PSFL: Parallel-Sequential Federated Learning with Convergence Guarantees

Jinrui Zhou*, Yu Zhao*, Yin Xu*, Mingjun Xiao*, Jie Wu†, and Sheng Zhang‡
*School of Computer Science and Technology & Suzhou Institute for Advanced Research
& State Key Laboratory of Cognitive Intelligenc, University of Science and Technology of China, China
†Department of Computer and Information Sciences, Temple University, USA
‡Department of Computer Science and Technology, Nanjing University, China
Corresponding Author: Yin Xu (yinxu@ustc.edu.cn)

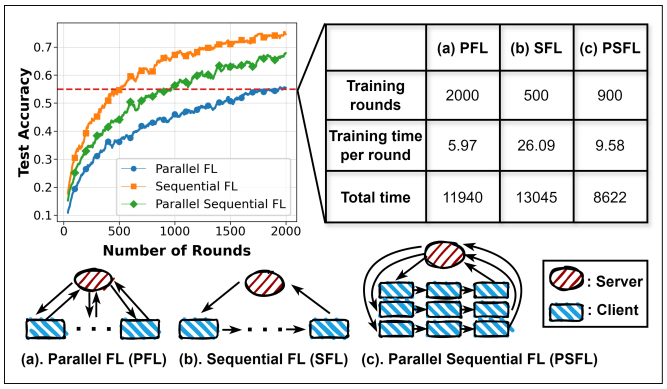
***Abstract*—Federated Learning (FL) is a novel distributed learning paradigm which can coordinate multiple clients to jointly train a machine learning model by using their local data samples. Existing FL works can be roughly divided into two categories according to the modes of model training: Parallel FL (PFL) and Sequential FL (SFL). PFL can speed up each round of model training time through parallel training, but it might suffer from the convergence degradation when facing the heterogeneity issue. SFL can deal with the heterogeneity issue well to reduce the number of training rounds, but it will spend more time in each round of local model training due to the sequential mode. In this paper, we propose a novel hybrid Parallel-Sequential Federated Learning (PSFL) framework by integrating the parallel and sequence training modes together. We derive the upper bounds of the model convergence and the expected total training time for the PSFL framework through theoretical analysis. Based on the results, we find out the optimal training structure and design a client sampling strategy, which can balance the two training modes and guarantee the unbiasedness. Extensive experiments validate our theoretical analysis and demonstrate the significant performance of the PSFL framework.**

***Index Terms*—Sequential Federated Learning, Heterogeneity, Client Sampling.**

I. Introduction

In recent years, Federated Learning (FL) [1] has emerged as a novel distributed learning paradigm that enables the training of machine learning models across multiple devices (a.k.a., clients) holding local data samples. FL addresses the critical issues of data security by keeping the raw data localized and aggregating the local models under the coordination of a central server. Many works have been devoted to investigating different FL issues, such as heterogeneity [2]–[15], communication [16]–[18], security [19]–[24], and so on.

Existing FL works can be roughly divided into two categories according to the modes of model training: Parallel FL (PFL) and Sequential FL (SFL) [25]. In the PFL mode, clients train local models in parallel before aggregating them to produce the global model, e.g., the well-known FedAvg algorithm [1] and its diverse variants. Many techniques are adopted to deal with the convergence degradation of model training incurred by heterogeneous clients, such as adaptive client or data sampling [4]–[8], grouped aggregation [9]–[13], configuration parameter tuning [14]–[17], etc. In the SFL mode, clients train and aggregate their local models in sequence. Such mode naturally has the ability to solve the heterogeneity issue, especially for small datasets training [26].



	(a) PFL	(b) SFL	(c) PSFL
Training rounds	2000	500	900
Training time per round	5.97	26.09	9.58
Total time	11940	13045	8622

Fig. 1. An example of FL with different training modes: (a). a standard FL setting where clients train models in a parallel manner; (b). the sequential FL setting where clients train models in a sequential manner; (c). a hybrid parallel-sequential FL framework where clients train models in both parallel and sequential manners.

The PFL and SFL modes exhibit two opposing characteristics when facing the heterogeneity issue. PFL can speed up each round of model training time while SFL can reduce the number of training rounds. Fig. 1 illustrates a typical heterogeneous FL example with different training modes (the detailed settings of FL are described in Sec. VI), in which PFL requires about four times as many training rounds as SFL to achieve the same accuracy. Conversely, the single-round training time of SFL is more than four times that of PFL. Both training modes exhibit their extremes that are not conducive to achieving the goal of minimizing total training time. To fill the gap between PFL and SFL, we propose a novel hybrid Parallel-Sequential Federated Learning (PSFL) framework that adopts parallel and sequential training simultaneously: clients are grouped into multiple sequences for model training and all sequences conduct model training tasks in parallel, as shown in Fig. 1.(c). Actually, there are two crucial challenges to design an efficient PSFL framework.

First, there is a trade-off between parallel and sequential training for PSFL to minimize the total training time. Specifically, increasing the size of each training sequence (i.e., the number of clients in the sequence) can reduce the number of training rounds to make model convergence, but it will

also increase the training time per round due to the sequential training mode. On the contrary, decreasing the size of each training sequence will reduce the training time per round, but it will also increase the heterogeneity of parallel training, so that more training rounds are required to make model convergence. Then, the first challenge is how to balance the trade-off between parallel and sequential training so that PSFL can minimize the total training time.

Second, PSFL needs to schedule clients onto different training sequences by dynamically sampling partial clients for each training round. Due to the heterogeneous capabilities of clients in terms of computation and communication, the scheduling issue can be transformed into a variant of the multi-machine scheduling problem [27], where each sequence acts as a "machine" and each client as a specific "job" to be scheduled. However, most existing approaches, which will schedule clients to various sequences based on their training time, might result in biased sampling probabilities. That is, the datasets of sampled clients might fail to accurately represent those of all clients [28], making the trained model deviate from the optimization objective. Thus, the second challenge is how to develop a novel client sampling strategy that minimizes training time while ensuring unbiasedness.

To address the above challenges, we formulate an optimization problem of minimizing the expected total training time for PSFL, taking into consideration the trade-off between parallel and sequential training as well as unbiased client sampling. Then, we derive a convergence upper bound and a closed formula about the upper bound of the expected total training time through theoretical analysis, whereby the problem is converted into a non-convex integer optimization problem to be solved. Finally, we design a training time-based partitioning and sampling strategy to guarantee the unbiasedness of client sampling. The major contributions are summarized as follows:

- We propose a novel hybrid PSFL framework by integrating the parallel and sequential model training modes together, which can speed up the convergence of model training in heterogeneous scenarios.
- We conduct a rigorous theoretical analysis to derive the upper bounds of the model convergence and the expected total training time for the PSFL framework, based on which we turn the balance issue between parallel and sequential training to a non-convex integer optimization problem to be solved.
- We design a time-based partitioning and sampling strategy for the PSFL framework, which can guarantee the unbiasedness of client sampling.
- We conduct extensive simulations to verify the theoretical analysis results and demonstrate the significant performance of the PSFL framework.

II. Related Works

We review the related works from the following aspects:

Parallel Federated Learning (PFL): PFL, represented by the predominant and standard FedAvg [1], faces system and statistical heterogeneity challenges. Statistical heterogeneity complicates model convergence, while the waiting time caused by system heterogeneity hampers training speed. A large number of works have been proposed to solve statistical heterogeneity, such as meta-learning [29], multi-task learning [30], knowledge distillation [31], regularization [32], and so on. To tackle system heterogeneity, researchers have explored solutions such as client sampling [4]–[6], adaptive training rounds [14], compression [16]–[18], [33], [34], and joint methods [35]. Additionally, some research efforts have explored alternative training topology, such as clustered FL [9]–[12], hierarchical FL [13], [36]–[40], and decentralized FL [41]–[45]. Although diverse model training designs are adopted in these works, all of them are still based on the parallel training mode or its variants.

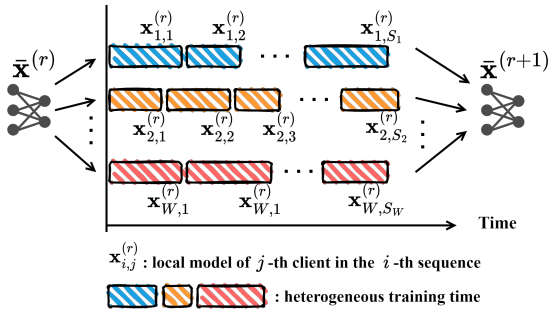


Fig. 2. Illustration of a PSFL training round with system heterogeneity.

Sequential Federated Learning (SFL): SFL is a new paradigm of federated learning, which demonstrates advantages on training speed (in terms of training rounds) both empirically [26], [46]–[49] and theoretically [25]. Kamp *et al.* in [26] enhanced model performance on small datasets by periodically redistributing local models across clients through the server. Zaccone *et al.* proposed an algorithm in [46] to enable sequential training of subgroups of heterogeneous clients. Lee *et al.* introduced a ring-based architecture for model training in [47]. Overall, none of these works takes the parallel and sequential training modes into consideration simultaneously, unlike our PSFL framework.

III. Framework and Problem Formulation

A. The PSFL Framework

We consider a typical FL system, consisting of a server and a crowd of clients, indexed by $\mathcal{N} = \{1, 2, ..., N\}$. Each client $n \in \mathcal{N}$ holds its local dataset $\mathcal{D}_n = \{\xi_i^n\}_{i=1}^{|\mathcal{D}_n|}$, where ξ_i^n denotes a data sample from $\mathcal{D}_n$, and $|\mathcal{D}_n|$ is the number of samples. We define $f(\mathbf{x}; \xi)$ as a loss function to measure how well a machine learning model $\mathbf{x}$ performs on the data sample ξ. The local loss function of client n is defined as $F_n(\mathbf{x}) \triangleq \mathbb{E}_{\xi \sim \mathcal{D}_n}[f(\mathbf{x}; \xi)]$. The global loss function is a linear combination of the local loss functions of all N clients, and the goal of FL is to train an optimal model $\mathbf{x}^*$ with minimum global loss function [25], [50], i.e.,

$$\mathbf{x}^* \triangleq \arg\min_{\mathbf{x}} F(\mathbf{x}) \triangleq \arg\min_{\mathbf{x}} \frac{1}{N} \sum\nolimits_{n=1}^{N} F_n(\mathbf{x}). \tag{1}$$

Different from traditional FL systems, we propose the PSFL framework, structured with multiple parallel sequences. As

shown in Fig. 2, several clients form a sequence, in which the local model trained by each client is transmitted to the next client. Multiple sequences are trained in parallel and the local models of each sequence are finally aggregated together to form a global model. Before the detailed training process, we first define a concept of training structure for the PSFL framework as follows.

Definition 1 (***Training Structure***)**.** *The training structure of PSFL is described as a 2-tuple $\mathcal{A} \triangleq (\mathcal{W}, \{S_w\}_{w\in\mathcal{W}})$, where $\mathcal{W} = \{1, 2, \ldots, w, \ldots\}$ represents a set of sequences, and $W = |\mathcal{W}|$ is called the parallel width. Each sequence $w \in \mathcal{W}$ comprises a set of clients with a cardinality of S_w, called the sequence length. Meanwhile, we let S denote the longest sequence length, i.e., $S = \max_{w\in\mathcal{W}}\{S_w\}$.*

We consider the whole FL process consisting of R rounds, and let $r \in \{0, 1, 2, \ldots, R-1\}$ indicate the rounds. Before the training process, the set of clients participating in round r is randomly sampled without replacement from the client set $\mathcal{N}$, denoted as $\Pi_{\mathcal{A}}^{(r)} = \{\pi_1^w, \pi_2^w, \ldots, \pi_{S_w}^w\}_{w\in\mathcal{W}}$, with respect to the training structure $\mathcal{A}$. Here, π_m^w is the index of the m-th sampled client in the w-th sequence. The m-th trained local model in the w-th sequence during round r is denoted by $\mathbf{x}_{w,m}^{(r)}$. At the beginning of round r, the server distributes the global model $\bar{\mathbf{x}}^{(r)}$ to the first client in each sequence for initialization, setting $\mathbf{x}_{w,0}^{(r)} = \bar{\mathbf{x}}^{(r)}$ for all $w \in \mathcal{W}$. Subsequently, the sequences are then trained in parallel, and the clients of each sequence are trained sequentially. The local models in sequence w are updated using Stochastic Gradient Descent (SGD) as follows:

$$\mathbf{x}_{w,m}^{(r)} = \mathbf{x}_{w,m-1}^{(r)} - \eta \mathbf{g}_{\pi_m^w}^{(r)},\ m \in \{1, 2, \ldots, S_w\}, \tag{2}$$

where η is the learning rate, and $\mathbf{g}_{\pi_m^w}^{(r)} \triangleq \nabla f(\mathbf{x}_{w,m-1}^{(r)}; \xi_{\pi_m^w})$ denotes the stochastic gradient of $F_{\pi_m^w}$ with respect to the parameter vector $\mathbf{x}_{w,m-1}^{(r)}$ and the sample $\xi_{\pi_m^w}$. After receiving all local models uploaded by all sequences, the server aggregates these local models to update the global model, i.e.,

$$\bar{\mathbf{x}}^{(r+1)} = \frac{1}{W} \sum\nolimits_{w\in\mathcal{W}} \mathbf{x}_{w,S_w}^{(r)}. \tag{3}$$

Here, we use the basic method of global aggregation. Actually, our PSFL framework can be easily extended to incorporate other variant algorithms of the PFL framework.

B. Problem Formulation

In the above model, we primarily consider two types of heterogeneity: i) statistical heterogeneity: the training data are distributed in an unbalanced and non-iid fashion among clients; ii) system heterogeneity: clients exhibit heterogeneous capabilities in both computing and communication. Let $t_n^{(r)}$ be the training time for client n in round r, which encompasses both local model computation time and the time taken to transmit the trained model to the next client or server. For simplicity, we assume that the training time $t_n^{(r)}$ follows an unknown distribution with a mean value t_n, and the average expected training time is denoted by $\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n$. Consequently, the total training time for a sequence w is calculated as $\sum_{m=1}^{S_w} t_{\pi_m^w}^{(r)}$. Due to the synchronization barrier, the total time for a training round is formulated as follows:

$$T_{total}^{(r)} = \max\nolimits_{w\in\mathcal{W}} \sum\nolimits_{m=1}^{S_w} t_{\pi_m^w}^{(r)}. \tag{4}$$

Our goal is to determine a client sampling strategy $\Pi_{\mathcal{A}} = \{\Pi_{\mathcal{A}}^{(r)}\}_{r=0}^{R-1}$ based on a training structure $\mathcal{A}$, so as to minimize the expected total training time $\mathbb{E}[\sum_{r=0}^{R-1} T_{total}^{(r)}]$, while ensuring that the expected global loss $\mathbb{E}[F(\bar{\mathbf{x}}^{(R)})]$ converges to the optimal value $F(\mathbf{x}^*)$ with an ϵ precision, with $\bar{\mathbf{x}}^{(R)}$ being the aggregated global model after R rounds. Considering that partial client participation has more practical interest than full client participation [2], we also restrict the number of clients sampled in each round. Therefore, the optimization problem can be formulated as follows:

$$\textbf{P1:}\quad \min_{\Pi_{\mathcal{A}}}\quad \mathbb{E}[\sum\nolimits_{r=0}^{R-1} T_{total}^{(r)}], \tag{5}$$

$$s.t.\quad \mathbb{E}[F(\bar{\mathbf{x}}^{(R)})] - F(\mathbf{x}^*) \le \epsilon, \tag{6}$$

$$\sum\nolimits_{w\in\mathcal{W}} S_w \le \widetilde{N}. \tag{7}$$

Here, the expectations in Eqs. (5) and (6) are taken over the potential randomness in client sampling, local SGD, and training time. Eq. (7) means that the number of selected clients in each round is not larger than a bound $\widetilde{N}$.

Solving Problem **P1**, however, is challenging in two aspects.

1) Before actually training the model, it is generally unclear how the chosen training structure $\mathcal{A}$, the sampling strategy $\Pi_{\mathcal{A}}$, and the number of training rounds R will affect the final model $\bar{\mathbf{x}}^{(R)}$ and the corresponding loss function $F(\bar{\mathbf{x}}^{(R)})$. Thus, we need to derive an analytical expression to explore the complex impact of $\mathcal{A}, \Pi_{\mathcal{A}}$, and R on model performance.

2) It is complicated to optimize $\mathbb{E}[\sum_{r=0}^{R} T_{total}^{(r)}]$. Actually, Problem **P1** can be simplified to the mainstream multi-machine scheduling problem. Hence, Problem **P1** is NP-hard. Existing multi-machine scheduling methods often rely on prioritization, but we must ensure the unbiasedness of client sampling to achieve the optimization of the global model. That is, we need a new client sampling approach.

IV. Convergence Analysis

In this section, we address the first challenge by deriving a new convergence bound for any given training structure. To facilitate a tractable convergence analysis, we state several assumptions on the local objective functions $F_n(\mathbf{x})$.

Assumption 1. *(L-smooth). For each client n, its local objective function F_n is L-smooth, i.e., there exists a constant $L > 0$ such that $\|\nabla F_n(\mathbf{x}) - \nabla F_n(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|$ for all $\mathbf{x}$ and $\mathbf{y}$.*

Assumption 2. *(Convex). For each client n, its local objective function F_n is convex, i.e., there is $F_n(\mathbf{x}) \ge F_n(\mathbf{y}) + (\mathbf{x} - \mathbf{y})^T \nabla F_n(\mathbf{y})$ for all $\mathbf{x}$ and $\mathbf{y}$.*

Assumption 3. *(Unbiased gradient and bounded variance). For each client n, the stochastic gradient is an unbiased estimator of the local gradient: $\mathbb{E}_{\xi\sim\mathcal{D}_n}[\nabla f_n(\mathbf{x}; \xi)] = \nabla F_n(\mathbf{x})$ and has bounded variance $\mathbb{E}_{\xi\sim\mathcal{D}_n}[\|\nabla f_n(\mathbf{x}; \xi) - \nabla F_n(\mathbf{x})\|^2 | \mathbf{x}] \le \sigma^2$, where ξ is sampled from the n-th client's local dataset $\mathcal{D}_n$ uniformly at random.*

Assumption 4. *(Bounded statistical heterogeneity). There exists a constant $B > 0$ such that $\frac{1}{N}\sum_{n=1}^{N}\|\nabla F_n(\mathbf{x}^*)\|^2 = B$.*

Assumptions 1-3 are commonly used in existing studies of convex federated learning problems, such as [2], [5], [25]. Furthermore, the experimental results to be presented in Section VI show that our approach also works for non-convex loss functions. Assumption 4 is followed by [3], [25], [50] for quantifying the degree of non-iid data distribution across clients. Before analyzing the convergence bound, we define the unbiasedness of the client sampling strategy.

Definition 2. *(Unbiasedness) In each round r, the sampling policy $\Pi_{\mathcal{A}}$ for any client $n \in \mathcal{N}$ is unbiased if the probability that client n is selected is $Pr[\pi_m^w = n] = \frac{1}{N}$, $\forall w \in \mathcal{W}, m \in \{1, 2, \ldots, S_w\}$. Consequently, this ensures that the expected value of the local loss function for any sampled client matches the global loss function, i.e., $\mathbb{E}[F_{\pi_m^w}(\mathbf{x})] = F(\mathbf{x})$.*

Based on the above definition, we present the theoretical analysis of the convergence bound. For brevity, the proofs of **Theorem 1** and **Corollary 1** are moved to the Appendix.

Theorem 1. *Let Assumptions 1 to 4 hold, and the values of L, σ^2, $\mathcal{A}$ are given. If the client sampling strategy $\Pi_{\mathcal{A}}$ is unbiased in the PSFL framework, and the learning rate satisfies $\eta \leq \frac{c_0}{LS}$, where $0 < c_0 < \frac{1}{5}$ is a constant, then the weighted average of the global parameters $\widetilde{\mathbf{x}} = \frac{1}{R+1}\sum_{r=0}^{R}\bar{\mathbf{x}}^{(r)}$ satisfies:*

$$\mathbb{E}[F(\widetilde{\mathbf{x}}) - F(\mathbf{x}^*)] \leq \frac{r_0}{b\tilde{\eta}R} + \frac{\tilde{\eta}}{b}(\alpha W + \beta), \tag{8}$$

where $b = \frac{16c_0^3 - 28c_0^2 - 24c_0 + 6}{3(1-2c_0^2)}$, $\tilde{\eta} = \frac{\eta N_0}{W}$, $r_0 = \|\bar{\mathbf{x}}^{(0)} - \mathbf{x}^*\|^2$, $\alpha = \frac{4c_0(1+2c_0)}{1-2c_0^2}\frac{(\sigma^2+B)}{N_0} + \frac{4B}{N_0}$, $\beta = \frac{4\sigma^2}{N_0}$, and $N_0 = \sum_{w\in\mathcal{W}} S_w$.

The convergence bound in Eq. (8) characterizes the relationship between the number of rounds R and the parallel width W to reach the target precision. In order to simplify the analysis process, we use the weighted average of the global parameters $\widetilde{\mathbf{x}} = \frac{1}{R+1}\sum_{r=0}^{R}\bar{\mathbf{x}}^{(r)}$ to replace the global model $\bar{\mathbf{x}}^{(R)}$ in Problem **P1**. Besides, we notice that a larger $\tilde{\eta}$ can make the first term vanish at a higher rate, yet results in the increase of the second term. Hence, we need to choose an appropriate $\tilde{\eta}$ to reach a balance between the two terms.

Corollary 1. *By choosing an appropriate learning rate $\tilde{\eta} = \min\{\sqrt{\frac{r_0}{R(\alpha W+\beta)}}, \frac{c_0 N_0}{LSW}\}$, we can obtain the convergence bound:*

$$\mathbb{E}[F(\widetilde{\mathbf{x}}) - F(\mathbf{x}^*)] \leq \mathcal{O}\left(\frac{1}{R}\frac{r_0 LSW}{bc_0 N_0} + \frac{1}{b}\sqrt{\frac{r_0(\alpha W+\beta)}{R}}\right), \tag{9}$$

where $b = \frac{16c_0^3 - 28c_0^2 - 24c_0 + 6}{3(1-2c_0^2)}$, $r_0 = \|\bar{\mathbf{x}}^{(0)} - \mathbf{x}^*\|^2$, $N_0 = \sum_{w\in\mathcal{W}} S_w$, $\alpha = \frac{4c_0(1+2c_0)}{1-2c_0^2}\frac{(\sigma^2+B)}{N_0} + \frac{4B}{N_0}$, and $\beta = \frac{4\sigma^2}{N_0}$.

The convergence bound in Eq. (9) contains two terms, which come from two different values of the learning rate $\tilde{\eta}$. In fact, when R is sufficiently large, that is, $\tilde{\eta} = \sqrt{\frac{r_0}{R(\alpha W+\beta)}}$, this convergence bound is reduced to $\mathcal{O}(\sqrt{r_0(\alpha W+\beta)/R})$. Similar to existing works [5], [14], [17], we adopt the upper bound of the convergence as an approximation of the actual error between $\mathbb{E}[F(\bar{\mathbf{x}}^{(R)})]$ and $F(\mathbf{x}^*)$.

V. Framework Design

In this section, we propose a novel hybrid Parallel-Sequential Federated Learning (PSFL) framework, which mainly consists of the optimal training structure determination module and the client sampling module. The former yields the optimal sequence length and parallel width, based on which the latter determines the set of participating clients for model training in each round. Specifically, we first provide a bound for the expected total training time to approximate the optimization goal in Problem **P1**. Then, we find the optimal training structure by solving the approximate problem. Finally, we design a time-based partitioning and sampling strategy, followed by a detailed description of the PSFL framework.

A. Bound for the Expected Total Training Time

Before providing the bound for the expected total training time, we first make a general assumption about the distribution of client training times, shown as follows.

Assumption 5. *(Subgaussian training time). For each client n, the local training time $t_n^{(r)}$ in each round is a κ^2-subgaussian random variable with mean t_n, where κ is a constant called the subgaussian parameter. In other words, for any $\lambda \in \mathbb{R}$, the inequality $\mathbb{E}[e^{\lambda(t_n^{(r)} - t_n)}] \leq \exp(\frac{\lambda^2\kappa^2}{2})$ holds.*

Assumption 5 implies that the deviation of the local training time $t_n^{(r)}$ from its mean t_n has a tail probability that decays at least as fast as that of a normal distribution with variance κ^2. Based on Assumption 5, we present our bound for the expected total training time as follows.

Theorem 2. *Let Assumption 5 hold, and assume that the client sampling strategy $\Pi_{\mathcal{A}}$ is unbiased, then the expected total training time is bounded as follows:*

$$\mathbb{E}[\sum_{r=0}^{R-1} T_{total}^{(r)}] \geq R\frac{N_0}{W}\frac{1}{N}\sum_{n=1}^{N} t_n, \tag{10}$$

$$\mathbb{E}[\sum_{r=0}^{R-1} T_{total}^{(r)}] \leq R[S\frac{1}{N}\sum_{n=1}^{N} t_n + \sqrt{2\kappa^2 S\log W}], \tag{11}$$

where $N_0 = \sum_{w\in\mathcal{W}} S_w$ and κ is a constant.

Proof. The first inequality is trivial since there is

$$\mathbb{E}[\max_{w\in\mathcal{W}}\sum_{m=1}^{S_w} t_{\pi_m^w}^{(r)}] \geq \mathbb{E}[\frac{1}{W}\sum_{w\in\mathcal{W}}\sum_{m=1}^{S_w} t_{\pi_m^w}^{(r)}] = \frac{N_0}{W}\frac{1}{N}\sum_{n=1}^{N} t_n.$$

For clarity and brevity, we define $X_w = \sum_{m=1}^{S_w}(t_{\pi_m^w}^{(r)} - t_{\pi_m^w})$ and $Y = \max_{w\in\mathcal{W}} X_w$. Based on the conclusion in [51], we can further establish the subgaussian property of X_w, i.e.,

$$\mathbb{E}[\exp(\lambda X_w)] = \mathbb{E}[\mathbb{E}[\exp(\lambda X_w)|\pi_m^w]] \leq \mathbb{E}[\exp(\frac{1}{2}\lambda^2 S_w\kappa^2)], \tag{12}$$

$$\begin{aligned}\exp(\lambda\mathbb{E}[Y]) &\leq \mathbb{E}[\exp(\lambda Y)] = \mathbb{E}[\max_{w\in\mathcal{W}}\exp(\lambda X_w)] \\ &\leq \sum_{w\in\mathcal{W}}\mathbb{E}[\exp(\lambda X_w)] \leq W\exp(\frac{1}{2}\lambda^2 S\kappa^2).\end{aligned} \tag{13}$$

By taking the logarithm of both sides of Eq. (13), we can directly obtain $\mathbb{E}[Y] \leq \log W/\lambda + \lambda S\kappa^2/2$. According to the AM-GM inequality, the values of $\log W/\lambda + \lambda S\kappa^2/2$ can be minimized by setting $\lambda = \sqrt{2\log W/S\kappa^2}$. Therefore, we have $\mathbb{E}[Y] \leq \sqrt{2\kappa^2 S\log W}$. Under the assumption that the client sampling strategy $\Pi_{\mathcal{A}}$ is unbiased, there is $\mathbb{E}[\sum_{m=1}^{S_w} t_{\pi_m^w}^{(r)}] = S_w\frac{1}{N}\sum_{n=1}^{N} t_n$. Furthermore, we obtain

$$\mathbb{E}[\sum_{r=0}^{R-1} T_{total}^{(r)}] \leq R(\mathbb{E}[Y] + S\frac{1}{N}\sum_{n=1}^{N} t_n)$$
$$\leq R(S\frac{1}{N}\sum_{n=1}^{N} t_n + \sqrt{2\kappa^2 S\log W}). \quad (14)$$

The proof of the theorem is now completed. □

B. Training Structure Determination

Inspired by the proofs of **Theorem 1** and **Theorem 2**, we find that with a fixed number of sampled clients, more balanced sequence lengths result in tighter convergence bounds and a lower upper bound for the expected total training time. This suggests that the optimal solution to our problem is achieved when the sequence lengths are equal, i.e., $S_w = S, \forall w \in \mathcal{W}$. Recall that N_0 is defined as $\sum_{w\in\mathcal{W}} S_w$, thus we have $N_0 = SW$. According to the convergence bound and the upper bound of the expected total training time, i.e., Eqs. (9) and (11), Problem **P1** can be transformed into Problem **P2**:

$$\textbf{P2:} \quad \min_{S,W} \quad R(S\frac{1}{N}\sum_{n=1}^{N} t_n + \sqrt{2\kappa^2 S\log W}), \quad (15)$$
$$s.t. \quad \frac{1}{R}(\alpha W + \beta) \leq \epsilon', N_0 \leq \widetilde{N}, \quad (16)$$
$$\alpha = \frac{4c_0(1+2c_0)(\sigma^2+B)}{(1-2c_0^2)N_0} + \frac{4B}{N_0}, \beta = \frac{4\sigma^2}{N_0}, \quad (17)$$

where ϵ' is a constant derived from ϵ. It is worth noting that integer optimization problems are difficult to solve, we relax W, S, and R as continuous variables in Problem **P2**. To solve this problem, we first transform the constraint in Eq. (16) into the form of a minimum value of R, i.e., $R \geq \frac{1}{\epsilon'}(\alpha W + \beta)$. Afterwards, the optimal parallel width W^* in the Problem **P2** can be solved as follows:

$$W^* \triangleq \arg\min R\left[\frac{N_0\bar{t}}{W} + \sqrt{2\kappa^2\frac{N_0}{W}\log W}\right] \quad (18)$$
$$= \arg\min(\alpha W+\beta)\left[\frac{N_0\bar{t}}{W} + \sqrt{2\kappa^2\frac{N_0}{W}\log W}\right] \quad (19)$$
$$= \arg\min \frac{\beta N_0\bar{t}}{W} + (\alpha W+\beta)\sqrt{2\kappa^2\frac{N_0}{W}\log W} \quad (20)$$
$$= \arg\min \alpha\sqrt{2\kappa^2 N_0 W\log W} + \beta(\frac{N_0\bar{t}}{W} + \sqrt{2\kappa^2\frac{N_0}{W}\log W}). \quad (21)$$

Here, we omit the constant terms in Eqs. (19) and (20). When the parallel width W is fixed, the objective in Eq. (21) decreases monotonically with respect to the number of participating clients (i.e., N_0). Therefore, we conclude that the optimal sequence length S^* and the optimal parallel width W^* always satisfy $S^*W^* = \widetilde{N}$.

In Eq. (21), we divide the optimization objective into two terms, each of which has the coefficient α or β. These two terms exhibit opposite behaviors with respect to the parallel width W. Specifically, the term $\sqrt{2\kappa^2 N_0 W\log W}$ increases monotonically along with the growth of W, while the term $\frac{N_0\bar{t}}{W} + \sqrt{2\kappa^2\frac{N_0}{W}\log W}$ would decrease monotonically with the growth of W. Obviously, the ratio $\frac{\alpha}{\beta}$ reflects the importance of these two terms in the optimization process. Due to $\frac{\alpha}{\beta} = \frac{1+c_0}{1-2c_0^2}\frac{B}{\sigma^2} + \frac{c_0(1+2c_0)}{1-2c_0^2}$, we find that the statistical heterogeneity bound B is closely associated with the ratio $\frac{\alpha}{\beta}$. In order to clearly reflect the impact of different statistical bounds on the optimal parallel width solution, we solve Problem **P2** separately under the following three cases.

(i) **Case 1:** When the statistical heterogeneity bound B is sufficiently large (i.e., $B \to \infty$), the ratio $\frac{\alpha}{\beta}$ tends to infinty. Thus, we mainly focus on the first term of the optimization objective in Eq. (21) and get the following optimal solution:

$$W^* = \arg\min \alpha\sqrt{2\kappa^2 N_0 W\log W} + \beta(\frac{N_0\bar{t}}{W} + \sqrt{2\kappa^2\frac{N_0}{W}\log W})$$
$$= \arg\min \alpha\sqrt{2\kappa^2 N_0 W\log W} \to 1. \quad (22)$$

Based on Eq. (22), we conclude the optimal parallel width is $W^* = 1$ and the optimal sequence length is $S^* = \widetilde{N}$. This result is consistent with the situation when statistical heterogeneity is very strong. In this case, the global model in the traditional PFL framework is difficult to converge quickly, and we tend to choose the traditional SFL framework to serialize the training process and speed up convergence.

(ii) **Case 2:** When the statistical heterogeneity bound B is sufficiently small (i.e., $B \to 0$), the ratio tends to a constant (i.e., $\frac{\alpha}{\beta} \to \frac{c_0(1+2c_0)}{1-2c_0^2}$). Therefore, we get

$$W^* = \arg\min \alpha\sqrt{2\kappa^2 N_0 W\log W} + \beta(\frac{N_0\bar{t}}{W} + \sqrt{2\kappa^2\frac{N_0}{W}\log W})$$
$$= \arg\min \frac{c_0(1+2c_0)}{1-2c_0^2}\sqrt{W\log W} + \frac{\sqrt{N_0}\bar{t}}{\sqrt{2\kappa^2 W}} + \sqrt{\frac{\log W}{W}}. \quad (23)$$

In Eq. (23), We remove a constant $\sqrt{2\kappa^2 N_0}$ to simplify the equation without affecting the optimization problem. In the following analysis, we omit the last term $\sqrt{\log W/W}$, since it is too small. For concise, we define a function $h(W) = a\sqrt{W\log W} + \frac{b}{W}$,where $a = \frac{c_0(1+2c_0)}{1-2c_0^2}$, and $b = \frac{\sqrt{N_0}\bar{t}}{\sqrt{2\kappa^2}}$. Then we calculate the derivative of $h(W)$ with regard to W:

$$\frac{\partial h(W)}{\partial W} = \frac{a}{2}\frac{1+\log W}{\sqrt{W\log W}} - \frac{b}{W^2} = \frac{aW^{3/2}(1+\log W) - 2b\sqrt{\log W}}{2W^2\sqrt{\log W}}.$$

We obtain the critical point W^* by setting $\partial h(W)/\partial W = 0$, satisfying $\frac{2b}{a} = W^{*3/2}(\sqrt{\log W^*} + 1/\sqrt{\log W^*})$. Therefore, when $\frac{b}{a} > \frac{3e^{3/4}}{2\sqrt{2}} \approx 2.245$, the optimal parallel width W^* is larger than $\sqrt{e}$, which indicates that the traditional SFL framework is not suitable for weak heterogeneity. In contrast, the global model in the traditional PFL framework can converge quickly due to the power of parallel training. It is noteworthy that the PFL framework is not always optimal. Specifically, we observe that only when $N_0\sqrt{\log N_0/2} < \frac{4\bar{t}}{a\sqrt{2\kappa^2}}$, the optimal training structure satisfies $W^* > \frac{N_0}{2}$ and $S^* = 1$. This implies that the PFL framework is optimal only when the number of participating clients (i.e., N_0) is not very large.

(iii) **Case 3:** For general heterogeneity, we need to numerically solve the optimization problem in Eq. (21). Notably, there are some variables, such as σ^2, B, $\bar{t}$ and κ^2, whose values are unknown at the beginning. To adequately explore the values of these variables, we design a predefined warm-up phase in the PSFL framework. During this phase, all clients participate in the K training rounds to estimate parameters. Similar to existing work [42], in each round k, each client n estimates the stochastic gradient variance bound (denoted by $\hat{\sigma}_{n,k}^2$) during local training. After the local update is completed, each client sends its local model gradients and the parameter estimate

Algorithm 1: Parallel-Sequential Federated Learning

input : number of total training rounds R.
output: aggregated global model $\bar{\mathbf{x}}^{(R)}$.

1 **//Warm-up Phase:**
2 Initialize: the global model $\mathbf{x}^0$;
3 **for** *training round* $k = 0, 1, \ldots, K-1$ **do**
4 **for** *client* $n = 1, \ldots, N$ *in parallel* **do**
5 Initialize: $\mathbf{x}_n^k = \mathbf{x}^k$;
6 Local update: $\mathbf{x}_n^{k+1} = \mathbf{x}_n^k - \eta \nabla F_n(\mathbf{x}^k)$;
7 Estimate $\hat{\sigma}_{n,k}^2 = \mathbb{E}[\|\nabla f(\mathbf{x}^k, \xi_n) - \nabla F_n(\mathbf{x}^k)\|^2]$;
8 Global aggregation: $\mathbf{x}^{k+1} = \mathbf{x}^k - \eta \frac{1}{N} \sum_{n=1}^N \nabla F_n(\mathbf{x}^k)$;
9 Estimate $\hat{B}_k = \mathbb{E}[\|\nabla F_n(\mathbf{x}^k) - \nabla F(\mathbf{x}^k)\|^2]$;
10 Estimate $\hat{\sigma}^2$, $\hat{B}$, $\hat{\bar{t}}$, and $\hat{\kappa}^2$ based on Eq. (24) $\sim$ Eq. (26);
11 Solve Eq. (21) to get optimal sequence length S and optimal parallel width W;
12 **//Training Phase:**
13 Initialize: $\bar{\mathbf{x}}^{(0)}$ and the estimates of training time $\hat{t}_n$;
14 **for** *training round* $r = 0, 1, \ldots, R-1$ **do**
15 Sort the clients according to estimate $\hat{t}_n$;
16 Sample clients $\{\pi_1^w, \pi_2^w, \ldots, \pi_S^w\}_{w \in \mathcal{W}}$ based on time-based partitioning and sampling strategy;
17 **for** *sequence* $w = 1, \ldots, W$ *in parallel* **do**
18 Initialize: $\mathbf{x}_{w,0}^{(r)} = \bar{\mathbf{x}}^{(r)}$;
19 **for** *client* $m = 1, \ldots, S$ *in sequence* **do**
20 Local update: $\mathbf{x}_{w,m}^{(r)} = \mathbf{x}_{w,m-1}^{(r)} - \eta \mathbf{g}_{\pi_m^w}^{(r)}$;
21 Update the estimate of training time $\hat{t}_{\pi_m^w}$;
22 Global aggregation: $\bar{\mathbf{x}}^{(r+1)} = \frac{1}{W} \sum_{w=1}^W \mathbf{x}_{w,S}^{(r)}$.

$\hat{\sigma}_{n,k}^2$ to the server. The server then takes the average of the parameters $\hat{\sigma}_{n,k}^2$ to obtain $\hat{\sigma}^2$. Moreover, the server estimates the heterogeneity bound in each round k (denoted by $\hat{B}_k$) based on the gradients sent by the clients, and estimates the statistical property of training time (i.e., $\bar{t}$ and κ^2) according to the local training time of each client in each round. We use $\hat{B}$, $\hat{\bar{t}}$ and $\hat{\kappa}^2$ to represent the estimates of B, $\bar{t}$ and κ^2, respectively. Therefore, we have the following equations:

$$\hat{\sigma}_{n,k}^2 = \mathbb{E}[\|\nabla f(\mathbf{x}^k, \xi_n) - \nabla F_n(\mathbf{x}^k)\|^2], \hat{\sigma}^2 = \frac{1}{K} \sum_{k=1}^K \frac{1}{N} \sum_{n=1}^N \sigma_{n,k}^2, \quad (24)$$

$$\hat{B}_k = \mathbb{E}[\|\nabla F_n(\mathbf{x}^k) - \nabla F(\mathbf{x}^k)\|^2], \hat{B} = \frac{1}{K} \sum_{k=1}^K \hat{B}_k, \quad (25)$$

$$\hat{\bar{t}} = \frac{1}{K} \sum_{k=1}^K \frac{1}{N} \sum_{n=1}^N t_n^k, \hat{\kappa}^2 = \frac{1}{N} \sum_{n=1}^N \mathbb{E}[\|t_n^k - \frac{1}{K} \sum_{k=1}^K t_n^k\|^2], \quad (26)$$

where $\mathbf{x}^k$ is the global model in the k-th round.

Based on the above estimated parameters, it is still hard to acquire a closed-form solution, but there are many existing tools and methods (e.g., the bisection or Newton's method) to obtain an approximate numerical solution of the optimization problem in Eq. (21). After determining the optimal parallel width W^*, our optimal sequence length can be calculated as $S^* = \frac{\widetilde{N}}{W^*}$. In practice, we round these values to the nearest integers to determine the final optimal training structure.

C. Time-based Partitioning and Sampling Strategy

In the previous sections, we establish and solve an approximate optimization problem with regard to the training structure, containing the parallel width and sequence length. In this section, we determine an unbiased client sampling strategy to significantly reduce the training time in a round.

We note that any unbiased sampling strategy cannot reduce the training time of a sequence. This is because for any unbiased sampling strategy $\Pi_{\mathcal{A}}$, the expected training time of a sequence is fixed, i.e., $\mathbb{E}[\sum_{m=1}^{S_w} t_{\pi_m^w}^{(r)}] = S_w \frac{1}{N} \sum_{n=1}^N t_n$. To reduce the maximum training time across multiple sequences, it is essential to minimize the variance between sequences, as suggested by **Theorem 2**. Given the optimal sequence length S^* and the optimal parallel width W^*, the process of our sampling strategy unfolds as follows:

1. We first sort the clients according to their estimates of training time, denoted as $\hat{t}_n$ for each client n, which is calculated by averaging their historical training times.
2. We then divide the clients into S^* groups based on this sorting, with each class containing $\lfloor \frac{N}{S^*} \rfloor$ or $\lceil \frac{N}{S^*} \rceil$ clients.
3. Next, we uniformly sample a client from each group without replacement and generate a random permutation of these selected clients to form a sequence.
4. Finally, we repeat the above steps W^* times to generate W^* sequences.

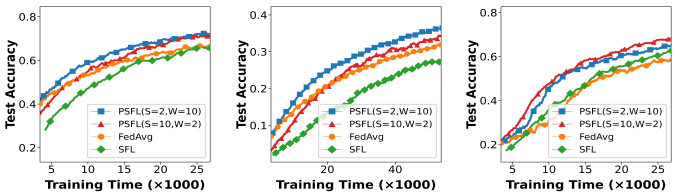
Theorem 3. *The time-based partitioning and sampling strategy is approximately unbiased.*

Proof. The probability that a client n is sampled to sequence w is $\frac{1}{\lfloor \frac{N}{S^*} \rfloor}$ or $\frac{1}{\lceil \frac{N}{S^*} \rceil}$, which is approximately $\frac{S^*}{N}$. Due to the random permutation of sampled clients in a sequence, we get $Pr[n = \pi_m^w] = \frac{S^*}{N} \cdot \frac{1}{S^*} = \frac{1}{N}$. The proof is finished. □

Algorithm 1 shows the pseudocode of the PSFL framework, which is divided into two main phases: the warm-up phase (Lines 1-11) and the training phase (Lines 12-22). During the warm-up phase, we focus on estimating parameters related to statistical and system heterogeneity (Lines 7-10). Based on these estimates, the server determines the optimal training structure by solving the optimization problem in Eq. (21) (Line 11). In the training phase, we focus on sampling clients according to the structure obtained in the warm-up phase (Lines 15-16) and training them to achieve global model convergence (Lines 17-22). The computational complexity of PSFL is dominated by the sorting operation on clients' training time estimates, i.e., $O(N \log N)$.

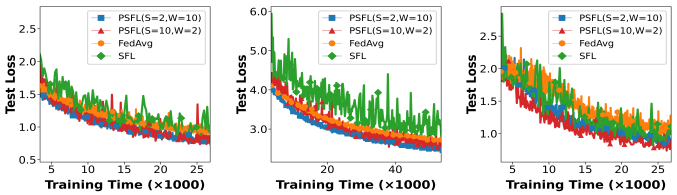
VI. Simulation Results

In this section, we conduct experiments on real datasets to evaluate the performance of our proposed PSFL framework. The main findings are: 1) PSFL can significantly reduce the total training time for the model to converge. 2) Our training time-based partitioning and sampling strategy can significantly reduce the single-round training time by reducing the imbalance of training time between sequences. First, we present our experimental settings. Then we display the evaluation results.



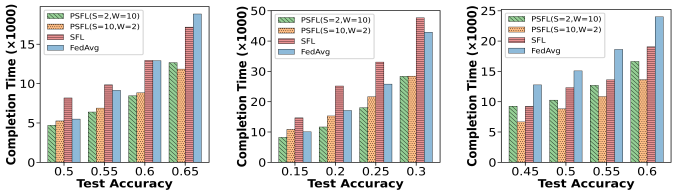
(a) CIFAR-10 (b) CIFAR-100 (c) HAM

Fig. 3. Test accuracy of three frameworks on the three datasets.



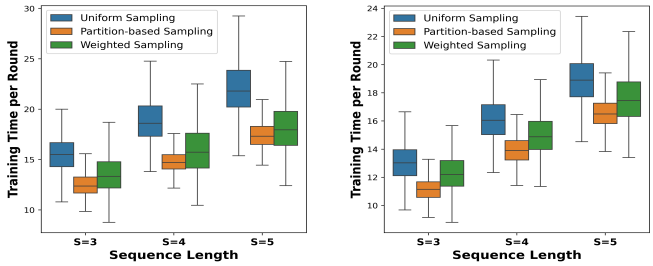
(a) CIFAR-10 (b) CIFAR-100 (c) HAM

Fig. 4. Test loss of three frameworks on the three datasets.



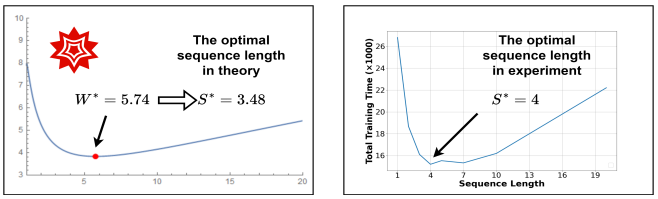
(a) CIFAR-10 (b) CIFAR-100 (c) HAM

Fig. 5. Completion time of three frameworks under different target accuracies.



(a) Discrete Distribution (b) Gaussian Distribution

Fig. 6. Comparison of sampling strategies under different distributions.



(a) Theoretical Result (b) Experimental Result

Fig. 7. Comparison of theoretical and experimental optimal sequence length.

A. Experimental Setup

1) Datasets and models: We conduct the experiments over three common real-world datasets: CIFAR-10, CIFAR-100, and HAM10000 [52]. We train models on CIFAR-10 and CIFAR-100 using a 5-layer CNN with two 5×5 convolutional layers, each followed by ReLU activation and max pooling, and three fully connected layers. The model for HAM10000 is a CNN consisting of three 3×3 convolutional layers (with ReLU activation and max pooling) and two fully connected layers with dropout regularization.

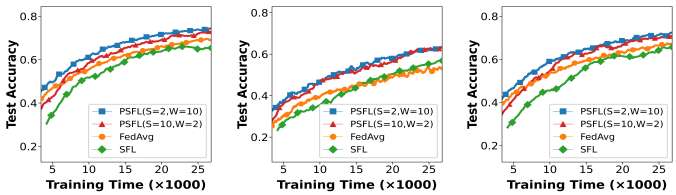
2) Heterogeneous data: In our experiment, we will allocate the dataset to $N = 500$ clients. To simulate the non-iid data, we use the widely used Dirichlet-based strategy and an extended Dirichlet strategy [25] to partition the datasets. In the Dirichlet-based strategy, the proportion of data allocated to each client follows a Dirichlet distribution $Dir(\alpha \mathbf{p})$, where $\mathbf{p}$ characterizes a prior distribution. A smaller α value increases the heterogeneity among the clients' data distributions. The extended Dirichlet strategy, denoted by $ExDir(C, \alpha)$ introduces an additional parameter C to determine the number of classes per client. Specifically, before allocating samples via Dirichlet distribution (with parameter α), this strategy allocates C different classes to each client, obtaining the specific prior distribution $\mathbf{p}_c$ for each class c. For example, $\mathbf{p}_c = [1, 1, 0, 0, \ldots]$ means that the samples of class c are only allocated to the first 2 clients. This method enhances the control over the heterogeneity of the data distribution across different clients in federated learning environments.

3) Heterogeneous system: The simulation experiments are conducted on an AMAX deep learning workstation equipped with an Intel(R) Xeon(R) Platinum 8358P CPU, one NVIDIA A40 (48GB) GPU and 80 GB RAM. To simulate the system heterogeneity, we use four distributions to model the mean value of training time (i.e., t_n) for each client: uniform distribution $U(0.5, 4.5)$, exponential distribution $Exp(2.5)$, Gaussian distribution $\mathcal{N}(2.5, 1)$ and an extreme discrete distribution where values in $\{0.5, 1, 2, 4, 5\}$ have equal probability. These distributions help represent the various situations of device computing and network communication capabilities in reality. Additionally, to simulate the randomness in reality, the actual training time in each round (i.e., $t_n^{(r)}$) follows a Gaussian distribution with a mean of t_n and a standard variance of $0.2t_n$, denoted as $t_n^{(r)} \sim \mathcal{N}(t_n, (0.2t_n)^2)$.

4) Baselines and metrics: Since other variant algorithms of PFL can be incorporated into PSFL by modifying the global model aggregation method, we only select two basic frameworks as baselines: i) FedAvg [1] transmits and trains the local models in parallel for all sampled clients. ii) SFL [25] transmits and trains the local models in sequence for all sampled clients. We employ the following metrics to evaluate the performance of PSFL and baselines. Firstly, *test accuracy* is measured as the proportion of correctly classified test samples to the total number of test samples. Secondly, *test loss* is calculated as the average cross-entropy loss over the test dataset. Thirdly, *total training time* is calculated by accumulating the time used in each training round.

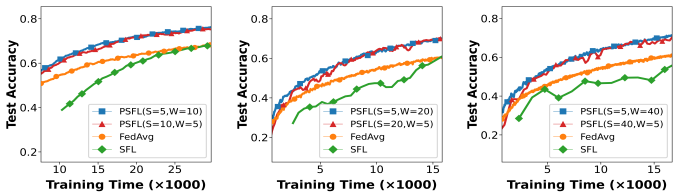
B. Performance Results

1) Comparing to baselines: We implement PSFL and the baselines on three datasets to observe their performance, examining test accuracy and test loss under the same total training time, as well as the training time required to achieve the same test accuracy. In the experiments, we maintain a consistent system setup, including the number of clients participating in each round and the simulation of the system and statistical heterogeneity. Specifically, we set $N_0 = 20$, allocate data according to $ExDir(C=2, \alpha=10)$, and let t_n



(a) Dir(0.2) (b) ExDir(1,10) (c) ExDir(2,5)

Fig. 8. Test accuracy of three frameworks under different non-iid settings.



(a) $N_0 = 50$ (b) $N_0 = 100$ (c) $N_0 = 200$

Fig. 9. Test accuracy of three frameworks at different N_0.

follow the discrete distribution. Figs. 3 and 4 demonstrate that PSFL achieves better convergence performance under the same training time compared to the baselines across all datasets. Moreover, Fig. 5 shows that PSFL achieves the same target test accuracy with significantly less training time than the baselines. For the CIFAR-10 dataset, PSFL requires approximately 33% less time than PFL and SFL to achieve the same target test accuracy of 60%.

2) The efficiency of sampling strategy: We empirically evaluate the performance of our proposed training time-based partitioning and sampling strategy (referred to as partition-based sampling) and compare it with two benchmarks within the PSFL framework: uniform sampling and weighted sampling [5]. In the weighted sampling strategy, the probability of selecting a client is inversely proportional to the square root of the client's training time, i.e., $p_n \propto \frac{1}{\sqrt{t_n}}$. Consequently, this approach introduces a bias in our problem, prioritizing clients with shorter training times. Fig. 6 illustrates the impact of different sampling strategies on training time per round across various distributions (with discrete and Gaussian distributions as representatives) and sequence lengths ($S=3, S=4, S=5$). Overall, our sampling strategy demonstrates greater stability and lower training time across all distributions and sequence lengths, outperforming both uniform and weighted sampling.

3) The efficiency of optimal structure: We evaluate the performance of our theoretically optimal training structure and compare it with the experimental values. We test the optimal values for different numbers of clients participating in each training round ($N_0=10, 20, 50, 100, 200$), while keeping other configurations unchanged. In Fig. 7, we present the theoretical and experimental optimal sequence lengths for $N_0 = 20$. Specifically, the theoretical optimal sequence length calculated in the real domain is 3.48, while the experimental value is 4, demonstrating their close alignment. Results for other client numbers are displayed in Table 1.

4) Comparing different configurations: To demonstrate the robustness of PSFL to non-iid data, we show the test accuracy of PSFL and the baselines at different non-iid settings. Due to space limitation, we only present the test accuracy plots

TABLE I

THEORETICAL AND EXPERIMENTAL OPTIMAL SEQUENCE LENGTHS.

Settings	Theoretical values		Experimental values	
	S	W	S	W
$N_0 = 10$	2.18	4.59	3	3
$N_0 = 20$	3.48	5.74	4	5
$N_0 = 50$	6.54	7.65	7	7
$N_0 = 100$	10.56	9.47	10	10
$N_0 = 200$	16.95	11.73	25	8

for CIFAR-10 across three allocation strategies: $Dir(0.2)$, $ExDir(C{=}1, \alpha = 10)$, and $ExDir(C{=}2, \alpha = 5)$. The results in Fig. 8 demonstrate that PSFL achieves better convergence performance under the same training time compared to the baselines in all non-iid settings, indicating high robustness against statistical heterogeneity. Moreover, to demonstrate the robustness of PSFL to different numbers of participating clients, we present the test accuracy plots for CIFAR-10 across three settings: $N_0 = 50, 100, 200$. Fig. 9 shows that under all settings, PSFL achieves the same target test accuracy with significantly less training time, compared to the baselines.

5) Experiment settings in Fig. 1: We compare the convergence of the three frameworks under the same settings. Specifically, we set $\widetilde{N}{=}9$, allocate data according to $ExDir(C{=}2, \alpha = 10)$, and let t_n follow the discrete distribution. The experiments are run on the CIFAR-10 dataset. Notably, in the PSFL framework, we set the sequence length and parallel width to 3, i.e. $S = 3, W = 3$.

VII. CONCLUSION

In this paper, we propose a novel FL framework, called PSFL, to minimize the total training time by optimizing both the training structure and client sampling strategy. First, we provide a theoretical analysis of the convergence bounds for the PSFL framework under common assumptions. Next, inspired by this analysis, we find out the optimal training structure and introduce a novel training time-based partitioning and sampling strategy. Extensive experiments validate our theoretical analysis and demonstrate the significant performance improvements of the PSFL framework.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62436010, 62172386, and 61872330, in part by the Natural Science Foundation of Jiangsu Province in China under Grants BK20231212 and BK20191194, and in part by the National Natural Science Foundation of USA under Grants CNS 2128378 and CNS 2107014.

APPENDIX

A. Proof of Theorem 1

We first define the *Bregman Divergence* with respect to the function h and arbitrary $\mathbf{x}$, $\mathbf{y}$ as

$$D_h(\mathbf{x}, \mathbf{y}) \triangleq h(\mathbf{x}) - h(\mathbf{y}) - \langle \nabla h(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle. \tag{27}$$

Before proving the **Theorem 1**, we present two lemmas.

Lemma 1. *Let Assumptions 1 to 4 hold and assume that the sampling policy $\Pi_{\mathcal{A}}$ is unbiased, then it holds that*

$$\mathbb{E}[\|\bar{\mathbf{x}}^{(r+1)}-\mathbf{x}^*\|^2] \le \|\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\|^2 + (\frac{2L\eta}{W}+\frac{4L^2\eta^2N_0}{W^2})\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2 + (\frac{8L\eta^2N_0^2}{W^2}-\frac{2\eta N_0}{W})D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)+\frac{4\eta^2N_0}{W}B+\frac{4\eta^2\sigma^2N_0}{W^2}. \quad (28)$$

Proof. The overall model updates after a complete training round is $\bar{\mathbf{x}}^{(r+1)}-\bar{\mathbf{x}}^{(r)} = -\frac{\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbf{g}_{\pi_m^w}^{(r)}$. Then,

$$\mathbb{E}[\|\bar{\mathbf{x}}^{(r+1)}-\mathbf{x}^*\|^2] = \|\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\|^2+\frac{\eta^2}{W^2}\mathbb{E}[\|\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbf{g}_{\pi_m^w}^{(r)}\|^2] - \frac{2\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}[\langle\nabla F_{\pi_m^w}(\mathbf{x}_{w,m-1}^{(r)}),\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\rangle]. \quad (29)$$

Applying Lemma 2 in [25] with $\mathbf{x}=\mathbf{x}_{w,m-1}^{(r)}$, $\mathbf{y}=\mathbf{x}^*$, $\mathbf{z}=\bar{\mathbf{x}}^{(r)}$, $\mu=0$ and $h=F_{\pi_m^w}$ for the third term on the right-hand side in Eq. (29), we get

$$\begin{aligned}
&-\frac{2\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}[\langle\nabla F_{\pi_m^w}(\mathbf{x}_{w,m-1}^{(r)}),\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\rangle]\\
\le&-\frac{2\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}[F_{\pi_m^w}(\bar{\mathbf{x}}^{(r)})-F_{\pi_m^w}(\mathbf{x}^*)-L\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2] \quad (30)\\
=&-\frac{2\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}[D_{F_{\pi_m^w}}(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)+\langle\nabla F_{\pi_m^w}(\mathbf{x}^*),\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\rangle]\\
&+\frac{2L\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2\\
=&-\frac{2\eta N_0}{W}D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)+\frac{2L\eta}{W}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2 \quad (31)
\end{aligned}$$

Here, in the last equation, we use the fact that the sampling policy $\Pi_{\mathcal{A}}$ is unbiased, i.e., $\mathbb{E}[F_{\pi_m^w}(\mathbf{x})]=F(\mathbf{x})$. Moreover, the gradient of F at the optimal model $\mathbf{x}^*$ is zero. For the second term, we apply Jensen's inequality to decompose it:

$$\begin{aligned}
&\frac{\eta^2}{W^2}\mathbb{E}[\|\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbf{g}_{\pi_m^w}^{(r)}\|^2]\\
\le&\frac{4\eta^2}{W^2}\mathbb{E}[\|\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbf{g}_{\pi_m^w}^{(r)}-\nabla F_{\pi_m^w}(\mathbf{x}_{w,m-1}^{(r)})\|^2]\\
&+\frac{4\eta^2}{W^2}\mathbb{E}[\|\sum_{w=1}^{W}\sum_{m=1}^{S_w}\nabla F_{\pi_m^w}(\mathbf{x}_{w,m-1}^{(r)})-\nabla F_{\pi_m^w}(\bar{\mathbf{x}}^{(r)})\|^2]\\
&+\frac{4\eta^2}{W^2}\mathbb{E}[\|\sum_{w=1}^{W}\sum_{m=1}^{S_w}\nabla F_{\pi_m^w}(\bar{\mathbf{x}}^{(r)})-\nabla F_{\pi_m^w}(\mathbf{x}^*)\|^2]\\
&+\frac{4\eta^2}{W^2}\mathbb{E}[\|\sum_{w=1}^{W}\sum_{m=1}^{S_w}\nabla F_{\pi_m^w}(\mathbf{x}^*)\|^2] \quad (32)\\
\le&\frac{4\eta^2N_0\sigma^2}{W^2}+\frac{4L^2\eta^2N_0}{W^2}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}[\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2]\\
&+\frac{8L\eta^2N_0}{W^2}\sum_{w=1}^{W}\sum_{m=1}^{S_w}\mathbb{E}[D_{F_{\pi_m^w}}(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)]\\
&+\frac{4\eta^2}{W}\sum_{w=1}^{W}\mathbb{E}[\|\sum_{m=1}^{S_w}\nabla F_{\pi_m^w}(\mathbf{x}^*)\|^2]. \quad (33)
\end{aligned}$$

We clarify the fact used for each term in the last inequality: (i) Under Assumption 3, the vectors $\{\mathbf{g}_{\pi_m^w}^{(r)}-\nabla F_{\pi_m^w}(\mathbf{x}_{w,m-1}^{(r)})\}$ form a martingale difference sequence, i.e., the conditional expectation is $\mathbb{E}_{\pi_m^w}[\mathbf{g}_{\pi_m^w}^{(r)}|\mathbf{g}_{\pi_{m-1}^w}^{(r)},\ldots,\mathbf{g}_{\pi_1^w}^{(r)}] = \nabla F_{\pi_m^w}(\mathbf{x}_{w,m-1}^{(r)})$, and Lemma 1 in [25] is applied to it. (ii) Assumption 1 holds. (iii) If h is L-smooth and convex, then we have $D_h(\mathbf{x},\mathbf{y})\ge\frac{1}{2L}\|\nabla h(\mathbf{x})-\nabla h(\mathbf{y})\|^2$. (iv) The norm squared $\|\cdot\|^2$ is convex and Jensen's inequality is applied to it. Finally, by substituting Eq. (31) and Eq. (33) into Eq. (29), we establish the lemma. □

Lemma 2. *Let Assumptions 1 to 4 hold and assume that the sampling policy $\Pi_{\mathcal{A}}$ is unbiased. If the learning rate satisfies $\eta \le \frac{c_0}{LS_w}$ for all $w\in\mathcal{W}$ then it holds that*

$$\sum_{m=1}^{S_w}\mathbb{E}\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2 \le \frac{1}{1-2c_0^2}[2\eta^2\sigma^2S_w^2 + \frac{8}{3}L\eta^2S_w^3D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)+2\eta^2S_w^2B]. \quad (34)$$

Proof. Building on the approach used in the proof of Lemma 1, we apply Jensen's inequality to divide the term on the left-hand side in Eq. (34) into four parts and execute similar approximations. Then, we can get

$$\begin{aligned}
&\sum_{m=1}^{S_w}\mathbb{E}\|\mathbf{x}_{w,m-1}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2 = \sum_{m=1}^{S_w}\mathbb{E}[\|-\eta\sum_{i=1}^{m-1}\mathbf{g}_{\pi_i^w}^{(r)}\|^2]\\
\le&\,4\eta^2\sigma^2\sum_{m=1}^{S_w}(m-1)+4L^2\eta^2\sum_{m=1}^{S_w}(m-1)\sum_{i=1}^{m-1}\mathbb{E}\|\mathbf{x}_{w,i}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2\\
&+8L\eta^2\sum_{m=1}^{S_w}(m-1)^2D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)\\
&+4\eta^2\sum_{m=1}^{S_w}(m-1)\frac{1}{N}\sum_{n=1}^{N}\|\nabla F_n(\mathbf{x}^*)\|^2. \quad (35)
\end{aligned}$$

Given the fact $\sum_{i=1}^{m-1}\mathbb{E}\|\mathbf{x}_{w,i}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2 \le \sum_{m=1}^{S_w}\mathbb{E}\|\mathbf{x}_{w,m}^{(r)}-\bar{\mathbf{x}}^{(r)}\|^2$ and $4L^2\eta^2\sum_{m=1}^{S_w}(m-1)\le 2L^2\eta^2S_w^2\le 2c_0^2$, we transfer this term to the left side of the inequality and obtain the lemma after rearrangment. □

Then, we come back to the proof of **Theorem 1**. Since $\eta\le\frac{c_0}{LS_w}$ for all $w\in\mathcal{W}$, we have $\frac{L\eta N_0}{W}\le c_0$. Substituting Eq. (34) to Eq. (28), the coefficient of the $\mathbb{E}[D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)]$ is:

$$\begin{aligned}
&(\frac{2L\eta}{W}+\frac{4L^2\eta^2N_0}{W^2})\frac{1}{1-2c_0^2}\sum_{w=1}^{W}\frac{8}{3}L\eta^2S_w^3+\frac{8L\eta^2N_0^2}{W^2}-\frac{2\eta N_0}{W}\\
\le&\,(4c_0+2)\frac{L\eta}{W}\frac{1}{1-2c_0^2}\frac{8}{3L}c_0^2\sum_{w=1}^{W}S_w+(8c_0-2)\frac{\eta N_0}{W}. \quad (36)
\end{aligned}$$

The coefficient of the constant term is:

$$\begin{aligned}
&(\frac{2L\eta}{W}+\frac{4L^2\eta^2N_0}{W^2})\frac{1}{1-2c_0^2}\sum_{w=1}^{W}(2\eta^2\sigma^2S_w^2+2\eta^2BS_w^2)\\
&+\frac{4\eta^2N_0}{W}B+\frac{4\eta^2\sigma^2N_0}{W^2}\\
\le&\,\frac{\eta^2N_0^2}{W^2}[\frac{4c_0(1+2c_0)}{1-2c_0^2}(\sigma^2\frac{W}{N_0}+\frac{BW}{N_0})+\frac{4BW}{N_0}+\frac{4\sigma^2}{N_0}]. \quad (37)
\end{aligned}$$

We get the recursion as follows:

$$\mathbb{E}[\|\bar{\mathbf{x}}^{(r+1)}-\mathbf{x}^*\|^2]\le\|\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\|^2-b\tilde{\eta}\mathbb{E}[D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)]+c\tilde{\eta}^2. \quad (38)$$

where $b=\frac{16c_0^3-28c_0^2-24c_0+6}{3(1-2c_0^2)}$, $\tilde{\eta}=\frac{\eta N_0}{W}$, and $c=\frac{4c_0(1+2c_0)}{1-2c_0^2}(\sigma^2\frac{W}{N_0}+\frac{BW}{N_0})+\frac{4BW}{N_0}+\frac{4\sigma^2}{N_0}$.

Therefore, we can derive the upper bound, i.e.,

$$\begin{aligned}
&\mathbb{E}[F(\tilde{\mathbf{x}})-F(\mathbf{x}^*)]\\
\le&\frac{1}{R+1}\sum_{r=0}^{R}\mathbb{E}[F(\bar{\mathbf{x}}^{(r)})-F(\mathbf{x}^*)]=\frac{1}{R+1}\sum_{r=0}^{R}\mathbb{E}[D_F(\bar{\mathbf{x}}^{(r)},\mathbf{x}^*)]\\
=&\frac{1}{R+1}\sum_{r=0}^{R}\frac{1}{b\tilde{\eta}}[\mathbb{E}[\|\bar{\mathbf{x}}^{(r)}-\mathbf{x}^*\|^2]-\mathbb{E}[\|\bar{\mathbf{x}}^{(r+1)}-\mathbf{x}^*\|^2]]+c\tilde{\eta} \quad (39)\\
\le&\frac{\|\bar{\mathbf{x}}^{(0)}-\mathbf{x}^*\|^2}{b\tilde{\eta}R}+\frac{c\tilde{\eta}}{b}. \quad (40)
\end{aligned}$$

B. Proof of Corollary 1

Let $\tilde{\eta}=\min\{\sqrt{\frac{r_0}{cR}},\frac{c_0N_0}{LSW}\}\le\frac{c_0N_0}{LSW}$, there are two cases: If $\tilde{\eta}=\sqrt{\frac{r_0}{cR}}$, there is

$$\mathbb{E}[F(\tilde{\mathbf{x}})-F(\mathbf{x}^*)]\le\frac{r_0}{b\tilde{\eta}R}+\frac{c\tilde{\eta}}{b}=\frac{2}{b}\sqrt{\frac{cr_0}{R}}. \quad (41)$$

If $\tilde{\eta}=\frac{c_0N_0}{LSW}$, we have $\frac{c_0N_0}{LSW}\le\sqrt{\frac{r_0}{cR}}$, then

$$\mathbb{E}[F(\tilde{\mathbf{x}})-F(\mathbf{x}^*)]\le\frac{r_0}{b\tilde{\eta}R}+\frac{c\tilde{\eta}}{b}\le\frac{1}{R}\frac{r_0LSW}{bc_0N_0}+\frac{1}{b}\sqrt{\frac{cr_0}{R}}. \quad (42)$$

REFERENCES

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS*, 2017.

[2] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *ICLR*, 2020.

[3] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *NeurIPS*, 2020.

[4] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," in *AISTATS*, 2022.

[5] B. Luo, W. Xiao, S. Wang, J. Huang, and L. Tassiulas, "Tackling system and statistical heterogeneity for federated learning with adaptive client sampling," in *IEEE INFOCOM*, 2022.

[6] F. Lai, X. Zhu, H. V. Madhyastha, and M. Chowdhury, "Oort: Efficient federated learning via guided participant selection," in {*USENIX*} *OSDI*, 2021.

[7] E. Rizk, S. Vlaski, and A. H. Sayed, "Federated learning under importance sampling," *IEEE Transactions on Signal Processing*, vol. 70, pp. 5381–5396, 2022.

[8] F. Wu, S. Guo, Z. Qu, S. He, Z. Liu, and J. Gao, "Anchor sampling for federated learning with partial client participation," in *ICML*, 2023.

[9] W. Bao, H. Wang, J. Wu, and J. He, "Optimizing the collaboration structure in cross-silo federated learning," in *ICML*, 2023.

[10] H. Kim, H. Kim, and G. De Veciana, "Clustered federated learning via gradient-based partitioning," in *ICML*, 2024.

[11] F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 8, pp. 3710–3722, 2020.

[12] R. Zhou, J. Yu, R. Wang, B. Li, J. Jiang, and L. Wu, "A reinforcement learning approach for minimizing job completion time in clustered federated learning," in *IEEE INFOCOM*, 2023.

[13] Z. Zhong, Y. Zhou, D. Wu, X. Chen, M. Chen, C. Li, and Q. Z. Sheng, "P-fedavg: Parallelizing federated learning with theoretical guarantees," in *IEEE INFOCOM*, 2021.

[14] 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.

[15] Y. Liao, Y. Xu, H. Xu, L. Wang, and C. Qian, "Adaptive configuration for heterogeneous participants in decentralized federated learning," in *IEEE INFOCOM*, 2023.

[16] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *IEEE INFOCOM*, 2021.

[17] L. Cui, X. Su, Y. Zhou, and J. Liu, "Optimal rate adaption in federated learning with compressed communications," in *IEEE INFOCOM*, 2022.

[18] J. Perazzone, S. Wang, M. Ji, and K. S. Chan, "Communication-efficient device scheduling for federated learning using stochastic optimization," in *IEEE INFOCOM*, 2022.

[19] Y. Xu, M. Xiao, J. Wu, H. Tan, and G. Gao, "A personalized privacy preserving mechanism for crowdsourced federated learning," *IEEE Transactions on Mobile Computing*, vol. 23, no. 2, pp. 1568–1585, 2023.

[20] Z. Wang, Z. Chang, J. Hu, X. Pang, J. Du, Y. Chen, and K. Ren, "Breaking secure aggregation: Label leakage from aggregated gradients in federated learning," *CoRR*, vol. abs/2406.15731, 2024.

[21] T.-A. Nguyen, J. He, L. T. Le, W. Bao, and N. H. Tran, "Federated pca on grassmann manifold for anomaly detection in iot networks," in *IEEE INFOCOM*, 2023.

[22] J. Wang, S. Guo, X. Xie, and H. Qi, "Protect privacy from gradient leakage attack in federated learning," in *IEEE INFOCOM*, 2022.

[23] S. Shi, C. Hu, D. Wang, Y. Zhu, and Z. Han, "Federated anomaly analytics for local model poisoning attack," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 596–610, 2021.

[24] Z. Wang, M. Song, Z. Zhang, Y. Song, Q. Wang, and H. Qi, "Beyond inferring class representatives: User-level privacy leakage from federated learning," in *IEEE INFOCOM*, 2019.

[25] Y. Li and X. Lyu, "Convergence analysis of sequential federated learning on heterogeneous data," in *NeurIPS*, 2024.

[26] M. Kamp, J. Fischer, and J. Vreeken, "Federated learning from small datasets," in *ICLR*, 2023.

[27] E. M. Arkin and R. O. Roundy, "Weighted-tardiness scheduling on parallel machines with proportional weights," *INFORMS Operations Research*, vol. 39, no. 1, pp. 64–81, 1991.

[28] M. Tang and V. W. Wong, "Tackling system induced bias in federated learning: Stratification and convergence analysis," in *IEEE INFOCOM*, 2023.

[29] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *NeurIPS*, 2020.

[30] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *NeurIPS*, 2017.

[31] Z. Zhu, J. Hong, and J. Zhou, "Data-free knowledge distillation for heterogeneous federated learning," in *ICML*, 2021.

[32] Y. Huang, L. Chu, Z. Zhou, L. Wang, J. Liu, J. Pei, and Y. Zhang, "Personalized cross-silo federated learning on non-iid data," in *AAAI*, 2021.

[33] A. M. Abdelmoniem and M. Canini, "Dc2: Delay-aware compression control for distributed machine learning," in *IEEE INFOCOM*, 2021.

[34] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," in *AISTATS*, 2021.

[35] S. Wang, J. Perazzone, M. Ji, and K. S. Chan, "Federated learning with flexible control," in *IEEE INFOCOM*, 2023.

[36] L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *IEEE ICC*, 2020.

[37] Z. Wang, H. Xu, J. Liu, H. Huang, C. Qiao, and Y. Zhao, "Resource-efficient federated learning with hierarchical aggregation in edge computing," in *IEEE INFOCOM*, 2021.

[38] T. Qi, Y. Zhan, P. Li, J. Guo, and Y. Xia, "Hwamei: A learning-based synchronization scheme for hierarchical federated learning," in *IEEE ICDCS*, 2023.

[39] Z. Yang, S. Fu, W. Bao, D. Yuan, and B. Zhou, "Hierarchical federated learning with adaptive momentum in multi-tier networks," in *IEEE ICDCS*, 2023.

[40] Y. Deng, J. Ren, C. Tang, F. Lyu, Y. Liu, and Y. Zhang, "A hierarchical knowledge transfer framework for heterogeneous federated learning," in *IEEE INFOCOM*, 2023.

[41] Z. Tang, S. Shi, B. Li, and X. Chu, "Gossipfl: A decentralized federated learning framework with sparsified and adaptive communication," *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 3, pp. 909–922, 2022.

[42] Y. Liao, Y. Xu, H. Xu, L. Wang, and C. Qian, "Adaptive configuration for heterogeneous participants in decentralized federated learning," in *IEEE INFOCOM*, 2023.

[43] H. Xu, M. Chen, Z. Meng, Y. Xu, L. Wang, and C. Qiao, "Decentralized machine learning through experience-driven method in edge networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 515–531, 2021.

[44] D. Su, Y. Zhou, L. Cui, and S. Guo, "Boosting dynamic decentralized federated learning by diversifying model sources," *IEEE Transactions on Services Computing*, 2024.

[45] Y. Duan, X. Li, and J. Wu, "Topology design and graph embedding for decentralized federated learning," *Intelligent and Converged Networks*, vol. 5, no. 2, pp. 100–115, 2024.

[46] R. Zaccone, A. Rizzardi, D. Caldarola, M. Ciccone, and B. Caputo, "Speeding up heterogeneous federated learning with sequentially trained superclients," in *IEEE ICPR*, 2022.

[47] J. Lee, J. Oh, S. Lim, S. Yun, and J. Lee, "Tornadoaggregate: Accurate and scalable federated learning via the ring-based architecture," *CoRR*, vol. abs/2012.03214, 2020.

[48] S. Hosseinalipour, S. Wang, N. Michelusi, V. Aggarwal, C. G. Brinton, D. J. Love, and M. Chiang, "Parallel successive learning for dynamic distributed model training over heterogeneous wireless networks," *IEEE/ACM Transactions on Networking*, 2023.

[49] S. Zeng, Z. Li, H. Yu, Y. He, Z. Xu, D. Niyato, and H. Yu, "Heterogeneous federated learning via grouped sequential-to-parallel training," in *DASFAA*, 2022.

[50] S. Wang and M. Ji, "A unified analysis of federated learning with arbitrary client participation," in *NeurIPS*, 2022.

[51] O. Papaspiliopoulos, "High-dimensional probability: An introduction with applications in data science," 2020.

[52] P. Tschandl, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," 2018.