High Probability Guarantees For Federated Learning

Sravani Ramishetty and Abolfazl Hashemi

Abstract

Federated learning (FL) has emerged as a promising approach for training machine learning models on distributed data while ensuring privacy preservation and data locality. However, one key challenge in FL optimization is the lack of high-probability guarantees, which can undermine the trustworthiness of FL solutions. To address this critical issue, we introduce Federated Averaging with post-optimization (`FedAvg-PO`) method, a modification to the Federated Averaging (FedAvg) algorithm. The proposed algorithm applies a post-optimization phase to evaluate a short list of solutions generated by several independent runs of the FedAvg method. These modifications allow to significantly improve the large-deviation properties of FedAvg which improve the reliability and robustness of the optimization process. The novel complexity analysis shows that `FedAvg-PO` can compute accurate and statistically guaranteed solutions in the federated learning context. Our result further relaxes the restrictive assumptions in FL theory by developing new technical tools which may be of independent interest. The insights provided by the computational requirements analysis contribute to the understanding of the scalability and efficiency of the algorithm, guiding its practical implementation.

Index Terms

Federated learning, probability guarantees, trustworthiness, reliability, scalability.

I. Introduction

Federated learning (FL) is an innovative edge-computing approach that enables training statistical models directly on remote devices, harnessing the computational resources available on each device while ensuring privacy and data locality. In a typical FL setting, there exist a group of n clients, each equipped with its own local training data, and a central server responsible for coordinating the model training process. The model is parameterized by a vector $w \in \mathbb{R}^d$ and, the data distribution of the i^{th} client is denoted as $\mathcal{D}_i$. Consequently, the i^{th} client possesses its own objective function $f_i(w)$, which quantifies the expected loss incurred by the model when applied to data drawn from $\mathcal{D}_i$. Typically, this objective function is defined using a specific loss function, denoted as ℓ, which measures the discrepancy between the predicted and actual values. The primary goal of the central server in FL is to optimize the average loss, denoted as $f(w)$, across all n clients. This involves minimizing the following objective function:

$$f(w) := \frac{1}{n}\sum_{i=1}^{n} f_i(w) \ \& \ f_i(w) = \mathbb{E}_{x\sim\mathcal{D}_i}[\ell(x,w)]. \tag{1}$$

By collaboratively optimizing this objective function $f(w)$, FL enables the central server to train a model that captures the knowledge and patterns present in the distributed data, without requiring the raw data to be centrally pooled or shared. By distributing the model training process across multiple devices and aggregating their local updates, FL offers significant benefits, such as enhanced privacy, reduced communication overhead, and the ability to leverage diverse data sources. This approach holds tremendous potential for a wide range of applications and paves the way for more privacy-preserving and efficient machine learning in the era of edge computing.

The FL setting where the data distributions of all the clients are identical, i.e. $\mathcal{D}_1 = \cdots = \mathcal{D}_n$, is typically known as the "homogeneous" setting. However, if the data distributions are not identical, it is referred to as the "heterogeneous" setting, which is the focus on the present paper.

The core algorithmic idea of FL, known as FedAvg, was initially introduced by [1]. FedAvg involves a subset of clients performing multiple rounds of local updates on their respective local data using gradient descent. These clients then communicate their updates back to the central server, which aggregates and averages them to update the global model. The goal of FedAvg is to reduce the communication overhead while ensuring that the global model benefits from the collective knowledge of all participating clients.

Since the introduction of FL, extensive research efforts have focused on analyzing the FedAvg algorithm across various settings. Researchers have explored modifications to FedAvg by incorporating concepts from centralized optimization to accelerate training or reduce communication costs. However, analyzing these FL algorithms poses unique technical challenges compared to centralized optimization methods, including noisy and limited information [2]–[4] and privacy issues [5], [6].

One significant technical challenge, in particular, arises from the inclusion of multiple local updates performed by the clients which introduces complexities in analyzing FL algorithms. Understanding the interplay between local updates and global model aggregation is crucial for convergence analysis and performance guarantees. Additionally, heterogeneity in data distribution among clients further complicates FL analysis, impacting convergence speed, model quality, and fairness. Overcoming these challenges requires novel techniques for effective and equitable model training. Moreover, the assumption of all clients receiving the same final FL model can introduce unfairness and hinder the widespread adoption of FL and its trustworthiness. Addressing

this limitation calls for algorithms that improve the large-deviation properties in FL. By enhancing the robustness and reliability of FL models in extreme or rare scenarios, such algorithms can enhance fairness, trustworthiness, and performance.

In centralized optimization, the worst-case stochastic first-order complexity of any algorithm aiming to achieve an ϵ-stationary point (where the expected norm of the gradient is bounded by ϵ) i.e., $\mathbb{E}\|\nabla f(\widetilde{w})\|^2 \leq \epsilon$, for smooth non-convex functions is known to be $\Omega(\epsilon^{-2})$ and can be achieved by vanilla stochastic gradient descent (SGD) [7]. However, vanilla SGD lacks high probability guarantees due to its limited ability to mitigate the high variance of stochastic gradient noise without requiring strong assumptions that are not met in practice [8]. Furthermore, SGD and its variants rely on randomization in their solution output, and their performance is typically analyzed in terms of expected behavior, without providing high probabilistic guarantees. To address the lack of high probability guarantees in non-convex optimization problems, authors [9] proposed 2-RSGD method. This method aims to improve the reliability and robustness of optimization algorithms by incorporating randomness and post optimization techniques that provide high probability guarantees. More recently [10] adopted a proximal-based method in centralized setting which strongly relies on the strongly convex assumption.

In the context of FL where many algorithms are inspired from centralized-optimization, currently there is a lack of methods that provide high probability guarantees. Existing algorithms mainly offer guarantees based on expectation [11]–[14]. To address this limitation, we propose a modification to the conventional FedAvg method by taking inspiration from 2-RSGD method. We refer to this modified approach as Federated Averaging with post optimization (FedAvg-PO).

Instead of simply averaging the iterates to output a solution, our approach adopts a different strategy by randomly selecting a solution $\widetilde{w}$ from the set of K rounds of FedAvg method $(w_1, \ldots, w_K)$. This random selection is based on a probability distribution and the selected $\widetilde{w}$ serves as the output for an independent FedAvg run. Furthermore, we introduce a post-optimization phase that selects a solution from the set of shortlisted solutions $(\widetilde{w}_1, \ldots, \widetilde{w}_S)$ generated over S independent runs of the FedAvg method. This consideration of several independent runs strengthens the robustness and reliability of the overall optimization process. By incorporating these modifications and addressing technical challenges arising from doing so, our approach (`FedAvg-PO`) improves large deviation properties of FedAvg method which in-turn contributes to the improved effectiveness and trustworthiness of the optimization process in non-convex FL settings.

A. Contribution

We propose `FedAvg-PO`, in which we describe a variant of the FedAvg method which can considerably improve the complexity bound of seeking stationary points of (1). The design of `FedAvg-PO` is motivated by lack of high probability guarantee algorithms for FedAvg to alleviate the trustworthiness concerns in FL. Through empirical evaluations and theoretical analysis, we demonstrate the algorithm's capability to compute accurate and statistically guaranteed solutions in the federated learning context. We show that the complexity of the `FedAvg-PO` method for computing an (ε, Λ)-solution of problem i.e., a point $\widetilde{w}^*$ such that $\mathrm{Prob}\{\|\nabla f(\widetilde{w}^*)\|^2 \leq \epsilon\} \geq 1 - \Lambda$ for some $\epsilon > 0$ and $\Lambda \in (0, 1)$ can be bounded by

$$\mathscr{O}\left\{\frac{\sigma^2 r}{\varepsilon}\log(1/\Lambda) + \frac{1}{r\varepsilon^2}\log\frac{1}{\Lambda} + \frac{\log^2(1/\Lambda)\sigma^2}{\Lambda\varepsilon}\right\},$$

where σ^2 is the maximum variance of local (client-level) stochastic gradients, r is the number of clients that the server accesses in each communication round. By considering the unique characteristics of distributed learning scenarios, our algorithm aims to compute an approximate stationary point of the federated learning problem with high probability. Additionally, our analysis does not assume the somewhat unrealistic assumption of *bounded client dissimilarity*, and secondly, it only requires smooth non-convex loss functions, a function class better suited for modern machine learning and deep learning models. This addresses the challenges associated with the trustworthiness and reliability of the FedAvg method and further bridges the gap between the theory and practice of FL. We establish the complexity of this algorithm, and provide insights into its computational requirements, shedding light on the resources needed to achieve desirable solutions in FL. This analysis helps in understanding the scalability and efficiency of the algorithm, guiding its practical implementation.

B. Related Work and Significance

High probability bounds for stochastic optimization. From a theoretical perspective, the existing literature has extensively studied the expected guarantees of stochastic gradient descent (SGD) [12], [15]–[23]. Various adaptive gradient methods have been introduced to achieve better performance guarantees for stochastic optimization tasks in nonconvex settings [24], [25]. These expectation bounds provide valuable insights into the average behavior of SGD across multiple runs. However, relying solely on expectation bounds may not fully capture the behavior of SGD within a single or a few runs, which is particularly important considering the probabilistic nature of SGD and the fact the in practice, the final learned model should reliably deliver a high-performing solution in every single usage.

In practical applications, such as deep learning, it is often the case that model is trained only once due to the significant computational and time costs associated with the training process, but used numerous times. Consequently, obtaining high probability bounds becomes essential to ensure the performance of the algorithm in single runs. High probability bounds can

Algorithm 1 `FedAvg-PO`

Optimization phase:
Input: Initial point w_0, number of runs S, # of communication rounds K, period E, learning rates $\{\eta_k\}_{k=0}^{K-1}$ and global batch size r.
for $s = 0, \ldots, S-1$ **do**
  for $k = 0, \ldots, K-1$ **do**
    Server sends w_k to a set $\mathcal{S}_k$ of r clients chosen uniformly at random w/o replacement.
    for client $i \in \mathcal{S}_k$ **do**
      Set $w_{k,0}^{(i)} = w_k$.
      for $\tau = 0, \ldots, E-1$ **do**
        Pick a random batch of samples in client i, $\mathcal{B}_{k,\tau}^{(i)}$. Compute the stochastic gradient of f_i at $w_{k,\tau}^{(i)}$ over $\mathcal{B}_{k,\tau}^{(i)}$, viz. $\widetilde{\nabla} f_i(w_{k,\tau}^{(i)}; \mathcal{B}_{k,\tau}^{(i)})$.
        Update $w_{k,\tau+1}^{(i)} = w_{k,\tau}^{(i)} - \eta_k \widetilde{\nabla} f_i(w_{k,\tau}^{(i)}; \mathcal{B}_{k,\tau}^{(i)})$.
      end for
      Send $(w_k - w_{k,E}^{(i)})$ to the server.
    end for
    Update $w_{k+1} = w_k - \frac{1}{r} \sum_{i \in \mathcal{S}_k} (w_k - w_{k,E}^{(i)})$.
  end for
end for
Output: $\widetilde{w}_0, \widetilde{w}_1, \ldots, \widetilde{w}_{S-1}$
Post-Optimization phase: Choose a solution $\widetilde{w}^\star$ from the candidate list $\{\widetilde{w}_0, \ldots \widetilde{w}_{S-1}\}$ such that $\|g(\widetilde{w}^\star)\| = \min_{s=1,\ldots S-1} \|g(\widetilde{w}_s)\|$, $g(\widetilde{w}_s) = \frac{1}{T} \sum_{k=1}^{T} \widetilde{\nabla} f(\widetilde{w}_s, \xi_k)$ where $\widetilde{\nabla} f(\widetilde{w}, \xi_k), k = 1, \ldots T$ are the stochastic gradients returned by the stochastic first-order oracle (SFO) where T is the sample size assuming full client participation.

provide a level of confidence that the algorithm's behavior will satisfy certain criteria with a high degree of certainty. While the literature has mainly focused on proving bounds of SGD in expectation, high probability bounds have been primarily explored for convex learning problems, encompassing optimization and generalization performance. In comparison, there is a scarcity of high probability studies in the context of nonconvex learning. However, some related work [9] has provided high probability bounds for SGD, but this work is limited to the centralized case.

FedAvg and related methods Several optimization techniques have been proposed for FL, aiming to improve convergence guarantees and handle data heterogeneity. FedAvg, introduced by [1] is one of the early and influential approaches in this field. It performs model averaging by aggregating client updates but lacks theoretical convergence guarantees and does not account for data heterogeneity. To address these limitations, several methods have been developed. [26] proposed FedProx, which incorporates a proximal term to control client parameter deviation from the global server parameter. This method provides convergence guarantees for heterogeneous data while considering systems heterogeneity. [27] introduced FedPAQ, a variant of FedAvg that employs quantized client-to-server communication and establishes convergence guarantees in the homogeneous case. [28] demonstrated the convergence of FedAvg for strongly convex functions with heterogeneity, assuming bounded client dissimilarity. Other methods focus on specific challenges in federated learning optimization. [29] proposed FedCOMGATE, which incorporates gradient tracking and quantized communication to address data heterogeneity. [11] introduced SCAFFOLD, leveraging control variates to mitigate client-drift caused by heterogeneity. While these algorithms and related methods have made significant contributions to the optimization of FL, their guarantees are primarily based on expected performance in homogeneous or heterogeneous settings. The absence of high probability guarantees in these methods raises concerns regarding the trustworthiness and reliability of FL methods.

In this paper, we propose a novel algorithm that addresses the limitations of existing approaches and aims to compute an (ε, Λ)-solution for the optimization problem defined in (1) under challenging federated learning settings characterized by partial-device participation and data heterogeneity. Unlike previous methods, our approach does not rely on the assumption of bounded client dissimilarity and also bounds the number of calls to the stochastic first-order oracle. By introducing this algorithm, we strive to bridge the gap between expected guarantees and high probability guarantees in the context of FL optimization. Our work contributes to enhancing the trustworthiness and reliability of federated learning algorithms by providing more rigorous guarantees and considerations for real-world scenarios.

II. Federated Averaging with post optimization method

From Proposition 1, it is known that `FedAvg` algorithm requires $K = \mathcal{O}(\frac{1}{r\epsilon^2})$ rounds of communication to achieve an expected gradient norm of $\mathbb{E}[\|\nabla f(w_{k^*})\|^2] \leq \epsilon$. However, this iteration complexity is an upper bound that holds on expectation.

We are interested in establishing its complexity for computing an (ε, Λ)-solution of problem, which is a point $\widetilde{w}^*$ that satisfies $\text{Prob}\{\|\nabla f(\widetilde{w}^*\|^2 \geq \varepsilon\} \leq \Lambda$, where $\varepsilon > 0$ and $\Lambda \in (0,1)$ i.e., finding a first-order stationary point with high probability. A simple approach to establish this notion of complexity is to use $K = \mathcal{O}(\frac{1}{r\epsilon^2})$ bound and Markov's inequality which states that, $\text{Prob}\{X \geq \lambda\} \leq \frac{\mathbb{E}[X]}{\lambda}$ for a positive random variable X and $\lambda > 0$. Applying this, one can simply derive the following bound for FedAvg:

$$\text{Prob}\{\|\nabla f(\widetilde{w}^*\|^2 \geq \frac{\mathscr{B}_N}{\Lambda}\} \leq \Lambda.$$

Here, we set $\varepsilon = \frac{\mathscr{B}_N}{\Lambda}$ which implies that the number of required calls to the oracle to find an (ε, Λ)-solution is $K = \mathcal{O}(\frac{1}{r\Lambda^2\epsilon^2})$. However this complexity bound is rather unsatisfactory in terms of its dependence on Λ.

Instead, we propose a variant of the FedAvg method, which is summarized in `FedAvg-PO` that significantly improves the complexity bound for computing an (ε, Λ)-solution of the problem. `FedAvg-PO` involves a two-phase approach which is motivated by the existing techniques in the centralized setting [9], [10]: an optimization phase and a post-optimization phase. During the optimization phase, we conduct multiple independent runs of the FedAvg method, considering partial client participation and intermittent communication (i.e., local updates). We assume that each client has access to unbiased stochastic gradients of their individual losses, with a maximum variance of σ^2 for their local (client-level) stochastic gradients. The stochastic gradient of the i^{th} client, computed by the model parameterized by a vector $\widetilde{w}$, over a batch of samples $\mathcal{B}$, is denoted as $\widetilde{\nabla} f_i(\widetilde{w}; \mathcal{B})$. In `FedAvg-PO`, we generate a list of candidate solutions $\{\widetilde{w}_0, \dots \widetilde{w}_{S-1}\}$ through these independent runs, with a total of S runs. The optimization phase involves K communication rounds, where each round includes E local updates per round and the server accesses r clients in each round. These independent runs capture different sources of randomness and enable us to improve the complexity bound.

In the post-optimization phase, we select a solution $\widetilde{w}^\star$ from the candidate list $\{\widetilde{w}_0, \dots \widetilde{w}_{S-1}\}$ such that the solution has the lowest gradient norm i.e., $\|g(\widetilde{w}^\star)\| = \min\limits_{s=1,\dots S-1} \|g(\widetilde{w}_s)\|$, $g(\widetilde{w}_s) = \frac{1}{T}\sum_{k=1}^{T} \widetilde{\nabla} f(\widetilde{w}_s, \xi_k)$ where $\widetilde{\nabla} f(\widetilde{w}, \xi_k), k = 1, \dots T$ are the stochastic gradients returned by the stochastic first-order oracle (SFO) where T is the sample size assuming full client participation. Alternatively, we can choose a solution $\widetilde{w}^\star$ from $\{\widetilde{w}_0, \dots \widetilde{w}_{S-1}\}$ such that the solution has the lowest loss i.e., $f(\widetilde{w}^\star) = \min\limits_{s=1,\dots S-1} f(\widetilde{w}_s)$, $f(\widetilde{w}_s) = \frac{1}{T}\sum_{k=1}^{T} F(\widetilde{w}_s, \xi_k)$. Given the stochastic nature of the objective, we sample a few data points such that we are able to produce a high confidence solution which has low gradient norm with high probability instead of finding the total gradient norm. This statement is made precise in our theoretical results.

In the `FedAvg-PO` method (Algorithm 1), the number of calls to the SFO are given by $S \times K$ and $S \times T$, respectively for the optimization phase and post-optimization phase. We will provide below certain bounds on S, K and T, to compute an (ε, Λ)-solution of problem in Theorem 1.

III. Convergence Analysis of FedAvg-PO

We now discuss the assumptions we made in our analysis of the proposed algorithm. They are standard assumptions used in the analysis of Federated Averaging algorithms [11], [13].

Assumption 1 (Smoothness). *The objective function $\ell(x, w)$ is L-smooth with respect to w, for all x. Thus, each $f_i(w)$ ($i \in [n]$) is L-smooth, and so is $f(w)$.*

Assumption 2 (Bounded Variance). *The variance of the stochastic gradient of each client i is bounded,*

$$\mathbb{E}\Big[\Big\|\widetilde{\nabla} f_i(\widetilde{w}_{k,\tau}^{(i)}, \xi_k^i) - \nabla f_i(\widetilde{w}_{k,\tau}^{(i)})\Big\|^2\Big] \leq \sigma^2, \tag{2}$$

where ξ_k^i denotes random batch of samples in client i for $(k)^{th}$ round, $(\tau)^{th}$ local step and $\widetilde{\nabla} f_i(\widetilde{w}_{k,\tau}^{(i)}, \xi_k^i)$ denotes the stochastic gradient. In addition, we also assume the stochastic gradient is unbiased, i.e., $\mathbb{E}[\|\widetilde{\nabla} f_i(\widetilde{w}_{k,\tau}^{(i)}, \xi_k^i))\|^2] = \nabla f_i(\widetilde{w}_{k,\tau}^{(i)})$.

Assumption 3 (Non-negativity). *Each $f_i(w)$ is non-negative and therefore, $f_i^\star \triangleq \min f_i(w) \geq 0$.*

Most loss functions used in practice satisfy this anyways and if not, we can just add a constant offset to achieve non-negativity. Here, we provide a convergence result for `FedAvg-PO` (1) in the absence of the bounded client dissimilarity assumption. Instead, we assume that Assumption 3 holds for `FedAvg-PO` method.

Assumption 4. *Suppose all clients participate, i.e. $r = n$, in the $(k+1)^{st}$ round of `FED-AVG` ((1)). Let $w_{k,\tau}^{(i)}$ be the i^{th} client's local parameter at the $(\tau+1)^{st}$ local step of the $(k+1)^{st}$ round of `FED-AVG`, for $i \in [n]$. Define $\widetilde{e}_{k,\tau}^{(i)} \triangleq \nabla f_i(w_{k,\tau}^{(i)}) - \nabla f_i(\overline{w}_{k,\tau})$, where $\overline{w}_{k,\tau} \triangleq \frac{1}{n}\sum_{i\in[n]} w_{k,\tau}^{(i)}$. Then for some $\alpha \ll n$:*

$$\mathbb{E}\Big[\Big\|\sum_{i\in[n]} \widetilde{e}_{k,\tau}^{(i)}\Big\|^2\Big] \leq \alpha \sum_{i\in[n]} \mathbb{E}\Big[\Big\|\widetilde{e}_{k,\tau}^{(i)}\Big\|^2\Big], \ \forall\ \tau \in [E]. \tag{3}$$

In the worst case, this assumption will always hold with $\alpha = n$.

We now describe the main convergence analysis of `FedAvg-PO` method. Theorem 1.a) shows the convergence rate of the algorithm for a given set of parameters (S,K,T), while Theorem 1.b) establishes the complexity of the `FedAvg-PO` method for computing an (ε, Λ)-solution of problem in (1).

Theorem 1 (Iteration complexity). *Let Assumptions 1,2,3 and 4 hold for the `FedAvg-PO` method applied to problem* (1). *`FedAvg-PO` finds an (ε, Λ)-solution after*

$$\mathscr{O}\left\{\frac{\sigma^2 r}{\varepsilon}\log(1/\Lambda) + \frac{1}{r\varepsilon^2}\log\frac{1}{\Lambda} + \frac{\log^2(1/\Lambda)\sigma^2}{\Lambda\varepsilon}\right\}$$

calls to the SFO.

a) Let $\mathscr{B}_N$ be the bound on the expected gradient norm defined in Proposition 1. We have $\forall\ \lambda > 0$

$$\text{Prob}\left\{\|\nabla f(\widetilde{w}^*)\|^2 \geq 2\left(4\mathscr{B}_N + \frac{3\lambda\sigma^2}{nT}\right)\right\} \leq \frac{S+1}{\lambda} + 2^{-S}. \tag{4}$$

b) Let $\varepsilon > 0$ and $\Lambda \in (0,1)$ be given. If the parameters (S, K, T) are set to

$$S = S(\Lambda) := \lceil log(2/\Lambda)\rceil, \tag{5}$$

$$K = K(\varepsilon) := \Big\lceil max\Big\{\frac{32\sigma^2 r}{\epsilon n}(\frac{8\alpha}{9} + \frac{1}{E}), \Big[\frac{32}{\varepsilon\sqrt{r}}\Big(4Lf(w_0) + \sigma^2(\frac{1}{E} + \frac{n-r}{3(n-1)})\Big)\Big]^2\Big\}\Big\rceil, \tag{6}$$

$$T = T(\varepsilon, \Lambda) := \Big\lceil \frac{24(S+1)\sigma^2}{n\Lambda\varepsilon}\Big\rceil, \tag{7}$$

then the `FedAvg-PO` method can compute (ε, Λ)-solution of problem after taking at most

$$S(\Lambda)[K(\varepsilon) + T(\varepsilon, \Lambda)] \tag{8}$$

calls to the stochastic first-order oracle.

Proof. We want to show that the gradient norm ($\|\nabla f(\widetilde{w}^*)\|^2$) evaluated at output is small with high probability and we only have access to the stochastic gradients and not the true gradient. For part a), observe that by the definition $\widetilde{w}^*$ has the lowest gradient norm, from this we have

$$\begin{aligned}
\|g(\widetilde{w}^*)\|^2 &= \min_{s=1,\ldots,S}\|g(\widetilde{w}_s)\|^2 \\
&= \min_{s=1,\ldots,S}\|\nabla f(\widetilde{w}_s) + g(\widetilde{w}_s) - \nabla f(\widetilde{w}_s)\|^2 \\
&\leq \min_{s=1,\ldots,S}\{2\|\nabla f(\widetilde{w}_s)\|^2 + 2\|g(\widetilde{w}_s) - \nabla f(\widetilde{w}_s)\|^2\} \\
&\leq 2\min_{s=1,\ldots,S}\|\nabla f(\widetilde{w}_s)\|^2 + 2\max_{s=1,\ldots,S}\|g(\widetilde{w}_s) - \nabla f(\widetilde{w}_s)\|^2,
\end{aligned}$$

which implies that

$$\begin{aligned}
\|\nabla f(\widetilde{w}^*)\|^2 &\leq 2\|g(\widetilde{w}^*)\|^2 + 2\|\nabla f(\widetilde{w}^*) - g(\widetilde{w}^*)\|^2 \\
&\leq 4\underbrace{\min_{s=1,\ldots,S}\|\nabla f(\widetilde{w}_s)\|^2}_{\text{Term I}} + 4\underbrace{\max_{s=1,\ldots,S}\|g(\widetilde{w}_s) - \nabla f(\widetilde{w}_s)\|^2}_{\text{Term II}} + 2\underbrace{\|\nabla f(\widetilde{w}^*) - g(\widetilde{w}^*)\|^2}_{\text{Term III}}.
\end{aligned} \tag{9}$$

We now provide certain probabilistic upper bounds to the three terms in the right hand side of the above inequality. Firstly, using the fact that $\widetilde{w}_s, 1 \leq s \leq S$, are independent, and using the bound on the norm of the true gradient of FedAvg in Proposition 1 (20) and Markov's inequality (with $\lambda = 2$), we have

$$\begin{aligned}
\mathbb{E}[\|\nabla f(\widetilde{w}_s)\|^2] &\leq \mathscr{B}_N, \\
\text{Prob}\{\|\nabla f(\widetilde{w}_s)\|^2 \geq 2\mathscr{B}_N\} &\leq \frac{1}{2}.
\end{aligned}$$

For any mutually independent events $A_1, A_2, \ldots, A_n$, we have

$$\begin{aligned}\text{Prob}\left\{\min A_i \geq \beta\right\} &= \text{Prob}\left\{\cap_i A_i \geq \beta\right\} \\ &= \prod_i \text{Prob}\left\{A_i \geq \beta\right\}.\end{aligned}$$

Using the above idea,

$$\text{Prob}\left\{\min_{s=1,\ldots,S} \left\|\nabla f\left(\widetilde{w}_s\right)\right\|^2 \geq 2\mathscr{B}_N\right\} = \prod_{s=1}^{S} \text{Prob}\left\{\left\|\nabla f\left(\widetilde{w}_s\right)\right\|^2 \geq 2\mathscr{B}_N\right\} \leq 2^{-S}. \tag{10}$$

Since the gradient norm of each instance of s is bounded and independent, the probability of minimum gradient norm is product of bounds for each instance. This means, by employing S instances of `FedAvg` , the probability of the gradient norm exceeding $\mathscr{B}_N$ is reduced from $\frac{1}{\lambda}$ to $\frac{1}{\lambda^S}$. This tighter bound on the probability ensures a higher likelihood of the gradient norm being small.

Next, the third term is bounded by using Assumption 2 and Lemma 1, i.e., by bounding the variance of the random variable $\|g(\widetilde{w}_s) - \nabla f(\widetilde{w}_s\|$. For any $s = 1, \ldots, S$ and $\forall \lambda > 0$

$$\text{Prob}\left\{\left\|g\left(\widetilde{w}_s\right) - \nabla f\left(\widetilde{w}_s\right)\right\|^2 \geq \frac{\lambda\sigma^2}{nT}\right\} = \text{Prob}\left\{\left\|\sum_{k=1}^{T} \delta_{s,k}\right\|^2 \geq \lambda T\sigma^2\right\} \leq \frac{1}{\lambda},$$

$$\text{Prob}\left\{\left\|g\left(\widetilde{w}^*\right) - \nabla f\left(\widetilde{w}^*\right)\right\|^2 \geq \frac{\lambda\sigma^2}{nT}\right\} \leq \frac{1}{\lambda}. \tag{11}$$

By using union bound,for any event $A_1, A_2, \ldots, A_n$, we have

$$\text{Prob}(\cup_{i=1}^{n} A_i) \leq \sum_{i=1}^{n} \text{Prob}(A_i),$$

which implies that $\forall \lambda > 0$

$$\text{Prob}\left\{\max_{s=1,\ldots,S} \left\|g\left(\widetilde{w}_s\right) - \nabla f\left(\widetilde{w}_s\right)\right\|^2 \geq \frac{\lambda\sigma^2}{nT}\right\} \leq \sum_{s=1}^{S} \frac{1}{\lambda} \leq \frac{S}{\lambda}. \tag{12}$$

The result then follows by combining (10), (11), (12) with (9).

$$\text{Prob}\left\{\left\|\nabla f\left(\widetilde{w}^*\right)\right\|^2 \geq 2\left(4\mathscr{B}_N + \frac{3\lambda\sigma^2}{nT}\right)\right\} \leq \frac{S+1}{\lambda} + 2^{-S}.$$

We now show that part b) holds. Since the `FedAvg-PO` method needs to call the Fed-Avg method S times with iteration limit $K(\varepsilon)$ in the optimization phase, and estimate the gradients $g\left(\widetilde{w}_s\right), s = 1, \ldots, S$ with sample size $T(\varepsilon)$ in the post-optimization phase, the total number of calls to the stochastic first-order oracle is bounded by $S[K(\varepsilon) + T(\varepsilon)]$. It remains to show that $\widetilde{w}^*$ is an (ε, Λ)-solution of problem. Noting that by the definitions of $\mathscr{B}_N$ and K, respectively, in Proposition 1 and equation (6), we have

$$\begin{aligned}\mathscr{B}_{K(\varepsilon)} =& \frac{1}{\sqrt{rK}}\Big(4Lf(w_0) + \frac{\sigma^2}{E}\Big(1 + \frac{n-r}{3(n-1)}\Big)\Big) + \frac{1}{K}\Big(\frac{8\sigma^2 r}{9} + \frac{\sigma^2}{E}\Big) && (13)\\ \leq& \frac{\varepsilon}{32} + \frac{\varepsilon}{32} = \frac{\varepsilon}{16}. && (14)\end{aligned}$$

Using the above observation, equation (6) and setting $\lambda = [\frac{2(S+1)}{\Lambda}]$ in (4), we have

$$4B_{K(\varepsilon)} + \frac{3\lambda\sigma^2}{T(\varepsilon)} = \frac{\varepsilon}{4} + \frac{\lambda\Lambda\varepsilon}{8(S+1)} = \frac{\varepsilon}{2},$$

which, together with relations (10) and (11), and the selection of λ, then imply that

$$\text{Prob}\left\{\left\|\nabla f\left(\widetilde{w}^*\right)\right\|^2 \geq \varepsilon\right\} \leq \frac{\Lambda}{2} + 2^{-S} \leq \Lambda.$$

In view of (5), (6) and (7), the complexity bound in (8), after disregarding a few constant factors, is equivalent to

$$\mathscr{O}\left\{\frac{\sigma^2 r}{\varepsilon}\log(1/\Lambda)+\frac{1}{r\varepsilon^2}\log\frac{1}{\Lambda}+\frac{\log^2(1/\Lambda)\sigma^2}{\Lambda\varepsilon}\right\}.$$

The proposed `FedAvg-PO` method with the post-optimization step can yield a considerably smaller bound compared to using just the Markov inequality ($K=\mathcal{O}(\frac{1}{r\Lambda^2\epsilon^2})$) by up to a factor of $1/\left[\Lambda^2\log(1/\Lambda)\right]$, when the second terms dominate in both bounds.

■

Lemma 1. *We bound the norm of the random variable $\|g(\widetilde{w}_s)-\nabla f(\widetilde{w}_s)\|$ where $g(\widetilde{w}_s)=\frac{1}{T}\sum_{k=1}^{T}\widetilde{\nabla}f(\widetilde{w}_s,\xi_k)$ is the stochastic gradient returned by the stochastic first-order oracle for a sample size T and $\nabla f(\widetilde{w}_s)$ is the full gradient using client level variance.*

$$\mathbb{E}[\|g(\widetilde{w}_s)-\nabla f(\widetilde{w}_s)\|^2]\leq\frac{\sigma^2}{nT}.$$

Proof.

$$\begin{aligned}\mathbb{E}[\|g(\widetilde{w}_s)-\nabla f(\widetilde{w}_s)\|^2]=&\mathbb{E}\Big[\Big\|\frac{1}{T}\sum_{k=1}^{T}\widetilde{\nabla}f(\widetilde{w}_s,\xi_k^i)-\nabla f(\widetilde{w}_s)\Big\|^2\Big]\\ =&\mathbb{E}\Big[\Big\|\frac{1}{T}\sum_{k=1}^{T}\Big(\frac{1}{n}\sum_{i=1}^{n}\widetilde{\nabla}f_i(\widetilde{w}_s,\xi_k^i)\Big)-\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\widetilde{w}_s)\Big\|^2\Big]\\ =&\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{1}{T}\sum_{k=1}^{T}\widetilde{\nabla}f_i(\widetilde{w}_s,\xi_k^i)-\nabla f_i(\widetilde{w}_s)\Big)\Big\|^2\Big].\end{aligned}\tag{15}$$

Moreover, denoting

$$\delta_{s,k}^i=\widetilde{\nabla}f_i(\widetilde{w}_s,\xi_k^i)-\nabla f_i(\widetilde{w}_s),\tag{16}$$

for $k=1,\ldots,T$, we have

$$\frac{1}{T}\sum_{k=1}^{T}\widetilde{\nabla}f_i(\widetilde{w}_s,\xi_k^i)-\nabla f_i(\widetilde{w}_s)=\frac{1}{T}\sum_{k=1}^{T}\delta_{s,k}^i.\tag{17}$$

This is the sum of martingale-difference sequence because each term has zero mean and bounded variance (since we use stochastic oracle of bounded variance σ^2). Using this observation, and from concentration lemma of the sequence of random variables which is a martingale-difference sequence with bounded second moments (Lemma 2.3.a in [9]), we can bound the sum below using Jensen's inequality.

$$\begin{aligned}\mathbb{E}\Big[\Big\|\delta_{s,k}^i\Big\|^2\Big]\leq&\sigma^2\\ \mathbb{E}\Big[\Big\|\frac{1}{T}\sum_{k=1}^{T}\delta_{s,k}^i\Big\|^2\Big]\leq&\frac{\sigma^2}{T}.\end{aligned}\tag{18}$$

From independence of each client

$$\begin{aligned}\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i=1}^{n}\Big(\frac{1}{T}\sum_{k=1}^{T}\delta_{s,k}^i\Big)\Big\|^2\Big]\leq&\frac{1}{n^2}\sum_{i=1}^{n}\mathbb{E}\Big[\Big\|\Big(\frac{1}{T}\sum_{k=1}^{T}\delta_{s,k}^i\Big)\Big\|^2\Big]\\ \leq&\frac{1}{n^2}\sum_{i=1}^{n}\frac{\sigma^2}{T}=\frac{\sigma^2}{nT}\\ \mathbb{E}[\|g(\widetilde{w}_s)-\nabla f(\widetilde{w}_s)\|^2]\leq&\frac{\sigma^2}{nT}.\end{aligned}\tag{19}$$

■

Proposition 1 (Iteration complexity for smooth non-convex case). *Let Assumptions 1, 2, 3 and 4 hold. Let σ^2 be the maximum variance of the local (client-level) stochastic gradients. In `FED-AVG`, set $\eta_k=\frac{1}{\gamma LE}\sqrt{\frac{r}{K}}$ for all k, where $\gamma>4$.*

Define a distribution $\mathbb{P}$ *for* $k \in \{0, \ldots, K-1\}$ *such that* $\mathbb{P}(k) = \frac{(1+\zeta)^{(K-1-k)}}{\sum_{k=0}^{K-1}(1+\zeta)^k}$ *where* $\zeta := 16\eta^2L^2E^2\Big(\frac{(n-r)}{6r(n-1)} + \frac{\eta LE}{9}\Big)$. *Sample* k^* *from* $\mathbb{P}$. *Then for* $K \geq \max\Big(\frac{256}{9\gamma^6}r^3, \frac{4r}{\gamma^2}\Big)$

$$\begin{aligned}\mathbb{E}[\|\nabla f(w_{k^*})\|^2] \leq & \mathscr{B}_N \\ :=& \frac{4Lf(w_0)}{\sqrt{rK}} + \frac{\sigma^2}{\sqrt{rKE}}\Big(1 + \frac{(n-r)}{3(n-1)}\Big) + \frac{8\sigma^2 r}{9K} + \frac{\sigma^2}{EK}. \end{aligned} \tag{20}$$

So FED-AVG *needs* $K = \mathcal{O}(\frac{1}{r\epsilon^2})$ *rounds of communication to achieve* $\mathbb{E}[\|\nabla f(w_{k^*})\|^2] \leq \epsilon$, *for* $\epsilon < \mathcal{O}\big(\max\big(\frac{1}{r}, \frac{n/\alpha}{r^2}\big)\big)$.

Thus, we recover the same complexity for FED-AVG/Local SGD (which is basically FED-AVG with full-client participation). Note that the iteration complexity of FED-AVG is $\mathcal{O}(\epsilon^{-2})$, even with $r = n$.

Proof. Using lemma 2, for $\eta_k LE \leq \frac{1}{2}$, we can bound the per-round progress as:

$$\begin{aligned}\mathbb{E}[f(w_{k+1})] \leq & \mathbb{E}[f(w_k)] - \frac{\eta_k E}{2}\mathbb{E}[\|\nabla f(w_k)\|^2] + \eta_k^2 LE^2\Big(\frac{(n-r)}{6r(n-1)} + \frac{8\alpha\eta_k LE}{9n}\Big)\Big(\frac{1}{n}\sum_{i\in[n]}\mathbb{E}[\|\nabla f_i(w_k)\|^2]\Big) \\ &+ \frac{\eta_k^2 LE}{2}\Big(\frac{\eta_k LE}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{r} + \frac{(n-r)}{3r(n-1)}\Big)\sigma^2. \end{aligned} \tag{21}$$

Now applying our earlier trick of using the L-smoothness and non-negativity of the f_i's, we get:

$$\begin{aligned}\sum_{i\in[n]}\|\nabla f_i(w_k)\|^2 &\leq \sum_{i\in[n]} 2L(f_i(w_k) - f_i^*) \\ &\leq 2nLf(w_k) - 2L\sum_{i\in[n]} f_i^* \\ &\leq 2nLf(w_k) \end{aligned} \tag{22}$$

Putting this in 21, we get for a constant learning rate of $\eta_k = \eta$:

$$\begin{aligned}\mathbb{E}[f(w_{k+1})] \leq & \Big(1 + \eta^2L^2E^2\Big(\frac{(n-r)}{6r(n-1)} + \frac{8\alpha\eta LE}{9n}\Big)\Big)\mathbb{E}[f(w_k)] - \frac{\eta E}{2}\mathbb{E}[\|\nabla f(w_k)\|^2] \\ &+ \frac{\eta^2 LE}{2}\Big(\frac{\eta LE}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{r} + \frac{(n-r)}{3r(n-1)}\Big)\sigma^2. \end{aligned} \tag{23}$$

For ease of notation, define $\zeta := \eta^2L^2E^2\Big(\frac{(n-r)}{6r(n-1)} + \frac{8\alpha\eta LE}{9n}\Big)$ and $\zeta_2 := \Big(\frac{\eta LE}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{r} + \frac{(n-r)}{3r(n-1)}\Big)$. Then, unfolding the recursion of 23 from $k = 0$ through to $k = K-1$, we get:

$$\mathbb{E}[f(w_K)] \leq (1+\zeta)^K f(w_0) - \frac{\eta E}{2}\sum_{k=0}^{K-1}(1+\zeta)^{(K-1-k)}\mathbb{E}[\|\nabla f(w_k)\|^2] + \frac{\eta^2 LE}{2}\zeta_2\sigma^2\sum_{k=0}^{K-1}(1+\zeta)^{(K-1-k)}. \tag{24}$$

Let us define $p_k := \frac{(1+\zeta)^{(K-1-k)}}{\sum_{k'=0}^{K-1}(1+\zeta)^{(K-1-k')}}$. Then, re-arranging 24 and using the fact that $\mathbb{E}[f(w_K)] \geq 0$, we get:

$$\sum_{k=0}^{K-1} p_k\mathbb{E}[\|\nabla f(w_k)\|^2] \leq \frac{2(1+\zeta)^K f(w_0)}{\eta E\sum_{k'=0}^{K-1}(1+\zeta)^{k'}} + \eta L\zeta_2\sigma^2 \tag{25}$$

$$= \frac{2\zeta f(w_0)}{\eta E(1-(1+\zeta)^{-K})} + \eta LE\Big(\frac{\eta L}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{rE} + \frac{(n-r)}{3Er(n-1)}\Big)\sigma^2, \tag{26}$$

where the last step follows by using the fact that $\sum_{k'=0}^{K-1}(1+\zeta)^{k'} = \frac{(1+\zeta)^K - 1}{\zeta}$ and plugging in the value of ζ_2. Now,

$$(1+\zeta)^{-K} < 1 - \zeta K + \zeta^2\frac{K(K+1)}{2} < 1 - \zeta K + \zeta^2K^2 \implies 1 - (1+\zeta)^{-K} > \zeta K(1 - \zeta K).$$

Plugging this in 26, we have for $\zeta K < 1$:

$$\sum_{k=0}^{K-1} p_k\mathbb{E}[\|\nabla f(w_k)\|^2] \leq \frac{2f(w_0)}{\eta EK(1-\zeta K)} + \eta LE\Big(\frac{\eta L}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{rE} + \frac{(n-r)}{3Er(n-1)}\Big)\sigma^2. \tag{27}$$

In this case, note that the optimal step size will be $\eta = \mathcal{O}(\frac{1}{LE\sqrt{K}})$, even for $r = n$. So let us pick $\eta = \frac{1}{LE}\sqrt{\frac{r}{K}}$. Note that we need to have $\eta LE \leq \frac{1}{2}$; this happens for $K \geq 4r$. Further, let us ensure $\zeta K < \frac{1}{2}$; this happens for $K \geq \frac{64r^3}{9}(\frac{\alpha}{n})^2$. Thus, we should have $K \geq \max\Big(\frac{64r^3}{9}(\frac{\alpha}{n})^2, 4r\Big)$. Putting $\eta = \frac{1}{LE}\sqrt{\frac{r}{K}}$ in 27 and also using $1 - \zeta K \geq \frac{1}{2}$, we get:

$$\sum_{k=0}^{K-1} p_k \mathbb{E}[\|\nabla f(w_k)\|^2] \leq \frac{4Lf(w_0)}{\sqrt{rK}} + \frac{\sigma^2}{\sqrt{rK}E}\Big(1 + \frac{(n-r)}{3(n-1)}\Big) + \frac{8\sigma^2 r}{9K}\Big(\frac{\alpha}{n}\Big) + \frac{\sigma^2}{EK}\Big(\frac{r}{n}\Big). \tag{28}$$

This finishes the proof. ■

Lemma 2. *For $\eta_k LE \leq \frac{1}{2}$, we can bound the per-round progress as:*

$$\begin{aligned}\mathbb{E}[f(w_{k+1})] \leq & \mathbb{E}[f(w_k)] - \frac{\eta_k E}{2}\mathbb{E}[\|\nabla f(w_k)\|^2] + \eta_k^2 LE^2\Big(\frac{(n-r)}{6r(n-1)} + \frac{8\alpha\eta_k LE}{9n}\Big)\Big(\frac{1}{n}\sum_{i\in[n]}\mathbb{E}[\|\nabla f_i(w_k)\|^2]\Big) \\ & + \frac{\eta_k^2 LE}{2}\Big(\frac{\eta_k LE}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{r} + \frac{(n-r)}{3r(n-1)}\Big)\sigma^2. \end{aligned} \tag{29}$$

Proof. Define

$$\widehat{u}_{k,\tau}^{(i)} := \nabla \widetilde{f}_i(w_{k,\tau}^{(i)}; \mathcal{B}_{k,\tau}^{(i)}),\ \widehat{u}_{k,\tau} := \frac{1}{n}\sum_{i\in[n]} \widehat{u}_{k,\tau}^{(i)},$$

$$u_{k,\tau} := \frac{1}{n}\sum_{i\in[n]} \nabla f_i(w_{k,\tau}^{(i)}),$$

$$\overline{w}_{k,\tau} := \frac{1}{n}\sum_{i\in[n]} w_{k,\tau}^{(i)},\ \widetilde{e}_{k,\tau}^{(i)} = \nabla f_i(w_{k,\tau}^{(i)}) - \nabla f_i(\overline{w}_{k,\tau}).$$

Then:

$$w_{k+1} = w_k - \eta_k \sum_{\tau=0}^{E-1}\Big(\frac{1}{r}\sum_{i\in\mathcal{S}_k} \widehat{u}_{k,\tau}^{(i)}\Big). \tag{30}$$

$$\overline{w}_{k+1,\tau} = w_k - \eta_k \sum_{t=0}^{\tau-1} \widehat{u}_{k,t}. \tag{31}$$

$$\mathbb{E}_{\{\mathcal{B}_{k,\tau}^{(i)}\}_{i=1}^n}[\widehat{u}_{k,\tau}] = u_{k,\tau}. \tag{32}$$

$$\mathbb{E}\Big[\Big\|\sum_{t=0}^{\tau-1}\widehat{u}_{k,t}\Big\|^2\Big] \leq \tau\sum_{t=0}^{\tau-1}\mathbb{E}[\|u_{k,t}\|^2] + \frac{\tau\sigma^2}{n}. \tag{33}$$

$$\mathbb{E}\Big[\Big\|\sum_{t=0}^{\tau-1}\widehat{u}_{k,t}^{(i)}\Big\|^2\Big] \leq \tau\sum_{t=0}^{\tau-1}\mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] + \tau\sigma^2. \tag{34}$$

Recall that σ^2 is the maximum variance of the local (client-level) stochastic gradients. In (33), the expectation is w.r.t. $\{\mathcal{B}_{k,t}^{(i)}\}_{i=1,t=0}^{n,\tau-1}$ and it follows due to the independence of the noise in each local update of each learner. Similarly, (34), the expectation is w.r.t. $\{\mathcal{B}_{k,t}^{(i)}\}_{t=0}^{\tau-1}$ and it follows due to the independence of the noise in each local update.

Next, using the L-smoothness of f and (30), we get

$$\mathbb{E}[f(w_{k+1})] \leq \mathbb{E}[f(w_k)] - \mathbb{E}\Big[\Big\langle \nabla f(w_k), \eta_k \sum_{\tau=0}^{E-1} \Big(\frac{1}{r} \sum_{i \in \mathcal{S}_k} \widehat{u}_{k,\tau}^{(i)}\Big)\Big\rangle\Big] + \frac{L}{2}\mathbb{E}\Big[\Big\|\eta_k \sum_{\tau=0}^{E-1} \Big(\frac{1}{r} \sum_{i \in \mathcal{S}_k} \widehat{u}_{k,\tau}^{(i)}\Big)\Big\|^2\Big] \tag{35}$$

$$= \mathbb{E}[f(w_k)] - \mathbb{E}[\langle \nabla f(w_k), \sum_{\tau=0}^{E-1} \eta_k \widehat{u}_{k,\tau}\rangle] + \frac{\eta_k^2 L}{2}\Big\{\frac{n(r-1)}{r(n-1)}\mathbb{E}[\|\sum_{\tau=0}^{E-1} \widehat{u}_{k,\tau}\|^2] + \frac{(n-r)}{r(n-1)}\Big(\frac{1}{n}\sum_{i\in[n]}\mathbb{E}[\|\sum_{\tau=0}^{E-1}\widehat{u}_{k,\tau}^{(i)}\|^2]\Big)\Big\} \tag{36}$$

$$\leq \mathbb{E}[f(w_k)] - \eta_k \mathbb{E}[\langle \nabla f(w_k), \sum_{\tau=0}^{E-1} u_{k,\tau}\rangle] + \frac{\eta_k^2 LE}{2}\Big\{\frac{n(r-1)}{r(n-1)}\Big(\sum_{\tau=0}^{E-1}\mathbb{E}[\|u_{k,\tau}\|^2] + \frac{\sigma^2}{n}\Big)$$
$$+ \frac{(n-r)}{r(n-1)}\Big(\frac{1}{n}\sum_{i\in[n]}\sum_{\tau=0}^{E-1}\mathbb{E}[\|\nabla f_i(w_{k,\tau}^{(i)})\|^2] + \sigma^2\Big)\Big\} \tag{37}$$

Note that (36) follows by taking expectation w.r.t. $\mathcal{S}_k$ in (35), while (37) follows from (33), (34) and (35). For any 2 vectors a and b, we have that $\langle a, b\rangle = \frac{1}{2}(\|a\|^2 + \|b\|^2 - \|a-b\|^2)$. Using this:

$$\langle \nabla f(w_k), \sum_{\tau=0}^{E-1} u_{k,\tau}\rangle = \sum_{\tau=0}^{E-1}\langle \nabla f(w_k), u_{k,\tau}\rangle$$
$$= \frac{1}{2}\sum_{\tau=0}^{E-1}(\|\nabla f(w_k)\|^2 + \|u_{k,\tau}\|^2 - \|\nabla f(w_k) - u_{k,\tau}\|^2). \tag{38}$$

Putting this in (37), we get:

$$\mathbb{E}[f(w_{k+1})] \leq \mathbb{E}[f(w_k)] - \frac{\eta_k E}{2}\mathbb{E}[\|\nabla f(w_k)\|^2] - \frac{\eta_k}{2}\Big(1 - \eta_k LE\frac{n(r-1)}{r(n-1)}\Big)\sum_{\tau=0}^{E-1}\mathbb{E}[\|u_{k,\tau}\|^2]$$
$$+ \frac{\eta_k}{2}\underbrace{\sum_{\tau=0}^{E-1}\mathbb{E}[\|\nabla f(w_k) - u_{k,\tau}\|^2]}_{\text{(A)}} + \frac{\eta_k^2 LE}{2r}\sigma^2 + \frac{(n-r)}{r(n-1)}\frac{\eta_k^2 LE}{2}\underbrace{\Big(\frac{1}{n}\sum_{i\in[n]}\sum_{\tau=0}^{E-1}\mathbb{E}[\|\nabla f_i(w_{k,\tau}^{(i)})\|^2]\Big)}_{\text{(B)}}. \tag{39}$$

We upper bound (A) and (B) using Lemma (3) and Lemma (4), respectively. Plugging in these bounds, we get:

$$\mathbb{E}[f(w_{k+1})] \leq \mathbb{E}[f(w_k)] - \frac{\eta_k E}{2}\mathbb{E}[\|\nabla f(w_k)\|^2] - \frac{\eta_k}{2}\underbrace{\Big(1 - \eta_k LE\frac{n(r-1)}{r(n-1)} - \eta_k^2 L^2 E^2\Big)}_{\text{(C)}}\sum_{\tau=0}^{E-1}\mathbb{E}[\|u_{k,\tau}\|^2]$$
$$+ \eta_k^2 LE^2\Big(\frac{8(n-r)}{6r(n-1)} + \frac{8\alpha\eta_k LE}{9n}\Big)\Big(\frac{1}{n}\sum_{i\in[n]}\mathbb{E}[\|\nabla f_i(w_k)\|^2]\Big)$$
$$+ \frac{\eta_k^2 LE}{2}\Big(\frac{\eta_k LE}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{r} + \frac{(n-r)}{3r(n-1)}\Big)\sigma^2, \tag{40}$$

for $\eta_k LE \leq \frac{1}{2}$. Note that (C) ≥ 0 for $\eta_k LE \leq \frac{1}{2}$. Thus, for $\eta_k LE \leq \frac{1}{2}$, we have:

$$\mathbb{E}[f(w_{k+1})] \leq \mathbb{E}[f(w_k)] - \frac{\eta_k E}{2}\mathbb{E}[\|\nabla f(w_k)\|^2] + \eta_k^2 LE^2\Big(\frac{8(n-r)}{6r(n-1)} + \frac{8\alpha\eta_k LE}{9n}\Big)\Big(\frac{1}{n}\sum_{i\in[n]}\mathbb{E}[\|\nabla f_i(w_k)\|^2]\Big)$$
$$+ \frac{\eta_k^2 LE}{2}\Big(\frac{\eta_k LE}{n}\Big(1 + \frac{8\alpha E}{9}\Big) + \frac{1}{r} + \frac{(n-r)}{3r(n-1)}\Big)\sigma^2. \tag{41}$$

■

Lemma 3. *For $\eta_k LE \leq \frac{1}{2}$:*

$$\sum_{\tau=0}^{E-1}\mathbb{E}[\|\nabla f(w_k) - u_{k,\tau}\|^2] \leq \eta_k^2 L^2 E^2 \sum_{\tau=0}^{E-1}\mathbb{E}[\|u_{k,\tau}\|^2] + \frac{16\alpha\eta_k^2 L^2 E^3}{9n^2}\sum_{i\in[n]}\mathbb{E}[\|\nabla f_i(w_k)\|^2] + \frac{\eta_k^2 L^2 E^2}{n}\Big(1 + \frac{8\alpha E}{9}\Big)\sigma^2.$$

Proof. We have:

$$\begin{aligned}
\mathbb{E}[\|\nabla f(w_k) - u_{k,\tau}\|^2] =& \mathbb{E}[\|\nabla f(w_k) - \nabla f(\overline{w}_{k,\tau}) + \nabla f(\overline{w}_{k,\tau}) - u_{k,\tau}\|^2] \\
\leq& 2\mathbb{E}[\|\nabla f(w_k) - \nabla f(\overline{w}_{k,\tau})\|^2] + 2\mathbb{E}[\|\nabla f(\overline{w}_{k,\tau}) - u_{k,\tau}\|^2] && (42) \\
\leq& 2L^2\mathbb{E}[\|w_k - \overline{w}_{k,\tau}\|^2] + 2\mathbb{E}\Big[\Big\|\frac{1}{n}\sum_{i\in[n]} \underbrace{(\nabla f_i(\overline{w}_{k,\tau}) - \nabla f_i(w_{k,\tau}^{(i)}))}_{=-\widetilde{e}_{k,\tau}^{(i)}}\Big\|^2\Big] && (43) \\
\leq& 2\eta_k^2 L^2 \mathbb{E}\Big[\Big\|\sum_{t=0}^{\tau-1}\widehat{u}_{k,t}\Big\|^2\Big] + \frac{2\alpha}{n^2}\sum_{i\in[n]}\mathbb{E}[\|\widetilde{e}_{k,\tau}^{(i)}\|^2] && (44) \\
\leq& 2\eta_k^2 L^2\Big(\tau\sum_{t=0}^{\tau-1}\mathbb{E}[\|u_{k,t}\|^2] + \frac{\tau\sigma^2}{n}\Big) + \frac{2\alpha L^2}{n^2}\sum_{i\in[n]}\mathbb{E}[\|w_{k,\tau}^{(i)} - \overline{w}_{k,\tau}\|^2]. && (45)
\end{aligned}$$

Equation(43) follows from the L-smoothness of f and the definition of $u_{k,\tau}$. Equation(44) follows from (31) and Assumption 4. Equation(45) follows from (33) and the L-smoothness of f_i.

But:

$$\begin{aligned}
\sum_{i\in[n]}\mathbb{E}[\|w_{k,\tau}^{(i)} - \overline{w}_{k,\tau}\|^2] =& \sum_{i\in[n]}\mathbb{E}[\|(w_{k,0}^{(i)} - \eta_k\sum_{t=0}^{\tau-1}\widehat{u}_{k,t}^{(i)}) - (\overline{w}_{k,0} - \eta_k\sum_{t=0}^{\tau-1}\widehat{u}_{k,t})\|^2] && (46) \\
=& \eta_k^2\sum_{i\in[n]}\mathbb{E}[\|\sum_{t=0}^{\tau-1}\widehat{u}_{k,t} - \sum_{t=0}^{\tau-1}\widehat{u}_{k,t}^{(i)}\|^2] && (47) \\
\leq& \eta_k^2\tau\sum_{i\in[n]}\sum_{t=0}^{\tau-1}\mathbb{E}[\|\widehat{u}_{k,t} - \widehat{u}_{k,t}^{(i)}\|^2] && (48) \\
=& \eta_k^2\tau\sum_{t=0}^{\tau-1}\sum_{i\in[n]}\mathbb{E}[\|\widehat{u}_{k,t}\|^2 + \|\widehat{u}_{k,t}^{(i)}\|^2 - 2\langle\widehat{u}_{k,t}, \widehat{u}_{k,t}^{(i)}\rangle]. && (49)
\end{aligned}$$

Equation (46) follows because $w_{k,0}^{(i)} = w_k \;\forall\; i \in [n]$, due to which $\overline{w}_{k,0} = w_k$. Next, using the fact that $\widehat{u}_{k,\tau} = \frac{1}{n}\sum_{i\in[n]}\widehat{u}_{k,\tau}^{(i)}$, we can simplify (49) to:

$$\begin{aligned}
\sum_{i\in[n]}\mathbb{E}[\|w_{k,\tau}^{(i)} - \overline{w}_{k,\tau}\|^2] \leq& \eta_k^2\tau\sum_{t=0}^{\tau-1}\sum_{i\in[n]}(\mathbb{E}[\|\widehat{u}_{k,\tau}^{(i)}\|^2] - \mathbb{E}[\|\widehat{u}_{k,t}\|^2]) && (50) \\
\leq& \eta_k^2\tau\sum_{t=0}^{\tau-1}\sum_{i\in[n]}\mathbb{E}[\|\widehat{u}_{k,\tau}^{(i)}\|^2] && (51) \\
\leq& \eta_k^2\tau\sum_{t=0}^{\tau-1}\sum_{i\in[n]}(\mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] + \sigma^2). && (52)
\end{aligned}$$

Next, using (1) for $\eta_k LE \leq \frac{1}{2}$ in (52), we get:

$$\sum_{i\in[n]}\mathbb{E}[\|w_{k,\tau}^{(i)} - \overline{w}_{k,\tau}\|^2] \leq \frac{4\eta_k^2\tau^2}{3}\sum_{i\in[n]}(2\mathbb{E}[\|\nabla f_i(w_k)\|^2] + \sigma^2). \tag{53}$$

Plugging (53) back in (45), we get:

$$\begin{aligned}
\mathbb{E}[\|\nabla f(w_k) - u_{k,\tau}\|^2] \leq& 2\eta_k^2 L^2\Big(\tau\sum_{t=0}^{\tau-1}\mathbb{E}[\|u_{k,t}\|^2] + \frac{\tau\sigma^2}{n}\Big) + \frac{8\alpha\eta_k^2L^2\tau^2}{3n^2}\sum_{i\in[n]}(2\mathbb{E}[\|\nabla f_i(w_k)\|^2] + \sigma^2) && (54) \\
=& 2\eta_k^2L^2\tau\sum_{t=0}^{\tau-1}\mathbb{E}[\|u_{k,t}\|^2] + \frac{16\alpha\eta_k^2L^2\tau^2}{3n^2}\sum_{i\in[n]}\mathbb{E}[\|\nabla f_i(w_k)\|^2] + \frac{\eta_k^2L^2\tau\sigma^2}{n}\Big(2 + \frac{8\alpha}{3}\tau\Big). && (55)
\end{aligned}$$

Summing up (55) for $\tau \in \{0, \ldots, E-1\}$, we get:

$$\sum_{\tau=0}^{E-1} \mathbb{E}[\|\nabla f(w_k) - u_{k,\tau}\|^2] \leq \eta_k^2 L^2 E^2 \sum_{\tau=0}^{E-1} \mathbb{E}[\|u_{k,\tau}\|^2] + \frac{16\alpha\eta_k^2 L^2 E^3}{9n^2} \sum_{i\in[n]} \mathbb{E}[\|\nabla f_i(w_k)\|^2] + \frac{\eta_k^2 L^2 E^2}{n}\Big(1 + \frac{8\alpha E}{9}\Big)\sigma^2. \quad (56)$$

■

Lemma 4. *For $\eta_k LE \leq \frac{1}{2}$, we have:*

$$\sum_{t=0}^{\tau-1} \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] \leq \frac{8\tau}{3}\mathbb{E}[\|\nabla f_i(w_k)\|^2] + \frac{1}{3}\sigma^2.$$

Proof.

$$\begin{aligned}
\mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] =& \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)}) - \nabla f_i(w_k) + \nabla f_i(w_k)\|^2] \\
\leq& 2\mathbb{E}[\|\nabla f_i(w_k)\|^2] + 2\mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)}) - \nabla f_i(w_k)\|^2] \quad (57)\\
\leq& 2\mathbb{E}[\|\nabla f_i(w_k)\|^2] + 2L^2\mathbb{E}[\|w_{k,t}^{(i)} - w_k\|^2]. \quad (58)
\end{aligned}$$

But:

$$\mathbb{E}[\|w_k - w_{k,t}^{(i)}\|^2] = \mathbb{E}\Big[\Big\|\eta_k \sum_{t'=0}^{t-1} \nabla \widetilde{f}_i(w_{k,t'}^{(i)}; \mathcal{B}_{k,t'}^{(i)})\Big\|^2\Big] \leq \eta_k^2 t \sum_{t'=0}^{t-1} \mathbb{E}[\|\nabla f_i(w_{k,t'}^{(i)})\|^2] + \eta_k^2 t\sigma^2. \quad (59)$$

Putting this back in (57), we get:

$$\mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] \leq 2\mathbb{E}[\|\nabla f_i(w_k)\|^2] + 2\eta_k^2 L^2 t \sum_{t'=0}^{t-1} \mathbb{E}[\|\nabla f_i(w_{k,t'}^{(i)})\|^2] + 2\eta_k^2 L^2 t\sigma^2. \quad (60)$$

Now summing up (60) for all $t \in \{0, \ldots, \tau-1\}$, we get:

$$\begin{aligned}
\sum_{t=0}^{\tau-1} \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] \leq& 2\tau(\mathbb{E}[\|\nabla f_i(w_k)\|^2]) + 2\eta_k^2 L^2 \sum_{t=0}^{\tau-1} t \sum_{t'=0}^{t-1} \mathbb{E}[\|\nabla f_i(w_{k,t'}^{(i)})\|^2] + 2\eta_k^2 L^2\sigma^2 \sum_{t=0}^{\tau-1} t \\
\leq& 2\tau(\mathbb{E}[\|\nabla f_i(w_k)\|^2) + \eta_k^2 L^2 \tau^2 \sum_{t=0}^{\tau-1} \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] + \eta_k^2 L^2 \tau^2 \sigma^2. \quad (61)
\end{aligned}$$

Let us set $\eta_k LE \leq 1/2$. Then:

$$\sum_{t=0}^{\tau-1} \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] \leq 2\tau(\mathbb{E}[\|\nabla f_i(w_k)\|^2) + \frac{1}{4}\sum_{t=0}^{\tau-1} \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] + \frac{\sigma^2}{4}. \quad (62)$$

Simplifying, we get:

$$\sum_{t=0}^{\tau-1} \mathbb{E}[\|\nabla f_i(w_{k,t}^{(i)})\|^2] \leq \frac{8\tau}{3}\mathbb{E}[\|\nabla f_i(w_k)\|^2] + \frac{\sigma^2}{3}. \quad (63)$$

■

IV. EMPIRICAL EVIDENCE

In this section, we validate and assess the effectiveness of our proposed theory through empirical analysis. For this purpose, we devise two categories of experiments, (i) synthetic experiments and (ii) deep learning experiments on the MNIST dataset.

TABLE I

TRAINING MNIST WITH FEDAVG−PO APPROACH, WE SEE THAT AVERAGE TRAINING LOSS AND STANDARD DEVIATION DECREASES AND TEST ACCURACY INCREASE AS THE INDEPENDENT RUNS INCREASE.

S	Train Loss	Test accuracy
1	0.058 ± 0.0157	98.43 ± 0.49
2	0.051 ± 0.0077	98.44 ± 0.38
3	0.047 ± 0.0063	98.57 ± 0.35
4	0.046 ± 0.0062	98.63 ± 0.34

A. Synthetic experiment

In the synthetic experiment, we train a linear regression model with m = 30000 samples. These samples $\{(x_j, y_j)_{j=1}^m\}$ are generated based on a predefined model, where $y_j = \langle \theta^*, x_j \rangle + c_j$, where $\theta^* \in \mathbb{R}^{60}$, the j^{th} input x_j is generated randomly from a multivariate Gaussian distribution with a mean of zero and a Covariance matrix that ensures the features are independent and have unit variance ($x_j \sim \mathcal{N}(0, I_{60})$). The noise c_j is drawn at random from zero-mean Gaussian distribution with variance 0.05 ($c_j \sim \mathcal{N}(0, 0.05)$). This dataset is generated such that the $(samples \times features)$ matrix has the ℓ_2 norm of its Hessian equal to 1. This design choice facilitates the exploration of the algorithm's behavior and performance in a controlled setting. These samples are then distributed over 100 clients resulting in 300 samples/client. For training the linear regression model, we employ the mean squared error loss function.

To assess the performance and trustworthiness of the algorithm, we conducted the experiment for a total of 50 Monte Carlo (MC) simulations. In each simulation, during the optimization phase, we generate a list of solutions by several independent runs(S). We set $\gamma = 18$ based on the result in Proposition 1,L-smooth constant $L = 1$, local number of iterations, $E = 5$, global communication rounds, $K = 100$, local batch size, $BS = 16$. In each communication round of this independent run, we randomly select 20% of the clients to participate, resulting in $r = 20$. Based on Proposition 1, we calculate the learning rate as $\eta_k = \frac{1}{\gamma LE}\sqrt{\frac{r}{K}}$, which yields $\eta_k = 0.005$. In the post-optimization phase, we selected a solution from the list of these S candidate values. To make this selection, we computed the norm of the gradient for each candidate over a batch of T stochastic gradients and picked the one with the lowest gradient norm. To evaluate the concentration behavior of the true gradient norm, we repeated the above process for each of the 50 MC simulations. After each simulation, we recorded the true gradient norm of the selected solution from the post optimization phase. By considering this true gradient norm over the 50 MC simulations, we generate a probability density plot shown in Fig. 1. Our results demonstrate a notable trend: as we increased the number of independent runs S, the true gradient norm exhibited a clear concentration phenomenon. This observation aligns with the theoretical findings presented in Theorem 1, which states that as the number of independent runs increases, the norm of the gradient converges with a high probability.

B. Deep learning experiment

To evaluate the validity of our proposed theory on real-world dataset, we conducted deep learning experiments using the popular MNIST dataset, for non-IID setting. The MNIST dataset which consists of 60,000 handwritten digital images, are equally distributed over a set of $n = 100$ clients. In each round, 20% of the clients were randomly selected to participate, resulting in $r = 20$. To emulate the non-IID setting, we assigned 1 or 2 labels randomly to each client. For the deep learning model, we employed a convolutional neural network (CNN) architecture. The model architecture consisted of two 5×5 convolutional layers, with 32 and 64 channels respectively. Each convolutional layer was followed by a 2×2 max pooling operation. The output of the convolutional layers was fed into a fully connected layer with 512 units, followed by a ReLU activation function. Finally, a softmax layer was used as the output layer for classification. To mitigate overfitting, we included a dropout layer with a dropout rate of 0.2. During the training process, we used the following parameters : local number of iterations, $E = 5$, global communication rounds, $K = 100$, local batch size, $BS = 16$, and learning rate, $\eta_k = 0.01$.

Following the same experimental setup as the synthetic experiment, we conducted an experiment to analyze the effect of increasing the number of parallel runs S during the optimization phase on the training loss variance. In the post-optimization phase, we selected a solution from the list of S candidates based on the least gradient norm computed over a batch of $T = 16$ stochastic gradients. Table I provides a summary of the results on the training loss during the final round for 30 Monte Carlo (MC) simulations while varying S from 1 to 4. Our results show that as we increase the number of parallel runs S during the optimization phase and set the batch size $T = 16$ during the post-optimization phase, the overall training loss and its variance decreases. Additionally, we observed an increase in test accuracy and a decrease in its variance. These findings suggest that increasing the number of parallel runs during the optimization phase and using a larger batch size during the post-optimization phase can lead to improved stability in the training process, resulting in higher test accuracy and reduce variability in the training loss.

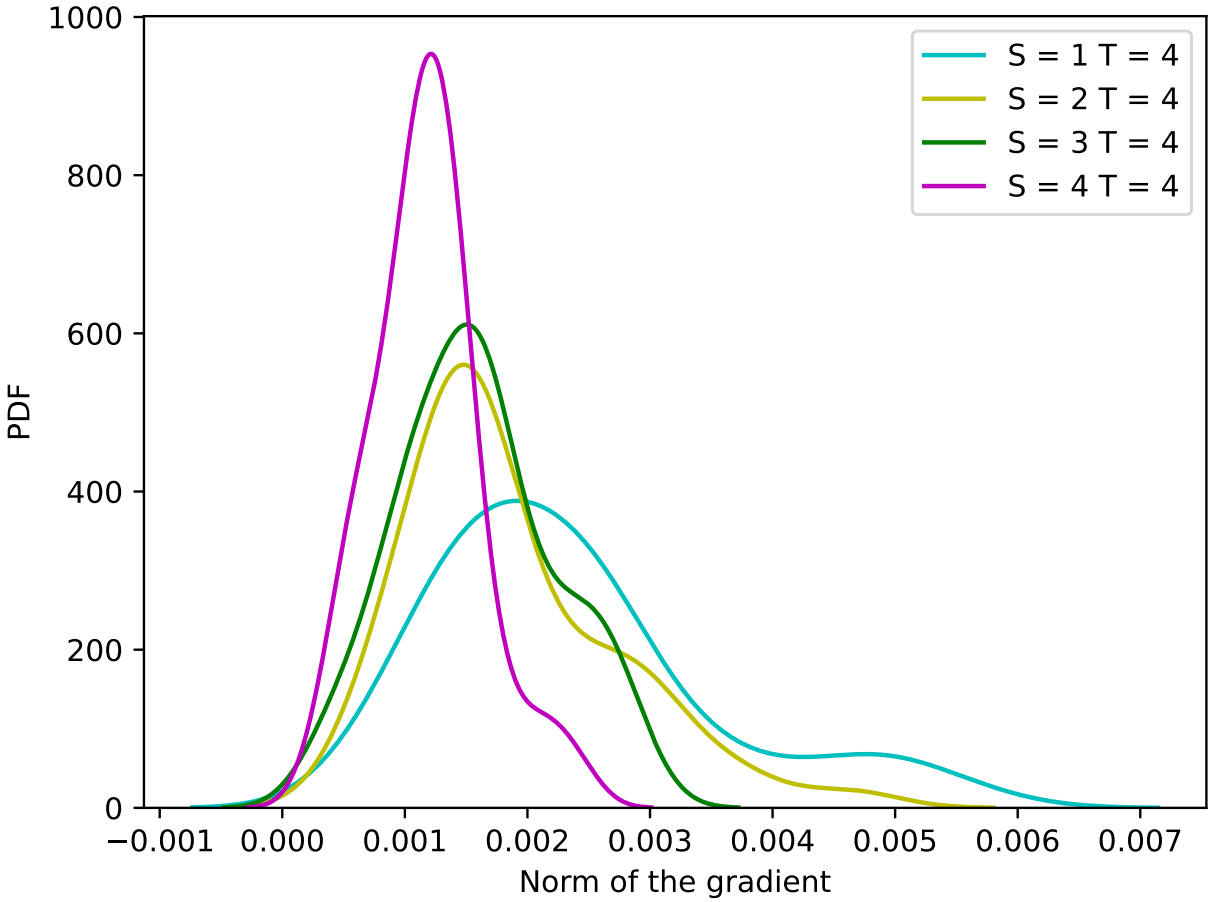

Fig. 1. Training Linear regression on synthetic data with `FedAvg-PO` approach, we see that norm of the true gradient concentrates as the independent runs increases.

V. Conclusion

In this paper, we studied the large deviation properties of the non-convex Federated Learning algorithms and introduce a post-optimization to improve these properties. By incorporating randomness and a post-optimization phase, `FedAvg-PO` enhances the reliability and robustness of the optimization process. The complexity analysis shows that `FedAvg-PO` can compute accurate and statistically guaranteed solutions in the FL context.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR, 2017.

[2] E. Kaya, M. Sahin, and A. Hashemi, "Communication-efficient zeroth-order distributed online optimization: Algorithm, theory, and applications," *IEEE Access*, 2023.

[3] A. Upadhyay and A. Hashemi, "Improved convergence analysis and snr control strategies for federated learning in the presence of noise," *IEEE Access*, 2023.

[4] V. P. Chellapandi, A. Upadhyay, A. Hashemi, and S. H. Zak, "On the convergence of decentralized federated learning under imperfect information sharing," *arXiv preprint arXiv:2303.10695*, 2023.

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

[6] R. Das, A. Hashemi, S. Sanghavi, and I. S. Dhillon, "Privacy-preserving federated learning via normalized (instead of clipped) updates," *arXiv preprint arXiv:2106.07094*, 2021.

[7] Y. Arjevani, Y. Carmon, J. C. Duchi, D. J. Foster, N. Srebro, and B. Woodworth, "Lower bounds for non-convex stochastic optimization," *arXiv preprint arXiv:1912.02365*, 2019.

[8] X. Li and F. Orabona, "A high probability analysis of adaptive sgd with momentum," *arXiv preprint arXiv:2007.14294*, 2020.

[9] S. Ghadimi and G. Lan, "Stochastic first- and zeroth-order methods for nonconvex stochastic programming," 2013.

[10] D. Davis, D. Drusvyatskiy, L. Xiao, and J. Zhang, "From low probability to high confidence in stochastic convex optimization," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 2237–2274, 2021.

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

[12] S. U. Stich, "Local sgd converges fast and communicates little," *arXiv preprint arXiv:1805.09767*, 2018.

[13] R. Das, A. Acharya, A. Hashemi, S. Sanghavi, I. S. Dhillon, and U. Topcu, "Faster non-convex federated learning via global and local momentum," in *Uncertainty in Artificial Intelligence*, pp. 496–506, PMLR, 2022.

[14] A. Hashemi, A. Acharya, R. Das, H. Vikalo, S. Sanghavi, and I. Dhillon, "On the benefits of multiple gossip steps in communication-constrained decentralized federated learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 11, pp. 2727–2739, 2021.

[15] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations," in *Advances in Neural Information Processing Systems*, pp. 14695–14706, 2019.

[16] A. K. R. Bayoumi, K. Mishchenko, and P. Richtárik, "Tighter theory for local sgd on identical and heterogeneous data," in *International Conference on Artificial Intelligence and Statistics*, pp. 4519–4529, 2020.

[17] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized sgd with changing topology and local updates," in *International Conference on Machine Learning*, pp. 5381–5393, PMLR, 2020.

[18] X. Liang, S. Shen, J. Liu, Z. Pan, E. Chen, and Y. Cheng, "Variance reduced local sgd with lower communication complexity," *arXiv preprint arXiv:1912.12844*, 2019.

[19] K. K. Patel and A. Dieuleveut, "Communication trade-offs for synchronized distributed sgd with large step size," *arXiv preprint arXiv:1904.11325*, 2019.

[20] S. U. Stich and S. P. Karimireddy, "The error-feedback framework: Better rates for sgd with delayed gradients and compressed communication," *arXiv preprint arXiv:1909.05350*, 2019.

[21] J. Wang and G. Joshi, "Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms," *arXiv preprint arXiv:1808.07576*, 2018.

[22] B. Woodworth, K. K. Patel, S. U. Stich, Z. Dai, B. Bullins, H. B. McMahan, O. Shamir, and N. Srebro, "Is local sgd better than minibatch sgd?," *arXiv preprint arXiv:2002.07839*, 2020.

[23] M. Zinkevich, M. Weimer, L. Li, and A. Smola, "Parallelized stochastic gradient descent," *Advances in neural information processing systems*, vol. 23, pp. 2595–2603, 2010.

[24] X. Li and F. Orabona, "On the convergence of stochastic gradient descent with adaptive stepsizes," in *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics* (K. Chaudhuri and M. Sugiyama, eds.), vol. 89 of *Proceedings of Machine Learning Research*, pp. 983–992, PMLR, 16–18 Apr 2019.

[25] J. Chen and Q. Gu, "Closing the generalization gap of adaptive gradient methods in training deep neural networks," *CoRR*, vol. abs/1806.06763, 2018.

[26] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[27] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization," in *International Conference on Artificial Intelligence and Statistics*, pp. 2021–2031, 2020.

[28] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[29] F. Haddadpour, M. M. Kamani, A. Mokhtari, and M. Mahdavi, "Federated learning with compression: Unified analysis and sharp guarantees," *arXiv preprint arXiv:2007.01154*, 2020.