CFedAvg: Achieving Efficient Communication and Fast Convergence in Non-IID Federated Learning

Haibo Yang
Department of ECE
The Ohio State University
Columbus, OH, U.S.A
yang.5952@osu.edu

Jia Liu
Department of ECE
The Ohio State University
Columbus, OH, U.S.A
liu@ece.osu.edu

Elizabeth S. Bentley
Information Directorate
Air Force Research Laboratory
Rome, NY, U.S.A
elizabeth.bentley.3@us.af.mil

ABSTRACT

Federated learning (FL) is a prevailing distributed learning paradigm, where a large number of workers jointly learn a model without sharing their training data. However, high communication costs could arise in FL due to large-scale (deep) learning models and bandwidth-constrained connections. In this paper, we introduce a communication-efficient algorithmic framework called CFedAvg for FL with non-i.i.d. datasets, which works with general (biased or unbiased) SNR-constrained compressors. We analyze the convergence rate of CFedAvg for non-convex functions with constant and decaying learning rates. The CFedAvg algorithm can achieve an $O(1/\sqrt{mKT} + 1/T)$ convergence rate with a constant learning rate, implying a linear speedup for convergence as the number of workers increases, where K is the number of local steps, T is the number of total communication rounds, m is the total worker number. This matches the convergence rate of distributed/federated learning without compression, thus achieving high communication efficiency while not sacrificing learning accuracy in FL. Furthermore, we extend CFedAvg to heterogeneous local steps with convergence guarantees, which allows different workers to perform a distinct number of local steps to better adapt to their own circumstances. The interesting observation in general is that the noise/variance introduced by compressors does not affect the overall convergence rate order for non-i.i.d. FL. We verify the effectiveness of our CFedAvg algorithm on three datasets with two gradient compression schemes of different compression ratios.

CCS CONCEPTS

• **Computing methodologies** → **Distributed algorithms**; • **Theory of computation** → **Distributed algorithms**.

KEYWORDS

Distributed/federated learning, communication efficient, convergence analysis, Non-IID Data

MobiHoc '20, June 30 – July 03, 2020, Shanghai, China

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Haibo Yang, Jia Liu, and Elizabeth S. Bentley. 2018. CFedAvg: Achieving Efficient Communication and Fast Convergence in Non-IID Federated Learning. In *MobiHoc '20: ACM Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, June 30 – July 03, 2020, Shanghai, China.* ACM, New York, NY, USA, 19 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In recent years, advances in machine learning (ML) have sparked many new and emerging applications that transform our society, which include a collection of widely used deep learning models for computer vision, text prediction, and many others [20, 29]. Traditionally, the training of ML applications often relies on cloud-based large data-centers to collect and process a vast amount of data. However, with the rise of Internet of Things (IoT), data and demands for ML are increasingly being generated from mobile devices in wireless edge networks. Due to high latency, low bandwidth, and privacy/security concerns, aggregating all data to the cloud for ML training may no longer be desirable or may even be infeasible. In these circumstances, Federated Learning (FL) has emerged as a prevailing ML paradigm, thanks to the rapidly growing computation capability of modern mobile devices. Generally speaking, FL is a network-based ML architecture, under which a large number of local devices (often referred to as workers/nodes) collaboratively train a model based on their local datasets and coordinated by a central parameter server. In FL, the parameter server is responsible for aggregating and updating model parameters without requiring the knowledge of the data located at each worker. Workers process their computational tasks independently with decentralized data, and communicate to the parameter server to update the model. By doing so, not only can FL significantly alleviate the risk of exposing data privacy, it also fully utilizes the idle computation resources at the workers. This constitutes a win-win situation that have led to a variety of successful real-world applications (see [13] for a comprehensive survey).

Despite the aforementioned advantages of FL, a number of technical challenges also arise due to the unique characteristics of FL. One key challenge in FL is the high communication cost due to the every-increasing sizes of learning models and datasets [13]. For example, modern models in deep learning (e.g., ResNet [10], VGG [30], etc.) typically contain millions of parameters, which implies a large amount of data being injected into the network that supports FL. The problem of a high communication load in FL is further exacerbated by the fact that, in many wireless edge networks, the communication links are often bandwidth-constrained and their

link capacities are highly dynamic due to stochastic channel fading effects. As a result, information exchanges between the parameter server and workers could be highly inefficient, rendering a major bottleneck in FL [13]. If this problem is not handled appropriately, FL could perform far worse than its centralized counterparts in scenarios where the communication-to-computation cost ratio is high and/or communication resources are constrained. Given the rapidly growing demands of FL and the restricted communication environments in reality, there is a compelling need to significantly reduce the communication cost in FL without substantially decaying the learning performance.

Generally speaking, the total communication cost during the training process is determined by two factors: the number of communication rounds and the size (or amount) of the update parameters in each communication round. There are some algorithms in FL (e.g., federated averaging (FedAvg) [25]) that take more local gradient steps at each node and communicate infrequently with the parameter server, thus decreasing the total number of communication rounds and in turn reducing the total communication cost. However, this does not completely solve the problem since the communicated parameter/gradient vectors could still be high-dimensional. On the other hand, in the literature, there exist a variety of gradient compression techniques (e.g., signSGD [5], gradient dropping [1], TernGrad [40], etc.) that were originally proposed for centralized/distributed learning and shown to be effective in reducing the size of exchanged parameters in each communication round. For example, Lin *et al.* [24] numerically demonstrated that 99.9% of the gradient exchange in distributed learning are redundant and proposed Deep Gradient Compression (DGC) to largely reduce the communication bandwidth requirement. Similar compression ideas have also been extended to decentralized learning over networks *without* dedicated parameter servers. In [45, 46], Zhang *et al.* developed a series of decentralized learning algorithms with differential-coded compressions. Koloskova *et al.* [17] proposed an algorithm that could achieve a linear speedup with respect to the number of workers for convergence in decentralized learning with arbitrary gradient compression.

Given the above encouraging results of information compression in distributed/decentralized learning, an interesting question naturally arises: *Could we combine compression with infrequent communication to further reduce the communication cost of FL?* However, answering this question turns out to be highly non-trivial. One key challenge stems from the heterogeneity of the local datasets among different workers. In the traditional distributed learning literature, the dataset at each node is usually well-shuffled and hence can be assumed to be independent and identically distributed (i.i.d.). However, the dataset at each worker in FL could be generated based on the local environment and cannot be shuffled with other workers due to privacy protection. Thus, the i.i.d. assumption often fails to hold. In some circumstances, the dataset distributions at different workers could vary dramatically due to factors such as geographic location differences, time window gaps, among many others. Upon integrating compression with FL, the already-complicated non-i.i.d. dataset problem is further worsened by the significant loss of information due to the use of compression operators. In addition, under infrequent communication, the multiple local steps in each worker introduce further "model drift" to the non-i.i.d. datasets. It has been shown that this "model drift" results in extra variances that may lead to deterioration or even failures of training [21]. Due to these complex randomness couplings between compression, local steps, and non-i.i.d. datasets, results on non-i.i.d. compressed FL remain limited. This motivates us to fill this gap and rigorously investigate the algorithmic design that integrates compression in non-i.i.d. FL.

Moreover, workers in FL system vary tremendously in terms of computation capabilities and resources limits (e.g., memory, battery capacity). Hence, using a predefined constant number of local steps for all workers (assumed in most existing work in FL) may not be a good design strategy, which may result in the faster workers idling and slower workers causing straggler problems. As a result, another important question in FL emerges: *Could we use heterogeneous local steps for workers to further improve flexibility and efficiency in FL?*

In this paper, we answer the above open questions by proposing a communication-efficient algorithm called *CFedAvg* (compressed FedAvg) with error-feedback. Our CFedAvg algorithm reduces both the communication rounds and link capacity requirement in each communication round. It also allows the use of heterogeneous local steps (i.e., different workers perform different local steps to better adapt to their own computing environments). Our main contributions and results are summarized as follows:

- We show that, under general signal-to-noise-ratio (SNR) constrained compressors, the convergence rate of our CFedAvg algorithm is $O(1/\sqrt{mKT} + 1/T)$ and $\tilde{O}(1/\sqrt{mKT}) + O(1/\sqrt{T})$ with constant and decaying learning rates, respectively, for general non-convex functions and non-i.i.d. datasets in FL, where K is the number of local steps, T is the number of total communication rounds, and m is the total number of workers. For a sufficiently large T, this implies that CFedAvg achieves an $O(1/\sqrt{mKT})$ convergence rate with a constant learning rate and enjoys the linear speedup effect as the number of workers increases.[1] Note that this matches the convergence rate of uncompressed distributed/federated learning algorithm orderly [3, 7, 14], thus achieving high communication efficiency while not sacrificing learning accuracy in FL.
- We extend CFedAvg to heterogeneous local steps among workers, which allows each worker performs different local steps based on its own computation capability and other conditions. To our knowledge, our paper is the first to show that FL can still offer theoretical performance guarantee without requiring the same predefined constant number of local steps among all workers.
- We show that the use of SNR-constrained compressors in CFedAvg only slightly increases the local variance constant and does not affect the overall convergence rate order for non-i.i.d. FL with infrequent communication. Moreover, we show that the convergence results of CFedAvg hold for either unbiased or biased SNR-constrained compressors, which is far more flexible than previous works that require unbiased compressors.
- We verify the effectiveness of CFedAvg on MNIST, FMNIST and CIFAR-10 datasets with different SNR-constrained compressors.

[1] To attain an ϵ accuracy for an algorithm, it takes $O(1/\epsilon^2)$ steps with a convergence rate $O(1/\sqrt{T})$. In contrast, it takes $O(1/m\epsilon^2)$ steps if the convergence rate is $O(1/\sqrt{mT})$ (the hidden constant in Big-O is the same). In this sense, it is a *linear speedup* with respect to the number of workers m.

We find that CFedAvg can reduce up to 99% of information exchange with minimal impacts on learning accuracy, which confirms the communication-efficiency advantages of training large learning models in non-i.i.d. FL with information compression.

The rest of the paper is organized as follows. In Section 2, we review the literature to put our work in comparative perspectives. In Section 3, we introduce the system model, problem formulation, and preliminaries on general SNR-constrained compressors. In Section 4, we present the details of our algorithm design and the convergence analysis. Numerical results are provided in Section 5 and Section 6 concludes this paper.

2 RELATED WORK

For distributed/federated learning in communication-constrained environments, a variety of communication-efficient algorithms have been proposed. We categorize the existing work into two classes: one is to use infrequent communication to reduce the communication rounds and the other is to compress the information to reduce the size of parameters transmitted from each worker to the parameter server in each communication round.

Infrequent-Communication Approaches: One notable algorithm of FL is the federated averaging (FedAvg) algorithm, which was first proposed by McMahan *et al.* [25] as a heuristic to improve both communication efficiency and data privacy. In FedAvg, every worker performs multiple SGD steps independently to update the model locally before communicating with the parameter server, which is different from traditional distributed learning with only one local step. It has been shown that the number of local steps can be up to 100 without significantly affecting the convergence speed in i.i.d. datasets for various convolution and recurrent neural network models. Since then, this work has sparked many follow-ups that focus on FL with i.i.d. datasets (referred to as LocalSGD) [16, 23, 31, 33, 36, 44, 48] and non-i.i.d. datasets [3, 11, 12, 14, 21, 23, 28, 37, 47]. These studies heuristically demonstrated the effectiveness of FedAvg and its variants on reducing communication cost. Also, researchers have theoretically shown that FedAvg and its variants can achieve the same convergence rate order as the traditional distributed learning (see, e.g., [3, 13, 14, 22]).

Compression-Based Approaches: Although FedAvg and its variants save communication costs by utilizing multiple local steps to reduce the total number of communication rounds, it has to transmit the full amount of model parameters in each communication round at every worker. Thus, it could still induce high latency and communication overhead in networks with low connection speeds or large channel variations. To address this challenge, a natural idea is to compress the parameters to reduce the amount of transmitted data from each worker to the parameter server. Compression-based approaches have attracted increasing attention in recent years in distributed and decentralized learning [17, 24], which have enabled the training of large-size models over networks with low-speed connections. Broadly speaking, compression-based approaches can be classified into the following two main categories:

- **Quantization:** The basic idea of quantization is to project a vector from a high-dimensional space to a low-dimensional subspace, so that the projected vector can be represented by a fewer number of bits. Notable examples of quantization-based algorithms in the learning literature include, e.g., signSGD [5], QSGD [2]. Note that these commonly seen quantization schemes could be either unbiased [2, 40] or biased [5].
- **Sparsification:** Given a high-dimensional vector, the basic idea of sparsification is to select only a part of its components to transmit. The component selection could be based on a predefined threshold. For example, Strom [34] proposed to only send components in the vector that are larger than a predefined constant, while Aji *et al.* [1] chose to send a fixed proportion of components. The component selection could also be randomized [1, 39]. Other variants include, e.g., adaptive threshold [8] and unbiased random dropping [39].

In fact, these two approaches are closely related, and there are works that combine them to achieve better compression results [2, 28, 40]. Qsparse-local-SGD proposed by Basu *et al.* [4] is the most related work to this paper, which combines quantization, sparsification and local steps to be more communication-efficient. However, our algorithm has a better convergence rate with more relaxed assumptions. We also propose an algorithm design that allows more flexible heterogeneous local steps. Please see Section 4.2 for further details.

Error Feedback: With the information loss of model parameters transmitted from the workers to server due to gradient compression, training accuracy could be significantly affected. To address this problem, the error feedback technique has been proposed [15, 24, 33, 35] for both distributed and decentralized learning. It has been shown that error feedback improves the convergence performance for cases with high compression ratios. It has also been theoretically shown that distributed and decentralized learning with gradient compression and error feedback could achieve the same convergence rate as that of the classical distributed SGD [15, 17, 33, 35] and enjoy the linear speedup effect.

So far, however, it remains unknown whether the same convergence rate (with linear speedup) could be achieved in FL with compression, particularly under the asynchrony due to non-i.i.d. dataset and the use of heterogeneous local steps at each worker. Answering this question constitutes the rest of this paper.

3 SYSTEM MODEL, PROBLEM FORMULATION, AND PRELIMINARIES

In this section, we first introduce the system model and problem formulation in Section 3.1. Then in Section 3.2, we provide some necessary background on the notion of general SNR-constrained compressors to facilitate the discussions in the rest of this paper.

Notation. In this paper, we use boldface to denote matrices and vectors. We use $\|\cdot\|_2$ to denote the ℓ^2-norm. For a positive integer m, we use $[m]$ to represent the set $\{1, \cdots, m\}$.

3.1 System Model and Problem Formulation

Consider an FL system with m workers who collaboratively learn a model with decentralized data and under the coordination of a central parameter server. The goal of the FL system 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 $F_i(\mathbf{x}) \triangleq \mathbb{E}_{\xi_i\sim D_i}[F_i(\mathbf{x},\xi_i)]$ denotes the local (non-convex) loss function, which evaluates the average discrepancy between the learning model's output and the ground truth corresponding to a random training sample ξ_i that follows a local data distribution D_i. In (1), the parameter d represents the dimensionality of the training model. For the i.i.d. setting, each local dataset is assumed to sample from some common latent distribution, i.e., $D_i = D, \forall i \in [m]$. In practice, however, the local dataset at each worker in FL could be generated based on its local environment and thus being non-i.i.d., i.e., $D_i \neq D_j$ if $i \neq j$. Note that the i.i.d. setting can be viewed as a special case of the non-i.i.d. setting. Hence, our results for the non-i.i.d. setting are directly applicable to the i.i.d. setting.

3.2 General SNR-Constrained Compressors

To facilitate the discussions of our CFedAvg algorithm, we will first formally define the notion of *general SNR (signal-to-noise ratio)-constrained compressors*, which has been used in the literature (see, e.g., [15, 32]):

Definition 1. *(General SNR-Constrained Compressor) An operator $C(\cdot) : \mathbb{R}^d \to \mathbb{R}^d$ is said to be constrained by an SNR threshold $\gamma \geq 1$ if it satisfies:*

$$\mathbb{E}_C\|C(\mathbf{x}) - \mathbf{x}\|^2 \leq (1/\gamma)\|\mathbf{x}\|^2, \quad \forall \mathbf{x} \in \mathbb{R}^d.$$

It is clear from Definition 1 that, for a given compressor, γ is its lowest SNR guarantee yielded by its largest compression noise power $\|C(\mathbf{x}) - \mathbf{x}\|^2$. The γ-threshold can be viewed as a proxy of compression rate of the compressor. For $\gamma = \infty$, we have $C(\mathbf{x}) = \mathbf{x}$, which means no compression and zero information loss. On the other hand, $\gamma \to 1$ implies that the compression rate is arbitrarily high and the output contains no information of $\mathbf{x}$. It is worth pointing out that, in Definition 1, the SNR-constrained compressor is not assumed to be unbiased, hence the term "general". Definition 1 covers a large class of compression schemes, e.g., the Top-k compressor [1, 24] that selects k coordinates with the largest absolute values, and the random sparsifier [39] that randomly selects components.

4 COMPRESSED FEDAVG (CFEDAVG) FOR NON-IID FEDERATED LEARNING

In this section, we will first introduce our CFedAvg (compressed FedAvg) algorithm in Section 4.1. Then, we will present the main theoretical result and their key insights/interpretations in Section 4.2. Due to space limitation, we provide proof sketches for the main results in Section 4.3 and relegate the full proofs of all theoretical results in our online technical report [42].

4.1 The CFedAvg Algorithmic Framework

The general CFedAvg algorithmic framework is stated in Algorithm 1. We aim to not only reduce the total communication rounds, but also compress the gradients transmitted in each communication round. The algorithm contains four key stages:

Algorithm 1 The General CFedAvg Algorithmic Framework.

1: Initialize $\mathbf{x}_0$.
2: **for** $t = 0, \cdots, T-1$ **do**
3: Initialize $\mathbf{e}_0^i = 0, i \in [m]$ if $t = 0$.
4: **for** each worker $i \in [m]$ in parallel **do**
5: $\mathbf{x}_{t,0}^i = \mathbf{x}_t$
6: **for** $k = 0, \cdots, K_i - 1$ **do**
7: Compute an unbiased stochastic gradient estimate $\mathbf{g}_{t,k}^i = \nabla F_i(\mathbf{x}_{t,k}^i, \xi_{t,k}^i)$ of $\nabla F_i(\mathbf{x}_{t,k}^i)$.
8: Local update: $\mathbf{x}_{t,k+1}^i = \mathbf{x}_{t,k}^i - \eta_{L,t}\mathbf{g}_{t,k}^i$.
9: **end for**
10: For Homogeneous Local Steps: $\mathbf{g}_t^i = \mathbf{x}_{t,K}^i - \mathbf{x}_t$.
11: For Heterogeneous Local Steps: $\mathbf{g}_t^i = \frac{1}{K_i}(\mathbf{x}_{t,K_i}^i - \mathbf{x}_t)$.
12: $\mathbf{p}_t^i = \mathbf{g}_t^i + \mathbf{e}_t^i$.
13: $\tilde{\Delta}_t^i = C(\mathbf{p}_t^i)$ ($C(\cdot)$ is an SNR-constrained compressor).
14: Send $\tilde{\Delta}_t^i$ to server.
15: $\mathbf{e}_{t+1}^i = \mathbf{p}_t^i - \tilde{\Delta}_t^i$.
16: **end for**
At Parameter Server:
17: Receive $\tilde{\Delta}_t^i, i \in [m]$.
18: $\tilde{\Delta}_t = \frac{1}{m}\sum_{i\in[m]} \tilde{\Delta}_t^i$.
19: Server Update: $\mathbf{x}_{t+1} = \mathbf{x}_t + \eta\tilde{\Delta}_t$.
20: Broadcast $\mathbf{x}_{t+1}$ to each worker.
21: **end for**

1. *Local Computation:* Line 6 says that, in each communication round, each worker runs K_i local updates before communicating with the parameter server. As shown in Line 7, each local update step takes an unbiased gradient estimator (e.g., vanilla SGD). A local learning rate $\eta_{L,t}$ is adopted for each local step (Line 8).
2. *Gradient Compression:* We compress the model changes $\mathbf{g}_t^i$ instead of the last model $\mathbf{x}_{t,K_i}^i$ in each worker, where $\mathbf{g}_t^i = \mathbf{x}_{t,K}^i - \mathbf{x}_t$ for homogeneous local step ($K_i = K, \forall i \in [m]$) in Line 10 or $\mathbf{g}_t^i = \frac{1}{K_i}(\mathbf{x}_{t,K_i}^i - \mathbf{x}_t)$ for heterogeneous local step (different local steps $K_i, \forall i \in [m]$) in Line 11. Before compressing the parameters, we add the error term to compensate the parameter in each worker in Line 12, i.e., $\mathbf{p}_t^i = \mathbf{g}_t^i + \mathbf{e}_t^i, \forall i \in [m]$. Then, we compress the parameter $\mathbf{p}_t^i$ and send the result $\tilde{\Delta}_t^i = C(\mathbf{p}_t^i)$ to the server, where $C(\cdot)$ denotes a general SNR-constrained compressor.
3. *Error Feedback:* We update the error term after gradient compression in each communication round in Line 15, which represents the information loss due to compression. This would be used later to compensate the parameters in the next communication round to ensure not too much parameter information is lost.
4. *Global Update:* Upon the reception of all returned parameters, the parameter server updates the parameters using a global learning rate η and broadcasts the new model parameters to all workers.

4.2 Main Theoretical Results

In this subsection, we will establish the convergence results of our proposed CFedAvg algorithmic framework. Our convergence results are proved under the following mild assumptions:

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

Assumption 2. *(Unbiased Local Gradient Estimator) Let ξ_t^i be a random local sample in the t-th round at worker i. The local gradient estimator is unbiased, i.e., $\mathbb{E}[\nabla F_i(\mathbf{x}_t, \xi_t^i)] = \nabla F_i(\mathbf{x}_t)$, $\forall i \in [m]$.*

Assumption 3. *(Bounded Local and Global Variance) There exist two constants $\sigma_L > 0$ and $\sigma_G > 0$, such that the variance of each local gradient estimator is bounded by $\mathbb{E}[||\nabla F_i(\mathbf{x}_t, \xi_t^i) - \nabla F_i(\mathbf{x}_t)||^2] \leq \sigma_L^2$, $\forall i \in [m]$, and the global variability of the local gradient of the cost function is bounded by $\|\nabla F_i(\mathbf{x}_t) - \nabla f(\mathbf{x}_t)\|^2 \leq \sigma_G^2$, $\forall i \in [m]$.*

The first two assumptions and the bounded local variance assumption in Assumption 3 are standard assumptions in the convergence analysis of stochastic gradient-type algorithms in non-convex optimization (e.g.,[6, 9]). We use a universal bound σ_G to quantify the heterogeneity of the *non-i.i.d. datasets* among different workers. Note that $\sigma_G = 0$ corresponds to i.i.d. datasets. This assumption has also been used in other works for FL with non-i.i.d. datasets [26, 38, 44] as well as in decentralized optimization [13]. It is worth noting that we do *not* require a bounded gradient assumption, which is often used in FL optimization analysis [13]. With the above assumptions, we are now in a position to present our main theoretical results. First, we state a useful result in Lemma 4.1:

Lemma 4.1. *(Bounded Error). For any local learning rate satisfying $\eta_{L,t} \leq \frac{1}{8LK}$, the error term can be upper bounded by $\sum_{t=0}^{T-1} \|\mathbf{e}_t\|^2 \leq h(\gamma) \sum_{t=0}^{T-1} \|\Delta_t\|^2$, where $\mathbf{e}_t = \frac{1}{m}\sum_{i\in[m]} \mathbf{e}_t^i$, $\Delta_t = \frac{1}{m}\sum_{i\in[m]} \mathbf{g}_t^i$, $h(\gamma) = (1/\gamma)(1+1/a)b$, and a and b are constants such that $\gamma\epsilon - 1 \geq a$ and $\frac{\alpha_t}{1-\epsilon} \leq b$ for $\epsilon \in (0,1)$.*

Lemma 4.1 implies that the error term cannot grow arbitrarily large with a proper SNR threshold γ (determined by compression rate) for which $h(\gamma)$ does not go to infinity. The error is upper bounded by the accumulated parameters Δ_t and the SNR threshold γ. This suggests that the total information loss due to compression is only a fraction of the total of the accumulated parameters.

1) CFedAvg with Constant Learning Rates (Homogeneous Local Steps): As a first step, we consider the simpler case where CFedAvg uses constant learning rates (i.e., $\eta_{L,t} \equiv \eta_L$, $\forall t$) and homogeneous local steps (i.e., $K_i \equiv K$, $\forall i$). In this case, by Lemma 4.1, we can establish the convergence result as follows:

Theorem 4.2. *(Convergence Rate of CFedAvg). Choose constant local and global learning rates η_L and η such that $\eta_L \leq \frac{1}{8LK}$, $\eta\eta_L < \frac{1}{KL}$ and $\eta\eta_L K(L^2 h(\gamma) + 1 + L) \leq 1$. Under Assumptions 1–3, the sequence of outputs $\{\mathbf{x}_t\}$ generated by Algorithm 1 satisfies:*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L KT} + \Phi, \tag{2}$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2 h(\gamma) + \frac{1}{2} + \frac{L}{2})\frac{\eta\eta_L}{m}\sigma_L^2 + \frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)]$, c is a constant, $h(\gamma)$ is defined the same as that in Lemma 4.1, $f_0 = f(\hat{\mathbf{x}}_0)$ and $f_* = f(\mathbf{x}_*)$.

Remark 1 (Decomposition of The Bound). The bound of the convergence rate in (2) contains two terms: the first term is a vanishing term $\frac{f_0 - f_*}{c\eta\eta_L KT}$ as T increase, and the second term is a constant Φ independent of T. Note that Φ depends on three factors: local variance σ_L, global variance σ_G, and the number of local steps K. We can further decompose the constant term Φ into two parts. The first part of Φ is due to the local variance of the stochastic gradient in each local SGD step for each worker. It shrinks at the rate $\frac{1}{m}$ with respect to the number of workers m, which favors large distributed systems. This makes intuitive sense since more workers means more training samples in one communication round, thus decreasing the local variance due to stochastic gradients. It can also be viewed as having a larger batch size to decrease the variance in SGD. The cumulative variance of K local steps contributes to the second term of Φ. This term depends on the number of local steps K, local learning rate η_L, local variance σ_L^2 and global variance σ_G^2 (data heterogeneity), but independent of the number of workers m.

Remark 2 (Comparison with FedAvg without Compression). Compared to the results of generalized FedAvg without compression, i.e., $\Phi \triangleq \frac{1}{c}[\frac{L\eta\eta_L}{2m}\sigma_L^2 + \frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)]$ (cf. [3]), we have two key observations. First, the compressor, which significantly reduces the communication cost, only slightly increases the constant Φ by $\frac{1}{c}[(\frac{1}{2}L^2\eta_L^2 h(\gamma) + \frac{1}{2})\frac{\eta_L}{m}\sigma_L^2]$ and does not change the convergence rate $O(1/T)$. This extra variance comes from increased local variance due to more noisy stochastic gradients after compression, and is independent of the number of local steps K. This insight means that one can *safely* use more local steps K without worrying about any accumulative effect due to compression, which is a somewhat surprising and counter-intuitive insight. On the other hand, this extra variance shrinks at rate $\frac{1}{m}$, which favors large FL systems with more workers. This makes intuitive sense since the server could obtain more information from the model updates with more workers, although each worker's information is noisy due to compression. In other words, as the number of worker m increases, the extra variance due to compression becomes negligible.

Second, the extra variance due to compression is irrelevant to the global variance σ_G^2 from the non-i.i.d. datasets. The global variance measures the heterogeneity among the loss functions of the workers with non-i.i.d. local datasets. Intuitively speaking, the compressor only introduces extra noise to the model's information. Thus, the compression operation only increases the local variance of the stochastic gradient and has nothing to do with the non-i.i.d. datasets and how many local steps taken in FL.

Remark 3 (Choice of Compressor). Our analysis also shows that many compression methods (e.g., [2, 5, 39]) that work well in traditional distributed and decentralized learning can also be used with the CFedAvg algorithm in FL. With error feedback, CFedAvg enjoys the same benefits as traditional distributed learning even with the use of local steps in FL and under non-i.i.d. datasets. Moreover, we do not restrict the choice of compressors. Thus, CFedAvg works with both biased and unbiased compressors, as long as they satisfy Definition 1. However, care must still be taken when one chooses a compressor in CFedAvg since it does not mean any compression methods with arbitrary compression rate could work. Consider a compressor with arbitrary compression rate such that $\gamma \to 1$ and then $\epsilon \to 1, b \to \infty$ so $h(\gamma) \to \infty$. This compressed model is too noisy to be trained since no useful information is transmitting to the parameter server. This will also be empirically verified in Section 5, where significant performance degradation due to a too aggressive compression rate can be observed. Note

also that our results are directly applicable to the i.i.d. setting for $\sigma_G = 0$.

From Theorem 4.2, we immediately have the following convergence rate with a proper choice of learning rates, which further implies a *linear speedup effect* for convergence:

Corollary 4.3. *(Linear Speedup for Convergence). If* $\eta_L = \frac{1}{\sqrt{T}KL}$ *and* $\eta = \sqrt{Km}$*, the outputs* $\{\mathbf{x}_t\}$ *generated by Algorithm 1 satisfies:* $\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 = O(\frac{h(\gamma)}{\sqrt{mKT}} + \frac{1}{T}) = O(\frac{1}{\sqrt{mKT}} + \frac{1}{T})$.

Remark 4. With a proper SNR-threshold γ such that $h(\gamma) = O(1)$, CFedAvg achieves a linear speedup for convergence for non-i.i.d. datasets, i.e., $O(\frac{1}{\sqrt{mKT}})$ convergence rate for $T \geq mK$. This matches the convergence rate in distributed learning and FL without compression [3, 13, 14, 43], which indicates that CFedAvg achieves high communication efficiency while not sacrificing learning accuracy in FL. When degenerating to i.i.d. case, CFedAvg still achieves the linear speedup effect, matching the results of previous work in distributed and decentralize learning [17, 35]. The most related work to this paper is Qsparse-local-SGD [4] which combines unbiased quantization, sparsification and local steps together and is able to recover or generalize other compression methods. It achieves $O(\frac{1}{\sqrt{mKT}} + \frac{mK}{T})$ convergence, implying a linear speedup for $T \geq m^3K^3$. However, for large systems (m) and large local steps (K), their T will be very large. Besides a better convergence rate in this paper, we have a weak assumption (no bounded gradient assumption). The challenges arising from the weak assumption is addressed by new approaches (see proofs in Section 4.3 and [42]).

2) CFedAvg with Decaying Learning Rates (Homogeneous Local Steps): We can see from Corollary 4.3 that the choice of constant learning rate requires the knowledge of time horizon T before running the algorithm, which may not be available in practice. In other words, the constant-learning-rate version of CFedAvg is not an *"anytime"* algorithm. To address this limitation, we propose CFedAvg with decaying learning rate, which is an anytime algorithm.

Theorem 4.4. *(Convergence with Decaying Learning Rate). Choose decaying local learning rate* $\eta_{L,t}$ *and constant global learning rates* η *such that* $\eta_{L,t} \leq \frac{1}{8LK}$, $\eta\eta_{L,t} < \frac{1}{KL}$ *and* $\eta\eta_{L,t}K(L^2h(\gamma) + 1 + L) \leq 1, \forall t \in [T]$*. Under Assumptions 1–3, the sequence of outputs* $\{\mathbf{x}_t\}$ *generated by Algorithm 1 satisfies:*

$$\mathbb{E}\|\nabla f(\mathbf{z})\|_2^2 \leq \frac{f_0 - f_*}{c\eta K H_T} + \Phi,$$

where $\Phi \triangleq (\frac{1}{2}L^2h(\gamma) + \frac{1}{2} + \frac{L}{2})\frac{\eta}{cmH_T}\sigma_L^2 \sum_{t=0}^{T-1}\eta_{L,t}^2 + \frac{5KL^2}{2cH_T}(\sigma_L^2 + 6K\sigma_G^2)\sum_{t=0}^{T-1}\eta_{L,t}^3$, $H_T = \sum_{t=0}^{T-1}\eta_{L,t}$ *and* $\mathbf{z}$ *is sampled from* $\{\mathbf{x}_t\}, \forall t \in [T]$ *with probability* $\mathbb{P}[\mathbf{z} = \mathbf{x}_t] = \frac{\eta_{L,t}}{H_T}$*. Here* c *is a constant,* $h(\gamma)$ *is defined the same as that in Lemma 4.1,* $f_0 = f(\hat{\mathbf{x}}_0)$ *and* $f_* = f(\mathbf{x}_*)$.

Based on Theorem 4.4, the following result immediately follows:

Corollary 4.5. *Let* $\eta_L = \frac{1}{\sqrt{t+a}KL}$ *for some constant* $a > 0$ *and let* $\eta = \sqrt{Km}$*. The outputs* $\{\mathbf{x}_t\}$ *generated by Algorithm 1 satisfies:* $\mathbb{E}\left\|\nabla f(\mathbf{z})\right\|^2 = \tilde{O}(\frac{1}{\sqrt{mKT}}) + O(\frac{1}{\sqrt{T}})$.

Remark 5. When $h(\gamma)$ is a constant, our CFedAvg algorithm achieves an $O(\frac{1}{\sqrt{mKT}}\ln(T) + \frac{1}{\sqrt{T}})$ convergence rate, which is slower than that with constant learning rates. However, it is still better than $O(\frac{1}{\ln(T)})$ proved in Qsparse-local-SGD [4].

3) CFedAvg with Heterogeneous Local Steps (Constant Learning Rates): In practice, FL systems are often formed by heterogeneous devices with various computing capabilities and resource limits (e.g., computation speed, memory size). Hence, fixing the same number of local steps at all workers results in: i) fast workers are idling after finishing computation in each round and thus wasting resources and ii) slow workers become the stragglers in the FL system. Next, we show it is possible to perform heterogeneous local steps among workers in CFedAvg while still offering theoretical performance guarantees.

Theorem 4.6. *(Convergence of Heterogeneous Local Steps). Choose constant local and global learning rates* η_L *and* η *such that* $\eta_L \leq \frac{1}{8LK_i}$, $\eta\eta_L < \frac{1}{K_iL}, \forall i \in [m]$ *and* $\eta\eta_L(L^2h(\gamma) + 1 + L) \leq 1$*. Under Assumptions 1–3, the sequence of outputs* $\{\mathbf{x}_t\}$ *generated by Algorithm 1 with heterogeneous local steps satisfies:*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L T} + \Phi,$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2h(\gamma) + \frac{1}{2} + \frac{L}{2})\frac{\eta\eta_L}{m^2}\sum_{i=1}^m \frac{1}{K_i}\sigma_L^2 + \frac{5\eta_L^2L^2}{2}\frac{1}{m}\sum_{i=1}^m K_i(\sigma_L^2 + 6K_i\sigma_G^2)]$*,* c *is a constant,* $h(\gamma)$ *is defined the same as that in Lemma 4.1,* $f_0 = f(\hat{\mathbf{x}}_0)$ *and* $f_* = f(\mathbf{x}_*)$.

Based on Theorem 4.6, the following result immediately follows:

Corollary 4.7. *(Linear Speedup with Heterogeneous Local Steps). Let* $\eta_L = \frac{1}{\sqrt{TL}}$ *and* $\eta = \sqrt{K_{min}m}$*. The convergence rate of Algorithm 1 with heterogeneous local steps and constant learning rate is*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 = O(\frac{1}{\sqrt{mK_{min}T}}) + O(\frac{K_{max}^2}{T}),$$

where $K_{min} = \min\{K_i, \forall i \in [m]\}$ *and* $K_{max} = \max\{K_i, \forall i \in [m]\}$.

Remark 6. Allowing heterogeneous local steps at different workers entails efficient FL implementations in practice. Specifically, instead of waiting for all workers to finish the same number of local steps, the server can broadcast a periodic "time-out" signal to all workers to interrupt their local updates. All worker can simply submit their current computation results even if they are in different stages in their local updates. We can see from Corollary 4.7 that CFedAvg with heterogeneous local steps can still achieve the same linear speedup for convergence $O(\frac{1}{\sqrt{mK_{min}T}})$ for $T \geq mK_{min}K_{max}^4$, while having the flexibility of choosing different number of local steps at each worker. Although this result is for CFedAvg, our theoretical analysis is general and can be applied to uncompressed FL algorithms (e.g., FedAvg) to allow heterogeneous $K_i^t, \forall i \in [m]$.

To our knowledge, our work is the first to show that performing heterogeneous local steps among workers still achieves theoretical performance guarantees for FL. We note that, in asynchronous Qsparse-local-SGD [4], the workers synchronize with the server at different times based on the workers but require each worker follows the same rate and performs the same local steps. Rizk et al. [27] proposed dynamic federated learning (without compression)

to use heterogeneous local steps at each worker. But they require the local steps to be known in advance in order to scale the gradient in each local step. For our CFedAvg, K_i can be set in an ad-hoc fashion without being known in advance.

4.3 Proof of the Main Results

Due to space limitation, we provide a proof sketch for Theorem 4.2, which is the foundation of other convergence results in this paper. The complete proofs of all theorems and corollaries can be found in our online technical report [42].

Proof Sketch for Theorem 4.2. For convenience, we define the following notation: $\mathbf{e}_t \triangleq \frac{1}{m}\sum_{i=1}^m \mathbf{e}_t^i$, $\hat{\mathbf{x}}_t \triangleq \mathbf{x}_t + \eta\mathbf{e}_t$ and $\Delta_t = \frac{1}{m}\sum_{i\in[m]}\Delta_t^i = \frac{1}{m}\sum_{i\in[m]}\mathbf{g}_t^i$. As mentioned earlier, a key feature of our analysis in this paper is that we do not assume bounded gradients. This is because the bounded gradient assumption may not be appropriate for non-i.i.d. datasets, which is our main focus in this paper. However, relaxing this assumption poses two challenges. One key difficulty is to bound the error feedback term $\mathbf{e}_t^i, \forall i \in [m]$ due to the compression. With bounded gradient assumption, each $\mathbf{e}_t^i$ is bounded individually based on the gradient during the local steps [4, 31]. However, $\mathbf{e}_t^i$ cannot be bounded in this fashion if the bounded gradient assumption is relaxed. In our analysis, we manage to bound $\mathbf{e}_t$ rather than $\mathbf{e}_t^i$ individually. By using $\|\mathbf{e}_t\|^2 = \left\|\frac{1}{m}\sum_{i=1}^m \mathbf{e}_t^i\right\|^2 \le \frac{1}{m}\sum_{i=1}^m \left\|\mathbf{e}_t^i\right\|^2 = \left\|\bar{\mathbf{e}}_t\right\|^2$ and the recursion $\left\|\bar{\mathbf{e}}_t\right\|^2 \le (1/\gamma)\frac{1}{\epsilon}\left\|\bar{\mathbf{e}}_{t-1}\right\|^2 + (1/\gamma)\frac{1}{1-\epsilon}\left\|\bar{\Delta}_{t-1}\right\|^2$, where $\|\bar{\Delta}_t\|^2 = \frac{1}{m}\sum_{i=1}^m \left\|\Delta_t^i\right\|^2$, $\sum_{t=0}^{T-1}\|\mathbf{e}_t\|^2$ can be bounded in Lemma 4.1. Second, the lack of bounded gradient assumption also results in difficulty to bound the model drift stemming from the non-i.i.d. datasets and the increase of the local steps, i.e., $\|\mathbf{x}_i - \bar{\mathbf{x}}\|, \forall i \in [m]$, where $\bar{\mathbf{x}} = \frac{1}{m}\sum_{i=1}^m \mathbf{x}_i$. To address this challenge, we derive the recursive relation based on communication round instead of local steps. Thanks to the virtual variable $\hat{\mathbf{x}}_t$, we have the recursive relation of $\hat{\mathbf{x}}_t$: $\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \eta\Delta_t$, where t is the index of communication round.

After addressing these two challenges, Assumption 1 yields per-communication-round descent as follows:

$$\begin{aligned}\mathbb{E}_t f(\hat{\mathbf{x}}_{t+1}) - f(\hat{\mathbf{x}}_t) &\le \mathbb{E}_t\left\langle \nabla f(\hat{\mathbf{x}}_t), [\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t]\right\rangle + \frac{L}{2}\mathbb{E}_t\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\ &= \underbrace{\mathbb{E}_t\left\langle \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t\right\rangle}_{A_1} + \frac{L\eta^2}{2}\underbrace{\mathbb{E}_t\|\Delta_t\|^2}_{A_2}. \end{aligned} \tag{3}$$

For A_1, the first step is to transform variable $\hat{\mathbf{x}}_t$ to $\mathbf{x}_t$ since only $\mathbf{x}_t$ is involved in the update process. Towards this end, we have:

$$\begin{aligned} A_1 =& \mathbb{E}_t\left\langle \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t\right\rangle \\ =& \mathbb{E}_t\left[\left\langle \nabla f(\hat{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t), \eta\Delta_t\right\rangle + \eta\left\langle \nabla f(\mathbf{x}_t), \Delta_t\right\rangle\right] \\ \le& \mathbb{E}_t\Big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2 - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 \\ &+ \eta\left\langle \nabla f(\mathbf{x}_t), \Delta_t + K\eta_L\nabla f(\mathbf{x}_t)\right\rangle\Big]. \end{aligned} \tag{4}$$

Further bounding the last term of equation (4) and rearranging, we simplify the above inequality as follows:

$$\begin{aligned} A_1 \le& \mathbb{E}_t\Big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2\Big] - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 \\ &+\eta\eta_L K(\frac{1}{2} + 15K^2\eta_L^2L^2)\|\nabla f(x_t)\|^2 + \frac{5\eta K^2\eta_L^3L^2}{2}\times \\ &(\sigma_L^2 + 6K\sigma_G^2) - \frac{\eta\eta_L}{2Km^2}\mathbb{E}_t\Big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\Big\|^2. \end{aligned} \tag{5}$$

For A_2, by using $\mathbb{E}[\|\mathbf{x}\|^2] = \mathbb{E}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|^2] + \|\mathbb{E}[\mathbf{x}]\|^2$ to decompose Δ_t and assumption 3, we have:

$$A_2 = \mathbb{E}_t[\|\Delta_t\|^2] \le \frac{K\eta_L^2}{m}\sigma_L^2 + \frac{\eta_L^2}{m^2}\Big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\Big\|^2. \tag{6}$$

Combining (5) and (6), the term $\mathbb{E}_t\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2$ is canceled with a proper learning rate. Then, we can simplify (8) as:

$$\begin{aligned}\mathbb{E}f(\hat{\mathbf{x}}_{t+1}) - \nabla f(\hat{\mathbf{x}}_t) \le& (\frac{1}{2}L^2\eta^2h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\frac{K\eta_L^2}{m}\sigma_L^2 \\ &- c\eta\eta_L K\big\|\nabla f(x_t)\big\|^2 + \frac{5\eta K^2\eta_L^3L^2}{2}(\sigma_L^2 + 6K\sigma_G^2).\end{aligned}$$

By Lemma 4.1, telescoping and rearranging, we have the convergence bound stated in Theorem 4.2. □

Theorems 4.4 and 4.6 can be proved following the similar approach and with more careful examinations on the effects of decaying learning rates and heterogeneous local steps. Due to space limitation, we relegate the full proofs of Theorems 4.4 and 4.6 to our online technical report [42].

5 EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to verify our theoretical results with a variety of datasets and compression methods.

1) Experiment Settings: *1-a) Datasets and Models:* We use three datasets (non-i.i.d. version) in FL settings, including MNIST [19], Fashion-MNIST [41] and CIFAR-10 [18]. Each of these three datasets contains 10 different classes of items. To induce non-i.i.d. datasets, we partition the data based on the classes of items (p) contained in these datasets. We distribute the dataset among $m = 100$ workers randomly and evenly in a class-based manner, such that the local dataset at each worker contains only a subset of classes of items with the same number of training/test samples. For example, for $p = 1$, each worker only has training/testing samples from one particular class, which induces heterogeneity among different workers. For $p = 10$, each worker has samples from 10 classes, which is essentially an i.i.d. setting, since the total number of classes for these three datasets are all 10. By doing so, we can use the number of classes in worker's local dataset, denoted as p, to control the non-i.i.d. degree of the datasets quantitatively. We set four levels of non-i.i.d. datasets for comparison: $p = 1, 2, 5$, and 10. We experiment two learning models: i) convolution neural network (CNN) (architecture is detailed in our online technical report) on MNIST and Fashion-MNIST, and ii) ResNet-18 [10] on CIFAR-10.

1-b) Compressors: We consider two compression methods: i) Top-k sparsification and ii) random dropping. These two compression methods have been empirically proved to be effective in distributed and decentralized learning.

1) *Top-k Sparsification [32]:* For a given vector $\mathbf{x}$, Top-K sparsification compresses $\mathbf{x}$ by retaining the k elements of this vector that have the largest absolute value and setting others to zero. Specifically, for vector $\mathbf{x} = [x_1, \cdots, x_d]^T \in \mathbb{R}^d$, $C(\mathbf{x})$ outputs a sparse vector with i-th coordinate $[C(\mathbf{x})]_i$ determined as follows:

$$\begin{cases} [C(\mathbf{x})]_i = \mathbf{x}_i, & i \in S(k), \\ [C(\mathbf{x})]_i = 0, & i \notin S(k), \end{cases}$$

where $S(k)$ is the index set whose corresponding values are the largest k elements in vector $abs(\mathbf{x})$. In this paper, we use a constant compression parameter $comp \in (0, 1]$. The corresponding size is $k = \lfloor d \times (1 - comp) \rfloor$. For example, for a d-dimensional vector and a compression method with $comp = 0.9$, the compressor picks a set S that contains only $\lfloor 0.1 \times d \rfloor$ elements, which means it can save about 90% of the bandwidth in each communication round. Therefore, we can use this parameter $comp$ to control the compression rate of the compression method.

2) *Random Dropping [1, 32]:* For a given vector, random dropping randomly drops each component with a fixed probability. That is, for a vector $\mathbf{x} = [x_1, \cdots, x_d]^T \in \mathbb{R}^d$ and a constant compression parameter $comp \in (0, 1]$, $C(\mathbf{x})$ outputs a sparse vector with the i-th coordinate $[C(\mathbf{x})]_i$ following a Bernoulli distribution:

$$\begin{cases} \mathbb{P}([C(\mathbf{x})]_i = \mathbf{x}_i) = 1 - comp, \\ \mathbb{P}([C(\mathbf{x})]_i = 0) = comp. \end{cases}$$

It is clear that both compressors satisfy Definition 1 for $\gamma = 1/comp$. However, the bound for Top-k is not tight, i.e., Top-k could contain more information than random dropping with the same amount of coordinates, since Top-k selects the largest elements with the same compression rate. This is verified in our numerical results that Top-k has better performance with the same $comp$.

1-c) Hyper-parameters: We set the default hyper-parameters as follows: the number of workers $m = 100$, local learning rate $\eta_L = 0.1$, global learning rate $\eta = 1.0$, batch size $B = 64$, local steps $K = 10$ epochs, communication rounds $T = 100$ for MNIST and Fashion-MNIST and $T = 200$ for CIFAR-10.

2) Numerical Results: We now present two types of experimental results. The first type is to show the effectiveness of our CFedAvg algorithm with significant communication cost reduction. The second type is to evaluate the importance of error feedback. We only show a part of the results here due to space limitation. Further results can be found in our online technical report [42].

2-a) Effectiveness of Compressors: As shown in Fig. 1, for the CNN model on different non-i.i.d. MNIST datasets, the figures in the left column are for training loss versus communication rounds and the figures in the right column are for test accuracy versus communication rounds. We can see that our CFedAvg algorithm with two compressors converges for all heterogeneity levels of non-i.i.d. datasets from top ($p = 10$) to bottom ($p = 1$). We can see that Top-k outperforms random dropping because the large coordinates may contain more information for the same communication load and the bound in Definition 1 may not be tight for Top-k. Note that it is not easy to get a tight SNR bound for Top-k since it varies case by case. Therefore, the impact of compressors on convergence rate depends on both the compression method and its compression rate parameter $comp$. For the random dropping (RD) method, it becomes worse

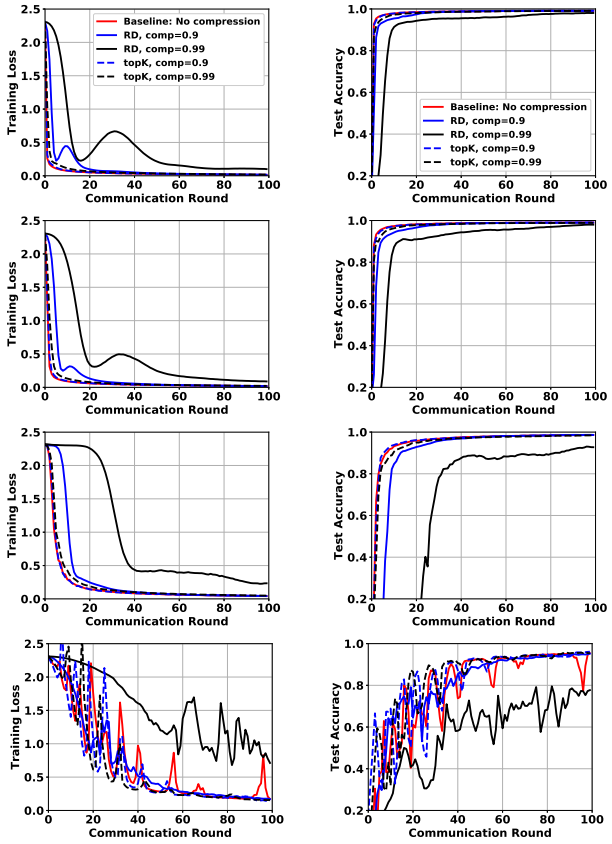
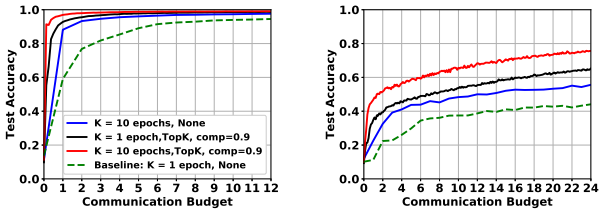


Figure 1: Training loss (left) and test accuracy (right) for the CNN model for MNIST. The non-i.i.d. levels are $p = 10, 5, 2, 1$ from top to bottom.



(a) MNIST. (b) CIFAR-10.

Figure 2: Comparison of test accuracy with same communication load for $p = 5$. The unit of the communication budget is the model size.

from $comp = 0.9$ to $comp = 0.99$. While for Top-k method, both cases ($comp = 0.9$ and $comp = 0.99$) can achieve almost the same convergence speed comparable to that without any compression. This indicates that we can reduce the communication cost in each communication round by about 99% with Top-k for MNIST, i.e., we only need to transmit 1% of coordinates in the gradient vector without significantly sacrificing the convergence rate. This will greatly facilitates FL on such communication-constrained devices.

Another interesting observation is that the compression methods (with error feedback) help stabilize the training process for non-i.i.d. case. Compared with i.i.d. datasets, the training curve is zigzagging

for non-i.i.d. case. As the heterogeneity level of non-i.i.d. datasets increases, this zigzagging phenomena of the curves is more pronounced as shown in the figures, which we believe is an inherent feature of non-i.i.d. dataset in FL. Meanwhile, the learning curves are smoother with compression and error feedback, particularly with highly non-i.i.d. datasets. For example, for RD with $comp = 0.9$ and Top-k with $comp = 0.99$, a slightly better convergence curve than that of the original FedAvg without any compression is shown in Figure 1 ($p = 1$). The intuition is that the compressor could filter some noises that lead to the instability of the learning curve due to model heterogeneity among workers originated from the non-i.i.d. datasets and local steps. However, this appears to require proper compression rate $comp$ based on the compression method. We do not rule out the possibility of mutual interactions that lead to poor performance between the compression and non-i.i.d. datasets. If a compressor drops too much useful information in the parameters, it could be impossible to train a model effectively in general. For example, for RD with $comp = 0.99$, the learning curve is more winding as shown in Figure 1.

In addition, we compare the test accuracy based on the same uplink communication budgets (the model size as the unit) among different compression methods under non-i.i.d. datasets $p = 5$ in Figure 2. The baseline is the FedAvg with one local steps and no compression. Consistent with previous work [24, 25], both compressors and local update steps are effective to reduce communication cost. In our algorithm, we combine these two methods together and achieve a better result, which confirms our theoretical analysis in Section 4.2.

2-b) Importance of Error Feedback: Although it is a natural idea to apply those compression methods that have been proved to be useful in traditional distributed learning to FL, there could be a significant information loss if one uses these compressors naively. It has been shown that the learning performance is poor without error feedback in compression under i.i.d. datasets [15, 33]. This conclusion is confirmed in our experiments as shown in Figure 3, where we can observe the gap between cases with and without error feedback (EF) under i.i.d. case ($p = 10$). As the heterogeneity degree of non-i.i.d. datasets increases from $p = 10$ to $p = 1$, the gap becomes larger and is no longer negligible. It is obvious that both compression methods, RD and Top-k, perform better with error feedback in Figure 3 for $p = 1$. This indicates the significant impact of error feedback. If naively applying compression, a huge amount of information could be lost, thus resulting in poor performance.

With error feedback, the error term accumulates the information that is not transmitted to the parameter server in the current communication round and then compensates the gradients in the next communication round. This is verified in Figure 4, which shows the mean of gradient norm changes $\frac{1}{m}\sum_{i=1}^{m}\|\Delta_t^i\|^2$ and the error term $\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{e}_t^i\|^2$ for the total worker number $m = 100$. One key observation is that the error term is bounded under appropriate compression methods and compression rates, which is usually several times of the gradient change term in general. Under the same condition, the error term is much larger with aggressive compression rate and RD usually has a larger error term than that of Top-k. However, with too aggressive compression as shown in Figure 4 for RD with $comp = 0.99$, the error term continues to grow. Thus, in

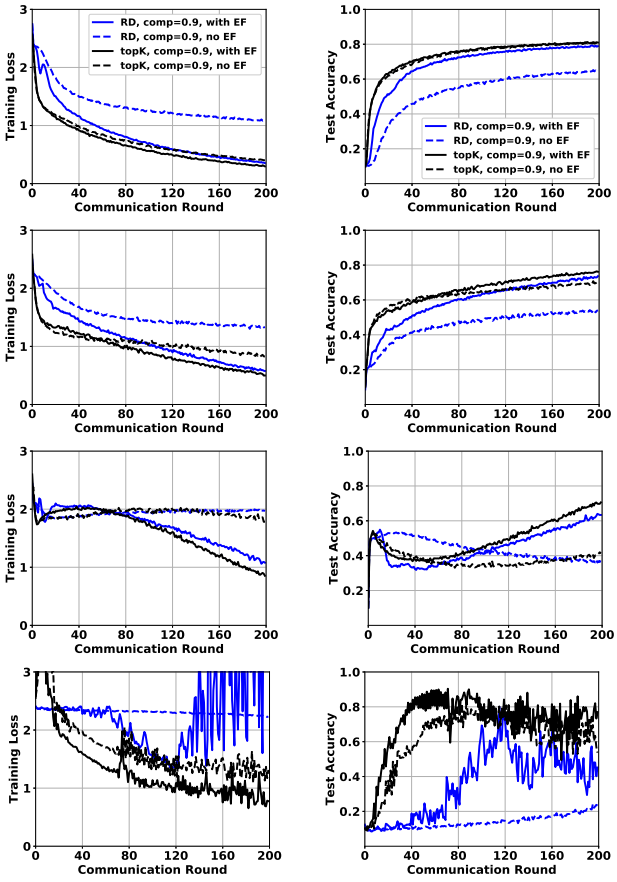
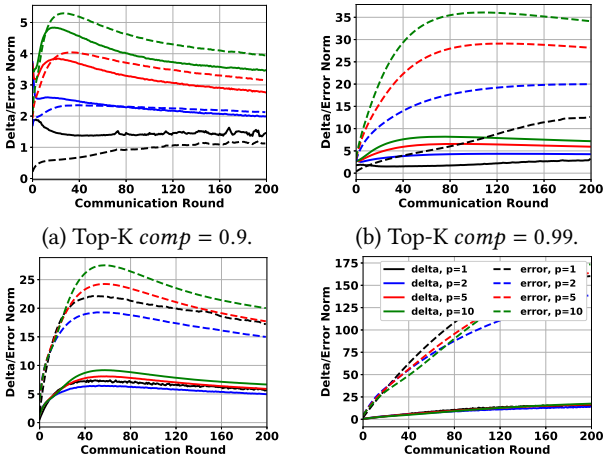


Figure 3: Training loss (left) and test accuracy (right) for ResNet-18 for CIFAR-10 with respect to error feedback (EF).



(a) Top-K $comp = 0.9$. (b) Top-K $comp = 0.99$.

(c) RD $comp = 0.9$. (d) RD $comp = 0.99$.

Figure 4: Mean of the norms of $\frac{1}{m}\sum_{i=1}^{100}\|\Delta_t^i\|^2$ and the error term $\frac{1}{m}\sum_{i=1}^{100}\|\mathbf{e}_t^i\|^2$ for the ResNet-18 on CIFAR-10.

general, the error feedback guarantees that not too much information is dropped due to compression, which verifies our theoretical

analysis. It has been shown that error feedback is effective in distributed/decentralized learning [15, 17, 33]. In this paper, we show its effectiveness continues to hold in non-i.i.d. compressed FL.

6 CONCLUSION

In this paper, we proposed a communication-efficient algorithmic framework called CFedAvg for FL on non-i.i.d. datasets. CFedAvg works with general (biased/unbiased) SNR-constrained compressors to reduce the communication cost and other techniques to accelerate the training. Theoretically, we analyzed the convergence rates of CFedAvg for non-convex functions with constant learning and decaying learning rates. The convergence rates of CFedAvg match that of distributed/federated learning without compression, thus achieving high communication efficiency while not significantly sacrificing learning accuracy in FL. Furthermore, we extended CFedAvg to heterogeneous local steps with convergence guarantees, which allows different workers perform different local steps to better adapt to their own circumstances. The key observation in this paper is that the noise/variance introduced by compressors does not affect the overall convergence rate order for non-i.i.d. FL. We verified the effectiveness of our CFedAvg algorithm on three datasets with two gradient compression schemes of different compression ratios. Our results contribute to the first step toward developing advanced compression methods for communication-efficient FL.

REFERENCES

[1] Alham Fikri Aji and Kenneth Heafield. 2017. Sparse communication for distributed gradient descent. *arXiv preprint arXiv:1704.05021* (2017).

[2] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*. 1709–1720.

[3] Anonymous. 2021. Achieving Linear Speedup with Partial Worker Participation in Non-{IID} Federated Learning. In *Submitted to International Conference on Learning Representations*. https://openreview.net/forum?id=jDdzh5ul-d under review.

[4] Debraj Basu, Deepesh Data, Can Karakus, and Suhas N Diggavi. 2020. Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations. *IEEE Journal on Selected Areas in Information Theory* 1, 1 (2020), 217–226.

[5] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. 2018. signSGD: Compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434* (2018).

[6] Léon Bottou, Frank E Curtis, and Jorge Nocedal. 2018. Optimization methods for large-scale machine learning. *Siam Review* 60, 2 (2018), 223–311.

[7] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. 2012. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research* 13 (2012), 165–202.

[8] Nikoli Dryden, Tim Moon, Sam Ade Jacobs, and Brian Van Essen. 2016. Communication quantization for data-parallel training of deep neural networks. In *2016 2nd Workshop on Machine Learning in HPC Environments (MLHPC)*. IEEE, 1–8.

[9] Saeed Ghadimi and Guanghui Lan. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization* 23, 4 (2013), 2341–2368.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[11] Li Huang, Yifeng Yin, Zeng Fu, Shifa Zhang, Hao Deng, and Dianbo Liu. 2018. Loadaboost: Loss-based adaboost federated machine learning on medical data. *arXiv preprint arXiv:1811.12629* (2018).

[12] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. 2018. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479* (2018).

[13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* (2019).

[14] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J Reddi, Sebastian U Stich, and Ananda Theertha Suresh. 2019. SCAFFOLD: Stochastic controlled averaging for on-device federated learning. *arXiv preprint arXiv:1910.06378* (2019).

[15] Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian U Stich, and Martin Jaggi. 2019. Error feedback fixes signsgd and other gradient compression schemes. *arXiv preprint arXiv:1901.09847* (2019).

[16] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. 2019. Better Communication Complexity for Local SGD. *arXiv preprint arXiv:1909.04746* (2019).

[17] Anastasia Koloskova, Tao Lin, Sebastian U Stich, and Martin Jaggi. 2019. Decentralized deep learning with arbitrary communication compression. *arXiv preprint arXiv:1907.09356* (2019).

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

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[20] He Li, Kaoru Ota, and Mianxiong Dong. 2018. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE network* 32, 1 (2018), 96–101.

[21] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2018. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* (2018).

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

[23] Tao Lin, Sebastian U Stich, Kumar Kshitij Patel, and Martin Jaggi. 2018. Don't Use Large Mini-Batches, Use Local SGD. *arXiv preprint arXiv:1808.07217* (2018).

[24] Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J Dally. 2017. Deep gradient compression: Reducing the communication bandwidth for distributed training. *arXiv preprint arXiv:1712.01887* (2017).

[25] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. 2016. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629* (2016).

[26] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecny, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive Federated Optimization. *arXiv preprint arXiv:2003.00295* (2020).

[27] Elsa Rizk, Stefan Vlaski, and Ali H Sayed. 2020. Dynamic federated learning. *arXiv preprint arXiv:2002.08782* (2020).

[28] Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. 2019. Robust and communication-efficient federated learning from non-iid data. *IEEE transactions on neural networks and learning systems* (2019).

[29] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE internet of things journal* 3, 5 (2016), 637–646.

[30] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[31] Sebastian U Stich. 2018. Local SGD converges fast and communicates little. *arXiv preprint arXiv:1805.09767* (2018).

[32] Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. 2018. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*. 4447–4458.

[33] Sebastian U Stich and Sai Praneeth Karimireddy. 2019. The error-feedback framework: Better rates for SGD with delayed gradients and compressed communication. *arXiv preprint arXiv:1909.05350* (2019).

[34] Nikko Strom. 2015. Scalable distributed DNN training using commodity GPU cloud computing. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[35] Hanlin Tang, Xiangru Lian, Shuang Qiu, Lei Yuan, Ce Zhang, Tong Zhang, and Ji Liu. 2019. DeepSqueeze : Decentralization Meets Error-Compensated Compression. *arXiv* (2019), arXiv–1907.

[36] Jianyu Wang and Gauri Joshi. 2018. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. *arXiv preprint arXiv:1808.07576* (2018).

[37] Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. 2019. SlowMo: Improving Communication-Efficient Distributed SGD with Slow Momentum. *arXiv preprint arXiv:1910.00643* (2019).

[38] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. 2019. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1205–1221.

[39] Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. 2018. Gradient sparsification for communication-efficient distributed optimization. In *Advances in Neural Information Processing Systems*. 1299–1309.

[40] Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2017. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In *Advances in neural information processing systems*. 1509–1519.

[41] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017).
[42] Haibo Yang, Jia Liu, and Elizabeth S. Bentley. 2020. *CFedAvg: Achieving Efficient Communication and Fast Convergence in Non-IID Federated Learning*. Technical Report. The Ohio State Univeristy. https://kevinliu-osu-ece.github.io/publications/CFedAvg_TR.pdf
[43] Hao Yu, Rong Jin, and Sen Yang. 2019. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *arXiv preprint arXiv:1905.03817* (2019).
[44] Hao Yu, Sen Yang, and Shenghuo Zhu. 2019. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5693–5700.
[45] Xin Zhang, Minghong Fang, Jia Liu, and Zhengyuan Zhu. 2020. Private and Communication-Efficient Edge Learning: A Sparse Differential Gaussian-Masking Distributed SGD Approach. *arXiv preprint arXiv:2001.03836* (2020).
[46] Xin Zhang, Jia Liu, Zhengyuan Zhu, and Elizabeth S Bentley. 2020. Communication-efficient network-distributed optimization with differential-coded compressors. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 317–326.
[47] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. 2018. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582* (2018).
[48] Fan Zhou and Guojing Cong. 2017. On the convergence properties of a K-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012* (2017).

A APPENDIX I: PROOF

A.1 Proof of Lemma

Lemma 4.1. *(Bounded Error). For any local learning rate satisfying $\eta_{L,t} \le \frac{1}{8LK}$, the error term can be upper bounded by $\sum_{t=0}^{T-1} \|\mathbf{e}_t\|^2 \le h(\gamma) \sum_{t=0}^{T-1} \|\Delta_t\|^2$, where $\mathbf{e}_t = \frac{1}{m}\sum_{i\in[m]} \mathbf{e}_t^i$, $\Delta_t = \frac{1}{m}\sum_{i\in[m]} \mathbf{g}_t^i$, $h(\gamma) = (1/\gamma)(1+1/a)b$, and a and b are constants such that $\gamma\epsilon - 1 \ge a$ and $\frac{\alpha_t}{1-\epsilon} \le b$ for $\epsilon \in (0,1)$.*

Proof of Lemma 4.1. Let $\|\bar{\mathbf{e}}_t\|^2 = \frac{1}{m}\sum_{i=1}^m \left\|\mathbf{e}_t^i\right\|^2$ and $\|\bar{\Delta}_t\|^2 = \frac{1}{m}\sum_{i=1}^m \left\|\Delta_t^i\right\|^2$. First we have:

$$\|\mathbf{e}_t\|^2 = \Big\|\frac{1}{m}\sum_{i=1}^m \mathbf{e}_t^i\Big\|^2 \le \frac{1}{m}\sum_{i=1}^m \left\|\mathbf{e}_t^i\right\|^2 = \left\|\bar{\mathbf{e}}_t\right\|^2$$

That is, we have the bound of $\|\mathbf{e}_t\|^2$ once we can bound $\|\bar{\mathbf{e}}_t\|^2$.

$$\begin{aligned}
\left\|\bar{\mathbf{e}}_t\right\|^2 &= \frac{1}{m}\sum_{i=1}^m \left\|\mathbf{e}_t^i\right\|^2 \\
&= \frac{1}{m}\sum_{i=1}^m \left\|\mathbf{P}_{t-1}^i - \tilde{\Delta}_{t-1}^i\right\|^2 \\
&\overset{(a1)}{\le} \frac{1}{m}\sum_{i=1}^m (1/\gamma)\left\|\mathbf{P}_{t-1}^i\right\|^2 \\
&\le \frac{1}{m}\sum_{i=1}^m (1/\gamma)\left\|\mathbf{g}_{t-1}^i + \mathbf{e}_{t-1}^i\right\|^2 \\
&\overset{(a2)}{\le} \frac{1}{m}(1/\gamma)\sum_{i=1}^m \Big[\frac{1}{1-\epsilon}\left\|\mathbf{g}_{t-1}^i\right\|^2 + \frac{1}{\epsilon}\left\|\mathbf{e}_{t-1}^i\right\|^2\Big] \\
&= (1/\gamma)\Big[\frac{1}{\epsilon}\frac{1}{m}\sum_{i=1}^m \left\|\mathbf{e}_{t-1}^i\right\|^2 + \frac{1}{1-\epsilon}\frac{1}{m}\sum_{i=1}^m \left\|\mathbf{g}_{t-1}^i\right\|^2\Big] \\
&= (1/\gamma)\frac{1}{\epsilon}\left\|\bar{\mathbf{e}}_{t-1}\right\|^2 + (1/\gamma)\frac{1}{1-\epsilon}\left\|\bar{\Delta}_{t-1}\right\|^2,
\end{aligned}$$

where $(a1)$ is due to the definition of the compressor, $(a2)$ follows from $\|\mathbf{x}+\mathbf{y}\|^2 \le \frac{1}{\epsilon}\|\mathbf{x}\|^2 + \frac{1}{1-\epsilon}\|\mathbf{y}\|^2$, where $\epsilon \in (0,1)$.

Recursively using this relationship that $\left\|\bar{\mathbf{e}}_t\right\|^2 \le (1/\gamma)\frac{1}{\epsilon}\left\|\bar{\mathbf{e}}_{t-1}\right\|^2 + (1/\gamma)\frac{1}{1-\epsilon}\left\|\bar{\Delta}_{t-1}\right\|^2$ and note $\bar{\mathbf{e}}_0 = 0$, then we have:

$$\|\bar{\mathbf{e}}_t\|^2 \le \sum_{p=0}^{t-1}[(1/\gamma)\frac{1}{\epsilon}]^{t-1-p}(1/\gamma)\frac{1}{1-\epsilon}\|\bar{\Delta}_p\|^2.$$

Summing from $t = 0$ to $t = T-1$:

$$\begin{aligned}
\sum_{t=0}^{T-1}\|\bar{\mathbf{e}}_t\|^2 &= [(1/\gamma)\frac{1}{1-\epsilon}\sum_{p=0}^{T-2}((1/\gamma)\frac{1}{\epsilon})^p\|\bar{\Delta}_0\|^2] \\
&\quad + [(1/\gamma)\frac{1}{1-\epsilon}\sum_{p=0}^{T-3}((1/\gamma)\frac{1}{\epsilon})^p\|\bar{\Delta}_1\|^2] + \cdots \\
&\quad + [(1/\gamma)\frac{1}{1-\epsilon}\sum_{p=0}^{1}((1/\gamma)\frac{1}{\epsilon})^p\|\bar{\Delta}_{T-3}\|^2] \\
&\quad + [(1/\gamma)\frac{1}{1-\epsilon}\sum_{p=0}^{0}((1/\gamma)\frac{1}{\epsilon})^p\|\bar{\Delta}_{T-2}\|^2] \\
&\le (1/\gamma)\frac{1}{1-\epsilon}\sum_{t=0}^{T-1}\sum_{p=0}^{\infty}((1/\gamma)\frac{1}{\epsilon})^p\|\bar{\Delta}_t\|^2 \\
&\overset{(a3)}{\le} (1/\gamma)\frac{1}{1-\epsilon}\sum_{t=0}^{T-1}\sum_{p=0}^{\infty}((1/\gamma)\frac{1}{\epsilon})^p\alpha_t\|\Delta_t\|^2 \\
&\le (1/\gamma)\sum_{t=0}^{T-1}(1+\frac{1}{\gamma\epsilon-1})\frac{\alpha_t}{1-\epsilon}\|\Delta_t\|^2 \\
&\overset{(a4)}{\le} (1/\gamma)(1+1/a)b\sum_{t=0}^{T-1}\|\Delta_t\|^2 \\
&= h(\gamma)\sum_{t=0}^{T-1}\|\Delta_t\|^2. \qquad (7)
\end{aligned}$$

$(a3)$ is due to $\alpha_t = \frac{\|\bar{\Delta}_t\|^2}{\|\Delta_t\|^2}$. For any given $\gamma > 1$, we can choose $\epsilon \in (0,1)$ such that $\gamma\epsilon - 1 \ge a$ and $\frac{\alpha_t}{1-\epsilon} \le b$ for two constant a and b, which yields $(a4)$.

This completes the proof of Lemma 4.1. □

Lemma 1 (Iterative Step). *By letting $\hat{\mathbf{x}}_t = \mathbf{x}_t + \eta\mathbf{e}_t$, we show the iterative relationship of $\hat{\mathbf{x}}_t$ as follows.*

$$\hat{\mathbf{x}}_{t+1} = \hat{\mathbf{x}}_t + \eta\Delta_t.$$

Proof of Lemma 1. We set a set of virtual variables for convenience. Denote $\mathbf{e}_t = \frac{1}{m}\sum_{i=1}^m \mathbf{e}_t^i$, $\Delta_t = \frac{1}{m}\sum_{i=1}^m \mathbf{g}_t^i$, $\tilde{\Delta}_t = \frac{1}{m}\sum_{i=1}^m \tilde{\Delta}_t^i =$

$\frac{1}{m}\sum_{i=1}^m C(\mathbf{P}_t^i)$, $\hat{\mathbf{x}}_t = \mathbf{x}_t + \eta \mathbf{e}_t$.

$$\begin{aligned}
\hat{\mathbf{x}}_{t+1} &= \mathbf{x}_{t+1} + \eta \mathbf{e}_{t+1} \\
&= \mathbf{x}_t + \eta\tilde{\Delta}_t + \eta \mathbf{e}_{t+1} \\
&= \mathbf{x}_t + \frac{1}{m}\eta\sum_{i=1}^m \tilde{\Delta}_t^i + \frac{1}{m}\eta\sum_{i=1}^m \mathbf{e}_{t+1}^i \\
&= \mathbf{x}_t + \frac{1}{m}\eta\sum_{i=1}^m \tilde{\Delta}_t^i + \frac{1}{m}\eta\sum_{i=1}^m (\mathbf{P}_t^i - \tilde{\Delta}_t^i) \\
&= \mathbf{x}_t + \frac{1}{m}\eta\sum_{i=1}^m \mathbf{P}_t^i \\
&= \mathbf{x}_t + \frac{1}{m}\eta\sum_{i=1}^m (\mathbf{g}_t^i + \mathbf{e}_t^i) \\
&= \mathbf{x}_t + \frac{1}{m}\eta\sum_{i=1}^m \mathbf{e}_t^i + \frac{1}{m}\eta\sum_{i=1}^m \mathbf{g}_t^i \\
&= \hat{\mathbf{x}}_t + \eta\Delta_t.
\end{aligned}$$

This completes the proof of Lemma 1. □

Lemma 2 (One Communication Round Descent). *Choose local and global learning rates η_L and η as $\eta_L \le \frac{1}{8LK}$, $\eta\eta_L < \frac{1}{KL}$ and $\eta\eta_L K(L^2h(\gamma)+1+L) \le 1$. Under Assumptions 1–3, one communication round descent of the virtual sequence of outputs $\{\hat{\mathbf{x}}_t = \mathbf{x}_t + \eta\mathbf{e}_t\}$ generated by our Algorithm satisfies:*

$$\begin{aligned}
&\mathbb{E}f(\hat{\mathbf{x}}_{t+1}) - \nabla f(\hat{\mathbf{x}}_t) \\
&\le (\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\frac{K\eta_L^2}{m}\sigma_L^2 \\
&\quad - c\eta\eta_L K\big\|\nabla f(x_t)\big\|^2 + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)
\end{aligned}$$

where $\hat{\mathbf{x}}_t = \mathbf{x}_t + \eta\mathbf{e}_t$, c is a constant, $h(\gamma)$ is defined the same as that in Lemma 4.1.

Proof of Lemma 2. Due to the Smoothness assumption 1, taking expectation of $f(\hat{\mathbf{x}}_{t+1})$ over the randomness at time step t, we have:

$$\begin{aligned}
&\mathbb{E}_t f(\hat{\mathbf{x}}_{t+1}) \\
&\le f(\hat{\mathbf{x}}_t) + \mathbb{E}_t < \nabla f(\hat{\mathbf{x}}_t), [\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t] > + \frac{L}{2}\mathbb{E}_t\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\
&= f(\hat{\mathbf{x}}_t) + \underbrace{\mathbb{E}_t < \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t >}_{A_1} + \frac{L\eta^2}{2}\underbrace{\mathbb{E}_t\|\Delta_t\|^2}_{A_2}. \qquad (8)
\end{aligned}$$

We bound A_1 and A_2 respectively in the following.

Bounding A_1:

$$\begin{aligned}
A_1 &= \mathbb{E}_t\big\langle \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t\big\rangle \\
&= \mathbb{E}_t\big\langle \nabla f(\hat{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t) + \nabla f(\mathbf{x}_t), \eta\Delta_t\big\rangle \\
&= \mathbb{E}_t\big[\big\langle \nabla f(\hat{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t), \eta\Delta_t\big\rangle + \eta\big\langle \nabla f(\mathbf{x}_t), \Delta_t\big\rangle\big] \\
&\overset{(b1)}{\le} \mathbb{E}_t\big[\frac{1}{2}(\|\nabla f(\hat{\mathbf{x}}_t) - \nabla f(\mathbf{x}_t)\|^2 + \eta^2\|\Delta_t\|^2) \\
&\qquad + \eta\big\langle \nabla f(\mathbf{x}_t), \Delta_t + K\eta_L\nabla f(\mathbf{x}_t) - K\eta_L\nabla f(\mathbf{x}_t)\big\rangle\big] \\
&\overset{(b2)}{\le} \mathbb{E}_t\big[\frac{1}{2}(L^2\|\hat{\mathbf{x}}_t - \mathbf{x}_t\|^2 + \eta^2\|\Delta_t\|^2) \\
&\qquad + \eta\big\langle \nabla f(\mathbf{x}_t), \Delta_t + K\eta_L\nabla f(\mathbf{x}_t)\big\rangle - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2\big] \\
&= \mathbb{E}_t\big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2 - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 \\
&\qquad + \eta\underbrace{\big\langle \nabla f(\mathbf{x}_t), \Delta_t + K\eta_L\nabla f(\mathbf{x}_t)\big\rangle}_{A_3}\big], \qquad (9)
\end{aligned}$$

where $(b1)$ follows from the fact that $\langle \mathbf{x}, \mathbf{y}\rangle = \frac{1}{2}[\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \|\mathbf{x}-\mathbf{y}\|^2] \le \frac{1}{2}[\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2]$, $(b2)$ is due to the Assumption 1.

We can bound A_3 as follows.

$$\begin{aligned}
A_3 &= \mathbb{E}_t\big\langle \nabla f(\mathbf{x}_t), \Delta_t + \eta_L K\nabla f(\mathbf{x}_t)\big\rangle \\
&= \big\langle \nabla f(\mathbf{x}_t), \mathbb{E}_t\big[-\frac{1}{m}\sum_{i=1}^m\sum_{k=0}^{K-1}\eta_L\mathbf{g}_{t,k}^i + \eta_L K\nabla f(x_t)\big]\big\rangle \\
&= \big\langle \nabla f(\mathbf{x}_t), \mathbb{E}_t\big[-\frac{1}{m}\sum_{i=1}^m\sum_{k=0}^{K-1}\eta_L\nabla F_i(\mathbf{x}_{t,k}^i) + \eta_L K\frac{1}{m}\sum_{i=1}^m\nabla F_i(\mathbf{x}_t)\big]\big\rangle \\
&= \big\langle \sqrt{\eta_L K}\nabla f(\mathbf{x}_t), -\frac{\sqrt{\eta_L}}{m\sqrt{K}}\mathbb{E}_t\sum_{i=1}^m\sum_{k=0}^{K-1}(\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t))\big\rangle \\
&\overset{(c1)}{=} \frac{\eta_L K}{2}\big\|\nabla f(\mathbf{x}_t)\big\|^2 - \frac{\eta_L}{2Km^2}\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2] \\
&\qquad + \frac{\eta_L}{2Km^2}\mathbb{E}_t\big[\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}(\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t))\big\|^2\big] \\
&\overset{(c2)}{\le} \frac{\eta_L K}{2}\big\|\nabla f(\mathbf{x}_t)\big\|^2 - \frac{\eta_L}{2Km^2}\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2] \\
&\qquad + \frac{\eta_L}{2m}\mathbb{E}_t\big[\sum_{i=1}^m\sum_{k=0}^{K-1}\big\|\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t)\big\|^2 \\
&\overset{(c3)}{\le} \frac{\eta_L K}{2}\big\|\nabla f(\mathbf{x}_t)\big\|^2 - \frac{\eta_L}{2Km^2}\mathbb{E}_t\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2 \\
&\qquad + \frac{\eta_L L^2}{2m}\mathbb{E}_t\sum_{i=1}^m\sum_{k=0}^{K-1}\big\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\big\|^2 \\
&\overset{(c4)}{\le} \eta_L K(\frac{1}{2} + 15K^2\eta_L^2L^2)\big\|\nabla f(x_t)\big\|^2 + \frac{5K^2\eta_L^3L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \\
&\qquad - \frac{\eta_L}{2Km^2}\mathbb{E}_t\big\|\sum_{i=1}^m\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\big\|^2, \qquad (10)
\end{aligned}$$

where $(c1)$ follows from the fact that $\langle \mathbf{x}, \mathbf{y} \rangle = \frac{1}{2}[\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \|\mathbf{x} - \mathbf{y}\|^2]$ for $\mathbf{x} = \sqrt{\eta_L K}\nabla f(\mathbf{x}_t)$ and $\mathbf{y} = -\frac{\sqrt{\eta_L}}{m\sqrt{K}} \sum_{i=1}^{m} \sum_{k=0}^{K-1} (\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t))$, $(c2)$ is due to that $\mathbb{E}[\|x_1 + \cdots + x_n\|^2] \leq n\mathbb{E}[\|x_1\|^2 + \cdots + \|x_n\|^2]$, $(c3)$ is due to Assumption 1 and $(c4)$ follows from Lemma 3.

Plugging the above results 10 into 9 yields:

$$
\begin{aligned}
A_1 &\leq \mathbb{E}_t\big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2 \\
&\quad + \eta\langle \nabla f(\mathbf{x}_t), \Delta_t + K\eta_L \nabla f(\mathbf{x}_t)\rangle - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2\big] \\
&\leq \mathbb{E}_t\big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2 - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2\big] \\
&\quad + \eta\eta_L K(\frac{1}{2} + 15K^2\eta_L^2 L^2)\|\nabla f(x_t)\|^2 \\
&\quad + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \\
&\quad - \frac{\eta\eta_L}{2Km^2}\mathbb{E}_t\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2 \\
&\overset{(*)}{\leq} \mathbb{E}_t\big[(\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2)\|\Delta_t\|^2\big] - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 \\
&\quad + \eta\eta_L K(\frac{1}{2} + 15K^2\eta_L^2 L^2)\|\nabla f(x_t)\|^2 \\
&\quad + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \\
&\quad - \frac{\eta\eta_L}{2Km^2}\mathbb{E}_t\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2. \qquad (11)
\end{aligned}
$$

Note that Lemma 4.1 does not necessarily leads to $\|\mathbf{e}_t\|^2 \leq h(\gamma)\|\Delta_t\|^2$ in general for $(*)$ to hold rigorously. We abuse the notation and inequality here in order to show the function descent in one communication round more clearly. In the precesses of proving convergence in the following different cases, the focus is the summation $\sum_{t=0}^{T-1}\|\mathbf{e}_t\|^2$ since a common approach is to sum the function descent from 0 to $T-1$. This gives $\sum_{t=0}^{T-1}\|\mathbf{e}_t\|^2 \leq h(\gamma)\sum_{t=0}^{T-1}\|\Delta_t\|^2$, thus the convergence rate results proven in the following rigorously hold.

Bounding A_2:

$$
\begin{aligned}
A_2 &= \mathbb{E}_t\big[\|\Delta_t\|^2\big] \\
&= \mathbb{E}_t\big[\|\frac{1}{m}\sum_{i=1}^{m}\Delta_t^i\|^2\big] \\
&\leq \frac{1}{m^2}\mathbb{E}_t\big[\|\sum_{i=1}^{m}\Delta_t^i\|^2\big] \\
&= \frac{\eta_L^2}{m^2}\mathbb{E}_t\big[\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\mathbf{g}_{t,k}^i\|^2\big] \\
&\overset{(d1)}{=} \frac{\eta_L^2}{m^2}\mathbb{E}_t\big[\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}(\mathbf{g}_{t,k}^i - \nabla F_i(\mathbf{x}_{t,k}^i))\|^2\big] \\
&\quad + \frac{\eta_L^2}{m^2}\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2 \\
&\overset{(d2)}{\leq} \frac{K\eta_L^2}{m}\sigma_L^2 + \frac{\eta_L^2}{m^2}\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2, \qquad (12)
\end{aligned}
$$

where $(d1)$ follows from the fact that $\mathbb{E}[\|\mathbf{x}\|^2] = \mathbb{E}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|^2] + \|\mathbb{E}[\mathbf{x}]\|^2$, $(d2)$ is due to the bounded variance assumption in Assumption 3 and the fact that $\mathbb{E}[\|x_1 + \cdots + x_n\|^2] = \mathbb{E}[\|x_1\|^2 + \cdots + \|x_n\|^2]$ if x_i are independent with mean 0.

By combining the results in 11, 12 and 8, we have:

$$
\begin{aligned}
&\mathbb{E}f(\hat{\mathbf{x}}_{t+1}) - \nabla f(\hat{\mathbf{x}}_t) \\
&\leq \underbrace{\mathbb{E}\langle \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t\rangle}_{A_1} + \frac{L\eta^2}{2}\underbrace{\mathbb{E}\|\Delta_t\|^2}_{A_2} \\
&\overset{(e1)}{\leq} \big[(\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\times \\
&\quad (\frac{K\eta_L^2}{m}\sigma_L^2 + \frac{\eta_L^2}{m^2}\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2)\big] \\
&\quad + \eta\eta_L K(\frac{1}{2} + 15K^2\eta_L^2 L^2)\|\nabla f(x_t)\|^2 \\
&\quad + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) - K\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 \\
&\quad - \frac{\eta\eta_L}{2Km^2}\|\sum_{i=1}^{m}\sum_{k=0}^{K-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2 \\
&\overset{(e2)}{\leq} (\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\frac{K\eta_L^2}{m}\sigma_L^2 \\
&\quad - \eta\eta_L K(\frac{1}{2} - 15K^2\eta_L^2 L^2)\|\nabla f(x_t)\|^2 \\
&\quad + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \\
&\overset{(e3)}{\leq} (\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\frac{K\eta_L^2}{m}\sigma_L^2 \\
&\quad - c\eta\eta_L K\|\nabla f(x_t)\|^2 + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2).
\end{aligned}
$$

$(e1)$ is due to the inequality 5 and 6. $(e2)$ holds if $(\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2})\frac{\eta_L^2}{m^2} - \frac{\eta\eta_L}{2Km^2} \leq 0$, that is, $\eta\eta_L K(L^2 h(\gamma) + 1 + L) \leq 1$. $(e3)$ holds for a constant c such that $(\frac{1}{2} - 15K^2\eta_L^2 L^2) > c > 0$ if $\eta_L < \frac{1}{\sqrt{30}KL}$.

This completes the proof of Lemma 2. □

A.2 Proof of Theorem 4.2

Theorem 4.2. *(Convergence Rate of CFedAvg). Choose constant local and global learning rates η_L and η such that $\eta_L \leq \frac{1}{8LK}$, $\eta\eta_L < \frac{1}{KL}$ and $\eta\eta_L K(L^2 h(\gamma) + 1 + L) \leq 1$. Under Assumptions 1–3, the sequence of outputs $\{\mathbf{x}_t\}$ generated by Algorithm 1 satisfies:*

$$
\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L KT} + \Phi, \qquad (2)
$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2 h(\gamma) + \frac{1}{2} + \frac{L}{2})\frac{\eta\eta_L}{m}\sigma_L^2 + \frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)]$, c is a constant, $h(\gamma)$ is defined the same as that in Lemma 4.1, $f_0 = f(\hat{\mathbf{x}}_0)$ and $f_* = f(\mathbf{x}_*)$.

Proof of Theorem 4.2. Summing from $t = 0, \cdots, T-1$ of the one communication round descent in Lemma 2 yields:

$$\begin{aligned}
&\mathbb{E}f(\hat{\mathbf{x}}_T) - \nabla f(\hat{\mathbf{x}}_0) \\
&\leq \sum_{t=0}^{T-1} \Big\{ (\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2}) \frac{K\eta_L^2}{m}\sigma_L^2 \\
&\quad - c\eta\eta_L K \big\|\nabla f(x_t)\big\|^2 + \frac{5\eta K^2\eta_L^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \Big\}.
\end{aligned}$$

By rearranging, we have:

$$\begin{aligned}
&\sum_{t=0}^{T-1} c\eta\eta_L K \|\nabla f(\mathbf{x}_t)\|^2 \\
&\leq \nabla f(\hat{\mathbf{x}}_0) - \mathbb{E}f(\hat{\mathbf{x}}_T) + T(\eta\eta_L K) \Bigg[\frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \\
&\quad + (\frac{1}{2}L^2\eta h(\gamma) + \frac{1}{2}\eta + \frac{L\eta}{2}) \frac{\eta_L}{m}\sigma_L^2 \Bigg].
\end{aligned}$$

By letting $f_0 = f(\hat{\mathbf{x}}_0)$ and $f_* = f(\mathbf{x}_*) \leq f(\hat{\mathbf{x}}_T)$, we have:

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L KT} + \Phi,$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2 h(\gamma) + \frac{1}{2} + \frac{L}{2})\frac{\eta\eta_L}{m}\sigma_L^2 + \frac{5K\eta_L^2 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2)]$.

This completes the proof of Theorem 4.2. □

Corollary 4.3. *(Linear Speedup for Convergence). If $\eta_L = \frac{1}{\sqrt{T}KL}$ and $\eta = \sqrt{Km}$, the outputs $\{\mathbf{x}_t\}$ generated by Algorithm 1 satisfies:* $\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 = O(\frac{h(\gamma)}{\sqrt{mKT}} + \frac{1}{T}) = O(\frac{1}{\sqrt{mKT}} + \frac{1}{T})$.

Proof of Corollary 4.3. Plugging the learning rate into the convergence rate bound in Theorem 4.2 completes the proof. □

A.3 Proof of Theorem 4.4

Theorem 4.4. *(Convergence with Decaying Learning Rate). Choose decaying local learning rate $\eta_{L,t}$ and constant global learning rates η such that $\eta_{L,t} \leq \frac{1}{8LK}$, $\eta\eta_{L,t} < \frac{1}{KL}$ and $\eta\eta_{L,t}K(L^2 h(\gamma) + 1 + L) \leq 1, \forall t \in [T]$. Under Assumptions 1–3, the sequence of outputs $\{\mathbf{x}_t\}$ generated by Algorithm 1 satisfies:*

$$\mathbb{E}\|\nabla f(\mathbf{z})\|_2^2 \leq \frac{f_0 - f_*}{c\eta KH_T} + \Phi,$$

where $\Phi \triangleq (\frac{1}{2}L^2 h(\gamma) + \frac{1}{2} + \frac{L}{2})\frac{\eta}{cmH_T}\sigma_L^2 \sum_{t=0}^{T-1}\eta_{L,t}^2 + \frac{5KL^2}{2cH_T}(\sigma_L^2 + 6K\sigma_G^2)\sum_{t=0}^{T-1}\eta_{L,t}^3$, $H_T = \sum_{t=0}^{T-1}\eta_{L,t}$ and $\mathbf{z}$ is sampled from $\{\mathbf{x}_t\}, \forall t \in [T]$ with probability $\mathbb{P}[\mathbf{z} = \mathbf{x}_t] = \frac{\eta_{L,t}}{H_T}$. Here c is a constant, $h(\gamma)$ is defined the same as that in Lemma 4.1, $f_0 = f(\hat{\mathbf{x}}_0)$ and $f_* = f(\mathbf{x}_*)$.

Proof of Theorem 4.4. Note that Lemma 4.1 and Lemma 1 still hold with decaying local learning rate. Thus the one communication round descent is as following:

$$\begin{aligned}
&\mathbb{E}f(\hat{\mathbf{x}}_{t+1}) - \nabla f(\hat{\mathbf{x}}_t) \\
&\leq (\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2}) \frac{K\eta_{L,t}^2}{m}\sigma_L^2 \\
&\quad - c\eta\eta_{L,t} K \big\|\nabla f(x_t)\big\|^2 + \frac{5\eta K^2\eta_{L,t}^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2),
\end{aligned}$$

where $\hat{\mathbf{x}}_t = \mathbf{x}_t + \eta\mathbf{e}_t$, c is a constant, $h(\gamma)$ is defined the same as that in Lemma 4.1.

By telescoping the above result, we have:

$$\begin{aligned}
&\mathbb{E}f(\hat{\mathbf{x}}_T) - \nabla f(\hat{\mathbf{x}}_0) \\
&\leq \sum_{t=0}^{T-1} \Big\{ (\frac{1}{2}L^2\eta^2 h(\gamma) + \frac{1}{2}\eta^2 + \frac{L\eta^2}{2}) \frac{K\eta_{L,t}^2}{m}\sigma_L^2 \\
&\quad - c\eta\eta_{L,t} K \big\|\nabla f(x_t)\big\|^2 + \frac{5\eta K^2\eta_{L,t}^3 L^2}{2}(\sigma_L^2 + 6K\sigma_G^2) \Big\}.
\end{aligned}$$

Rearranging the terms:

$$\begin{aligned}
&\sum_{t=0}^{T-1} \eta_{L,t} \big\|\nabla f(\mathbf{x}_t)\big\|^2 \\
&\leq \frac{\mathbb{E}f(\hat{\mathbf{x}}_0) - \nabla f(\hat{\mathbf{x}}_T)}{c\eta K} + \sum_{t=0}^{T-1} \Big\{ (\frac{1}{2}L^2 h(\gamma) + \frac{1}{2} + \frac{L}{2}) \frac{\eta\eta_{L,t}^2}{cm}\sigma_L^2 \\
&\quad + \frac{5K\eta_{L,t}^3 L^2}{2c}(\sigma_L^2 + 6K\sigma_G^2) \Big\}.
\end{aligned}$$

Let $H_T = \sum_{t=0}^{T-1}\eta_{L,t}$ and $\mathbf{z}$ is sampled from $\{\mathbf{x}_t\}, \forall t \in [T]$ with probability $\mathbb{P}[\mathbf{z} = \mathbf{x}_t] = \frac{\eta_{L,t}}{H_T}$ which results in $\mathbb{E}\big\|\nabla f(\mathbf{z})\big\|^2 = \frac{1}{H_T}\sum_{t=0}^{T-1}\eta_{L,t}\big\|\nabla f(\mathbf{x}_t)\big\|^2$. That is:

$$\begin{aligned}
&\mathbb{E}\big\|\nabla f(\mathbf{z})\big\|^2 \\
&\leq \frac{\mathbb{E}f(\hat{\mathbf{x}}_0) - \nabla f(\hat{\mathbf{x}}_T)}{c\eta KH_T} + (\frac{1}{2}L^2 h(\gamma) + \frac{1}{2} + \frac{L}{2}) \frac{\eta}{cmH_T}\sigma_L^2 \sum_{t=0}^{T-1}\eta_{L,t}^2 \\
&\quad + \frac{5KL^2}{2cH_T}(\sigma_L^2 + 6K\sigma_G^2) \sum_{t=0}^{T-1}\eta_{L,t}^3.
\end{aligned}$$

By letting $f_0 = f(\hat{\mathbf{x}}_0)$ and $f_* = f(\mathbf{x}_*) \leq f(\hat{\mathbf{x}}_T)$, we complete the proof of Theorem 4.4. □

Corollary 4.5. *Let $\eta_L = \frac{1}{\sqrt{t+a}KL}$ for some constant $a > 0$ and let $\eta = \sqrt{Km}$. The outputs $\{\mathbf{x}_t\}$ generated by Algorithm 1 satisfies:* $\mathbb{E}\big\|\nabla f(\mathbf{z})\big\|^2 = \tilde{O}(\frac{1}{\sqrt{mKT}}) + O(\frac{1}{\sqrt{T}})$.

Proof of Corollary 4.5.

$$\begin{aligned}
H_T &= \sum_{t=0}^{T-1}\eta_{L,t} = \sum_{t=0}^{T-1}\frac{1}{\sqrt{t+a}KL} = \frac{1}{KL}\Theta(T^{1/2}). \\
\sum_{t=0}^{T-1}\eta_{L,t}^2 &= \sum_{t=0}^{T-1}(\frac{1}{\sqrt{t+a}KL})^2 = \frac{1}{K^2L^2}O(\ln(T)) \\
\sum_{t=0}^{T-1}\eta_{L,t}^3 &= \sum_{t=0}^{T-1}(\frac{1}{\sqrt{t+a}KL})^3 = \frac{1}{K^3L^3}O(1)
\end{aligned}$$

So we have:

$$\begin{aligned}
\mathbb{E}\big\|\nabla f(\mathbf{z})\big\|^2 &= O(\frac{1}{\sqrt{mKT}}\ln(T)) + O(\frac{1}{\sqrt{T}}) \\
&= \tilde{O}(\frac{1}{\sqrt{mKT}}) + O(\frac{1}{\sqrt{T}}).
\end{aligned}$$

□

A.4 Proof of Theorem 4.6

Theorem 4.6. *(Convergence of Heterogeneous Local Steps). Choose constant local and global learning rates η_L and η such that $\eta_L \leq \frac{1}{8LK_i}$, $\eta\eta_L < \frac{1}{K_iL}, \forall i \in [m]$ and $\eta\eta_L(L^2h(\gamma)+1+L) \leq 1$. Under Assumptions 1–3, the sequence of outputs $\{\mathbf{x}_t\}$ generated by Algorithm 1 with heterogeneous local steps satisfies:*

$$\min_{t\in[T]} \mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2 \leq \frac{f_0 - f_*}{c\eta\eta_L T} + \Phi,$$

where $\Phi \triangleq \frac{1}{c}[(\frac{1}{2}L^2h(\gamma)+\frac{1}{2}+\frac{L}{2})\frac{\eta\eta_L}{m^2}\sum_{i=1}^m \frac{1}{K_i}\sigma_L^2 + \frac{5\eta_L^2L^2}{2}\frac{1}{m}\sum_{i=1}^m K_i(\sigma_L^2 + 6K_i\sigma_G^2)]$, *$c$ is a constant, $h(\gamma)$ is defined the same as that in Lemma 4.1, $f_0 = f(\hat{\mathbf{x}}_0)$ and $f_* = f(\mathbf{x}_*)$.*

Proof of Theorem 4.6. First, note that Lemma 4.1 still holds for distinct local steps among different workers. With gradients scaled by the local steps K_i, Δ_t is:

$$\Delta_t = \frac{1}{m}\sum_{i=1}^m \Delta_t^i = \sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1} \mathbf{g}_{t,k}^i$$

Due to the Smoothness assumption 1, taking expectation of $f(\hat{\mathbf{x}}_{t+1})$ over the randomness at time step t, we have:

$$\begin{aligned}
&\mathbb{E}_t f(\hat{\mathbf{x}}_{t+1}) \\
&\leq f(\hat{\mathbf{x}}_t) + \mathbb{E}_t < \nabla f(\hat{\mathbf{x}}_t), [\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t] > + \frac{L}{2}\mathbb{E}_t\|\hat{\mathbf{x}}_{t+1} - \hat{\mathbf{x}}_t\|^2 \\
&= f(\hat{\mathbf{x}}_t) + \underbrace{\mathbb{E}_t < \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t >}_{A_1} + \frac{L\eta^2}{2}\underbrace{\mathbb{E}_t\|\Delta_t\|^2}_{A_2}. \qquad (13)
\end{aligned}$$

Bounding A_2:

$$\begin{aligned}
A_2 &= \mathbb{E}_t\big[\|\Delta_t\|^2\big] \\
&= \mathbb{E}_t\big[\|\frac{1}{m}\sum_{i=1}^m \Delta_t^i\|^2\big] \\
&\leq \frac{1}{m^2}\mathbb{E}_t\big[\|\sum_{i=1}^m \Delta_t^i\|^2\big] \\
&= \frac{\eta_L^2}{m^2}\mathbb{E}_t\big[\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1} \mathbf{g}_{t,k}^i\|^2\big] \\
&\overset{(d1)}{=} \frac{\eta_L^2}{m^2}\mathbb{E}_t\big[\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}(\mathbf{g}_{t,k}^i - \nabla F_i(\mathbf{x}_{t,k}^i))\|^2\big] \\
&\quad + \frac{\eta_L^2}{m^2}\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2 \\
&\overset{(d2)}{\leq} \frac{\eta_L^2}{m^2}\sum_{i=1}^m \frac{1}{K_i}\sigma_L^2 + \frac{\eta_L^2}{m^2}\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2, \qquad (14)
\end{aligned}$$

where $(d1)$ follows from the fact that $\mathbb{E}[\|\mathbf{x}\|^2] = \mathbb{E}[\|\mathbf{x} - \mathbb{E}[\mathbf{x}]\|^2] + \|\mathbb{E}[\mathbf{x}]\|^2]$, $(d2)$ is due to the bounded variance assumption in Assumption 3 and the fact that $\mathbb{E}[\|x_1 + \cdots + x_n\|^2] = \mathbb{E}[\|x_1\|^2 + \cdots + \|x_n\|^2]$ if x_i are independent with mean 0.

Bounding A_1: We can have the same result as in 9.

$$\begin{aligned}
A_1 &= \mathbb{E}_t\big\langle \nabla f(\hat{\mathbf{x}}_t), \eta\Delta_t \big\rangle \\
&= \mathbb{E}_t\big[\frac{1}{2}L^2\eta^2\|\mathbf{e}_t\|^2 + \frac{1}{2}\eta^2\|\Delta_t\|^2 - \eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2 \\
&\quad + \eta\underbrace{< \nabla f(\mathbf{x}_t), \Delta_t + \eta_L\nabla f(\mathbf{x}_t) >}_{A_3}\big], \qquad (15)
\end{aligned}$$

We can bound A_3 as follows.

$$\begin{aligned}
A_3 &= \mathbb{E}_t\big\langle \nabla f(\mathbf{x}_t), \Delta_t + \eta_L\nabla f(\mathbf{x}_t)\big\rangle \\
&= \big\langle \nabla f(\mathbf{x}_t), \mathbb{E}_t\big[-\frac{1}{m}\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\eta_L\mathbf{g}_{t,k}^i + \eta_L\nabla f(x_t)\big]\big\rangle \\
&= \big\langle \nabla f(\mathbf{x}_t), \mathbb{E}_t\big[-\frac{1}{m}\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\eta_L\nabla F_i(\mathbf{x}_{t,k}^i) + \eta_L\frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_t)\big]\big\rangle \\
&= \big\langle \sqrt{\eta_L}\nabla f(\mathbf{x}_t), -\frac{\sqrt{\eta_L}}{m}\mathbb{E}_t\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}(\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t))\big\rangle \\
&\overset{(c1)}{=} \frac{\eta_L}{2}\|\nabla f(\mathbf{x}_t)\|^2 - \frac{\eta_L}{2m^2}\mathbb{E}_t\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2] \\
&\quad + \frac{\eta_L}{2m^2}\mathbb{E}_t\big[\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}(\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t))\|^2\big] \\
&\overset{(c2)}{\leq} \frac{\eta_L}{2m}\mathbb{E}_t\big[\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K-1}\|\nabla F_i(\mathbf{x}_{t,k}^i) - \nabla F_i(\mathbf{x}_t)\|^2\big] \\
&\quad + \frac{\eta_L}{2}\|\nabla f(\mathbf{x}_t)\|^2 - \frac{\eta_L}{2m^2}\mathbb{E}_t\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2 \\
&\overset{(c3)}{\leq} \frac{\eta_L}{2}\|\nabla f(\mathbf{x}_t)\|^2 + \frac{\eta_L L^2}{2m}\mathbb{E}_t\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\|^2 \\
&\quad - \frac{\eta_L}{2m^2}\mathbb{E}_t\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2 \\
&\overset{(c4)}{\leq} \eta_L(\frac{1}{2} + 15\eta_L^2L^2\frac{1}{m}\sum_{i=1}^m K_i^2)\|\nabla f(x_t)\|^2 \\
&\quad + \frac{5\eta_L^3L^2}{2}\frac{1}{m}\sum_{i=1}^m K_i(\sigma_L^2 + 6K_i\sigma_G^2) \\
&\quad - \frac{\eta_L}{2m^2}\mathbb{E}_t\|\sum_{i=1}^m \frac{1}{K_i}\sum_{k=0}^{K_i-1}\nabla F_i(\mathbf{x}_{t,k}^i)\|^2, \qquad (16)
\end{aligned}$$

where $(c1)$ follows from the fact that $\big\langle \mathbf{x}, \mathbf{y}\big\rangle = \frac{1}{2}[\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \|\mathbf{x} - \mathbf{y}\|^2]$, $(c2)$ is due to that $\mathbb{E}[\|x_1 + \cdots + x_n\|^2] \leq n\mathbb{E}[\|x_1\|^2 + \cdots + \|x_n\|^2]$, $(c3)$ is due to Assumption 1 and $(c4)$ follows from Lemma 3 (Here for each worker i, $\mathbb{E}\|\mathbf{x}_{t,k}^i - \mathbf{x}_t\|^2 \leq 5K_i\eta_L^2(\sigma_L^2 + 6K_i\sigma_G^2) + 30K_i^2\eta_L^2\|\nabla f(\mathbf{x}_t)\|^2$).

Following the same path for proof in Theorem 4.2, the following holds with $\eta\eta_L(L^2h(\gamma)+1+L)\leq 1$:

$$\begin{aligned}&\sum_{t=0}^{T-1} c\eta\eta_L\|\nabla f(\mathbf{x}_t)\|^2\\ &\leq \nabla f(\hat{\mathbf{x}}_0)-\mathbb{E}f(\hat{\mathbf{x}}_T)\\ &\quad+T(\eta\eta_L)\Bigg[\frac{5\eta_L^2L^2}{2}\frac{1}{m}\sum_{i=1}^m K_i(\sigma_L^2+6K_i\sigma_G^2)\\ &\quad+(\frac{1}{2}L^2h(\gamma)+\frac{1}{2}+\frac{L}{2})\frac{\eta\eta_L}{m^2}\sum_{i=1}^m\frac{1}{K_i}\sigma_L^2\Bigg],\end{aligned}$$

where c is a constant such that $(\frac{1}{2}-15\eta_L^2L^2\frac{1}{m}\sum_{i=1}^m K_i^2)>c>0$ if $\eta_L<\frac{1}{\sqrt{30}K_iL}$.

By letting $f_0=f(\hat{\mathbf{x}}_0)$ and $f_*=f(\mathbf{x}_*)\leq f(\hat{\mathbf{x}}_T)$, we have:

$$\min_{t\in[T]}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2\leq\frac{f_0-f_*}{c\eta\eta_LT}+\Phi,$$

where $\Phi\triangleq\frac{1}{c}[(\frac{1}{2}L^2h(\gamma)+\frac{1}{2}+\frac{L}{2})\frac{\eta\eta_L}{m^2}\sum_{i=1}^m\frac{1}{K_i}\sigma_L^2+\frac{5\eta_L^2L^2}{2}\frac{1}{m}\sum_{i=1}^m K_i(\sigma_L^2+6K_i\sigma_G^2)]$.

This completes the proof of Theorem 4.6. □

Corollary 4.7. *(Linear Speedup with Heterogeneous Local Steps). Let $\eta_L=\frac{1}{\sqrt{TL}}$ and $\eta=\sqrt{K_{min}m}$. The convergence rate of Algorithm 1 with heterogeneous local steps and constant learning rate is*

$$\min_{t\in[T]}\mathbb{E}\|\nabla f(\mathbf{x}_t)\|_2^2=O(\frac{1}{\sqrt{mK_{min}T}})+O(\frac{K_{max}^2}{T}),$$

where $K_{min}=\min\{K_i,\forall i\in[m]\}$ and $K_{max}=\max\{K_i,\forall i\in[m]\}$.

Proof of Corollary 4.7.

$$\begin{aligned}\frac{\eta\eta_L}{m^2}\sum_{i=1}^m\frac{1}{K_i}&=\frac{\sqrt{K_{min}m}}{m^2\sqrt{TL}}\sum_{i=1}^m\frac{1}{K_i}\\&=\frac{\sqrt{K_{min}}}{\sqrt{mTL}}\frac{1}{m}\sum_{i=1}^m\frac{1}{K_i}\\&\leq\frac{\sqrt{K_{min}}}{\sqrt{mTL}}\frac{1}{K_{min}}\\&=\frac{1}{\sqrt{mK_{min}TL}}\\ \eta_L^2\frac{1}{m}\sum_{i=1}^m K_i^2&=\frac{1}{TL^2}\frac{1}{m}\sum_{i=1}^m K_i^2\\&\leq\frac{K_{max}^2}{TL^2}.\end{aligned}$$

□

A.5 Auxiliary Lemma

Lemma 3. *[Lemma 4 in [26]] For any step-size satisfying $\eta_L\leq\frac{1}{8LK}$, we can have the following results:*

$$\mathbb{E}\|\mathbf{x}_{t,k}^i-\mathbf{x}_t\|^2\leq 5K\eta_L^2(\sigma_L^2+6K\sigma_G^2)+30K^2\eta_L^2\|\nabla f(\mathbf{x}_t)\|^2$$

Proof of Lemma.
For the completeness of the proof, we rewrite the proof of this lemma in [26].

For any worker $i\in[m]$ and $k\in[K]$, we have:

$$\begin{aligned}&\mathbb{E}\|\mathbf{x}_{t,k}^i-\mathbf{x}_t\|^2=\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t-\eta_Lg_{t,k-1}^t\|^2\\&\leq\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t-\eta_L(g_{t,k-1}^t-\nabla F_i(\mathbf{x}_{t,k-1}^i)\\&\quad+\nabla F_i(\mathbf{x}_{t,k-1}^i)-\nabla F_i(\mathbf{x}_t)+\nabla F_i(\mathbf{x}_t)-\nabla f(\mathbf{x}_t)+\nabla f(\mathbf{x}_t))\|^2\\&\leq(1+\frac{1}{2K-1})\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t\|^2+\mathbb{E}\|\eta_L(g_{t,k-1}^t-\nabla F_i(\mathbf{x}_{t,k-1}^i))\|^2\\&\quad+6K\mathbb{E}\|\eta_L(\nabla F_i(\mathbf{x}_{t,k-1}^i)-\nabla F_i(\mathbf{x}_t))\|^2\\&\quad+6K\mathbb{E}\|\eta_L(\nabla F_i(\mathbf{x}_t)-\nabla f(\mathbf{x}_t)))\|^2+6K\mathbb{E}\|\eta_L\nabla f(\mathbf{x}_t))\|^2\\&\leq(1+\frac{1}{2K-1})\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t\|^2+\eta_L^2\sigma_L^2\\&\quad+6K\eta_L^2L^2\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t\|^2+6K\eta_L^2\sigma_G^2+6K\mathbb{E}\|\eta_L\nabla f(\mathbf{x}_t))\|^2\\&=(1+\frac{1}{2K-1}+6K\eta_L^2L^2)\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t\|^2\\&\quad+\eta_L^2\sigma_L^2+6K\eta_L^2\sigma_G^2+6K\mathbb{E}\|\eta_L\nabla f(\mathbf{x}_t))\|^2\\&\leq(1+\frac{1}{K-1})\mathbb{E}\|\mathbf{x}_{t,k-1}^i-\mathbf{x}_t\|^2+\eta_L^2\sigma_L^2\\&\quad+6K\eta_L^2\sigma_G^2+6K\mathbb{E}\|\eta_L\nabla f(\mathbf{x}_t))\|^2\end{aligned}$$

Unrolling the recursion, we get:

$$\begin{aligned}&\mathbb{E}\|\mathbf{x}_{t,k}^i-\mathbf{x}_t\|^2\\&\leq\sum_{p=0}^{k-1}(1+\frac{1}{K-1})^p[\eta_L^2\sigma_L^2+6K\sigma_G^2+6K\eta_L^2\mathbb{E}\|\eta_L\nabla f(\mathbf{x}_t))\|^2]\\&\leq(K-1)\big[(1+\frac{1}{K-1})^K-1\big]\big[\eta_L^2\sigma_L^2+6K\sigma_G^2+6K\eta_L^2\mathbb{E}\|\eta_L\nabla f(\mathbf{x}_t))\|^2\big]\\&\leq 5K\eta_L^2(\sigma_L^2+6K\sigma_G^2)+30K^2\eta_L^2\|\nabla f(\mathbf{x}_t)\|^2\end{aligned}$$

B APPENDIX II: EXPERIMENTS

We provide the full detail of the experiments. We use three datasets in FL (non-i.i.d. version) settings, including MNIST, Fashion-MNIST and CIFAR-10. In the appendix, we show the results of Fashion-MNIST and CIFAR-10 while these of MNIST are shown in the paper.

We split the data based on the classes of items (p) they contain in their datasets. The system is containing 1 central server and $m=100$ workers, whose local dataset is distributed randomly and evenly in a class-based manner. The local dataset for every worker contains only certain classes of items with the same number of training/test samples. Each of these three datasets contains 10 different classes of items. So parameter p can be used to represent the non-i.i.d. degree of the datasets qualitatively. For example, for $p=1$, each worker only has training/testing samples labeled with one class, which causes heterogeneity among different workers. For $p=10$, each worker has samples with 10 classes, which is essentially i.i.d. case since the total classes for these three datasets are 10. We set four levels of non-i.i.d. version for comparison, i.e., $p=1,2,5,10$.

We run two models: convolutional neural network (CNN) on MNIST and Fashion-MNIST and Resnet-18 on CIFAR-10.

B.1 Effectiveness of Gradient Compression:

As shown in Figure 5 and 6 for CNN model and ResNet-18 on different non-i.i.d. Fashion-MNIST and CIFAR-10 datasets respectively, the top-row figures are for training loss versus communication

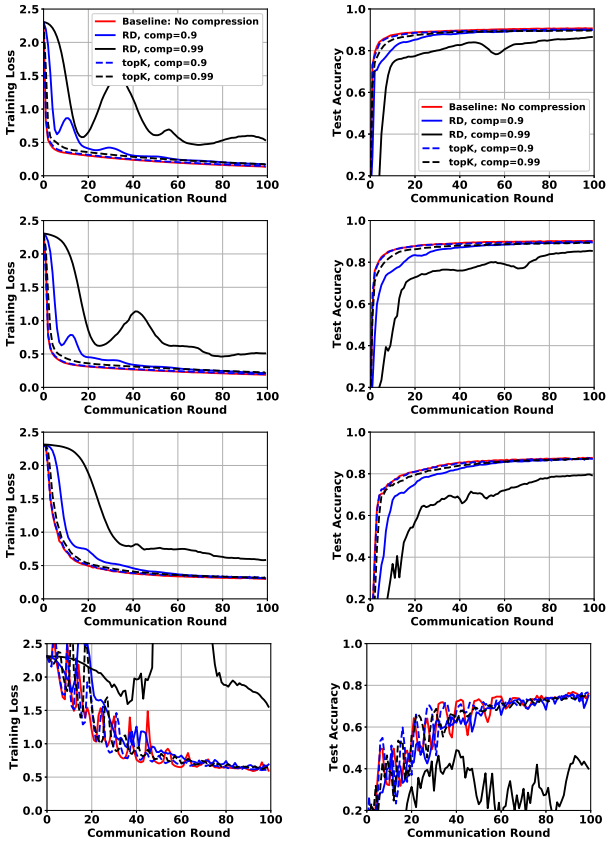


Figure 5: Training loss (left) and test accuracy (right) for the CNN model for Fashion-MNIST. The non-i.i.d. levels are p = 10, 5, 2, 1 from top to bottom.

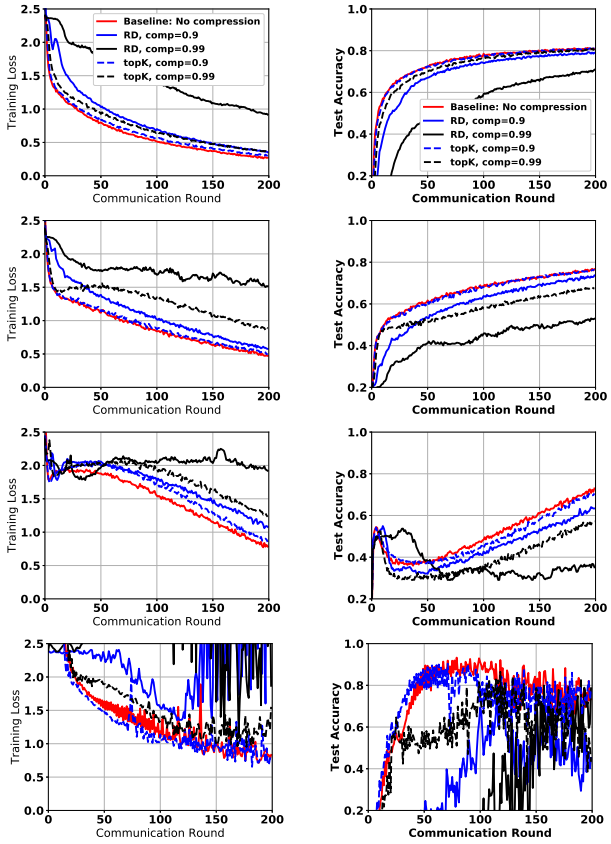


Figure 6: Training loss (top) and test accuracy (bottom) for the Resnet-18 model for CIFAR-10. The non-i.i.d. levels are p = 10, 5, 2, 1 from top to bottom.

Table 1: CNN Architecture for MNIST and Fashion-MNIST.

Layer Type	Size
Convolution + ReLu	$5 \times 5 \times 32$
Max Pooling	2×2
Convolution + ReLu	$5 \times 5 \times 64$
Max Pooling	2×2
Fully Connected + ReLU	1024×512
Fully Connected	512×10

round and the bottom-row are for test accuracy versus communication round. Key observations are as follows:

- We can see that our algorithm with two gradient compression methods converges at different levels of non-i.i.d.-ness from left (p = 10) to right (p = 1). In general, these cases preserving more information in each communication round usually have better results. It is shown that top-k precedes random dropping with same compression rate *comp* and that with low compression rate *comp* for the same compression method have faster convergence speed.
- The training curve is twisted for highly non-i.i.d. case (e.g., p = 1) compared to i.i.d. case. As the non-i.i.d.-ness increases, this zigzagging curve is more obvious as shown in the figure, which we believe is a instinct feature of non-i.i.d. dataset in FL. However, the learning curves are more smoothing with gradient compression and error feedback. For example, for RD with *comp* = 0.9 and top-K with *comp* = 0.99, a better convergence curve than that of the original one without any compression is shown in Figure 5(d) and 6 (d) (p = 1). The intuition is that gradient compression method might filter some noises that leads to the instability of the learning curve due to model asynchrony among workers originated from the non-i.i.d. datasets and local steps. We believe it is worthing exploring further.
- As discussed in the paper, our algorithm is compatible with any gradient compression method in theory. However, it does not mean that any gradient compression method with any degree of compression is feasible in practice. Intuitively, when losing too much information, it is impossible to train a valid model. There seems a unsuccessful training for random dropping with *comp* = 0.99 in p = 1 case. So in practice, the trade-off between communication cost and convergence of the model always exists.

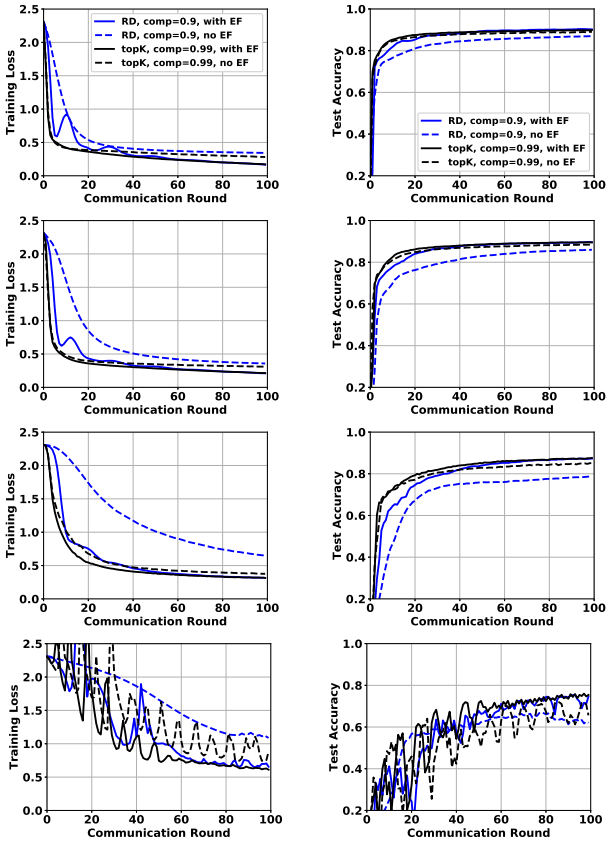
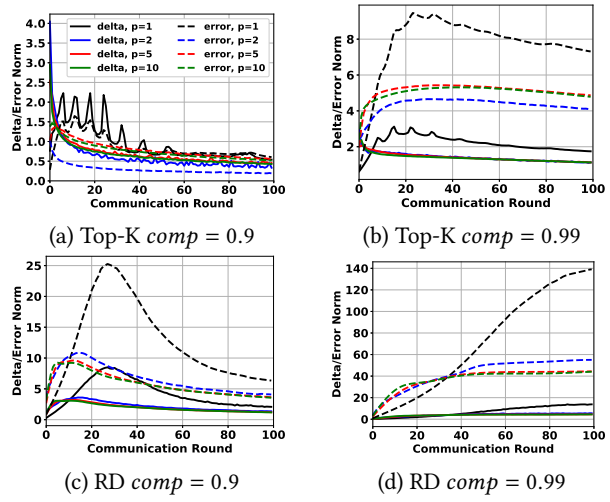


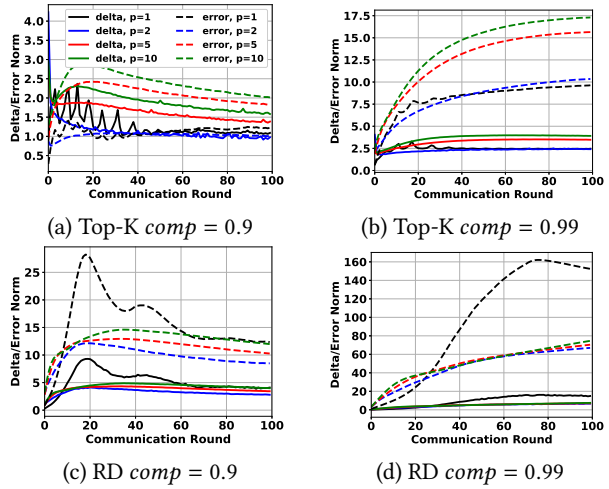
Figure 8: Training loss (top) and test accuracy (bottom) for the CNN model for Fashion-MNIST as comparison with respect to error feedback (EF). The non-i.i.d. levels are $p = 10, 5, 2, 1$ from top to bottom.



(a) Top-K $comp = 0.9$ (b) Top-K $comp = 0.99$

(c) RD $comp = 0.9$ (d) RD $comp = 0.99$

Figure 9: Mean of the norms of $\frac{1}{m}\sum_{i=1}^{100}\|\Delta_t^i\|^2$ and the error term $\frac{1}{m}\sum_{i=1}^{100}\|\mathbf{e}_t^i\|^2$ for the CNN model on MNIST.



(a) Top-K $comp = 0.9$ (b) Top-K $comp = 0.99$

(c) RD $comp = 0.9$ (d) RD $comp = 0.99$

Figure 10: Mean of the norms of $\frac{1}{m}\sum_{i=1}^{100}\|\Delta_t^i\|^2$ and the error term $\frac{1}{m}\sum_{i=1}^{100}\|\mathbf{e}_t^i\|^2$ for the CNN model on Fashion-MNIST.

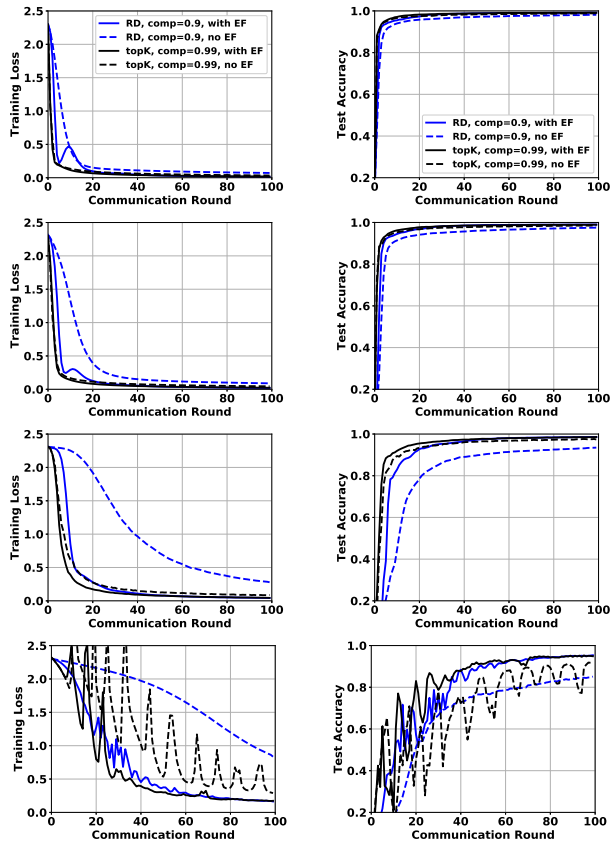


Figure 7: Training loss (top) and test accuracy (bottom) for the CNN model for MNIST as comparison with respect to error feedback (EF). The non-i.i.d. levels are $p = 10, 5, 2, 1$ from top to bottom.

B.2 Importance of Error Feedback:

As shown in Figure 7 and 8, error feedback helps the training in general. If naively applying gradient compression, huge amount of information will be lost, thus leading to relatively poor performance. With error feedback, the error term accumulates the information that is not transmitted to the server in the current communication round and then compensates the gradients in the next communication round. This is verified in Figure 9 and 10, which shows the mean of the norm of gradients changes $\frac{1}{m}\sum_{i=1}^{m}\|\triangle_t^i\|$ and the error term $\frac{1}{m}\sum_{i=1}^{m}\|\mathbf{e}_t^i\|$ for the total workers $m = 100$. One key observation is that the error term is usually several times of the gradients changes term in general. Under the same condition, the error term is much larger in high non-i.i.d. case. Another observation is that the error term is bounded under proper gradient compression methods and compression rate, which confirms our theoretical analysis. However, with too ambitious compression, the error term continues growing as shown in Figure 9(b) and 10(d). This is in accordance with the analysis in Figure 5 that learning curve is unstable. So in general, the error feedback could guarantee that not too much information is dropped due to the compression operator albeit with some delay.