# FAST: A Lightweight Mechanism Unleashing Arbitrary Client Participation in Federated Learning

**Zhe Li**[1] , **Seyedsina Nabavirazavi**[2] , **Bicheng Ying**[3] , **Sitharama Iyengar**[2] and **Haibo Yang**[1]
[1]Rochester Institute of Technology
[2]Florida International University
[3]Google Inc.
{zl4063, hbycis}@rit.edu, snabavir@fiu.edu, ybc@google.com, iyengar@cis.fiu.edu,

## Abstract

Federated Learning (FL) provides a flexible distributed platform where numerous clients with high data and system heterogeneity can collaborate to learn a model. While previous research has shown that FL can handle diverse data, it often completely assumes idealized conditions. In practice, real-world factors make it hard to predict or design individual client participation. This complexity results in an unknown participation pattern - *arbitrary client participation (ACP)*. Hence, key open problem is to understand the impact of client participation and develop a lightweight mechanism to support ACP in FL. In this paper, we first empirically investigate the client participation's influence in FL, revealing that FL algorithms is adversely impacted by ACP. To alleviate the impact, we propose a lightweight solution, Federated Average with Snapshot (FAST), that supports almost ACP for FL and can seamlessly integrate with other classic FL algorithms. Specifically, FAST enforces clients to take a snapshot once in a while and facilitates ACP for the majority of training processes. We prove that the convergence rates of FAST in non-convex and strongly-convex cases match those under ideal client participation. Furthermore, we empirically introduce an adaptive strategy to dynamically configure the snapshot frequency, tailored to accommodate diverse FL systems. Extensive experiments show that FAST significantly improves performance under ACP and high data heterogeneity.

## 1 Introduction

Federated Learning (FL) stands out as an emerging distributed machine learning framework where a large number of clients (i.e., computing nodes or devices) collaborate together to train a global model under the coordination of a central server [McMahan *et al.*, 2017; Kairouz *et al.*, 2021]. FL establishes itself as a powerful and flexible distributed platform, fostering collaboration among diverse clients characterized by substantial heterogeneity in data and system while preserving the privacy of raw data residing within each client. Hence, previous research endeavors have yielded a spectrum of efficient algorithms capable of achieving optimal convergence rates in theory and delivering great performance in some practical cases, in the presence of varying degrees of data heterogeneity [Kairouz *et al.*, 2021; Zhao *et al.*, 2018; Li *et al.*, 2019; Karimireddy *et al.*, 2020; Yang *et al.*, 2020; Wang *et al.*, 2021].

Nevertheless, realizing these favorable outcomes often hinges on the ideal system condition (i.e., ideal client participation). Specifically, most FL algorithms presume that client participation can be fully known, controlled, predicted or tracked. For example, [McMahan *et al.*, 2017; Acar *et al.*, 2021; Cho *et al.*, 2023] assume partial client participation, where participation follows a known or controllable random process, such as ergodic, mixing, or independent processes. [Yang *et al.*, 2022b; Gu *et al.*, 2021; Yan *et al.*, 2024] suppose that each client participates at least once within certain rounds.

In practice, however, each client's participation is ***highly dynamic, unknown and unpredictable*** [Bonawitz *et al.*, 2019; Soltani *et al.*, 2022] since clients frequently exhibit heterogeneous and dynamically shifting attributes, including computational power, communication capacity, and availability [Kairouz *et al.*, 2021; Bonawitz *et al.*, 2019; Yang *et al.*, 2021]. These variations stem from the unique characteristics of each individual client and the dynamics of distributed learning systems. The dynamic, unknown, and unpredictable intricacies of client participation make it challenging and even impossible to ascertain a priori beforehand. Moreover, in some FL systems, such as cross-device FL, tracking client participation is either infeasible or not permitted [Kairouz *et al.*, 2021]. We name these patterns as ***arbitrary client participation (ACP)***, reflecting its dependence on various system factors and the absence of explicit client tracking. Clearly, it leaves a substantial gap between algorithmic designs built on the premise of ideal client participation and the real-world applications of FL involving ACP. Also, without any conditions on client participation, a constant error arises for ACP as identified by the lower bound [Cho *et al.*, 2022; Wang *et al.*, 2020; Yang *et al.*, 2022b], implying that no algorithm can achieve stationary point convergence in such case. This observation motivates us to pose the following fundamental question:

**Question**: *Is it possible to design a lightweight mechanism for FL that can accommodate arbitrary client participation with theoretical guarantees?*

In this paper, we show an affirmative answer to this question by proposing a new client participation mechanism for FL, denoted as Federated Averaging with SnapshoT (FAST). In contrast to most FL algorithms that necessitate ideal client participation in each communication round, FAST imposes a minimal requirement for client participation by intermittently implementing a snapshot step. This approach significantly diminishes the requirement for individual client participation, enabling ACP for the majority of the training process. We highlight our contributions as follows:

- Through extensive experiments, we reveal that the mismatch between ideal client participation in algorithm design and ACP in practice leads to severe performance degradation, especially in highly heterogeneous data scenarios. These phenomena are universal and extend beyond specific algorithms, as observed across multiple FL algorithms.
- To address this issue, we introduce FAST, a lightweight FL framework that requires only intermittent snapshot steps, enforcing fully random client participation during these steps while accommodating ACP within the system at all other times. This requirement applies to the client cohort rather than individual clients, allowing the participating group to be statistically representative. This is a milder condition compared to existing works (see Table 1), as it eliminates the need to track each client individually.
- Theoretically, we demonstrate that, under mild conditions, FAST can achieve a convergence rate of $\mathcal{O}(1/\sqrt{mRK})$ for non-convex functions and $\widetilde{\mathcal{O}}(1/R)$ for strongly-convex functions, where $R$ is the number of communication rounds, $K$ is the number of local steps, and $m$ is the number of participating clients. These rates can match the rates of those under ideal client participation.
- Empirically, we further propose an adaptive strategy adjusting the snapshots' frequency dynamically and show that FAST can seamlessly integrate with other classic FL algorithms. Also, extensive experiments verify its effectiveness.

## 2 Related Work

**Ideal Client Participation: full client participation and uniformly random client participation.** In FL, client participation can be seen as a proxy for system heterogeneity. Due to the inherent complexity of real-world FL systems, explicitly modeling client participation proves challenging [Bonawitz *et al.*, 2019; Yang *et al.*, 2021]. Most existing FL algorithms often make an assumption about ideal client participation, typically relying on either full client participation [Gorbunov *et al.*, 2021; Haddadpour *et al.*, 2019; Lin *et al.*, 2018; Wang and Joshi, 2019; Wang and Joshi, 2021; Yu *et al.*, 2019] or uniformly random client participation [McMahan *et al.*, 2017; Li *et al.*, 2019; Karimireddy *et al.*, 2020; Yang *et al.*, 2020; Wu *et al.*, 2023; Zhang *et al.*, 2023; Wang *et al.*, 2023; Liu *et al.*, 2021; Jhunjhunwala *et al.*, 2022; Grudzień *et al.*, 2023]. This assumption requires that the server force all clients or at least uniformly and randomly sample a subset of clients to participate in each communication round. However, each client in FL is not entirely under the server's control. While the server may sample a client for a specific round, the client is highly likely not to participate due to various system factors such as drop-out, communication congestion, and other unpredictable factors [Kairouz *et al.*, 2021; Yang *et al.*, 2021]. It is worth noting that the server can invest additional resources to enforce uniform client participation, such as sampling more clients and extending the waiting time in each round. Yet, this approach leads to prolonged training times due to significant communication and computation overhead [Zhou *et al.*, 2022]. As shown in [Luo *et al.*, 2022], enforcing uniform client participation in every round by the server results in slow wall-clock time for FL training.

**Controllable Client Participation.** In addition to uniform client participation, another approach in the field involves modeling client participation as a controllable random process. One line of works utilizes predefined patterns or probabilities as the model of client participation [Chen *et al.*, 2022; Yang *et al.*, 2022b; Fraboni *et al.*, 2021; Ruan *et al.*, 2021; Gu *et al.*, 2021; Avdiukhin and Kasiviswanathan, 2021; Wang and Ji, 2022; Koloskova *et al.*, 2022]. The main idea is to allow asynchronous communication or fixed participation patterns (e.g., given probability) for clients to participate flexibly in training. However, existing works in this area often require extra assumptions, such as bounded delay and extra memory [Yang *et al.*, 2022b; Ruan *et al.*, 2021; Gu *et al.*, 2021; Koloskova *et al.*, 2022] and identical computation rate [Avdiukhin and Kasiviswanathan, 2021]. Moreover, several works explore some unique scenarios of client participation. For instance, [Chen *et al.*, 2022] introduced a novel client subsampling scheme considering the importance of updates, relying solely on the norm of the update. [Malinovsky *et al.*, 2023; Cho *et al.*, 2023] investigated cyclic client participation. [Wang and Ji, 2022] provided a unified analysis for various client participation, including regularized, ergodic, independent, and mixing participation. The implicit assumption in these studies is that client participation is either known, largely controllable or adheres to predefined patterns. It is also noteworthy to mention a related work [Wang and Ji, 2023], wherein the estimated probability of each client's participation was used for a re-weighting process under unknown participation statistics. However, estimating such probabilities can be challenging in practice, such as cross-device FL [Kairouz *et al.*, 2021].

Each of these approaches contributes to the diverse client participation strategies employed in FL. However, these strategies often necessitate adherence to specific patterns, which may not align seamlessly with practical FL scenarios characterized by *highly dynamic, unknown and unpredictable* nature. In this paper, we introduce a more general and practical pattern - *arbitrary client participation (ACP)*. This implies that we do not impose any assumptions on client participation for the majority of training rounds. Our aim is to offer a flexible and realistic framework that accommodates various client participation scenarios in real-world FL applications.

**Comparison of Related Work.** We compare some related work about ACP in Table 1. Except for differences in partici-

pation and convergence rates, we still need to compare some important points. For FedAmplify [Wang and Ji, 2022], it can achieve the convergence rate of $\mathcal{O}(\frac{1}{\sqrt{mKR}})$ only in some ideal cases (see Sec. 5 in [Wang and Ji, 2022]), and the server requires participation frequency for each client. For MIFA [Gu *et al.*, 2021], each client needs to participate in training at least once in the one-time window. For Anarchic Federated Learning (AFL) [Yang *et al.*, 2022b], the server needs to identify and store local models, and each client needs to participate in training at least once in the one-time window. In contrast, FAST framework has no extra assumptions for client participation and can achieve the ideal convergence rate. In addition, regular FAST does not demand to store extra information.

| Algorithm | Participation Condition | Client Track | Convergence Rate |
|---|---|---|---|
| MIFA | Bounded inactive rounds | ✓ | $\mathcal{O}(1/\sqrt{mKR})$ |
| AFL | Bounded inactive rounds | ✓ | $\mathcal{O}(1/\sqrt{mKR})$ |
| FedAU | Every client participates | ✓ | $\mathcal{O}(1/\sqrt{mKR})$ |
| FedAmplify | Regularized, mixing, independent process | ✗ | $\mathcal{O}(1/\sqrt{mKR})$ |
| FedAvg | Uniform participation in every round | ✗ | $\mathcal{O}(1/\sqrt{mKR})$ |
| FAST (ours) | Uniform participation occasionally | ✗ | $\mathcal{O}(1/\sqrt{mKR})$ |
| Lower Bounds | No assumptions | - | $\Omega(1)$ |

Table 1: Comparison of Client Participation in FL and Convergence Rate for Non-convex Functions.

# 3 The Impact of Client Participation in FL

In this section, our goal is to investigate the impact of client participation on FL performance. We first introduce the fundamental formulation and the standard FedAvg. Subsequently, we examine FedAvg's performance across various client participation scenarios and show the adverse effects of different ACP. This highlights the gap between current algorithm designs and practical FL systems, thus motivating us to develop a new framework to accommodate ACP for FL.

## 3.1 Federated Learning and Federated Averaging

**Problem Formulation.** In a FL system with $M$ clients, our goal is to minimize the objective function as:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} F(\boldsymbol{x}) := \frac{1}{M}\sum_{i=1}^{M} F_i(\boldsymbol{x}), \tag{1}$$

where $\boldsymbol{x}$ is a $d$-dimension model parameter, $M$ is the total number of clients, and $F_i(\boldsymbol{x}) := \frac{1}{|S_i|}\sum_{\xi\in D_i} F(\boldsymbol{x},\xi), \forall i \in [M]$ is the local loss function associated with local dataset $D_i$ that is IID sampled from one underlying distribution $P_i$. One of the critical features of FL is that each client has a subtly different local data distribution, i.e., $P_i \neq P_j$ if $i \neq j$. This leads to heterogeneous (or Non-IID) data in the FL system, causing model drift and non-trivial performance degradation [Kairouz *et al.*, 2021; Wang *et al.*, 2021].

**FedAvg Algorithm.** The Federated Average (FedAvg) algorithm [McMahan *et al.*, 2017] stands as the pioneering exemplar FL algorithm, inspiring numerous variants. Most of the FL algorithms follow the typical parameter-server architecture. In each communication round $r \in [R]$, the server first selects a subset of clients to participate and broadcasts the current global model $\boldsymbol{x}_r$ to each client. Upon receiving the global model, each participating client locally optimizes the loss function for some local steps using the local dataset without communication. For example, FedAvg takes $K$ local steps using the vanilla stochastic gradient descent method. That is, $\boldsymbol{x}^i_{r,k+1} = \boldsymbol{x}^i_{r,k} - \eta_c \nabla F_i(\boldsymbol{x}^i_{r,k}, \xi^i_{r,k}), k \in \{0, \cdots, K-1\}$ starting from $\boldsymbol{x}^i_{r,0} = \boldsymbol{x}_r$ where $\xi^i_{r,k} \sim D_i$. After local computation, the client sends the model update $\boldsymbol{x}^i_r = \boldsymbol{x}^i_{r,K}$ to the server. At the server side, the server updates the global model by aggregating all the returned local model, i.e., $\boldsymbol{x}_{r+1} = \frac{1}{|S_r|}\sum_{i\in S_r} \boldsymbol{x}^i_r$ where $S_r$ is the set of participated clients in the $r$-th round. Then, the next training round begins.

Undoubtedly, client participation, denoted as the set $S_r$, stands as a pivotal factor influencing the performance of FL models. While the majority of works in FL concentrate on mitigating data heterogeneity, the implications of client participation remain largely under-explored. To ensure convergence guarantees in FL algorithms, specific conditions must be imposed on client participation. Essentially, these algorithms necessitate a regulated form of client participation, such as participation through uniformly random sampling or a predetermined probability distribution, as detailed in Sec.2.

However, in real-world FL systems, client participation is inherently dynamic, prone to changes in each round [Bonawitz *et al.*, 2019; Yang *et al.*, 2021]. Even if the server employs an ideal sampling way, like uniformly random sampling, actual client participation remains unknown and largely uncontrollable. We term this as ***arbitrary client participation***, signifying that $S_r$ includes any sampling from the whole client set $[M]$, thereby incorporating a diverse array of participation schemes. This process is determined by various inherent system factors, such as client failures and status changes [Bonawitz *et al.*, 2019; Yang *et al.*, 2021]. Hence, there exists a conflict between existing algorithm designs with ideal client participation and practical FL systems with ACP. This motivates us to explore the impact of ACP on FL algorithms' performance.

## 3.2 The Impact of Client Participation in FL

**Arbitrary Client Participation Simulation.** We delve into FedAvg's performance across four client participations characterized by distinct distributions: uniform, Beta, Gamma, and Weibull. Uniform client participation entails the random client selection from the entire client set, which is an idealized scenario in current FL algorithms. The Beta distribution is commonly employed to model events constrained within an interval. The Gamma distribution finds application in characterizing the frequency of a sequence of events associated with time or distance. The Weibull distribution is widely utilized in reliability or survival analysis [Lai *et al.*, 2006]. In FL, the server often receives returns from clients within a given time window. Hence, it is reasonable to use uniform distribution as a baseline for ideal client participation. The latter three distributions are utilized to approximate different real-world scenarios, serving as representatives of ACP.

It is important to emphasize that our primary goal is not to precisely model client participation in FL but to explore the impact of different potential client participation scenarios. Also, we aim to highlight the adverse effects resulting from

the mismatch between the ideal client participation used in the current algorithm design and ACP observed in practical FL.

**Experiment Settings.** We perform extensive experiments on Fashion-MNIST [Xiao *et al.*, 2017] and CIFAR-10 [Krizhevsky *et al.*, 2009], considering various Non-IID degrees and utilizing the four distributions to simulate different client participation. As shown in Table 2, we scrutinize the model performance using FedAvg. For each case, we record the last five results and report the mean and standard deviation of test accuracy. Here, we only show key findings and delegate the detailed settings and results for other datasets and algorithms to Sec. 5 and Appendix.

**Observations.** We have three key observations. First, FedAvg's performance is significantly influenced by client participation. As shown in Table 2, the model accuracy varies across different client participation cases, with uniform participation yielding the best performance among these four cases. This performance difference is substantial, ranging from $3\%$ to $18\%$. These results align with practical FL simulations, where uncontrolled client participation induced by system heterogeneity leads to non-trivial model performance degradation [Yang *et al.*, 2021]. Second, this performance degradation strongly correlates with the degree of Non-IID data. In our setting, we adopt the common approach of generating Non-IID data using the Dirichlet distribution [Acar *et al.*, 2021], with the parameter $\alpha$ controlling the Non-IID degree. A smaller $\alpha$ corresponds to a higher Non-IID degree. For datasets with a higher degree of Non-IID data (smaller $\alpha$), the model accuracy gap between uniform and other cases becomes more pronounced. For instance, on the Fashion-MNIST dataset, the model behaves similarly for different client participation cases with less Non-IID data (i.e., $\alpha = 1$). However, as the Non-IID degree gets higher, such as $\alpha = 0.05$, the accuracy gap between uniform and other participation could be as large as $18\%$. Third, the performance degradation for ACP (in the latter three cases) is universal, which extends beyond FedAvg, as evidenced by consistent observations across other FL algorithms such as FedProx and SCAFFOLD.

It is essential to note that occasional enforcement of uniform client participation in FL is feasible. For instance, the server can sample a larger number of clients and allocate sufficient time for each communication round, allowing ample clients to complete local computations. However, this strategy inevitably demands more resources and significantly extends the training time due to longer waiting time. Thus, it becomes unrealistic to enforce uniform client participation in every round. In addition, without imposing any constraints on client participation, FedAvg is theoretically incapable of asymptotically converging to a stationary point [Yang *et al.*, 2022b; Yang *et al.*, 2022a] and experiences non-trivial performance degradation in practice, as shown above. This realization motivates us to develop a lightweight client participation mechanism, aiming to achieve performance similar to that of uniform participation while imposing fewer constraints on FL systems.

# 4 Federated Average with Snapshot (FAST)

We first introduce a lightweight client participation mechanism - Federated Average with SnapshoT (FAST). Then, we provide the convergence analysis for non-convex and strongly-convex cases. Lastly, to eliminate the need for predefining the snapshot frequency, we empirically propose a strategy to dynamically adjust the snapshot frequency for FAST.

**Algorithm 1** Federated Average with Snapshot (FAST)

**Initialize:** model parameter $\boldsymbol{x}_0$, learning rate $\eta_c$, local update steps $K$, communication rounds $R$, snapshot step interval $I$ (or probability $q$).
**for** $r = 0, \ldots, R-1$ **do**
  If $r\%I == 0$ (with $q = 1/I$): ▶ Snapshot
    Server enforces *uniformly* random clients $\mathcal{S}_r = \mathcal{S}_r^u$ ($|\mathcal{S}_r^u| = m$) to participate.
  Otherwise: ▶ Arbitrary
    Server allows *arbitrarily* random clients $\mathcal{S}_r = \mathcal{S}_r^a$ ($|\mathcal{S}_r^a| = n$) to participate.
  Each client $i \in \mathcal{S}_r$ computes in parallel:
    $\boldsymbol{x}_{r,k+1}^i = \boldsymbol{x}_{r,k}^i - \eta_c \nabla F_i(\boldsymbol{x}_{r,k}^i, \xi_{r,k}^i), k \in [K]$
    Send $\boldsymbol{x}_r^i = \boldsymbol{x}_{r,k+1}^i$ to the server
  Server aggregation: $\boldsymbol{x}_{r+1} = \frac{1}{|\mathcal{S}_r|} \sum_{i \in \mathcal{S}_r} \boldsymbol{x}_r^i$
**end for**
*Note: Lines 7-10 can be replaced by any other FL algorithms.*

## 4.1 Algorithm Description

As shown in Algo. 1, we introduce a lightweight client participation mechanism for FL. In each communication round $r \in [R]$, we design two client participation options. If $r\%I == 0$, the server takes a snapshot step that requires to enforce a round of uniform client participation denoted as client set $S_r^u$ with cardinality $m$ for that round, where $I$ is a hyper-parameter to control the snapshot frequency. Otherwise, the server does not put any constraints and can accommodate any system heterogeneity by allowing ACP denoted as set $S_r^a$ with cardinality $n$. On the client side, each participating client takes $K$ Stochastic Gradient Descent (SGD) steps and sends the returns back to the server, mirroring the procedure in FedAvg. Subsequently, after local computations, the server aggregates all the returns and updates the global model. Additionally, from a probabilistic perspective, in each round, there exist a probability $q$ of enforcing snapshots and a complementary probability of $1 - q$ to permit ACP. Here $q = 1/I$ can be regarded as the snapshot probability or frequency.

In general, the uniqueness of FAST is utilizing a snapshot step every $I$ rounds by enforcing a round of uniform client participation. The trade-offs of the snapshot are discussed as follows: 1) Resources. Although uniform client participation is an ideal situation in FL, it can still be achieved in practice by using some strategies. For instance, the server can initially sample $1.3 \times m$ clients and extend the waiting period [Bonawitz *et al.*, 2019]. This approach would make uniformly random client participation hold statistically, and mirrors practical FL simulations, such as 11.6% dropout rate and an optimal waiting time [Yang *et al.*, 2021]. Hence, enforcing uniform client participation is practical in reality. Unfortunately, this approach to achieve uniform participation consumes more resources, such as time and computation. However, in FAST,

| Participation \ $\alpha$ | Fashion-MNIST | | | | | CIFAR-10 | | |
|---|---|---|---|---|---|---|---|---|
| | 0.05 | 0.1 | 0.3 | 0.5 | 1.0 | 0.1 | 0.5 | 1.0 |
| Uniform | **84.10%±2.4** | **86.85%±1.9** | **89.39%±0.7** | **91.39%±0.3** | **92.21%±0.3** | **80.18%±0.6** | **80.49%±0.4** | **80.83%±0.7** |
| Beta | 74.84%±1.2 | 79.89%±4.0 | 86.40%±1.1 | 88.74%±0.4 | 89.43%±0.1 | 68.30%±0.9 | 72.27%±0.4 | 73.32%±0.6 |
| Gamma | 66.65%±4.7 | 81.81%±1.8 | 88.41%±0.5 | 87.79%±0.4 | 89.44%±0.2 | 70.90%±0.8 | 73.20%±0.4 | 73.04%±0.3 |
| Weibull | 73.15%±5.1 | 78.78%±1.6 | 88.80%±0.4 | 89.20%±0.6 | 89.53%±0.2 | 71.74%±0.7 | 73.21%±0.7 | 73.75%±0.3 |

Table 2: Test Accuracy Comparison of FedAvg

snapshots just occupy a small portion of entire training rounds, so FAST can save resources compared to completely uniform participation in other FL algorithms. 2) Benefits. By the snapshot, our FAST can simultaneously enjoy the optimal convergence rates as those with uniform client participation shown in Sec. 4.2 and achieve improved performance when compared with ACP shown in Sec. 5.

## 4.2 Convergence Analysis

We first state several standard assumptions commonly used in our work and other works about optimization and FL [Kairouz *et al.*, 2021; Wang *et al.*, 2021].

**Assumption 1 ($L$-Lipschitz Continuous Gradient)** *For any $\boldsymbol{x}$ and $\boldsymbol{y}$, there exists a constant $L > 0$ such that $\|\nabla F(\boldsymbol{x}) - \nabla F(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$ and $\|\nabla F_i(\boldsymbol{x}) - \nabla F_i(\boldsymbol{y})\| \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$.*

**Assumption 2 (Unbiased Stochastic Gradients with Bounded Variance)** *The stochastic gradient calculated by the client or server is unbiased with bounded variance: $\mathbb{E}[\nabla F_i(\boldsymbol{x}, \xi)] = \nabla F_i(\boldsymbol{x})$ and $\mathbb{E}[\|\nabla F_i(\boldsymbol{x}, \xi) - \nabla F_i(\boldsymbol{x})\|^2] \leq \sigma^2$, where $\xi$ is a data sample.*

**Assumption 3 (Bounded Gradient Dissimilarity)** *For any $i \in [M]$, $\|\nabla F_i(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\|^2 \leq \sigma_G^2$.*

Next, we offer FAST's convergence under non-convexity.

**Theorem 1** (Convergence of FAST for Non-convex Functions). *Under assumptions 1, 2 and 3, supposing that the probability $q \geq \frac{(2LK\eta_c-1)G_2+2K^2\sigma_G^2}{G_1+(2LK\eta_c-1)G_2-2LK\eta_cG_3+2K^2\sigma_G^2}$ and the learning rate $\eta_c \leq \min\left\{\frac{1}{8LK}, \frac{nq+m(1-q)}{5mnLK}\right\}$, then the sequence $\{\boldsymbol{x}_r\}$ generated by FAST satisfies:*

$$\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}\|\nabla F(\boldsymbol{x}_r)\|^2 \leq \underbrace{\frac{4\zeta}{KR\eta_c}}_{\textit{Optimization Error}} + \underbrace{\frac{4\big(qn+(1-q)m\big)L\eta_c}{mn}\sigma^2}_{\textit{Statistical Error}} + \underbrace{\Big(120(1-q)+60q\Big)L^2K^2\eta_c^2\sigma_G^2}_{\textit{Heterogeneity Error}},$$

*where $\zeta := F(\boldsymbol{x}_0) - F(\boldsymbol{x}^*)$, $\boldsymbol{x}^*$ is the optimal solution, and $G_{1-3}$ are defined in Appendix.*

With a proper learning rate, FAST achieves convergence rate:

**Corollary 1** *With $\eta_c = \mathcal{O}\left(\frac{\sqrt{mn}}{\sqrt{RK(nq+m(1-q))}}\right)$, the convergence rate of FAST is*

$$\frac{1}{R}\sum_{r=1}^{R}\mathbb{E}\|\nabla F(\boldsymbol{x}_r)\|^2 = \mathcal{O}\left(\sqrt{\frac{nq+m(1-q)}{nmKR}}\right) + \mathcal{O}\left(\frac{mnK}{(nq+(1-q)m)R}\right)$$

The convergence error of FAST comprises three components: 1) optimization error depending on the initial point $\boldsymbol{x}_0$, 2) statistical error associated with stochastic gradient noise $\sigma$, and 3) error arising from heterogeneous data and local updates in FL. Notably, the third error exhibits a quadratic relationship with the learning rate. Hence, the first two terms dominate when using a sufficiently small learning rate. With an appropriate learning rate, the convergence rate is $\mathcal{O}\Big(\sqrt{\frac{nq+m(1-q)}{nmKR}}\Big)$ for a suitably large round $R \geq \frac{(mnK)^3}{[nq+m(1-q)]^3}$. In a special case ($m = n$), the convergence rate becomes:

**Corollary 2** *With $m = n$, FAST achieves convergence rate:*

$$\frac{1}{R}\sum_{r=1}^{R}\|\nabla F(\boldsymbol{x}_r)\|^2 = \mathcal{O}\Big(\sqrt{\frac{1}{mRK}}\Big). \tag{2}$$

**Remark 1** In non-convex functions, this sublinear convergence rate shows the speedup in terms of clients' number $m$ and the local steps $K$, which matches the optimal convergence rate in FL with uniform client participation in every round [Karimireddy *et al.*, 2020; Yang *et al.*, 2020].

**Remark 2** It is worth noting that there exists a requirement of the snapshot probability/frequency $q$ (or $I$). Specifically, it depends on data heterogeneity in the FL system: $q \geq \frac{(2LK\eta_c-1)G_2+2K^2\sigma_G^2}{G_1+(2LK\eta_c-1)G_2-2LK\eta_cG_3+2K^2\sigma_G^2} = \frac{1}{1+(G_1-2LK\eta_cG_3)/(2K^2\sigma_G^2+2LK\eta_cG_2-G_2)}$. For every heterogeneous data in FL, we can choose a proper $q$ such that it can converge at such an optimal rate. We list two special cases to show FAST's generalization. 1) $\sigma_G \to 0$. If data is IID among clients, then $q \geq 0$, meaning that we can always avoid using the snapshot step and set $q = 0$. This situation corresponds to traditional distributed learning where each client has access to a shared dataset or IID datasets. In such cases, the choice of which subset of clients participates is inconsequential, as the training data used remains statistically identical. 2) $\sigma_G \to \infty$. If data is extremely highly Non-IID, the lower bound of $q$ will approach 1, requiring a high frequency of snapshots. In extreme cases, it might require uniform client participation in every round to guarantee convergence.

If we assume a strongly convex condition on the function, we can achieve a faster convergence rate.

**Assumption 4 (Strong Convexity)** *For any $\boldsymbol{x}$ and $\boldsymbol{y}$, $F_i$ is $\mu$-convex with a constant $\mu > 0$, if $F_i(\boldsymbol{y}) \geq F_i(\boldsymbol{x}) + \nabla F_i(\boldsymbol{x})^T(\boldsymbol{y} - \boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2, \forall i \in [M]$.*

**Theorem 2** (Convergence of FAST for Strongly Convex Functions). *Under assumptions 1,2,3 and 4, supposing that the learning rate* $0 < \eta_c \leq \min\left\{\frac{1}{20mLK}, \frac{1}{20nLK}\right\}$ *and the probability* $q \geq 1 - \left(\frac{\mu K \eta_c - 16L^2K^2\eta_c^2}{4\sigma_G^2}\right)$*, the sequence* $\{x_r\}$ *generated by FAST satisfies:*

$$\mathbb{E}\left\|\boldsymbol{x}_R - \boldsymbol{x}^*\right\|^2 \leq \exp\left(-\mu K R \eta_c\right)\kappa + \frac{(1-q)}{2\mu} K\eta_c + \frac{8}{\mu}K\eta_c\sigma_G^2 + \frac{2\big(qn+(1-q)m\big)}{mn\mu}\eta_c\sigma^2,$$

*where* $\kappa = \left\|\boldsymbol{x}_0 - \boldsymbol{x}^*\right\|^2$ *and* $\boldsymbol{x}^*$ *is the optimal solution.*

**Corollary 3** *For Theorem 2, supposing* $\mu > 0$*,* $m = n$*,* $\eta_c \leq \frac{1}{20mLK}$ *and* $R \geq 20mL$*, we can obtain*

$$\mathbb{E}\left\|\boldsymbol{x}_R - \boldsymbol{x}^*\right\|^2 = \widetilde{\mathcal{O}}\Big(\exp\big(-\frac{\mu R}{20mL}\big)\Big) + \widetilde{\mathcal{O}}\Big(\frac{1-q}{\mu R}\Big) + \widetilde{\mathcal{O}}\Big(\frac{1}{\mu R}\sigma_G^2\Big) + \widetilde{\mathcal{O}}\Big(\frac{1}{\mu mKR}\sigma^2\Big),$$

*where* $\widetilde{\mathcal{O}}(\cdot)$ *subsumes all log-terms and constants. Accordingly, FAST achieves a convergence rate of* $\widetilde{\mathcal{O}}(1/R)$*.*

**Remark 3** In strongly-convex functions, FAST can achieve a faster convergence rate of $\widetilde{\mathcal{O}}(1/R)$ compared to the non-convex case. It is worth mentioning that this rate can match those achieved in FL with ideal client participation [Li *et al.*, 2019]. In conjunction with Corollary 1, it is clear that, under appropriate hyper-parameter settings, FAST can achieve the same convergence rate under ACP as these FL algorithms with ideal client participation.

### 4.3 Adaptive FAST

As shown in Algo. 1, FAST introduces an extra hyper-parameter, $q$ (or $I$), representing the snapshot probability (or frequency). Obviously, the effective performance of our FAST is evidently contingent on the selection of an appropriate $q$, as indicated by the ablation study on $q$ in Sec. 5. In practice, obtaining prior knowledge to consistently set the optimal $q$ poses a challenge. To address this issue, we propose an adaptive strategy to dynamically update $q$ as shown in Algo. 2.

**Algorithm 2** Adaptive $q$ in FAST

1: **Initialize:** $q_0 = 0$, $\Delta = 0$, $\lambda(\textit{default} = 1)$.
2: **for** round $r = 0, 1, ..., R-1$ **do**
3: Obtain $acc_r$ from the FL system
4: $\Delta \leftarrow \Delta - acc_r$
5: $q_{r+1} \leftarrow \min(1, \max(0, q_r + \lambda\Delta))$
6: $\Delta \leftarrow acc_r$
7: **end for**

In more detail, we initiate with $q = 0$ to refrain from enforcing client participation at the beginning of training procedure. Meanwhile, $q$ is adjusted in each round based on the training accuracy difference $\Delta$ between the current and previous rounds. When $\Delta > 0$, indicating a decrease in training accuracy compared to the last round, we increase $q$ by $\lambda\Delta$. This adjustment aims to increase the probability of uniform client participation, improving performance. Conversely, when $\Delta < 0$, signifying an increase in training accuracy in the current round, we decrease $q$ by $\lambda\Delta$. This reduction aims to diminish the probability of uniform client participation, ensuring a more substantial contribution from arbitrary participation in the training process. Line 5 ensures that the frequency $q$ stays within the range of [0,1]. For the selection of $\lambda$, we conduct a series of experiments to assess the performance under different $\lambda$. Our results show that the adaptive FAST is less sensitive to the choice of $\lambda$, and choosing a default $\lambda = 1$ works well under different settings provided in Sec. 5 and Appendix.

## 5 Experiments

We provide our experiment settings and main results in Sec. 5.1, while leaving other details to Appendix.

**Datasets and Models.** We employ Fashion-MNIST [Xiao *et al.*, 2017] and CIFAR-10 datasets [Krizhevsky *et al.*, 2009] for image classification tasks, and we utilize the Shakespeare dataset [Caldas *et al.*, 2018] for the next character prediction task. For image classification tasks, we train convolutional neural network (CNN) models in our FL system, but the models are different for these two datasets, aiming to adapt to the characteristics of different tasks. For character prediction tasks, we train the Char-LSTM model. Comprehensive details regarding datasets and models can be found in Appendix.

**FL System.** Our FL system comprises 100 clients in total for Fashion-MNIST and CIFAR-10 and 139 clients for Shakespeare. In each round, only $10\%$ clients are chosen to participate in training. 1) Data Heterogeneity. The experiments on Fashion-MNIST and CIFAR-10 adhere to balanced and Non-IID datasets, implying that each client possesses an equal number of data, yet label distributions differ across clients. To establish this setup, we leverage the FedLab [Zeng *et al.*, 2023] for data partitioning and employ Dirichlet Distribution to generate label-based distributions for each client. By adjusting the concentration parameter $\alpha$, we can control the Non-IID degree of data. Generally, a smaller $\alpha$ corresponds to higher data heterogeneity. Shakespeare dataset is naturally Non-IID, so we directly distribute each user's data to each client. 2) Client Participation. We employ four distributions (i.e., uniform, Beta, Gamma and Weibull) to simulate various participation. The uniform distribution serves as ideal client participation, and the other three distributions act as proxies for ACP. 3) Algorithms. We implement three baselines: FedAvg, FedProx, and SCAFFOLD. Here, we primarily present FedAvg's results, deferring other results to Appendix. When $q = 0$, FAST becomes the classic FedAvg under various ACP. When $q = 1$, it is the FedAvg under ideal client participation.

**Note.** For simplicity and clarity, we declare the following notations across all tables in this paper: a) Ada.($\lambda$) means adaptive FAST with a fixed $\lambda$. $(def.)$ means the default $\lambda = 1$. b) *Ratio* = $\frac{\text{Rounds with ACP}}{\text{Total rounds}}$, representing the percentage of ACP. $(1 - \textit{Ratio})$ represents the percentage of the snapshot enforcement. c) For $A \pm B$, $A$ is the average of the last 5 test accuracy, and $B$ is the standard deviation.

| Participation | q | Fashion ($\alpha$=0.05) | | CIFAR-10 ($\alpha$=0.1) | | Shakespeare | |
|---|---|---|---|---|---|---|---|
| | | Test Accuracy | Ratio | Test Accuracy | Ratio | Test Accuracy | Ratio |
| Uniform (FedAvg) | 1 | 84.10%±2.4 | 0% | 80.18%±0.6 | 0% | 48.86%±0.3 | 0% |
| Beta (FAST) | Ada.(7) | 80.92%±3.1 | 60.3% | 76.83%±1.0 | 67.5% | 48.80%±0.3 | 54.2% |
| | Ada.(def.) | 77.93%±0.7 | 88.5% | 68.94%±4.0 | 96.6% | 47.51%±0.6 | 93.9% |
| | 0.5 | 80.74%±2.7 | 49.6% | 78.03%±1.3 | 50.7% | 48.63%±0.3 | 49.6% |
| | 0.3 | 75.88%±4.4 | 69.9% | 76.84%±0.6 | 70.1% | 48.31%±0.3 | 70.5% |
| | 0.1 | 74.42%±5.3 | 90.9% | 72.98%±1.4 | 89.9% | 47.45%±0.6 | 90.2% |
| Beta (FedAvg) | 0 | 74.84%±1.2 | 100% | 68.30%±0.9 | 100% | 46.84%±0.4 | 100% |
| Gamma (FAST) | Ada.(7) | 79.95%±4.9 | 59.3% | 76.26%±1.4 | 66.1% | 48.88%±0.3 | 50.8% |
| | Ada.(def.) | 71.48%±4.5 | 91.8% | 73.47%±0.5 | 97.3% | 45.37%±0.5 | 92.7% |
| | 0.5 | 77.39%±2.7 | 50.4% | 77.76%±0.5 | 49.6% | 48.66%±0.3 | 49.7% |
| | 0.3 | 76.87%±2.6 | 68.5% | 75.67%±1.1 | 70.7% | 47.69%±0.8 | 69.8% |
| | 0.1 | 72.23%±3.2 | 89.7% | 74.77%±0.6 | 89.7% | 45.91%±0.7 | 90.3% |
| Gamma (FedAvg) | 0 | 66.65%±4.7 | 100% | 70.90%±0.8 | 100% | 44.46%±1.0 | 100% |
| Weibull (FAST) | Ada.(7) | 77.89%±3.3 | 59.5% | 76.37%±1.3 | 66.6% | 48.38%±0.3 | 47.9% |
| | Ada.(def.) | 77.14%±2.7 | 90.4% | 72.91%±0.4 | 97.4% | 46.36%±0.8 | 89.0% |
| | 0.5 | 79.10%±4.2 | 50.7% | 79.17%±1.0 | 50.2% | 48.55%±0.3 | 50.1% |
| | 0.3 | 77.08%±4.2 | 71.3% | 75.80%±0.6 | 69.9% | 47.95%±0.4 | 70.4% |
| | 0.1 | 75.36%±2.6 | 89.5% | 74.14%±1.1 | 89.8% | 46.63%±1.4 | 90.6% |
| Weibull (FedAvg) | 0 | 73.15%±5.1 | 100% | 71.74%±0.7 | 100% | 45.18%±1.8 | 100% |

Table 3: Performance of FAST+FedAvg under Different Client Participation and Non-IID Degrees

## 5.1 Experiment Results

In this subsection, we provide four key findings to validate our algorithm and support theoretical analysis.

**1. FL Algorithms' Degraded Performance under ACP.** In Sec. 4, we show the non-trivial performance degradation of FedAvg under ACP. Notably, this performance degradation is a universal phenomenon extending beyond FedAvg. This is evident in the FedProx results that reveal a discernible gap between ideal client participation (uniform distribution) and ACP (other three distributions), with this gap significantly impacted by the level of data heterogeneity. In Appendix, we present more similar findings for other FL algorithms.

**2. Improved Performance of FAST under ACP.** In Table 3, we present a comparison between FedAvg and FAST across various client participation and Non-IID scenarios, leading to three key findings: 1) FAST improves performance by increasing the snapshot frequency $q$ across all tasks. We conducted experiments with different fixed values of $q$ and observed that when $q = 0.5$, FAST nearly matches the test accuracy of FedAvg under ideal client participation. In other words, we can enforce uniform client participation in only half of the rounds, enabling ACP in the remaining rounds. 2) Adaptive FAST proves effective, showcasing an increased test accuracy with the least snapshots. For instance, in Fashion-MNIST with $\alpha = 0.05$ and default $\lambda = 1$, FAST requires only $1 - 91.8\% = 8.2\%$ snapshot enforcement while elevating accuracy from $66.65\%$ to $71.48\%$ in the Gamma distribution. If we take a more aggressive $\lambda = 7$, the accuracy can be improved to $79.95\%$. 3) Adaptive FAST achieves a great balance between test accuracy and snapshot frequency. Across all cases in Table 3, the default adaptive strategy (Ada.(def.), with $\lambda = 1$) consistently requires less than $10\%$ snapshots while delivering notable improvements.

**3. Compatible FAST Framework to Integrate with Other FL Algorithms.** We highlight that the client participation mechanism in FAST constitutes a general and compatible framework which can seamlessly integrate with other FL algorithms. To show this, we adopt two additional FL algorithms, FedProx and SCAFFOLD, utilizing the FAST client participation mechanism, referred to as **FAST+**.

Detailed experimental results are provided in Appendix. In general, we observe that, under ACP, the performance of FAST+ significantly surpasses that of FedProx and SCAFFOLD. These results hold for both fixed $q$ and adaptive $q$. In specific cases, adaptive FAST+ achieves higher test accuracy than FAST+ with a fixed $q$ when their individual proportions of ACP are approximately equal. In other words, adaptive FAST+ can attain higher test accuracy with a higher percentage of ACP (bigger *Ratio* or smaller $q$). These observations align with the results in Table 3 for FedAvg, demonstrating that the client participation mechanism in FAST is general and compatible with other FL algorithms.

**4. Ablation Study for Hyper-parameters.** We conducted extensive ablation experiments on FL and hyper-parameters of FAST, including $\alpha$, distributions for modeling client participation, adaptive hyper-parameter $\lambda$, etc. Here, we specifically investigate the impact of $\lambda$ as a key hyper-parameter in adaptive FAST. All other results are offered in Appendix.

In Appendix, we present the test accuracy for Fashion-MNIST with $\lambda$ varying from 1 to 9. Overall, FAST's performance exhibits less sensitivity to the choices of different $\lambda$ under distinct distributions. As increasing $\lambda$, the snapshot frequency rises, resulting in a decreased ratio. This indicates that the $q$ increases with the increase of $\lambda$. However, we observe that the model performance remains stable. Notably, with our default choice of $\lambda = 1$, FAST attains good test accuracy with only a small percentage of snapshots. Across these three distributions, when $\lambda = 1$, we require less than $5\%$ of snapshot enforcement, validating the effectiveness of adaptive FAST.

## Acknowledgments

This work has been partially supported by the GWBC Award at RIT.

## References

[Acar *et al.*, 2021] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[Avdiukhin and Kasiviswanathan, 2021] Dmitrii Avdiukhin and Shiva Kasiviswanathan. Federated learning under arbitrary communication patterns. In *International Conference on Machine Learning*, pages 425–435. PMLR, 2021.

[Bonawitz *et al.*, 2019] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečnỳ, Stefano Mazzocchi, Brendan McMahan, et al. Towards federated learning at scale: System design. *Proceedings of machine learning and systems*, 1:374–388, 2019.

[Caldas *et al.*, 2018] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konečnỳ, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[Chen *et al.*, 2022] Wenlin Chen, Samuel Horváth, and Peter Richtárik. Optimal client sampling for federated learning. *Transactions on Machine Learning Research*, 2022.

[Cho *et al.*, 2022] Yae Jee Cho, Jianyu Wang, and Gauri Joshi. Towards understanding biased client selection in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 10351–10375. PMLR, 2022.

[Cho *et al.*, 2023] Yae Jee Cho, Pranay Sharma, Gauri Joshi, Zheng Xu, Satyen Kale, and Tong Zhang. On the convergence of federated averaging with cyclic client participation. *arXiv preprint arXiv:2302.03109*, 2023.

[Fraboni *et al.*, 2021] Yann Fraboni, Richard Vidal, Laetitia Kameni, and Marco Lorenzi. Clustered sampling: Low-variance and improved representativity for clients selection in federated learning. In *International Conference on Machine Learning*, pages 3407–3416. PMLR, 2021.

[Gorbunov *et al.*, 2021] Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. Local sgd: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pages 3556–3564. PMLR, 2021.

[Grudzień *et al.*, 2023] Michał Grudzień, Grigory Malinovsky, and Peter Richtárik. Can 5th generation local training methods support client sampling? yes! In *International Conference on Artificial Intelligence and Statistics*, pages 1055–1092. PMLR, 2023.

[Gu *et al.*, 2021] Xinran Gu, Kaixuan Huang, Jingzhao Zhang, and Longbo Huang. Fast federated learning in the presence of arbitrary device unavailability. *Advances in Neural Information Processing Systems*, 34:12052–12064, 2021.

[Haddadpour *et al.*, 2019] Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Local sgd with periodic averaging: Tighter analysis and adaptive synchronization. *Advances in Neural Information Processing Systems*, 32, 2019.

[Jhunjhunwala *et al.*, 2022] Divyansh Jhunjhunwala, Pranay Sharma, Aushim Nagarkatti, and Gauri Joshi. Fedvarp: Tackling the variance due to partial client participation in federated learning. In *Uncertainty in Artificial Intelligence*, pages 906–916. PMLR, 2022.

[Kairouz *et al.*, 2021] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

[Karimireddy *et al.*, 2020] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, pages 5132–5143. PMLR, 2020.

[Koloskova *et al.*, 2022] Anastasiia Koloskova, Sebastian U Stich, and Martin Jaggi. Sharper convergence guarantees for asynchronous sgd for distributed and federated learning. *Advances in Neural Information Processing Systems*, 35:17202–17215, 2022.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Lai *et al.*, 2006] Chin-Diew Lai, DN Murthy, and Min Xie. Weibull distributions and their applications. In *Springer Handbooks*, pages 63–78. Springer, 2006.

[Li *et al.*, 2019] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. In *International Conference on Learning Representations*, 2019.

[Lin *et al.*, 2018] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local sgd. *arXiv preprint arXiv:1808.07217*, 2018.

[Liu *et al.*, 2021] Su Liu, Jiong Yu, Xiaoheng Deng, and Shaohua Wan. Fedcpf: An efficient-communication federated learning approach for vehicular edge computing in 6g communication networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):1616–1629, 2021.

[Luo *et al.*, 2022] Bing Luo, Wenli Xiao, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. Tackling system and statistical heterogeneity for federated learning with adaptive client sampling. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, pages 1739–1748. IEEE, 2022.

[Malinovsky *et al.*, 2023] Grigory Malinovsky, Samuel Horváth, Konstantin Burlachenko, and Peter Richtárik. Federated learning with regularized client participation. *arXiv preprint arXiv:2302.03662*, 2023.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[Ruan *et al.*, 2021] Yichen Ruan, Xiaoxi Zhang, Shu-Che Liang, and Carlee Joe-Wong. Towards flexible device participation in federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3403–3411. PMLR, 2021.

[Soltani *et al.*, 2022] Behnaz Soltani, Venus Haghighi, Adnan Mahmood, Quan Z Sheng, and Lina Yao. A survey on participant selection for federated learning in mobile networks. In *Proceedings of the 17th ACM Workshop on Mobility in the Evolving Internet Architecture*, pages 19–24, 2022.

[Wang and Ji, 2022] Shiqiang Wang and Mingyue Ji. A unified analysis of federated learning with arbitrary client participation. *Advances in Neural Information Processing Systems*, 35:19124–19137, 2022.

[Wang and Ji, 2023] Shiqiang Wang and Mingyue Ji. A lightweight method for tackling unknown participation probabilities in federated averaging. *arXiv preprint arXiv:2306.03401*, 2023.

[Wang and Joshi, 2019] Jianyu Wang and Gauri Joshi. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update sgd. *Proceedings of Machine Learning and Systems*, 1:212–229, 2019.

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

[Wang *et al.*, 2020] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[Wang *et al.*, 2021] Jianyu Wang, Zachary Charles, Zheng Xu, Gauri Joshi, H Brendan McMahan, Maruan Al-Shedivat, Galen Andrew, Salman Avestimehr, Katharine Daly, Deepesh Data, et al. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*, 2021.

[Wang *et al.*, 2023] Lin Wang, Yongxin Guo, Tao Lin, and Xiaoying Tang. DELTA: Diverse client sampling for fasting federated learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[Wu *et al.*, 2023] Feijie Wu, Song Guo, Zhihao Qu, Shiqi He, Ziming Liu, and Jing Gao. Anchor sampling for federated learning with partial client participation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 37379–37416. PMLR, 23–29 Jul 2023.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Yan *et al.*, 2024] Yikai Yan, Chaoyue Niu, Yucheng Ding, Zhenzhe Zheng, Shaojie Tang, Qinya Li, Fan Wu, Chengfei Lyu, Yanghe Feng, and Guihai Chen. Federated optimization under intermittent client availability. *INFORMS Journal on Computing*, 36(1):185–202, 2024.

[Yang *et al.*, 2020] Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. In *International Conference on Learning Representations*, 2020.

[Yang *et al.*, 2021] Chengxu Yang, Qipeng Wang, Mengwei Xu, Zhenpeng Chen, Kaigui Bian, Yunxin Liu, and Xuanzhe Liu. Characterizing impacts of heterogeneity in federated learning upon large-scale smartphone data. In *Proceedings of the Web Conference 2021*, pages 935–946, 2021.

[Yang *et al.*, 2022a] Haibo Yang, Peiwen Qiu, Prashant Khanduri, and Jia Liu. With a little help from my friend: Server-aided federated learning with partial client participation. In *Workshop on Federated Learning: Recent Advances and New Challenges (in Conjunction with NeurIPS 2022)*, 2022.

[Yang *et al.*, 2022b] Haibo Yang, Xin Zhang, Prashant Khanduri, and Jia Liu. Anarchic federated learning. In *International Conference on Machine Learning*, pages 25331–25363. PMLR, 2022.

[Yu *et al.*, 2019] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

[Zeng *et al.*, 2023] Dun Zeng, Siqi Liang, Xiangjing Hu, Hui Wang, and Zenglin Xu. Fedlab: A flexible federated learning framework. *Journal of Machine Learning Research*, 24(100):1–7, 2023.

[Zhang *et al.*, 2023] Jianqing Zhang, Yang Hua, Hao Wang, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Fedala: Adaptive local aggregation for personalized federated learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 11237–11244, 2023.

[Zhao *et al.*, 2018] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[Zhou *et al.*, 2022] Xinyu Zhou, Jun Zhao, Huimei Han, and Claude Guet. Joint optimization of energy consumption and completion time in federated learning. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 1005–1017. IEEE, 2022.