# Taming Fat-Tailed ("Heavier-Tailed" with Potentially Infinite Variance) Noise in Federated Learning

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

In recent years, federated learning (FL) has emerged as an important distributed machine learning paradigm to collaboratively learn a global model with multiple clients, while keeping data local and private. However, a key assumption in most existing works on FL algorithms' convergence analysis is that the noise in stochastic first-order information has a finite variance. Although this assumption covers all light-tailed (i.e., sub-exponential) and some heavy-tailed noise distributions (e.g., log-normal, Weibull, and some Pareto distributions), it fails for many fat-tailed noise distributions (i.e., "heavier-tailed" with potentially infinite variance) that have been empirically observed in the FL literature. To date, it remains unclear whether one can design convergent algorithms for FL systems that experience fat-tailed noise. This motivates us to fill this gap in this paper by proposing an algorithmic framework called FAT-Clipping (federated averaging with two-sided learning rates and clipping), which contains two variants: FAT-Clipping per-round (FAT-Clipping-PR) and FAT-Clipping per-iteration (FAT-Clipping-PI). Specifically, for the largest $\alpha \in (1, 2]$ such that the fat-tailed noise in FL still has a bounded $\alpha$-moment, we show that both variants achieve $\mathcal{O}((mT)^{\frac{2-\alpha}{\alpha}})$ and $\mathcal{O}((mT)^{\frac{1-\alpha}{3\alpha-2}})$ convergence rates in the strongly-convex and general non-convex settings, respectively, where $m$ and $T$ are the numbers of clients and communication rounds. Moreover, at the expense of more clipping operations compared to FAT-Clipping-PR, FAT-Clipping-PI further enjoys a linear speedup effect with respect to the number of local updates at each client and being lower-bound-matching (i.e., order-optimal). Collectively, our results advance the understanding of designing efficient algorithms for FL systems that exhibit fat-tailed first-order oracle information.

## 1 Introduction

In recent years, federated learning (FL) has emerged as an important distributed machine learning paradigm, where, coordinated by a server, a set of clients collaboratively learn a global model, while keeping their training data local and private. With intensive research in recent years, researchers have developed many FL algorithms (e.g., FedAvg [1] and many follow-ups [2–12]) that have been theoretically shown to achieve fast convergence rates in the presence of various types of randomness and heterogeneity resulted from training data, network environments, computing resources at clients, etc. Moreover, many of these algorithms enjoy the so-called "linear speedup" effect, i.e., the convergence time to a first-order stationary point is inversely proportional to the number of workers and local update steps.

However, despite the recent advances in FL algorithm design and theoretical understanding, a "cloud that remains obscures the sky of FL" is a common assumption that can be found in almost all works on performance analysis of FL algorithms, which states that the random noise in stochastic first-order

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

oracles (e.g., stochastic gradients or associated estimators) has a *finite variance.* Although this assumption is not restrictive and can cover all light-tailed (i.e., sub-exponential) and some heavy-tailed noise distributions (e.g., log-normal, Weibull, and some Pareto distributions), it fails for many "fat-tailed" distributions (i.e., "heavier-tailed" with potentially infinite variance[1]). In fact, fat-tailed distributions have already been empirically observed under centralized learning settings [15–19], let alone in the more heterogeneous FL environments. Later in Section 3, we will also provide further empirical evidence that shows that fat-tailed noise distributions can be easily induced by FL systems with non-i.i.d. datasets and heterogeneous local updates across clients.

The presence of fat-tailed noise poses two major challenges in FL algorithm design and analysis: i) Experimentally, it has been shown in [20] that many existing FL algorithms suffer severely from fat-tailed noise and frequently exhibit the so-called "catastrophic failure of model performance" (i.e., sudden and dramatic drops of learning accuracy during the training phase); ii) Theoretically, the infinite variance of the random noise in the stochastic first-order oracles renders most of the proof techniques in existing FL algorithmic convergence analysis inapplicable, which necessitates new algorithmic ideas and proof strategies. In light of these empirical and theoretical challenges, two foundational questions naturally emerge in FL algorithm design and analysis: *1) Can we develop FL algorithms with convergence guarantee under fat-tailed noise? 2) If the answer to 1) is "yes," could we characterize their finite-time convergence rates?* In this paper, we provide affirmative answer to the above questions. Our major contributions in this paper are highlighted as follows:

- To address the challenges of the fat-tailed noise in FL algorithm design, we propose an algorithmic framework called FAT-Clipping (federated averaging with two-sided learning rates and clipping), which leverages a clipping technique to mitigate the impact of fat-tailed noise and uses a two-sided learning rate mechanism to lower communication complexity. Our FAT-Clipping framework contains two variants: FAT-Clipping per-round (FAT-Clipping-PR) and FAT-Clipping per-iteration (FAT-Clipping-PI). We show that, for the largest $\alpha \in (1, 2]$ such that the fat-tailed noise in FL still has a bounded $\alpha$-moment, both FAT-Clipping variants achieve $\mathcal{O}((mT)^{\frac{2-\alpha}{\alpha}})$ and $\mathcal{O}((mT)^{\frac{1-\alpha}{3\alpha-2}})$ convergence rates in the strongly-convex and general non-convex settings, respectively, where $m$ and $T$ are the numbers of clients and communication rounds.
- Between the proposed FAT-Clipping variants, FAT-Clipping-PR only performs one clipping operation in each communication round before client communicates to the server, while FAT-Clipping-PI performs clipping in each iteration of local model update. We show that, at the expense of more clipping operations compared to FAT-Clipping-PR, FAT-Clipping-PI further achieves a linear speedup effect with respect to the number local model updates at each client and is convergence rate lower-bound matching.
- In addition to theoretical analysis, we also conduct extensive numerical experiments to study the fat-tailed phenomenon in FL systems and verify the efficacy of our proposed FAT-Clipping algorithms for FL systems with fat-tailed noise. We first provide concrete empirical evidence that fail-tailed noise distributions are not uncommon in FL systems with non-i.i.d. datasets and heterogeneous local updates. We show that our FAT-Clipping algorithms render a much smoother FL training process, which effectively prevents the "catastrophic failure" in various FL settings.

For quick reference and easy comparisons, we summarize all convergence rate results in Table 1. The rest of the paper is organized as follows. In Section 2, we review the literature to put our work in comparative perspectives. In Sec 3, we provide empirical fat-tailed evidence for FL to further motivate this work. Section 4 presents our FAT-Clipping algorithms and their convergence analyses. Section 5 presents numerical results and Section 6 concludes this paper. Due to space limitation, all proof details and some experiments are provided in the supplementary material.

## 2 Related work

**1) Federated Learning:** As mentioned earlier, FL has recently emerged as an important distributed learning paradigm. The first and perhaps the most popular FL method, the federated averaging

[1] In the literature, the terminologies "heavy-tailed" and "fat-tailed" are not universally defined and could be interchangeable sometimes. In this paper, we follow the convention of those authors who reserve the term "fat-tailed" to mean the subclass of heavy-tailed distributions that exhibit power law decay behavior as well as infinite variance (see, e.g., [13, 14]). Thus, every fat-tailed distribution is heavy-tailed, but the reverse is not true.

Table 1: Convergence rate comparisons under fat-tailed noise distributions (shaded parts are our results; metrics: $f(\mathbf{x}) - f(\mathbf{x}^*) \leq \epsilon$ and $\|\nabla f(\mathbf{x})\| \leq \epsilon$ for strongly-convex and non-convex functions, respectively): $\alpha = 2$ and $\alpha \in (1, 2)$ correspond to non-fat-tailed and fat-tailed noises, respectively. Here, $R$ is the total number of iterations for centralized algorithms (SGD and GClip); $K$ and $T$ are local update steps and communication rounds in the FL setting, respectively; $m$ is the number of clients. N/A means no theoretical guarantee for convergence. Note that the total number of iterations $R$ in FL can be computed as $R = KT$, which relates to that in the centralized setting.

| Methods | Strongly Convex Objective Functions | | Nonconvex Objective Functions | |
|---|---|---|---|---|
| | Fat-Tailed | Non-Fat-Tailed | Fat-Tailed | Non-Fat-Tailed |
| SGD [21] | N/A | $\mathcal{O}(R^{-1})$ | N/A | $\mathcal{O}(R^{-\frac{1}{4}})$ |
| GClip [22] | $\mathcal{O}(R^{\frac{2-2\alpha}{\alpha}})$ | $\mathcal{O}(R^{-1})$ | $\mathcal{O}(R^{\frac{1-\alpha}{3\alpha-2}})$ | $\mathcal{O}(R^{-\frac{1}{4}})$ |
| FedAvg [3] | N/A | $\tilde{\mathcal{O}}((mKT)^{-1})$ | N/A | $\mathcal{O}((mKT)^{-\frac{1}{4}})$ |
| **FAT-Clipping-PR** | $\mathcal{O}((mT)^{\frac{2-2\alpha}{\alpha}} K^{\frac{2}{\alpha}})$ | $\tilde{\mathcal{O}}((mKT)^{-1})$ | $\mathcal{O}((mT)^{\frac{1-\alpha}{3\alpha-2}} K^{\frac{2-\alpha}{3\alpha-2}})$ | $\mathcal{O}((mKT)^{-\frac{1}{4}})$ |
| **FAT-Clipping-PI** | $\tilde{\mathcal{O}}((mKT)^{\frac{2-2\alpha}{\alpha}})$ | $\tilde{\mathcal{O}}((mKT)^{-1})$ | $\mathcal{O}((mKT)^{\frac{1-\alpha}{3\alpha-2}})$ | $\mathcal{O}((mKT)^{-\frac{1}{4}})$ |
| **Lower Bound** | $\Omega((mKT)^{\frac{2-2\alpha}{\alpha}})$ | $\Omega((mKT)^{-1})$ | $\Omega((mKT)^{\frac{1-\alpha}{3\alpha-2}})$ | $\Omega((mKT)^{-\frac{1}{4}})$ |

(FedAvg) algorithm [1], was initially proposed as a heuristic to improve communication efficiency and data privacy. Since then, FedAvg has sparked many follow-ups to further address the challenges of data/system heterogeneity and further reduce iteration and communication complexities. Notable approaches include adding regularization for the local loss function [2, 5, 6], using variance reduction techniques [3], taking adaptive learning rate strategy [8] or adaptive communication strategy [23], and many momentum variants [4, 9, 10]. Empirically, these algorithms are shown to be communication-efficient [1] and enjoy better generalization performance [24]. Moreover, many state-of-the-art algorithms enjoy the "linear speed" effect in terms of the numbers of clients and local update steps. We note, however, that all these theoretical results are built upon the finite variance assumption of stochastic gradient noise. Unfortunately, when the stochastic gradient noise is fat-tailed, the finite variance assumption no longer holds, and hence the associated theoretical analysis is also invalid. This motivates us to fill this gap in this paper and conduct the first theoretical analysis for FL systems that experience fat-tailed noise.

**2) Heavy-Tailed Noise in Learning:** Recently, heavy-tailed noise has been empirically observed in modern machine learning systems and theoretically analyzed [15, 16, 18, 22, 25–28]. Heavy-tailed noise significantly affects the learning dynamics and computational complexity, such as the first exit time escaping from saddle point [25] and iteration complexity [22]. This is dramatically different from classic dynamic analysis often based on sub-Gaussian noise assumption [29, 30] and algorithmic convergence analysis with bounded variance assumption [21, 31]. However, for FL, there exist few investigations about heavy-tailed behaviors. In this paper, we first demonstrate through extensive experiments that fat-tailed (i.e., heavier-tailed) noise in FL can be easily induced by data heterogeneity and local update steps. We then propose efficient algorithms to mitigate the impacts of fat-tails.

**3) The Clipping Technique:** Since our FAT-Clipping algorithms are based on the idea of clipping, here we provide an overview on this technique. As far as we know, dating back to at least 1985 [32], gradient clipping has been an effective technique to ensure convergence for optimization problems with fast-growing objective functions. In deep learning, clipping is a widely adopted technique to address the exploding gradient problem. Recently, gradient clipping was theoretically shown to be able to accelerate the training of centralized learning [17, 33–35]. Also, clipping is an effective approach to mitigate heavy-tailed noise [17, 18] in centralized learning. In FL, clipping has been used as the preconditioning step for preserving differential privacy (DP) [36–38]. Unlike these works, in this paper, we utilize clipping to address algorithmic divergence caused by fat-tailed noise in FL.

# 3 Fat-tailed noise phenomenon in federated learning

In this section, we first introduce the basic FL problem statement and the standard FedAvg algorithm for FL. Then, we provide some necessary background of fat-tailed distributions and provide empirical evidence to show that fat-tailed noise can be easily induced by heterogeneity of data and local updates in FL, which further motivates this work. Lastly, we demonstrate the algorithmic divergence and frequently catastrophic model failure under fat-tailed noise.

**Algorithm 1** Generalized FedAvg Algorithm (GFedAvg).

Initialize $\mathbf{x}_1$.
**for** $t = 1, \cdots, T$ (communication round) **do**
  **for** each client $i \in [m]$ in parallel **do**
    Update local model: $\mathbf{x}_{t,i}^1 = \mathbf{x}_t$.
    **for** $k = 1, \cdots, K$ (local update step) **do**
      Compute an unbiased estimate $\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ of $\nabla f_i(\mathbf{x}_{t,i}^k)$.
      Local update: $\mathbf{x}_{t,i}^{k+1} = \mathbf{x}_{t,i}^k - \eta_L \nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$.
    **end for**
    Send $\Delta_t^i = \sum_{k\in[K]} \nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ to the server.
  **end for**
  Global Aggregation At Server:
    Receive $\Delta_t^i, i \in [m]$.
    Server Update: $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta\eta_L}{m} \sum_{i\in[m]} \Delta_t^i$.
    Broadcasting $\mathbf{x}_{t+1}$ to clients.
**end for**

**1) Problem Statement of Federated Learning and the FedAvg Algorithm:** The goal of FL is to solve the following optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) := \frac{1}{m}\sum_{i=1}^{m} f_i(\mathbf{x}), \tag{1}$$

where $m$ is the number of clients and $f_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i\sim D_i}[f_i(\mathbf{x}, \xi_i)]$ is the local loss function associated with a local data distribution $D_i$. A key challenge in FL stems from data heterogeneity, i.e., $D_i \neq D_j, \forall i \neq j$. In FL, the standard and perhaps the most popular algorithm is the federated averaging (FedAvg) method. Here in Algorithm 1, we illustrate a more generalized version of the original FedAvg (GFedAvg) with separate learning rates on client and server sides [3, 7, 8] . Note that when $\eta = 1$, GFedAvg reduces to the original FedAvg [1]. In each communication round of GFedAvg, each client performs local update steps and returns the update difference $\Delta_t^i$. The server then aggregates these results and update the global model [2] and the updated model parameters will then be retrieved by the clients to start the next round of local updates.

**2) Empirical Evidence of Fat-Tailed Noise Phenomenon in Federated Learning:** With the basics of FL and the FedAvg algorithm, we are now in a position to demonstrate the empirical evidence of the existence of fat-tailed noise in FL systems. As mentioned earlier, in most performance analyses of FL algorithms, a common assumption is the bounded variance assumption of the local stochastic gradients: $\mathbb{E}[\|\nabla f_i(\mathbf{x}, \xi) - \nabla f_i(\mathbf{x})\|^2] \leq \sigma^2$. This assumption holds for all light-tailed noise distributions (i.e., the sub-exponential family) and some heavy-tailed distributions (e.g., log-normal, Weibull, and some Pareto distributions).

However, the finite-variance assumption fails to hold for many fat-tailed noise distributions. For instance, for a random variable $X$, if its density $p(x)$ has a power-law tail decreasing as $1/|x|^{\alpha+1}$ with $\alpha \in (0, 2)$, then only the $\alpha$-moment of this noise exists with $\alpha < 2$. To more precisely characterize fat-tailed distributions, in this paper, we adopt the notion of tail-index $\alpha$ [15] to parameterize fat-tailed and heavy-tailed distributions. More specifically, if the density of a random variable $X$'s distribution decays with a power law tail as $1/|x|^{\alpha+1}$ where $\alpha \in (0, 2]$, then $\alpha$ is called the *tail-infex*. This $\alpha$-parameter determines the behavior of the distribution: the smaller the $\alpha$-value, the heavier the tail of the distribution. Also, the $\alpha$-parameter also determines the moments: $\mathbb{E}[|X|^r] < \infty$ if and only if $r < \alpha$, which implies that $X$ has infinite variance when $\alpha < 2$, i.e., being *fat-tailed*.

Next, we investigate the tail property of model updates returned by clients in the GFedAvg algorithm. Due to multiple local steps in the GFedAvg algorithm, we view the whole update vector $\Delta_t^i$ returned by each client, which we called "pseudo-gradient," as a random vector and then analyze its statistical properties. Note that in the special case with the number of local update $K = 1$, $\Delta_t^i$ coincides with a single stochastic gradient of a random sample, (i.e., $\Delta_t^i = \nabla f_i(\mathbf{x}_t, \xi_t)$).

[2] We assume all clients participate in the training at each communication round, but the results can be extended to that with (uniformly random sampled) subset of clients in each communication round [3, 7].

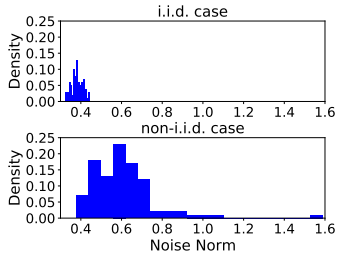


Figure 1: Distributions of the norms of the pseudo-gradient noises computed with CNN on CIFAR-10 dataset in i.i.d. case (top) and non-i.i.d. case (bottom). $m = 100$ clients participate in the training.

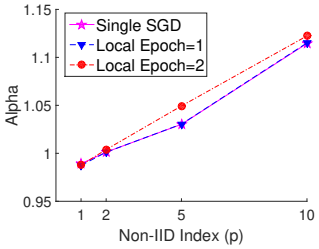


Figure 2: Estimation of $\alpha$ for CIFAR-10 dataset. The non-IID index $p$ represents the data heterogeneity level, and $p = 10$ is the IID case. The smaller the $p$, the more heterogeneous the data across clients.

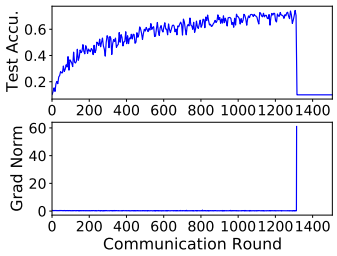


Figure 3: Catastrophic training failures happen when applying GFedAvg on CIFAR-10 dataset, where the test accuracy experiences a sudden and dramatic drop and the pseudo-gradient norm increases substantially.

We study the mismatch between the "non-fat-tailed" condition ($\alpha = 2$) and the empirical behavior of the stochastic psudo-gradient noise. In Fig. 1, we illustrate the distributions of the norms of the stochastic pseudo-gradient noises computed with convolutional neural network (CNN) on CIFAR-10 dataset in both i.i.d. and non-i.i.d. client dataset settings. We can clearly observe that the non-i.i.d. case exhibit a rather fat-tailed behavior, where the pseudo-gradient norm could be as large as $1.6$. Although the i.i.d. case appears to have a much lighter tail, our detailed analysis shows that it still exhibits a fat-tailed behavior. To see this, in Fig. 2, we estimate $\alpha$-value for CIFAR-10 dataset in different scenarios: 1) different local update steps, and 2) different data heterogeneity. We use a parameter $p$ to characterize the data heterogeneity level, with $p = 10$ corresponding to the i.i.d. case. The smaller the $p$, the more heterogeneous the data among clients. Fig. 2 shows that the $\alpha$-value is smaller than $1.15$ in all scenarios, and $\alpha$ increases as the non-i.i.d. index $p$ increases (i.e., closer to the i.i.d. case). This implies that the stochastic pseudo gradient noise is fat-tailed and the "fatness" increases as the clients' data become more heterogeneous.

**3) The Impacts of Fat-Tailed Noise on Federated Learning:** Next, we show that the fat-tailed noise could lead to a "catastrophic model failure" (i.e., a sudden and dramatic drop of learning accuracy), consistent with previous observations in the FL literature [20]. To demonstrate this, we apply GFedAvg on the CIFAR-10 dataset and randomly sample five clients among $m = 10$ clients in each communication round. In Fig. 3, we illustrate a trial where a catastrophic training failure occurred. Correspondingly, we can observe in Fig. 3 a spike in the norm of the pseudo-gradient. This exceedingly large pseudo-gradient norm motivates us to apply the clipping technique to curtail the gradient updates. It is also worth noting that even if the squared norm of stochastic gradient may not be infinitely large in practice (i.e., having a bounded support empirically), it could still be too large to cause catastrophic model failures. In fact, under fat-tailed noise, the FedAvg algorithm could *diverge*, which follows from the fact that there exists one function that SGD diverges under heavy-tailed noise (see Remark 1 in [22]). As a result, the returned value by one client might be exceedingly large, leading to divergence of the FedAvg-type algorithms.

# 4 The FAT-Clipping algorithmic framework for fat-tailed federated learning

Given the evidence of fat-tailed noise in FL and its potential catastrophic training failure as shown in Section 3, there is a compelling need to design an efficient FL algorithm with provable convergence guarantee under fat-tailed noise in FL. Interestingly, the observation of an exceedingly large pseudo-gradient norm in Fig. 3 suggests a natural idea to mitigate fat-tailed noise: *clipping*. Toward this end, in Section 4.1 we first propose a clipping-based algorithmic framework called FAT-Clipping, which contains two variants: FAT-Clipping per-round (FAT-Clipping-PR) and FAT-Clipping per-iteration (FAT-Clipping-PI). Then in Section 4.2, we analyze their convergence rate performance.

## 4.1 The FAT-Clipping-PR and FAT-Clipping-PI algorithms

We illustrate the FAT-Clipping-PR and FAT-Clipping-PI algorithms in Algorithms 2 and 3, respectively. It can be seen that both FAT-Clipping-PR and FAT-Clipping-PI share a similar algorithmic

**Algorithm 2** The FAT-Clipping-PR Algorithm.

1: Initialize $\mathbf{x}_1$.
2: **for** $t = 1, \cdots, T$ (communication round) **do**
3: **for** each client $i \in [m]$ in parallel **do**
4: Update local model: $\mathbf{x}_{t,i}^1 = \mathbf{x}_t$.
5: **for** $k = 1, \cdots, K$ (local update step) **do**
6: Compute an unbiased estimate $\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ of $\nabla f_i(\mathbf{x}_{t,i}^k)$.
7: Local update: $\mathbf{x}_{t,i}^{k+1} = \mathbf{x}_{t,i}^k - \eta_L \nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$.
8: **end for**
9: Let $\Delta_t^i = \sum_{k\in[K]} \nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ .
10: **Clipping:** $\tilde{\Delta}_t^i = \min\{1, \frac{\lambda}{\|\Delta_t^i\|}\}\Delta_t^i$, where $\Delta_t^i = \sum_{k\in[K]} \nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$.
11: Send $\tilde{\Delta}_t^i$ to the server.
12: **end for**
13: Global Aggregation At Server:
14: Receive $\tilde{\Delta}_t^i, i \in [m]$.
15: Server Update: $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta\eta_L}{m} \sum_{i\in[m]} \tilde{\Delta}_t^i$.
16: Broadcasting $\mathbf{x}_{t+1}$ to clients.
17: **end for**

**Algorithm 3** The FAT-Clipping-PI Algorithm.

1: Initialize $\mathbf{x}_1$.
2: **for** $t = 1, \cdots, T$ (communication round) **do**
3: **for** each client $i \in [m]$ in parallel **do**
4: Update local model: $\mathbf{x}_{t,i}^1 = \mathbf{x}_t$.
5: **for** $k = 1, \cdots, K$ (local update step) **do**
6: Compute an unbiased estimate $\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ of $\nabla f_i(\mathbf{x}_{t,i}^k)$.
7: **Clipping:** $\tilde{\nabla} f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k) = \min\{1, \frac{\lambda}{\|\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)\|}\}\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$.
8: Local update: $\mathbf{x}_{t,i}^{k+1} = \mathbf{x}_{t,i}^k - \eta_L \tilde{\nabla} f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$.
9: **end for**
10: Send $\tilde{\Delta}_t^i = \sum_{k\in[K]} \tilde{\nabla} f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ to the server.
11: **end for**
12: Global Aggregation At Server:
13: Receive $\tilde{\Delta}_t^i, i \in [m]$.
14: Server Update: $\mathbf{x}_{t+1} = \mathbf{x}_t - \frac{\eta\eta_L}{m} \sum_{i\in[m]} \tilde{\Delta}_t^i$.
15: Broadcasting $\mathbf{x}_{t+1}$ to clients.
16: **end for**

structure with GFedAvg, with the key differences lying in the additional clipping operations. In FAT-Clipping-PR, each client performs a clipping in each communication round on the returned $\Delta_t^i$:

$$\tilde{\Delta}_t^i = \min\left\{1, \frac{\lambda}{\|\Delta_t^i\|}\right\}\Delta_t^i, \tag{2}$$

and then sends $\tilde{\Delta}_t^i$ instead of $\Delta_t^i$ to the server (Line 10 in Algorithm 2). By contrast, in FAT-Clipping-PI, each client clips the stochastic gradient before each local update step (Line 7 in Algorithm 3):

$$\tilde{\nabla} f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k) = \min\left\{1, \frac{\lambda}{\|\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)\|}\right\}\nabla f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k), \tag{3}$$

$$\mathbf{x}_{t,k+1}^i = \mathbf{x}_{t,i}^k - \eta_L \tilde{\nabla} f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k). \tag{4}$$

Then, $\tilde{\Delta}_t^i = \sum_{k\in[K]} \tilde{\nabla} f_i(\mathbf{x}_{t,i}^k, \xi_{t,i}^k)$ is sent to the server for aggregation (Line 10 in Algorithm 3).

### 4.2 Convergence analysis of the FAT-Clipping algorithms

Before conducting the convergence analysis for the FAT-Clipping algorithms, we first state two standard assumptions that are commonly used in the literature of first-order stochastic methods.

**Assumption 1** ($L$-Lipschitz Continuous Gradient)**.** *There exists a constant $L > 0$, such that $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, and $i \in [m]$.*

**Assumption 2** (Unbiased Local Gradient Estimator)**.** *The local gradient estimator is unbiased, i.e., $\mathbb{E}[\nabla f_i(\mathbf{x}, \xi)] = \nabla f_i(\mathbf{x})$, $\forall i \in [m]$, where $\xi$ is a random local data sample at the $i$-th worker.*

Next, we state the key bounded $\alpha$-moment assumption for *fat-tailed* the stochastic first-order oracle, which leverages the notion of tail-index introduced in Section 3:

**Assumption 3** (Bounded $\alpha$-Moment)**.** *There exists a real number $\alpha \in (1, 2]$ and a constant $G \geq 0$, such that $\mathbb{E}[\|\nabla f_i(\mathbf{x}, \xi)\|^\alpha] \leq G^\alpha$, $\forall i \in [m], \mathbf{x} \in \mathbb{R}^d$.*

**1) Convergence Rates of the FAT-Clipping-PR Algorithm:** We first state the convergence rates of FAT-Clipping-PR for $\mu$-strongly convex and non-convex objective functions.

**Theorem 1.** (Convergence Rate of FAT-Clipping-PR in the Strongly Convex Case) *Suppose that $f(\cdot)$ is a $\mu$-strongly convex function. Under Assumptions 1–3, if $\eta\eta_L K \geq \frac{2}{\mu T}$, then the output $\bar{\mathbf{x}}_T$ of* FAT-Clipping-PR *being chosen in such a way that $\bar{\mathbf{x}}_T = \mathbf{x}_t$ with probability $\frac{w_t}{\sum_{j\in[T]} w_j}$, where $w_t = (1 - \frac{1}{2}\mu\eta\eta_L K)^{1-t}$, satisfies:*

$$f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) \leq \frac{\mu}{2}\exp\left(-\frac{1}{2}\mu\eta\eta_L KT\right) + \frac{\eta\eta_L K}{2}G^\alpha\lambda^{2-\alpha} + \frac{4}{\mu}\left[2G^{2\alpha}\lambda^{2-2\alpha} + 2L^2\eta_L^2 K^2 G^\alpha\lambda^{2-\alpha}\right],$$

*where $\mathbf{x}^*$ denotes the global optimal solution. Further, let $\eta\eta_L K = \frac{2c}{\mu}\frac{\ln(T)}{mKT}$, where $c \geq 1$ is a constant satisfying $m^{\frac{2-2\alpha}{\alpha}} K^{\frac{2}{\alpha}} T^{c+\frac{2-2\alpha}{\alpha}} \geq 1$, and let $\eta_L \leq (mKT)^{\frac{1-\alpha}{\alpha}}$. It then follows that*

$$f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) = \mathcal{O}((mT)^{\frac{2-2\alpha}{\alpha}} K^{\frac{2}{\alpha}}).$$

**Theorem 2.** (Convergence Rate of FAT-Clipping-PR in the Nonconvex Case) *Suppose that $f(\cdot)$ is a nonconvex function. Under Assumptions 1–3, if $\eta\eta_L KL \leq 1$, then the sequence of outputs $\{\mathbf{x}_k\}$ generated by* FAT-Clipping-PR *satisfies:*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2\left(f(\mathbf{x}_1) - f(x_T)\right)}{\eta\eta_L KT} + \left(L^2\eta_L^2 K^2 G^2 + K^2 G^{2\alpha}\lambda^{-2(\alpha-1)} + L\eta_L K^2 G^{1+\alpha}\lambda^{1-\alpha}\right) + \frac{L\eta\eta_L}{m}\left(KG^\alpha\lambda^{2-\alpha}\right).$$

*Further, choosing learning rates and clipping parameter in such a way that $\eta\eta_L = m^{\frac{2\alpha-2}{3\alpha-2}} K^{\frac{-\alpha-2}{3\alpha-2}} T^{\frac{-\alpha}{3\alpha-2}}$, $\eta_L \leq (mT)^{\frac{1-\alpha}{3\alpha-2}} K^{\frac{4-4\alpha}{3\alpha-2}}$, and $\lambda = (mK^4T)^{\frac{1}{3\alpha-2}}$, we have*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 = \mathcal{O}((mT)^{\frac{2-2\alpha}{3\alpha-2}} K^{\frac{4-2\alpha}{3\alpha-2}}).$$

**Remark 1.** We note that the above convergence rates for FAT-Clipping-PR does not generalize the results of FedAvg when $\alpha = 2$ (non-fat-tailed noise). Specifically, FedAvg is able to achieve $\tilde{\mathcal{O}}((mKT)^{-1})$ and $\mathcal{O}((mKT)^{-\frac{1}{4}})$ convergence rates for strongly convex and non-convex function, respectively [3, 39]. In contrast, FAT-Clipping-PR achieves $\mathcal{O}((mT)^{-1}K)$ and $\mathcal{O}((mT)^{-\frac{1}{4}})$ for strongly-convex and non-convex functions, respectively. These two rates are consistent with those of FedAvg in terms of $m$ and $T$, but not in terms of $K$.

Interestingly, with a separate proof for non-fat-tailed noise ($\alpha = 2$), we can show that clipping does not affect the dependence on $K$ in the convergence rates. Thus, FAT-Clipping-PR has the *same* convergence rates as those of FedAvg. Due to space limitation, we state an informal version of these theorems here. The full versions of Theorem 6,7 and their proofs are formally stated in Appendix.

**Theorem 6** & **7 (informal)** (Convergence Rates of FAT-Clipping-PR for Non-Fat-Tailed Noise): *For $\alpha = 2$, CPR-FedAvg achieves convergence rate $\tilde{\mathcal{O}}((mKT)^{-1})$ for strongly-convex and $\mathcal{O}((mKT)^{-\frac{1}{4}})$ for non-convex functions, respectively.*

**2) Convergence Rate of the FAT-Clipping-PI Algorithm:** Next, we provide the convergence rates of FAT-Clipping-PI for $\mu$-strongly convex and non-convex objective functions.

**Theorem 3.** (Convergence Rate of FAT-Clipping-PI in the Strongly Convex Case) *Suppose that $f(\cdot)$ is a $\mu$-strongly convex function. Under Assumptions 1–3, if $\eta\eta_L K \geq \frac{2}{\mu T}$, then the output $\bar{\mathbf{x}}_T$ of* FAT-Clipping-PI *being chosen in such a way that $\bar{\mathbf{x}}_T = \mathbf{x}_t$ with probability $\frac{w_t}{\sum_{j\in[T]} w_j}$, where $w_t = (1 - \frac{1}{2}\mu\eta\eta_L K)^{1-t}$, satisfies:*

$$f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) \leq \frac{\mu}{2}\exp\left(-\frac{1}{2}\mu\eta\eta_L KT\right) + \frac{\eta\eta_L K}{2}G^\alpha\lambda^{2-\alpha} + \frac{4}{\mu}[2G^{2\alpha}\lambda^{-2(\alpha-1)} + 2L^2\eta_L^2K^2G^\alpha\lambda^{2-\alpha}],$$

*where $\mathbf{x}^*$ denotes the global optimal solution. Further, let $\eta\eta_L K = \frac{2c}{\mu}\frac{\ln(T)}{mKT}$, where $c \geq 1$ is a constant satisfying $(mK)^{\frac{2-2\alpha}{\alpha}}T^{c+\frac{2-2\alpha}{\alpha}} \geq 1$, and let $\lambda = (mKT)^{\frac{1}{\alpha}}$, and $\eta_L \leq (mT)^{-\frac{1}{2}}K^{-\frac{3}{2}}$). It then follows that*

$$f(\bar{\mathbf{x}}_T) - f(\mathbf{x}^*) = \tilde{\mathcal{O}}((mKT)^{\frac{2-2\alpha}{\alpha}}).$$

**Theorem 4.** (Convergence Rate of FAT-Clipping-PI in the Nonconvex Case) *Suppose that $f(\cdot)$ is a non-convex function. Under Assumptions 1–3, if $\eta\eta_L KL \leq 1$, then the sequence of outputs $\{\mathbf{x}_k\}$ generated by* FAT-Clipping-PI *satisfies:*

$$\min_{t\in[T]}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \frac{2\left(f(\mathbf{x}_1) - f(x_T)\right)}{\eta\eta_L KT} + \left(2G^{2\alpha}\lambda^{-2(\alpha-1)} + 2L^2\eta_L^2K^2G^\alpha\lambda^{2-\alpha}\right) + \frac{L\eta\eta_L}{m}\left(G^\alpha\lambda^{2-\alpha}\right).$$

*Further, choosing learning rates and clipping parameter in such a way that $\eta\eta_L = m^{\frac{2\alpha-2}{3\alpha-2}}(KT)^{\frac{-\alpha}{3\alpha-2}}, \eta_L \leq (mKT)^{\frac{-\alpha}{6\alpha-4}}$, and $\lambda = (mKT)^{\frac{1}{3\alpha-2}}$, we have*

$$\min_{t\in[T]}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|^2 \leq \mathcal{O}((mKT)^{\frac{2-2\alpha}{3\alpha-2}}).$$

**Remark 2.** In comparison to FAT-Clipping-PR, convergence rates of FAT-Clipping-PI generalize the results of FedAvg for the non-fat-tailed noise case (i.e., $\alpha = 2$). Specifically, when $\alpha = 2$, FAT-Clipping-PI achieves $\tilde{\mathcal{O}}((mKT)^{-1})$ and $\mathcal{O}((mKT)^{-\frac{1}{4}})$ convergence rates for strongly convex and nonconvex objective functions, respectively. These two convergence rates are consistent with those of FedAvg in terms of $m$, $K$ and $T$ (ignoring logarithmic factors in the strongly-convex case).

Next, we show that the convergence rates for FAT-Clipping-PI is *order-optimal* for $\alpha \in (1, 2]$ by constructing the lower fundamental bounds.

**Theorem 5** (Convergence Rate Lower Bound)**.** *Given any $\alpha \in (1, 2]$, for any potentially randomized algorithm, there exists a stochastic strongly-convex function satisfying Assumption 3 with $G \leq 1$, such that the output of $\mathbf{x}_T$ after $T$ communication rounds has an expected error lower bounded by*

$$\mathbb{E}[f(\mathbf{x}_t)] - f(\mathbf{x}_*) = \Omega((mKT)^{\frac{2-2\alpha}{\alpha}}).$$

*Also, there exists a non-convex function satisfying Assumption 3, such that the output of $\mathbf{x}_T$ after $T$ communication rounds has an expected error lower bounded by*

$$\mathbb{E}[\|\nabla f(\mathbf{x}_t)\|]^2 = \Omega((mKT)^{\frac{2-2\alpha}{3\alpha-2}}).$$

With $T$ communication rounds, the total number of stochastic gradients is $mKT$. Thus, the lower bounds above can be obtained from the centralized SGD with fat-tailed noise [22, Theorems 5 and 6]). Clearly, the above lower bounds imply the optimality of the convergence rates of FAT-Clipping-PI .

## 5 Numerical results

In this section, we conduct numerical experiments to verify the theoretical findings in Section 4 using 1) a synthetic function, and 2) a convolutional neural network (CNN) with two convolutional

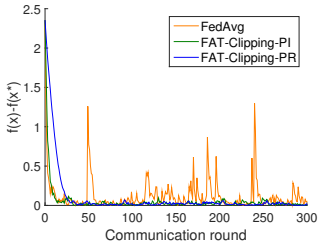


Figure 4: Convergence comparisons of FedAvg, FAT-Clipping-PI, and FAT-Clipping-PR for solving strongly convex models: synthetic data with $\xi$ having Cauchy tails (fat).

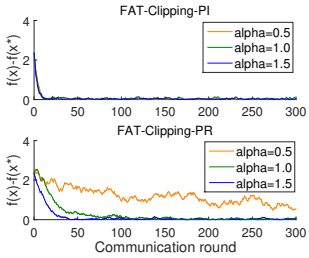


Figure 5: Convergence comparisons of FAT-Clipping-PI, and FAT-Clipping-PR for solving strongly convex models: synthetic data with $\xi$ having different fat tails represented by $\alpha$.

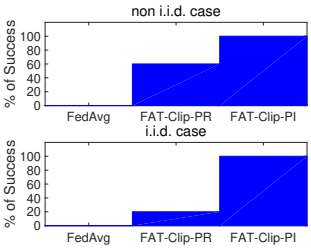


Figure 6: Percentage of successful training over 5 trials when applying FedAvg, FAT-Clipping-PR and FAT-Clipping-PI to CIFAR-10 dataset in non-i.i.d. case and i.i.d. case.

layers on CIFAR-10 dataset [40]. Due to space limitation, we relegate experiment details and extra experimental results to the supplementary material.

**1) Strongly Convex Model with Synthetic Data:** We consider a strongly convex model for Problem (1) as follows: $f_i(x) = \mathbb{E}_\xi[f_i(x,\xi)]$ and $f_i(x,\xi) = \frac{1}{2}\|x\|^2 + \langle \xi, x\rangle$, where $\xi$ is a random vector. We compare FedAvg, FAT-Clipping-PI, and FAT-Clipping-PR, where the noise $\xi$ is Cauchy distributed ($\alpha < 2$, fat-tailed). Also, we compare FAT-Clipping-PI and FAT-Clipping-PR with $\xi$ having different tail-indexes ($\alpha = 0.5, 1.0,$ and $1.5$). For each distribution, we use the same experimental setup, and $m = 5$ clients participate in the training. We show the trajectories of FedAvg, FAT-Clipping-PI, and FAT-Clipping-PR for solving Problem (1) with $\xi$ having Cauchy tails in Fig. 4 and with $\xi$ having different $\alpha$-values in Fig. 5. We can clearly observe from Fig. 4 that FAT-Clipping-PI and FAT-Clipping-PR converge rapidly in the Cauchy case, and FAT-Clipping-PI converges faster than FAT-Clipping-PR as our theoretical results predict. In contrast, FedAvg is not convergent in the Cauchy case. In Fig. 5, we can see that the convergence processes of FAT-Clipping-PR and FAT-Clipping-PI become slower as the $\alpha$-value increases as our theoretical results predict, but the differences in FAT-Clipping-PI is much less obvious compared to those of FAT-Clipping-PR.

**2) CNN (Non-convex Model) on the CIFAR-10:** This setting has $m = 10$ clients in total, and five clients are randomly selected to participate in each round of the training. We compare FAT-Clipping algorithms with FedAvg under different data heterogeneity. To simulate data heterogeneity across clients, we distribute the data to each client in a label-based partition following the same procedure as in existing works (e.g., [1, 41, 42]): we use a parameter $p$ to represent the number of labeled classes in each client, with $p = 10$ corresponding to the i.i.d. case and the rest corresponding to non-i.i.d. cases. The smaller the $p$-value, the more heterogeneous the data across clients. In Fig. 6, we present the percentage of successful training over 5 trials when applying FedAvg, FAT-Clipping-PR and FAT-Clipping-PI on CIFAR-10 in non-i.i.d. case ($p = 2$) and i.i.d. case ($p = 10$). FAT-Clipping-PI has 100% successful rates (i.e., no catastrophic model failures) in both non-i.i.d. and i.i.d cases, and FAT-Clipping-PR has 60% and 20% successful rates in non-i.i.d. and i.i.d. cases, respectively. However, FedAvg fails in all 5 trials. Thus, compared to FedAvg, FAT-Clipping methods (FAT-Clipping-PI in particular) significantly reduce catastrophic training failures.

## 6 Conclusions and future work

In this paper, we investigated the problem of designing efficient federated learning algorithms with convergence performance guarantee in the presence of fat-tailed noise in the stochastic first-order oracles. We first showed empirical evidence that fat-tailed noise in federated learning can be induced by data heterogeneity and heterogeneous local update steps. To address the fat-tailed noise challenge in FL algorithm design, we proposed a clipping-based algorithmic framework called FAT-Clipping . The FAT-Clipping framework contains two variants FAT-Clipping-PR and FAT-Clipping-PI , which perform clipping operations in each communication round and in each local update step, respectively. Then, we derived the convergence rate bounds of FAT-Clipping-PR and FAT-Clipping-PI for strongly convex and non-convex loss functions under fat-tailed noise. Not only does our work shed light on theoretical understanding of FL under fat-tailed noise, it also opens the doors to many new interesting questions in FL systems that experience fat-tailed noise.

# 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*. PMLR, 2017, pp. 1273–1282.

[2] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proceedings of Machine Learning and Systems*, I. Dhillon, D. Papailiopoulos, and V. Sze, Eds., vol. 2, 2020, pp. 429–450.

[3] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 5132–5143.

[4] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[5] X. Zhang, M. Hong, S. Dhople, W. Yin, and Y. Liu, "Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data," *arXiv preprint arXiv:2005.11418*, 2020.

[6] D. A. E. Acar, Y. Zhao, R. M. Navarro, M. Mattina, P. N. Whatmough, and V. Saligrama, "Federated learning based on dynamic regularization," in *International Conference on Learning Representations*, 2021.

[7] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," in *International Conference on Learning Representations*, 2021.

[8] S. J. Reddi, Z. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," in *International Conference on Learning Representations*, 2021.

[9] P. Khanduri, P. SHARMA, H. Yang, M. Hong, J. Liu, K. Rajawat, and P. Varshney, "STEM: A stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[10] X. Gu, K. Huang, J. Zhang, and L. Huang, "Fast federated learning in the presence of arbitrary device unavailability," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[11] M. Luo, F. Chen, D. Hu, Y. Zhang, J. Liang, and J. Feng, "No fear of heterogeneity: Classifier calibration for federated learning with non-IID data," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[12] S. P. Karimireddy, M. Jaggi, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Breaking the centralized barrier for cross-device federated learning," in *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[13] J. Nair, A. Wierman, and B. Zwart, *The Fundamentals of Heavy Tails*. Cambridge University Press, 2022, vol. 53.

[14] J. E. Taylor. Heavy-tailed distributions. [Online]. Available: https://math.la.asu.edu/~jtaylor/teaching/Spring2016/STP421/lectures/stable.pdf

[15] U. Simsekli, L. Sagun, and M. Gurbuzbalaban, "A tail-index analysis of stochastic gradient noise in deep neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5827–5837.

[16] M. Gurbuzbalaban, U. Simsekli, and L. Zhu, "The heavy-tail phenomenon in sgd," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3964–3975.

[17] J. Zhang, T. He, S. Sra, and A. Jadbabaie, "Why gradient clipping accelerates training: A theoretical justification for adaptivity," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=BJgnXpVYwS

[18] E. Gorbunov, M. Danilova, and A. Gasnikov, "Stochastic optimization with heavy-tailed noise via accelerated gradient clipping," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 042–15 053, 2020.

[19] A. Panigrahi, R. Somani, N. Goyal, and P. Netrapalli, "Non-gaussianity of stochastic gradient noise," *arXiv preprint arXiv:1910.09626*, 2019.

[20] Z. Charles, Z. Garrett, Z. Huo, S. Shmulyian, and V. Smith, "On large-cohort training for federated learning," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[21] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.

[22] J. Zhang, S. P. Karimireddy, A. Veit, S. Kim, S. Reddi, S. Kumar, and S. Sra, "Why are adaptive methods good for attention models?" *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 383–15 393, 2020.

[23] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update sgd," in *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia, Eds., vol. 1, 2019, pp. 212–229. [Online]. Available: https://proceedings.mlsys.org/paper/2019/file/c8ffe9a587b126f152ed3d89a146b445-Paper.pdf

[24] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local sgd," *arXiv preprint arXiv:1808.07217*, 2018.

[25] T. H. Nguyen, U. Simsekli, M. Gurbuzbalaban, and G. Richard, "First exit time analysis of stochastic gradient descent under heavy-tailed gradient noise," *Advances in neural information processing systems*, vol. 32, 2019.

[26] U. Simsekli, L. Zhu, Y. W. Teh, and M. Gurbuzbalaban, "Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8970–8980.

[27] L. Hodgkinson and M. Mahoney, "Multiplicative noise and heavy tails in stochastic optimization," in *International Conference on Machine Learning*. PMLR, 2021, pp. 4262–4274.

[28] H. Wang, M. Gurbuzbalaban, L. Zhu, U. Simsekli, and M. A. Erdogdu, "Convergence rates of stochastic gradient descent under infinite noise variance," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[29] S. Yaida, "Fluctuation-dissipation relations for stochastic gradient descent," *arXiv preprint arXiv:1810.00004*, 2018.

[30] W. Hu, C. J. Li, L. Li, and J.-G. Liu, "On the diffusion approximation of nonconvex stochastic gradient descent," *Annals of Mathematical Sciences and Applications*, vol. 4, no. 1, 2019.

[31] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[32] N. Z. Shor, "Minimization methods for non-differentiable functions," in *Springer Series in Computational Mathematics*, 1985.

[33] B. Zhang, J. Jin, C. Fang, and L. Wang, "Improved analysis of clipping algorithms for non-convex optimization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 511–15 521, 2020.

[34] X. Chen, S. Z. Wu, and M. Hong, "Understanding gradient clipping in private sgd: A geometric perspective," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 773–13 782, 2020.

[35] J. Qian, Y. Wu, B. Zhuang, S. Wang, and J. Xiao, "Understanding gradient clipping in incremental gradient methods," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 1504–1512.

[36] X. Zhang, X. Chen, M.-F. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," *ArXiv*, vol. abs/2106.13673, 2021.

[37] 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.

[38] G. Andrew, O. Thakkar, B. McMahan, and S. Ramaswamy, "Differentially private learning with adaptive clipping," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[39] 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.

[40] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[41] 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.

[42] H. Yang, M. Fang, and J. Liu, "Achieving linear speedup with partial worker participation in non-IID federated learning," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=jDdzh5ul-d

## Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] As a theoretical paper towards further understanding of federated optimization, we do not see a direct path to any negative applications.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4.2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] The dataset used in this paper is widely-used public datasets and we have provided detailed instruction for the partition about datasets.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We run the experiments multiple times and report the failure rates instead.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]