A Robust Training Method for Federated Learning with Partial Participation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Client weighting and partial participation are key techniques in federated learning. They reduce communication costs and maintain a balance in the data used for model training. Numerous strategies are well-established within the research community, leading to growing interest in developing a unified theory. In this paper, we explore this issue in detail. We propose a method that accumulates unused gradients from the current iteration locally and, after full aggregation, leverages them for effective training. Our framework supports a wide class of weighting and sampling heuristics. Furthermore, we show the proposed approach to be robust against clients' periodic disconnection. To validate it, we conduct a series of numerical experiments involving the training of convolutional and transformer-based architectures.

1 Introduction

Optimization is a cornerstone of training machine learning and neural network models. In a nutshell, almost every AI-based solution aims to minimize an empirical risk [Shalev-Shwartz et al., 2010], which evaluates how well the data is approximated. This process involves adjusting parameters to reduce the discrepancy between predicted outputs and ground truth labels, thereby improving generalization performance. Formally, the problem can be expressed as

$$\min_{x\in\mathbb{R}^d}\left[\frac{1}{n}\sum_{i=1}^{n}\ell(g(x,a_i),b_i)\right], \tag{1}$$

where x denotes the trainable parameters of the model g, (a_i, b_i) is the i-th sample from the dataset with size n, and ℓ is the loss function. Nowadays, there is a variety of methods developed to efficiently solve (1) [Robbins and Monro, 1951, Nesterov, 1983, Kingma and Ba, 2014, Defazio and Mishchenko, 2023]. The current successes of machine/deep learning owe much to the development of powerful numerical techniques that enable training on a huge amount of samples. Large-scale data processing became possible with the advancement of distributed optimization [Verbraeken et al., 2020]. Instead of solving the problem on a single machine, samples are shared among M nodes/devices/clients/machines connected via a server. Hence, the problem (1) transforms into

$$\min_{x\in\mathbb{R}^d}\left[f(x)=\frac{1}{M}\sum_{m=1}^{M}f_m(x)=\frac{1}{M}\sum_{m=1}^{M}\frac{1}{n_m}\sum_{i_m=1}^{n_m}\ell(g(x,a_{i_m}),b_{i_m})\right], \tag{2}$$

where n_m is the size of the dataset, stored on m-th device.

1.1 Client Weighting

Parallel data processing helps to reduce computational time significantly [Zinkevich et al., 2010, Abadi et al., 2016, Jouppi et al., 2017]. However, contemporary applications present new challenges.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Training samples are often accumulated locally by each specific machine, rather than being collected and distributed manually. This paradigm with data remaining on edge devices is called federated learning [Konečnỳ et al., 2016, McMahan et al., 2017, Bonawitz et al., 2019]. In such a setup, local datasets are typically heterogeneous – they vary in size, distribution, and quality. For instance, one device may hold unique objects that are poorly represented across the rest of the network, but are crucial for capturing more dependencies. This leads to the conclusion that some clients may be more useful than others. Modern approaches usually assign dynamic weights $\{\pi_m\}_{m=1}^M$ and use

$$f(x) = \sum_{m=1}^{M} \pi_m f_m(x), \quad \text{s.t. } \pi_m > 0, \sum_{m=1}^{M} \pi_m = 1 \tag{3}$$

to calculate statistics. If the devices are considered to be equivalent, this corresponds to the case where $\pi_1 = \ldots = \pi_M = 1/M$. As a result, more important nodes contribute more significantly to the global loss. There are many strategies to prioritize the clients known in the literature.

Weighting Based on Data Quality/Quantity. The most straightforward way to cope with data imbalance is to consider a number of local samples. McMahan et al. [2017] suggested setting each coefficient as the constant $\pi_m = n_m/n$. Since then, many modifications of this approach have been proposed, including federated averaging schemes with momentum [Wang et al., 2019, Reddi et al., 2020], variance reduction [Liang et al., 2019, Karimireddy et al., 2020] and proximal updates [Li et al., 2020]. However, this type of weighting ignores heterogeneity in terms of data quality, leading to bias, e.g. if some client holds an enormous amount of objects with the same labels. To support the diversity of training samples, Yurochkin et al. [2019] proposed to match the neurons of client neural networks before averaging. Building on the foundations laid by this work, subsequent works have explored more efficient approaches extensively [Wang et al., 2020a, Zhang et al., 2022, Yang et al., 2023, Wu et al., 2023, Kafshgari et al., 2023].

Learned Weighting Strategies. It is also common to learn weighting strategies instead of using fixed heuristics. Mohri et al. [2019] were among the first to present results in this direction. They proposed solving the saddle-point problem $\min_{x\in\mathbb{R}^d} \max_{\pi\in\triangle_1^M} \sum_{m=1}^M \pi_m f_m(x)$ to give small weights to well-trained devices. The idea of optimizing agnostic empirical loss was then generalized by Li et al. [2019a]. Their `q-FedAvg` can be reduced to agnostic optimization as one of the special cases. However, in practice, it is hard to search for appropriate saddle-points [Daskalakis and Panageas, 2018, Jin et al., 2020], especially in federated learning [Sharma et al., 2023]. As a result, the community has shifted towards softer adaptive approaches based on local losses [Zhang et al., 2020, Gao et al., 2022] and gradients [Wang et al., 2020b, Luo et al., 2024].

Robust Weighting. The idea of assigning weights to the devices found its application in robust optimization, where malicious clients can disrupt the learning process [Baruch et al., 2019, Xie et al., 2020, Fang et al., 2020]. To combat such attacks, advanced schemes usually compute $\{\pi_m\}_{m=1}^M$, as the trust scores of the devices based on their objectives decrease [Xie et al., 2019], local gradients [Cao et al., 2020, Yan et al., 2023], and the number of local samples [Cao and Lai, 2019]. Recently, researchers came up with the idea of using a Bayesian approach [Yang et al., 2024].

1.2 Client Sampling

Another significant issue of federated learning, on par with heterogeneity, is the communication bottleneck [Tang et al., 2020, Shi et al., 2020]. Sharing information between machines is costly and can limit the positive effect of parallelism, which is especially tangible when clients send messages to the server [Kairouz et al., 2021]. This issue is magnified in federated learning, where edge devices may have unstable network connectivity, and transmitting large updates may be prohibitively slow. Many techniques exist to reduce communication [Seide et al., 2014, Alistarh et al., 2017, Stich, 2018]. Partial participation is a special one among them [Li et al., 2019b, Yang et al., 2021]. In each communication round, only a random subset of clients participates in training, while the rest remain inactive. This approach offloads the server by decreasing the number of updates that need to be aggregated. Moreover, it provides significant advantages in edge computing, where communication channels are not equivalent, or some of them may be unavailable. Nowadays, there is a wide range of heuristics, which allows to choose subset of clients efficiently.

Data-Based Sampling Strategies. Methods from this class rely on zero- and first-order information of local functions. `Importance Sampling FedAvg` [Rizk et al., 2021] was one of the first such approaches. The authors suggested evaluating the relevance of a device by how large its gradient is relative to the others. Indeed, a small gradient makes a weak contribution to the step. Consequently, communication with this node can be neglected. Nguyen et al. [2020] proposed an orthogonal approach. Their `FOLB` measures the angle between local and average gradient. If it is negative, then such a device is useless at the current moment. This idea was then developed extensively in [Wu and Wang, 2022, Zhou et al., 2022]. In addition, techniques based on the norms of updates [Chen et al., 2020] and local loss decrease [Cho et al., 2022] were proposed. There are also a number of approaches that dynamically exploit data heterogeneity to maintain balance [Zhang et al., 2023] or support diversity [Chen and Vikalo, 2024].

System-Based Sampling Strategies. Another approach is to use information about the network itself. `FedCS` [Nishio and Yonetani, 2019] categorizes clients into groups based on their computational power. This strategy saves wall-clock time by avoiding frequent selection of weak devices. Another class of techniques optimizes energy consumption [Xu and Wang, 2020]. Most modern system heterogeneity techniques also incorporate local data considerations [Lai et al., 2021, Li et al., 2022]. `F3AST` [Ribero et al., 2022] learns an availability-dependent client selection strategy to minimize the impact of variance on the global model's convergence.

Thus, the community came up with various techniques for weighting and sampling to make partial participation as efficient as possible. The development of each new scheme was challenging in terms of algorithm design and convergence proof. Consequently, a number of papers appeared attempting to propose a theory without utilizing the properties of any particular strategy.

1.3 Unification of Sampling Strategies

Existing papers in this area of research are built around the federated averaging scheme [McMahan et al., 2017]. Li et al. [2019b] proposed an analysis for strongly convex objectives, obtaining a sublinear convergence rate $\mathcal{O}\left(\kappa^2/K\right)$, where κ is the condition number. However, they modeled the partial participation environment via unbiased sampling. Cho et al. [2022] were the first to study the unified case with biased devices selection. They derived $\mathcal{O}\left(\kappa^2/K + \kappa Q\right)$, where Q is a non-vanishing term that becomes zero solely in the absence of sampling bias. Thus, the authors recovered the results of Li et al. [2019b], but failed to extend the theory to weaker assumptions. The first success in this direction was achieved in [Luo et al., 2022]. This work resolved key questions regarding biased sampling in the strongly convex case. However, the non-convex analysis holds greater significance for applications. For this setting, Wang and Ji [2022] obtained $\mathcal{O}\left(\sqrt{L}/\sqrt{K} + \delta\right)$, where L is the smoothness constant and δ is the uniform bound on the difference between local gradients. This result contains the non-vanishing term and does not match the lower bound $\Omega\left(L/K\right)$ [Carmon et al., 2020].

Thus, current works in this field rely on `FedAvg`. As a consequence, their analysis requires boundedness of gradients [Li et al., 2019b, Cho et al., 2022, Luo et al., 2022] or their differences [Wang and Ji, 2022] even in the non-stochastic case. Therefore, there is still no flawless unified theory of partial participation.

1.4 Our Contribution

In contrast to prior works, where partial participation analysis was built upon `FedAvg`, we introduce our own scheme to leverage client sampling. While existing techniques ignore the information from inactive clients, our approach utilizes it for benefits. Namely, devices accumulate gradient surrogates locally, and the server accounts for them after the full aggregation round. The proposed approach allows weighting and sampling clients according to a variety of strategies, including biased ones. The convergence of our scheme can be proven in both strongly convex and non-convex cases without introducing unnatural assumptions. The obtained rates do not contain non-vanishing terms. To validate the theory, we conduct experiments with ResNet-18 and ViT.

2 Setup

We begin presenting our results with assumptions necessary to prove convergence. First of all, the objective is assumed to be smooth. This requirement is well-established in optimization.

Assumption 1. *The function f is L-smooth, i.e. for all $x, y \in \mathbb{R}^d$ it satisfies*

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\|.$$

Neural networks tend to have a complex loss landscape [Cybenko, 1989, Nguyen and Hein, 2018]. Since we are motivated by real-world scenarios, our main goal is to prove convergence in the non-convex case. For completeness, we also derive results under stronger assumptions.

Assumption 2. *The function f is:*

(a) ***non-convex*** *with at least one global minimum:*

$$\text{there exists may be not unique, } x^* \text{ s.t. } f(x^*) = \inf_{x \in \mathbb{R}^d} f(x) > -\infty.$$

(b) ***μ-strongly convex****, i.e. for all $x, y \in \mathbb{R}^d$ it satisfies*

$$f(y) \geqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

Federated learning methods usually require a bound on data heterogeneity to provide convergence guarantees [Khaled et al., 2020, Karimireddy et al., 2020]. In our work, we quantify it via gradients [Tang et al., 2018, Stich, 2020].

Assumption 3. *Each gradient ∇f_m is similar to the full gradient ∇f, i.e. for all $x \in \mathbb{R}^d$ it satisfies*

$$\frac{1}{M}\sum_{m=1}^{M} \|\nabla f_m(x) - \nabla f(x)\|^2 \leqslant \delta_1 \|\nabla f(x)\|^2 + \delta_2.$$

This assumption is not too strict, since we do not require uniform boundedness ($\delta_1 = 0$). The following one is imposed to derive convergence of our algorithm with local stochasticity. If one removes it, our theory still holds.

Assumption 4. *Each worker has access to a stochastic gradient $\nabla f_m(x, \xi_m)$. This is an unbiased random variable with bounded variance, i.e. for all $x \in \mathbb{R}^d$ it satisfies*

$$\mathbb{E}_{\xi_m}\left[\nabla f_m(x, \xi_m)\right] = \nabla f_m(x),$$
$$\mathbb{E}_{\xi_m}\left[\|\nabla f_m(x, \xi_m) - \nabla f_m(x)\|^2\right] \leqslant \sigma^2.$$

This assumption appears in different forms in a number of classic papers [Stich, 2018, Gower et al., 2019, Gorbunov et al., 2020]. Next, we consider that weights $\{\pi_m\}_{m=1}^M$ from (3) lie on the regularized simplex. Namely, $\pi \in \Delta_1^M \cap \left(\bigcap_{m=1}^M \left\{\pi : e_m^\top \pi + \frac{\alpha}{M} \geqslant 0\right\}\right)$, where $1 \leqslant \alpha \leqslant M$ is the regularization parameter and e is the unit basis. This technique is useful for solving a wide range of tasks [Mehta et al., 2024].

3 Algorithms and Analysis

3.1 Motivation

Existing papers on the unification of client sampling consider `FedAvg` without any modifications. Section 1.3 suggests that this approach is not promising due to poor results even under strong assumptions. A potential direction for future research could be to find a more suitable scheme. Below we propose an intuition that helps to address this issue.

To understand biased sampling, Cho et al. [2022] introduced the definition of selection skew and utilized it in the analysis. This is exactly the cause of the non-vanishing term in their rate. Indeed, there is no convergence if, for example, some devices are never selected for communication. However, we propose that the problem could be solved if we could somehow account for the error accumulated due to bias. To develop this idea, we formalize the sampling strategy as follows. First, we assign weights π_m to devices, as described in (3). Next, we define the selection rule of the server as a stochastic operator $\mathcal{R} : \mathbb{R}^M \to \mathbb{R}^M$ that zeros some entries of the input vector while retaining the others. Applying this operator to the introduced vector of weights, it can be seen that the wide variety of strategies described in Section 1.2 fits this formalism. This applies not only to simple cases of selecting clients with the highest weights but also to non-trivial ones, such as zeroing the weights of unavailable nodes.

Viewing partial participation as weight vector sparsification reveals connections to well-studied techniques [Beznosikov et al., 2023]. A state-of-the-art technique to handle it efficiently is error

feedback [Stich and Karimireddy, 2020, Richtárik et al., 2021]. Since sampling rules are represented as compressors, we believe that this idea may be extremely useful in our setting as well. However, we cannot apply the existing framework directly, as it requires all clients to account for the error at each algorithm iteration. Error feedback was designed to compress the information, while our goal is to exclude some clients from an entire epoch.

Thus, we have to address this challenge before proceeding to a unified analysis of partial participation.

3.2 Partial Participation without Unavailable Devices

To develop the idea proposed in Section 3.1, we present the **P**artial **P**articipation with **B**ias **C**orrection framework (PPBC, see Algorithm 1) that supports a wide class of weighting and sampling approaches. Since computing full-batch gradients is often impractical in modern applications, we also account for local stochasticity.

Algorithm 1 PPBC

1: **Input:** Start point $x^{-1,H^{-1}} \in \mathbb{R}^d$, $g^{-1,H^{-1}} \in \mathbb{R}^d$, epochs number K, number of devices M
2: **Parameters:** Stepsize $\gamma > 0$, momentum $0 < \theta < 1$, regularization $1 \leqslant \alpha \leqslant M$
3: **for** epochs $k = 0, \ldots, K-1$ **do**
4: Initialize π^k *// Server weighs clients using any procedure*
5: $\hat{\pi}^k = \widehat{\mathcal{R}}^k(\pi^k)$ *// Server selects clients to communicate through epoch using any rule* $\hat{\mathcal{R}}$
6: $g_m^{k,0} = 0$ *// Each client initializes the gradient surrogate*
7: $x^{k,0} = x^{k-1,H^{k-1}} - g^{k-1,H^{k-1}}$ *// Server initializes the initial point of the epoch*
8: Generate $H^k \sim \text{Geom}(p)$ *// Server generates number of iterations of k-th epoch*
9: **for** iterations $h = 0, \ldots, H^k - 1$ **do**
10: $\widetilde{\pi}^{k,h} = \widetilde{\mathcal{R}}^{k,h}\left(\hat{\pi}^k\right)$ *// Server selects clients to communicate at the current round using rule* $\widetilde{\mathcal{R}}$
11: **for** devices $m = 1 \ldots M$ in parallel **do**
12: $g_m^{k,h+1} = g_m^{k,h} + (1-\theta)\left(\frac{1}{M} - \widetilde{\pi}_m^{k,h}\right)\nabla f_m(x^{k,h}, \xi_m^{k,h})$ *// Update the gradient surrogate*
13: **end for**
14: **for** each device $m : \widetilde{\pi}_m^{k,h} \neq 0$ **do**
15: Send $\nabla f_m(x^{k,h}, \xi_m^{k,h})$ to the server
16: **end for**
17: $x^{k,h+1} = x^{k,h} - \gamma\left[(1-\theta)\sum\limits_{m=1}^{M}\widetilde{\pi}_m^{k,h}\nabla f_m(x^{k,h}, \xi_m^{k,h}) + \theta g^{k-1,H^{k-1}}\right]$ *// Server updates parameters*
18: **end for**
19: **for** devices $m = 1 \ldots M$ in parallel **do**
20: Send g_m^{k,H^k} to the server
21: **end for**
22: $g^{k,H^k} = \sum\limits_{m=1}^{M} g_m^{k,H^k}$ *// Server aggregates gradient surrogates*
23: **end for**

Description of Algorithm 1. In Algorithm 1, the weights $\pi^k = (\pi_1^k, \ldots, \pi_M^k)^\top$ are computed according to any of the mentioned strategies at the beginning of each epoch (Line 4). Next, the rule $\widehat{\mathcal{R}}$ is applied to determine the participating machines (Line 5). Its output $\hat{\pi}^k$ contains zeros at positions corresponding to nodes that are not chosen to communicate with the server. Note that $\widehat{\mathcal{R}}$ is not necessarily constant. There are no theoretical restrictions to change it during the execution. For example, one can vary the number of participating devices. We also allow additional client sampling at each iteration of the epoch by introducing a rule $\widetilde{\mathcal{R}}$ (Line 10). We propose to aggregate local gradient surrogates during the epoch (Line 12). To provide intuition beyond this update, we give a toy example where each π_m is equal to $1/M$. In this way, all inactive devices collect their gradients, while all active ones retain the vector g_m from the previous iteration. In the practical case with various weights, each device accounts for its deviation from the uniform distribution $\pi_u = \{1/M\}_{m=1}^M$. Next, we use the accumulated vectors during the following epoch (Line 17). To handle the magnitude imbalance between the gradient and its surrogate, we employ a smoothing scheme with a small parameter θ.

Analysis of Algorithm 1. We utilize virtual sequences to derive convergence rates of PPBC. The idea is to introduce an additional vector

$$\widetilde{x}^{k,h} = x^{k,h} - \gamma \sum_{m=1}^{M} g_m^{k,h}$$

and use it to prove convergence. Substituting Lines 10, 17 in this definition, we obtain

$$\widetilde{x}^{k,h+1} = \widetilde{x}^{k,h} - \gamma \left[(1-\theta) \frac{1}{M} \sum_{m=1}^{M} \nabla f_m(x^{k,h}, \xi_m^{k,h}) + \theta g^{k-1,H^{k-1}} \right].$$

This is an important technique for our method, since the sequence $\widetilde{x}$ is updated with the average of gradients from all devices, contrary to the original x. However, the virtual update also contains a combination of accumulated gradients from the previous epoch. We emphasize that handling $g^{k-1,H^{k-1}}$ is one of the main theoretical challenges we address. We set the epoch size H^k as a geometrically distributed random variable and provide the following lemma.

Lemma 1. *Suppose Assumptions 3, 4 hold. We consider the epoch size $H^k \sim Geom(p)$ and $1 \leqslant \alpha \leqslant M$. Then for Algorithm 1 it implies*

$$\mathbb{E}_{H^k}\mathbb{E}_{\xi_m^{k,0}} \ldots \mathbb{E}_{\xi_m^{k,H^k-1}} \left\| g^{k,H^k} \right\|^2 \leqslant \frac{24(1-\theta)^2\alpha(\delta_1+1)}{p^2} \mathbb{E}_{H^k} \left\| \nabla f(x^{k,H^k}) \right\|^2 + \frac{48(1-\theta)^2\alpha\delta_2}{p^2} + \frac{24(1-\theta)^2\alpha\sigma^2}{Mp^2}.$$

Assumption 4 is required only to handle local stochasticity. If the devices are able to compute exact gradients, Lemma 1 holds with $\sigma = 0$. For the details, see Appendix D. As a result, we obtain the convergence theorem.

Theorem 1. *Suppose Assumptions 1, 2(a), 3, 4 hold. Then for Algorithm 1 with $\theta \leqslant \frac{\gamma L p^2}{2}$ and $\gamma \leqslant \frac{p}{384L\alpha(\delta_1+1)}$ it implies that*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E} \left\| \nabla f(x^{k,0}) \right\|^2 \leqslant \frac{16\left(f(x^{0,0}) - f(x^*)\right)}{\gamma K} + \frac{768\gamma L\alpha\delta_2}{p} + \frac{384\gamma^2L^2\alpha\delta_2}{p^3} + \frac{400\gamma L\alpha\sigma^2}{Mp} + \frac{192\gamma^2L^2\alpha\sigma^2}{Mp^3}.$$

The main obstacle in proving Theorem 1 is the terms $\|g^{k,H^k}\|^2$ and $\|g^{k-1,H^{k-1}}\|^2$ that appear in the analysis. Using Lemma 1, they can be screwed to $\|\nabla f(x^{k,H^k})\|^2$ and $\|\nabla f(x^{k-1,H^{k-1}})\|^2$, respectively. The first norm is easy to analyze. Classically, it serves as a convergence criterion. Eliminating the second one turns out to be challenging. To cope with it, we incorporate the surrogate into the starting point of the epoch (Line 7). For the details, see Appendix D.1. With such an estimate, there is a technique to choose the stepsize γ appropriately to obtain convergence [Stich, 2019].

Corollary 1. *Under conditions of Theorem 1 Algorithm 1 with fixed rules $\widehat{\mathcal{R}}^k \equiv \widetilde{\mathcal{R}}^{k,h} \equiv \mathcal{R}$ needs*

$$\mathcal{O}\left(M\frac{M}{C}\left(\frac{\Delta L\alpha\delta_1}{\varepsilon^2} + \frac{\Delta L\alpha\delta_2}{\varepsilon^4} + \frac{\Delta L\alpha\sigma^2}{M\varepsilon^4}\right)\right)$$

number of devices communications to reach ε-accuracy, where $\varepsilon^2 = \frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2$, $\Delta = f(x^{0,0}) - f(x^*)$ and C is the number of devices participating in each epoch.

We also consider varying sampling rules $\widehat{\mathcal{R}}^k$ and $\widetilde{\mathcal{R}}^{k,h}$ to study corollaries of Theorem 1, see Appendix D.1 for the details. In our work, the analysis is extended to the strongly convex case.

Theorem 2. *Suppose Assumptions 1, 2(b), 3, 4 hold. Then for Algorithm 1 with $\theta \leqslant \frac{p\gamma\mu}{4}$ and $\gamma \leqslant \frac{p^2}{96L\alpha(\delta_1+1)}$ it implies that*

$$\mathbb{E}\left\|x^{K,0} - x^*\right\|^2 \leqslant \left(1 - \frac{\gamma\mu}{8}\right)^K \left\|x^{0,0} - x^*\right\|^2 + \frac{8\gamma\alpha}{\mu p^3}\left(144\delta_2 + \frac{74\sigma^2}{M}\right).$$

As well as for the non-convex objective, suitable γ can be chosen in Theorem 2.

Corollary 2. *Under conditions of Theorem 2 Algorithm 1 with fixed rules* $\widehat{\mathcal{R}}^{k,h} \equiv \widetilde{\mathcal{R}}^{k,h} \equiv \mathcal{R}$ *needs*

$$\widetilde{\mathcal{O}}\left(M\left(\frac{M}{C}\right)^2\left(\frac{L}{\mu}\alpha\delta_1\log\left(\frac{1}{\varepsilon}\right)+\frac{M}{C}\frac{\alpha\delta_2}{\mu^2\varepsilon}+\frac{\alpha\sigma^2}{\mu^2 C\varepsilon}\right)\right)$$

number of devices communications to reach ε*-accuracy, where* $\varepsilon^2 = \mathbb{E}\left\|x^{K,0}-x^*\right\|^2$ *and* C *is the number of devices participating in each epoch.*

3.3 Partial Participation with Unavailable Devices

The previous section addresses partial participation when all devices are available to communicate with the server. Indeed, in Algorithm 1 each node receives the current parameters at the end of the iteration, but does not send its gradient. This is motivated by the fact that forwarding a message from the client to the server is much more expensive than the other way around [Kairouz et al., 2021]. However, in practice, some devices can become inactive periodically [Li et al., 2019b, Yang et al., 2021]. Namely, these machines not only refrain from transmitting information but also do not perform local computations. In this section, we extend our theory to cover the case where the actual parameters are sent to only a fraction of the clients.

Description of Algorithm 2. In this section we present the part of Algorithm 2 (see Appendix A) that reflects key differences from Algorithm 1. To design it, we refuse using the biased sampling rule $\widetilde{\mathcal{R}}$ during the epoch. Instead, we simulate outage probability of the m-th device as a Bernoulli random variable $\eta_m^{k,h} \sim \text{Be}(q_m)$ [Chung, 2000] (Line 9). To describe client disconnection formally, $\eta_m^{k,h}$ is used to update the gradient surrogates (Line 11) and to perform the step (Line 16). Thus, in practice, it is not necessary for an inactive device to know the actual parameters. We also normalize the computed gradients by factors $\{q_m\}_{m=1}^M$ to balance their magnitudes.

9: Generate $\eta^{k,h}$

11: $g_m^{k,h+1} = g_m^{k,h} + (1-\theta)\frac{\eta_m^{k,h}}{q_m}\left(\frac{1}{M}-\hat{\pi}_m^{k,h}\right)\nabla f_m(x^{k,h},\xi_m^{k,h})$

16: $x^{k,h+1} = x^{k,h} - \gamma\left[(1-\theta)\sum\limits_{m=1}^{M}\frac{\eta_m^{k,h}}{q_m}\hat{\pi}_m^{k,h}\nabla f_m(x^{k,h},\xi_m^{k,h}) + \theta g^{k-1,H^{k-1}}\right]$

Analysis of Algorithm 2. We formulate the results for both non-convex and strongly-convex cases.

Corollary 3. *Suppose Assumptions 1, 2(a), 3, 4 hold. Algorithm 2 with fixed rules* $\widehat{\mathcal{R}}^k \equiv \widetilde{\mathcal{R}}^{k,h} \equiv \mathcal{R}$ *needs*

$$\mathcal{O}\left(M\frac{M}{C}\frac{1}{\min\limits_{1\leqslant m\leqslant M} q_m}\left(\frac{\Delta L\alpha\delta_1}{\varepsilon^2}+\frac{\Delta L\alpha\delta_2}{\varepsilon^4}+\frac{\Delta L\alpha\sigma^2}{\varepsilon^4}\right)\right)$$

number of devices communications to reach ε*-accuracy, where* $\varepsilon^2 = \frac{1}{K}\sum\limits_{k=0}^{K-1}\mathbb{E}\left\|\nabla f(x^{k,0})\right\|^2$, $\Delta = f(x^{0,0}) - f(x^*)$ *and* C *is the number of devices participating in each epoch.*

Corollary 4. *Suppose Assumptions 1, 2(b), 3, 4 hold. Algorithm 2 with fixed rules* $\widehat{\mathcal{R}}^k \equiv \widetilde{\mathcal{R}}^{k,h} \equiv \mathcal{R}$ *needs*

$$\widetilde{\mathcal{O}}\left(M\left(\frac{M}{C}\right)^2\frac{1}{\min\limits_{1\leqslant m\leqslant M} q_m}\left(\frac{L}{\mu}\alpha\delta_1\log\left(\frac{1}{\varepsilon}\right)+\frac{M}{C}\frac{\alpha\delta_2}{\mu^2\varepsilon}+\frac{M}{C}\frac{\alpha\sigma^2}{\mu^2\varepsilon}\right)\right)$$

number of devices communications to reach ε*-accuracy, where* $\varepsilon^2 = \mathbb{E}\left\|x^{K,0}-x^*\right\|^2$ *and* C *is the number of devices participating in each epoch.*

For more details, see Appendix E. Note that $\min_{1\leq m\leq M} q_m$ is a constant lying in the interval $(0,1]$. Thus, the rates of Algorithm 2 do not differ significantly from those for Algorithm 1. The only deterioration occurs in the variance term associated with local stochasticity. Thus, if each device has an access to its exact gradient, there is no asymptotical difference compared to Corollaries 1 and 2.

3.4 Discussion

We analyzed a wide class of sampling and weighting techniques and proposed algorithms for different network scenarios. Their rates asymptotically coincide with the optimal ones for SGD-like approaches

[Stich, 2019]. Due to considering biased strategies, we obtained an additional factor M/C. Again analogizing to compression, this multiplier signifies compression power. It is a well-known fact that there is no theoretical improvement for methods built upon error-feedback [Richtárik et al., 2021, Beznosikov et al., 2023]. However, we recover the convergence of SGD in the case of full participation. Comparing our non-convex rate regarding the main term $\mathcal{O}\left(1/\varepsilon^2\right)$ with prior works, we note that it surpasses that in [Wang and Ji, 2022] $\left(\mathcal{O}\left(1/\varepsilon^4 + \delta_2\right)\right)$ both asymptotically and by the absence of the non-vanishing term. Next, comparing strongly-convex rates $\left(\mathcal{O}\left(\kappa \log 1/\varepsilon\right)\right)$, we are superior to [Cho et al., 2022] $\left(\mathcal{O}\left(\kappa^2/\varepsilon + \kappa\delta_2\right)\right)$ and [Luo et al., 2022] $\left(\mathcal{O}\left(\kappa/\varepsilon\right)\right)$. Moreover, both of these works lack non-convex analysis. We highlight that we soften assumptions from all aforementioned works.

4 Experiments

To validate our theoretical findings, we conduct a systematic empirical study comparing three optimization approaches — FedAvg [Reddi et al., 2020], SCAFFOLD [Karimireddy et al., 2020], and PPBC (Algorithm 1) — each integrated with the same client sampling technique. Crucially, we maintain a fixed strategy across all methods, deliberately decoupling the sampling mechanism from algorithmic innovations to focus specifically on its interaction with different optimization approaches. Our experiments assess their relative performance under identical conditions, including model architectures, benchmark datasets, and hardware configurations. Firstly, we detail the experimental setup, including the neural network architectures, benchmark datasets, and computational hardware configurations employed in our analysis.

Experimental Setup. We evaluate each sampling strategy under three distinct data distribution scenarios: (**distr-1**) homogeneous (*i.i.d.*), (**distr-2**) heterogeneous with different classes on each client, and (**distr-3**) strongly heterogeneous configurations with varying client-specific data quantities and class distributions. Our benchmark experiments employ image classification on CIFAR-10 [Krizhevsky et al., 2009] using a ResNet-18 architecture [Meng et al., 2019], establishing a controlled testbed for comparative analysis of Algorithm 1 across the sampling strategies. Comprehensive details regarding data partitioning, model architecture, and dataset specifications are provided in Appendix B.

Client Selection Rule. Notably, not all strategies included in our comparative analysis inherently incorporate a client selection mechanism. To ensure a fair and consistent evaluation, we uniformly applied the following selection rule across all methods:

$$\widehat{\mathcal{R}}^k = \mathrm{Top}_C\left(\pi^k\right),$$

where Top_C denotes taking $C > 0$ clients with the highest weights π^k. Consequently, the remainder of our experiments will focus exclusively on the formulation and analysis of weight update rules, while treating the client selection process itself as a fixed component of the experimental framework.

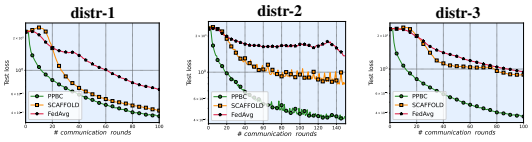
Loss-aware client sampling. Building upon previous work, Cho et al. [2022] introduced the Power-of-Choice (PoC) strategy, which employs a weighted client sampling mechanism based on local loss values. Formally, the weight update rule can be expressed as:

1. The server assigns to all clients the probabilities proportional to the data size fractions

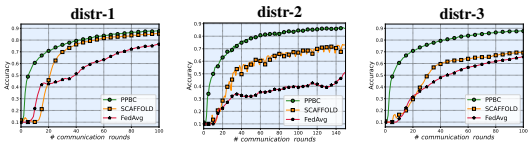
$$p_m = \frac{n_m}{\left(\sum_{m'=1}^{M} n_{m'}\right)}.$$

2. The global model is sent by the server to the selected C clients, which compute and return their local loss values based on their datasets. Subsequently, the weights are updated:

$$\pi^k = \left(\left[\frac{1}{n_m}\sum_{i_m=1}^{n_m} \ell(g(x, a_{i_m}), b_{i_m})\right]\right)_{m=1}^{M}.$$



(a) Convergence comparison.



(b) Metrics comparison.

Figure 1: Performance comparison for PoC strategy with different data distributions.

Trust-Score Sampling. The study by Xie et al. [2019] introduces the BANT, which implements

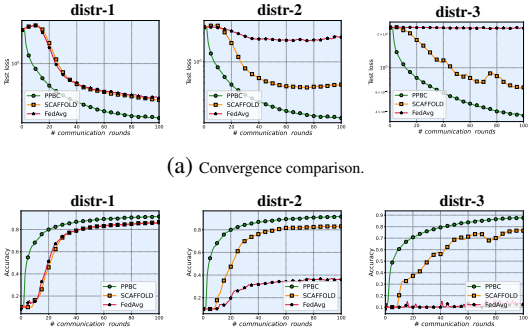
a trust-based sampling mechanism. This approach assigns dynamic trust scores to clients based on historical performance metrics. Thus, weight update rule can be described as:

1. The server assigns trust scores TS_m^k to each client m based on the alignment of their model updates with the performance on server-held ground truth data $\mathcal{V}$:

$$\mathrm{TS}_m^k = \exp\left[-\frac{1}{|\mathcal{V}|}\sum_{\xi\in\mathcal{V}} f_m(x^k,\xi)\right].$$

2. The weights are updated with a probability proportional to the trust scores assigned to each client:

$$\pi^k = \left(\frac{\mathrm{TS}_m^k}{\sum_{m'=1}^{M}\mathrm{TS}_{m'}^k}\right)_{m=1}^{M}.$$



(a) Convergence comparison.

(b) Metrics comparison.

Figure 2: Performance comparison for BANT strategy with different data distributions.

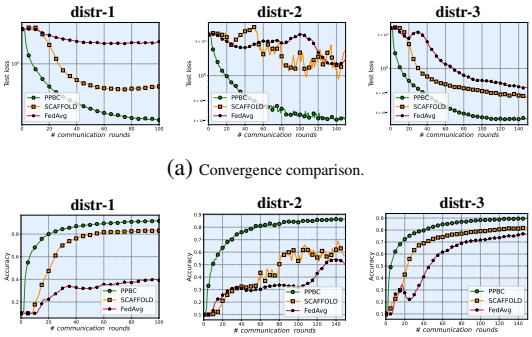
Importance Sampling. Nguyen et al. [2020] introduced FOLB, a theoretically grounded client selection framework for federated learning that optimizes convergence by sampling clients proportionally to the expected utility of their local updates. The core selection mechanism operates as follows:

1. Each client is assigned an importance score IS_m^k proportional to the inner product between its gradient $\nabla f_m(x^k,\xi_m^k)$ and the direction of the server model improvement (previous gradient d^k):

$$\mathrm{IS}_m^k = \left|\left\langle \nabla f_m(x^k,\xi_m^k), d^k\right\rangle\right|.$$

2. The weights are updated with a probability proportional to the trust scores for each client:

$$\pi^k = \left(\frac{\mathrm{IS}_m^k}{\sum_{m'=1}^{M}\mathrm{IS}_{m'}^k}\right)_{m=1}^{M}.$$



(a) Convergence comparison.

(b) Metrics comparison.

Figure 3: Performance comparison for FOLB strategy with different data distributions.

Discussion. Our experimental evaluation on the CIFAR-10 dataset using the ResNet18 architecture demonstrates a substantial performance gap between conventional approaches (FedAvg, SCAFFOLD) and Algorithm 1 (see Figures 1 2, and 3), providing strong empirical validation of our theoretical analysis. Notably, PPBC maintains consistent convergence rates and accuracy across all experimental configurations, with the observed performance variance remaining within 2% of theoretical predictions (see Figure 4). This robust empirical behavior confirms our key theoretical insight: PPBC's performance is strategy-agnostic, achieving stable convergence regardless of the underlying client selection mechanism.

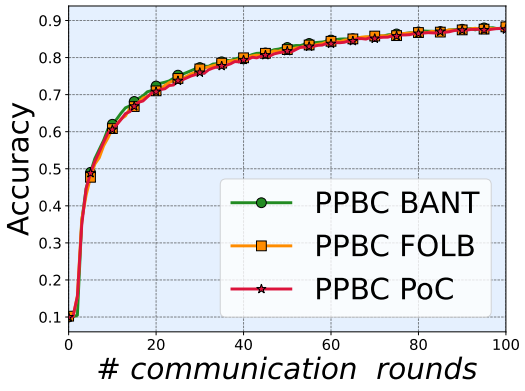


Figure 4: PPBC performace across all client-sampling strategies on the **distr-3**.

We present additional experiments in Appendix B. We consider advanced client selection techniques that utilize the rule $\widetilde{\mathcal{R}}^{k,h}$, provide the results for PPBC+ (Algorithm 2), and demonstrate the outcomes for ViT training.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: a system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

Zeyuan Allen-Zhu. Katyusha x: Practical momentum method for stochastic sum-of-nonconvex optimization. *arXiv preprint arXiv:1802.03866*, 2018.

Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

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.

Xiaoyu Cao, Minghong Fang, Jia Liu, and Neil Zhenqiang Gong. Fltrust: Byzantine-robust federated learning via trust bootstrapping. *arXiv preprint arXiv:2012.13995*, 2020.

Xinyang Cao and Lifeng Lai. Distributed gradient descent algorithm robust to an arbitrary number of byzantine attackers. *IEEE Transactions on Signal Processing*, 67(22):5850–5864, 2019.

Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points i. *Mathematical Programming*, 184(1):71–120, 2020. doi:10.1007/s10107-019-01406-y. URL https://doi.org/10.1007/s10107-019-01406-y.

Huancheng Chen and Haris Vikalo. Heterogeneity-guided client sampling: Towards fast and efficient non-iid federated learning. *Advances in Neural Information Processing Systems*, 37:65525–65561, 2024.

Wenlin Chen, Samuel Horvath, and Peter Richtarik. Optimal client sampling for federated learning. *arXiv preprint arXiv:2010.13723*, 2020.

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.

Kai Lai Chung. *A course in probability theory*. Elsevier, 2000.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Constantinos Daskalakis and Ioannis Panageas. The limit points of (optimistic) gradient descent in min-max optimization. *Advances in neural information processing systems*, 31, 2018.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. In *International Conference on Machine Learning*, pages 7449–7479. PMLR, 2023.

Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Gong. Local model poisoning attacks to {Byzantine-Robust} federated learning. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. Feddc: Federated learning with non-iid data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10112–10121, 2022.

Eduard Gorbunov, Filip Hanzely, and Peter Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690. PMLR, 2020.

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. Sgd: General analysis and improved rates. In *International conference on machine learning*, pages 5200–5209. PMLR, 2019.

Chi Jin, Praneeth Netrapalli, and Michael Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? In *International conference on machine learning*, pages 4880–4889. PMLR, 2020.

Norman P Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the 44th annual international symposium on computer architecture*, pages 1–12, 2017.

Zahra Hafezi Kafshgari, Chamani Shiranthika, Parvaneh Saeedi, and Ivan V Bajić. Quality-adaptive split-federated learning for segmenting medical images with inaccurate annotations. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2023.

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.

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.

Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *International conference on artificial intelligence and statistics*, pages 4519–4529. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

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

Fan Lai, Xiangfeng Zhu, Harsha V Madhyastha, and Mosharaf Chowdhury. Oort: Efficient federated learning via guided participant selection. In *15th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 21)*, pages 19–35, 2021.

Ya Le and Xuan S. Yang. Tiny imagenet visual recognition challenge. 2015. URL https://api.semanticscholar.org/CorpusID:16664790.

Chenning Li, Xiao Zeng, Mi Zhang, and Zhichao Cao. Pyramidfl: A fine-grained client selection framework for efficient federated learning. In *Proceedings of the 28th annual international conference on mobile computing and networking*, pages 158–171, 2022.

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019a.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450, 2020.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019b.

Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 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.

Ping Luo, Xiaoge Deng, Ziqing Wen, Tao Sun, and Dongsheng Li. Accelerating federated learning by selecting beneficial herd of local gradients. *arXiv preprint arXiv:2403.16557*, 2024.

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.

Ronak Mehta, Jelena Diakonikolas, and Zaid Harchaoui. Drago: Primal-dual coupled variance reduction for faster distributionally robust optimization. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Debin Meng, Xiaojiang Peng, Kai Wang, and Yu Qiao. Frame attention networks for facial expression recognition in videos. In *2019 IEEE international conference on image processing (ICIP)*, pages 3866–3870. IEEE, 2019.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *International conference on machine learning*, pages 4615–4625. PMLR, 2019.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate o (1/k2). In *Dokl akad nauk Sssr*, volume 269, page 543, 1983.

Hung T Nguyen, Vikash Sehwag, Seyyedali Hosseinalipour, Christopher G Brinton, Mung Chiang, and H Vincent Poor. Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39(1):201–218, 2020.

Quynh Nguyen and Matthias Hein. Optimization landscape and expressivity of deep cnns. In *International conference on machine learning*, pages 3730–3739. PMLR, 2018.

Takayuki Nishio and Ryo Yonetani. Client selection for federated learning with heterogeneous resources in mobile edge. In *ICC 2019-2019 IEEE international conference on communications (ICC)*, pages 1–7. IEEE, 2019.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*, 2020.

Mónica Ribero, Haris Vikalo, and Gustavo De Veciana. Federated learning under intermittent client availability and time-varying communication constraints. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):98–111, 2022.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. Ef21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34: 4384–4396, 2021.

Elsa Rizk, Stefan Vlaski, and Ali H Sayed. Optimal importance sampling for federated learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3095–3099. IEEE, 2021.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, volume 2014, pages 1058–1062. Singapore, 2014.

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 11:2635–2670, 2010.

Pranay Sharma, Rohan Panda, and Gauri Joshi. Federated minimax optimization with client heterogeneity. *arXiv preprint arXiv:2302.04249*, 2023.

Shaohuai Shi, Zhenheng Tang, Xiaowen Chu, Chengjian Liu, Wei Wang, and Bo Li. A quantitative survey of communication optimizations in distributed deep learning. *IEEE Network*, 35(3):230–237, 2020.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Sebastian U Stich. Unified optimal analysis of the (stochastic) gradient method. *arXiv preprint arXiv:1907.04232*, 2019.

Sebastian U Stich. On communication compression for distributed optimization on heterogeneous data. *arXiv preprint arXiv:2009.02388*, 2020.

Sebastian U Stich and Sai Praneeth Karimireddy. The error-feedback framework: Sgd with delayed gradients. *Journal of Machine Learning Research*, 21(237):1–36, 2020.

Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. *Advances in Neural Information Processing Systems*, 31, 2018.

Zhenheng Tang, Shaohuai Shi, Wei Wang, Bo Li, and Xiaowen Chu. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.

Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *Acm computing surveys (csur)*, 53(2): 1–33, 2020.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020a.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. Slowmo: Improving communication-efficient distributed sgd with slow momentum. *arXiv preprint arXiv:1910.00643*, 2019.

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, 2020b.

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.

Chenrui Wu, Zexi Li, Fangxin Wang, and Chao Wu. Learning cautiously in federated learning with noisy and heterogeneous clients. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 660–665. IEEE, 2023.

Hongda Wu and Ping Wang. Node selection toward faster convergence for federated learning on non-iid data. *IEEE Transactions on Network Science and Engineering*, 9(5):3099–3111, 2022.

Cong Xie, Sanmi Koyejo, and Indranil Gupta. Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance. In *International Conference on Machine Learning*, pages 6893–6901. PMLR, 2019.

Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Uncertainty in Artificial Intelligence*, pages 261–270. PMLR, 2020.

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

Haonan Yan, Wenjing Zhang, Qian Chen, Xiaoguang Li, Wenhai Sun, Hui Li, and Xiaodong Lin. Recess vaccine for federated learning: Proactive defense against model poisoning attacks. *Advances in Neural Information Processing Systems*, 36:8702–8713, 2023.

Haibo Yang, Minghong Fang, and Jia Liu. Achieving linear speedup with partial worker participation in non-iid federated learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:5974–5986, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/... NeurIPS 2021.

Mingkun Yang, Ran Zhu, Qing Wang, and Jie Yang. Fedtrans: Client-transparent utility estimation for robust federated learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhiqin Yang, Yonggang Zhang, Yu Zheng, Xinmei Tian, Hao Peng, Tongliang Liu, and Bo Han. Fedfed: Feature distillation against data heterogeneity in federated learning. *Advances in Neural Information Processing Systems*, 36:60397–60428, 2023.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In *International conference on machine learning*, pages 7252–7261. PMLR, 2019.

Jianyi Zhang, Ang Li, Minxue Tang, Jingwei Sun, Xiang Chen, Fan Zhang, Changyou Chen, Yiran Chen, and Hai Li. Fed-cbs: A heterogeneity-aware client sampling mechanism for federated learning via class-imbalance reduction. In *International Conference on Machine Learning*, pages 41354–41381. PMLR, 2023.

Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10174–10183, 2022.

Michael Zhang, Karan Sapra, Sanja Fidler, Serena Yeung, and Jose M Alvarez. Personalized federated learning with first order model optimization. *arXiv preprint arXiv:2012.08565*, 2020.

Pengyuan Zhou, Hengwei Xu, Lik Hang Lee, Pei Fang, and Pan Hui. Are you left out? an efficient and fair federated learning for personalized profiles on wearable devices of inferior networking conditions. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(2):1–25, 2022.

Martin Zinkevich, Markus Weimer, Lihong Li, and Alex Smola. Parallelized stochastic gradient descent. *Advances in neural information processing systems*, 23, 2010.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see Sections 1.4, 3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: all the assumptions are introduced in Section 2, further limitations are discussed in Section 3.4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: see Sections 2, 3, D, E.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: all necessary descriptions to understand the results of the experiments are provided. See Sections 4, B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: see Section B.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: see Sections 4, B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: since the experiments are more of a theoretical verification, we have no statistical effects associated with running the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: see Section B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: the paper follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: there is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: see Sections 4, B.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Justification: the paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: the core method development in this paper does not involve LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.