototype and Simulation Setups with variant $E$ and $K$, which show a similar performance as the experiments in this paper, and due to page limitations, we do not illustrate them all.




(a) Loss with wall-clock time (b) Accuracy with wall-clock time (c) Loss with number of rounds

Fig. 4. Performance of **Simulation Setup 1** with logistic regression model, $Synthetic\ (1,1)$ dataset, exponential communication time, and target loss 0.78.




(a) Loss with wall-clock time (b) Accuracy with wall-clock time (c) Loss with number of rounds

Fig. 5. Performance of **Simulation Setup 2** with CNN model, MNIST dataset, exponential communication time, and target loss 0.1.

*1) Loss with Wall-clock Time:* As predicted by our theory, Figs. 3(a)–5(a) show that *our proposed sampling scheme achieves the same target loss with significantly less time*, compared to the baseline sampling schemes. Specifically, for Prototype Setup in Fig. 3(a), our proposed sampling scheme spends around 73% less time than full sampling and uniform sampling and around 66% less time than weighted sampling and statistical sampling for reaching the same target loss. Fig. 5(a) highlights the fact that our proposed sampling works well with the non-convex CNN model, under which the naive uniform sampling cannot reach the target loss within 900 seconds, indicating the importance of a careful client sampling design. Table I summarizes the superior performances of our proposed sampling scheme in wall-clock time for reaching target loss in all three setups.

*2) Accuracy with Wall-clock Time:* As shown in Fig. 3(b)–5(b), our proposed sampling scheme *achieves the target test accuracy*[14] *much faster than the other benchmarks*. Notably, for Simulation Setup 1 with the target test accuracy of 75.3% in Fig. 4(b), our proposed sampling scheme takes around 70% less time than full sampling and around 46% less time than the other sampling schemes. We can also observe the superior test accuracy performance of our proposed sampling schemes in Prototype Setup and non-convex Simulation Setup 2 in Fig. 3(b) and Fig. 5(b), respectively.

[14]In Fig. 3(b), Fig. 4(b), and Fig. 5(b), the target test accuracy corresponds to the test accuracy result when our proposed scheme reaches the target loss.

*3) Loss with Number of Rounds:* Fig. 3(c)–5(c) show that our proposed sampling scheme requires more training rounds for reaching the target loss compared to baseline statistical sampling and full participation schemes. This observation is expected since our proposed sampling scheme *aims to minimize the wall-clock time instead of the number of rounds.* Nevertheless, we notice that statistical sampling performs better than the other sampling schemes, which verifies Corollary 1 since the performance of loss with respect to the number of rounds is equivalent to that with respect to wall-clock time for homogeneous systems.

## VII. Conclusion and Future Work

In this work, we studied the optimal client sampling strategy that addresses the system and statistical heterogeneity in FL to minimize the wall-clock convergence time. We obtained a new tractable convergence bound for FL algorithms with arbitrary client sampling probabilities. Based on the bound, we formulated a non-convex wall-clock time minimization problem. We developed an efficient algorithm to learn the unknown parameters in the convergence bound and designed a low-complexity algorithm to approximately solve the non-convex problem. Our solution characterizes the interplay between clients' communication delays (system heterogeneity) and data importance (statistical heterogeneity), and their impact on the optimal client sampling design. Experimental results validated the superiority of our proposed scheme compared to several baselines in speeding up wall-clock convergence time.

## REFERENCES


[1] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[2] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[4] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan *et al.*, "Towards federated learning at scale: System design," in *Proceedings of Machine Learning and Systems (MLSys)*, 2019.

[5] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[6] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

[7] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD for non-convex optimization with faster convergence and less communication," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[8] H. Yu, R. Jin, and S. Yang, "On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 7184–7193.

[9] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," in *Proceedings of Machine Learning and Systems (MLSys)*, 2019.

[10] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federated learning on user-held data," in *NeurIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[11] B. Avent, A. Korolova, D. Zeber, T. Hovden, and B. Livshits, "BLENDER: Enabling local search with a hybrid differential privacy model," in *USENIX Security Symposium (USENIX Security)*, 2017, pp. 747–764.

[12] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *NeurIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[13] M. Zhang, E. Wei, and R. Berry, "Faithful edge federated learning: Scalability and privacy," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3790–3804, 2021.

[14] P. Sun, H. Che, Z. Wang, Y. Wang, T. Wang, L. Wu, and H. Shao, "Pain-fl: Personalized privacy-preserving incentive for federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3805–3820, 2021.

[15] F. Haddadpour and M. Mahdavi, "On the convergence of local descent methods in federated learning," *arXiv preprint arXiv:1910.14425*, 2019.

[16] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for on-device federated learning," *arXiv preprint arXiv:1910.06378*, 2019.

[17] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-iid federated learning," *arXiv preprint arXiv:2101.11203*, 2021.

[18] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *Proceedings of the International Conference on Learning Representation (ICLR)*, 2019.

[19] Z. Qu, K. Lin, J. Kalagnanam, Z. Li, J. Zhou, and Z. Zhou, "Federated learning's blessing: Fedavg has linear speedup," *arXiv preprint arXiv:2007.05690*, 2020.

[20] R. Amirhossein, T. Isidoros, H. Hamed, M. Aryan, and P. Ramtin, "Straggler-resilient federated learning: Leveraging the interplay between statistical accuracy and system heterogeneity," *arXiv preprint arXiv:2012.14453*, 2020.

[21] P. Zhao and T. Zhang, "Stochastic optimization with importance sampling for regularized loss minimization," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2015, pp. 1–9.

[22] D. Needell, R. Ward, and N. Srebro, "Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm," in *Advances in neural information processing systems*, 2014, pp. 1017–1025.

[23] G. Alain, A. Lamb, C. Sankar, A. Courville, and Y. Bengio, "Variance reduction in SGD by distributed importance sampling," *arXiv preprint arXiv:1511.06481*, 2015.

[24] S. Gopal, "Adaptive sampling for SGD by exploiting side information," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2016, pp. 364–372.

[25] W. Chen, S. Horvath, and P. Richtarik, "Optimal client sampling for federated learning," *arXiv preprint arXiv:2010.13723*, 2020.

[26] E. Rizk, S. Vlaski, and A. H. Sayed, "Federated learning under importance sampling," *arXiv preprint arXiv:2012.07383*, 2020.

[27] H. T. Nguyen, V. Sehwag, S. Hosseinalipour, C. G. Brinton, M. Chiang, and H. V. Poor, "Fast-convergent federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 201–218, 2021.

[28] Y. J. Cho, J. Wang, and G. Joshi, "Client selection in federated learning: Convergence analysis and power-of-choice selection strategies," *arXiv preprint arXiv:2010.01243*, 2020.

[29] N. H. Tran, W. Bao, A. Zomaya, N. M. NH, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2019, pp. 1387–1395.

[30] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.

[31] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 453–467, 2021.

[32] T. Nishio and R. Yonetani, "Client selection for federated learning with heterogeneous resources in mobile edge," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2019, pp. 1–7.

[33] Z. Chai, A. Ali, S. Zawad, S. Truex, A. Anwar, N. Baracaldo, Y. Zhou, H. Ludwig, F. Yan, and Y. Cheng, "Tifl: A tier-based federated learning system," in *Proceedings of the International Symposium on High-Performance Parallel and Distributed Computing*, 2020, pp. 125–136.

[34] H. H. Yang, Z. Liu, T. Q. Quek, and H. V. Poor, "Scheduling policies for federated learning in wireless networks," *IEEE Transactions on Communications*, vol. 68, no. 1, pp. 317–333, 2019.

[35] Y. Jin, L. Jiao, Z. Qian, S. Zhang, S. Lu, and X. Wang, "Resource-efficient and convergence-preserving online participant selection in federated learning," in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2020.

[36] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3606–3621, 2021.

[37] H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing federated learning on non-iid data with reinforcement learning," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 1698–1707.

[38] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[39] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[40] Y. Tu, Y. Ruan, S. Wagle, C. G. Brinton, and C. Joe-Wong, "Network-aware optimization of distributed learning for fog computing," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2020.

[41] S. Wang, M. Lee, S. Hosseinalipour, R. Morabito, M. Chiang, and C. G. Brinton, "Device sampling for heterogeneous federated learning: Theory, algorithms, and implementation," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2021.

[42] S. U. Stich, "Local SGD converges fast and communicates little," in *Proceedings of the International Conference on Learning Representation (ICLR)*, 2018.

[43] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.