EFSkip: A New Error Feedback with Linear Speedup for Compressed Federated Learning with Arbitrary Data Heterogeneity

Hongyan Bao[1*], **Pengwen Chen**[1], **Ying Sun**[2], **Zhize Li**[1†]

[1]Singapore Management University, Singapore
[2]Pennsylvania State University, USA

Abstract

Due to the communication bottleneck in distributed and decentralized federated learning applications, algorithms using compressed communication have attracted significant attention. The Error Feedback (EF) is a widely-studied compression framework for convergence with biased compressors such as top-k sparsification. Although various improvements have been obtained in recent years, the theoretical guarantee for EF-type framework is still limited. Previous works either 1) rely on strong assumptions such as bounded gradient/dissimilarity assumptions, thus can not deal with arbitrary data heterogeneity and also slow the convergence speed, or 2) can not enjoy linear speedup in the number of clients. In this work, we propose a new EFSkip framework which removes the strong assumptions to allow arbitrary data heterogeneity and enjoys linear speedup for significantly improving upon previous results. In particular, EFSkip achieves the complexity result $O(\frac{\sigma^2}{n\epsilon^4} + \frac{1}{\epsilon^2})$ while previous EF21 only obtains $O(\frac{\sigma^2}{\delta^3\epsilon^4} + \frac{1}{\delta\epsilon^2})$, i.e., EFSkip enjoys the linear speedup in the number of clients n (reducing the result linearly using more clients) and also removes the compression factor δ (matching the result without compression). We also show that EFSkip enjoys linear speedup and achieves faster convergence for nonconvex problems satisfying Polyak-Łojasiewicz (PL) condition. We believe that the new EFSkip framework will have a large impact on the communication- and computation-efficient distributed and decentralized federated learning.

1 Introduction

With the proliferation of mobile and edge devices, federated learning (FL) (McMahan et al. 2017; Konečný et al. 2016b) has recently emerged as a disruptive paradigm for training large-scale machine learning models over a vast amount of distributed and heterogeneous devices/clients. FL is usually modeled as a distributed optimization problem (Konečný et al. 2016a,b; McMahan et al. 2017; Kairouz et al. 2019; Zhao, Li, and Richtárik 2021; Wang et al. 2021), aiming to solve

$$\min_{x\in\mathbb{R}^d} \Big\{ f(x;\mathcal{D}) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} f_i(x;\mathcal{D}_i) \Big\}, \tag{1}$$

where $f_i(x;\mathcal{D}_i) \overset{\text{def}}{=} \mathbb{E}_{\xi_i\sim\mathcal{D}_i}[f_i(x;\xi_i)]$. Here, n denotes the number of clients and each client $i \in [n]$ has a local (nonconvex) loss function f_i associated with a local data distribution $\mathcal{D}_i$. In particular, $f_i(x;\xi_i)$ denotes the loss function of model x on a random data sample ξ_i on client i. For simplicity, we may use $f(x)$ and $f_i(x)$ to denote $f(x;\mathcal{D})$ and $f_i(x;\mathcal{D}_i)$, respectively.

In this work, we consider both *distributed* (where a central server exists and communicates with all n clients) and *decentralized* (where there is no server and clients can only communicate with their neighbors over a network) settings.

1.1 Communication compression and error feedback

A standard approach to solve (1) in the distributed setting is using gradient-type algorithms, i.e., each client computes the (stochastic) gradient of the model on its *local* dataset and transmits the gradient to the server, and the server aggregates all gradient information to update the model and then broadcasts the updated model to all clients. The steps are repeated until a stopping criterion is achieved. However, modern machine learning models are often overparameterized and have a huge number of parameters (Arora, Cohen, and Hazan 2018), for instance, the language model GPT-3 (Brown et al. 2020) has billions of parameters. The communication cost forms a main bottleneck of the distributed training system. A typical method for communication-efficient distributed learning is *compression* (Alistarh et al. 2017; Li et al. 2020), i.e., compress the communicated messages with fewer bits to reduce the communication cost. For example, each client can compress its local gradient and transmit the compressed message to the server. However, naively integrating the compression framework into the communication steps of distributed algorithms cannot guarantee convergence, as shown in the following example.

A counter-example. The number of clients is $n = 3$. The local loss functions are given by $f_i(x) = \frac{1}{2}x^\top\Lambda_i x$, where the Λ_is are diagonal matrices with $\Lambda_1 = \mathrm{diag}(-4,3,3)$, $\Lambda_2 = \mathrm{diag}(3,-4,3)$, and $\Lambda_3 = \mathrm{diag}(3,3,-4)$. The algorithm is initialized at $x^0 = (1,1,1)^\top$ and the compressor $\mathcal{C}(\cdot)$ is top-1 (See Example 1 for the definition of top-k sparsification). The local gradients for these three clients at x^0 are $\nabla f_1(x^0) = \Lambda_1 x^0 = (-4,3,3)^\top$, $\nabla f_2(x^0) =$

[*] The work of Hongyan Bao was conducted during her stay in Zhize Li's group at SMU.

[†] Corresponding author (zhizeli@smu.edu.sg).

$\Lambda_2 x^0 = (3,-4,3)^\top$, and $\nabla f_3(x^0) = \Lambda_3 x^0 = (3,3,-4)^\top$, and becomes $\mathcal{C}(\nabla f_1(x^0)) = (-4,0,0)^\top$, $\mathcal{C}(\nabla f_2(x^0)) = (0,-4,0)^\top, \mathcal{C}(\nabla f_3(x^0)) = (0,0,-4)^\top$ after the top-1 sparsification. The server aggregates the compressed gradients to update the model for the next round, yielding $x^1 = x^0 - \eta\frac{1}{3}\sum_{i=1}^{3}\mathcal{C}(\nabla f_i(x^0)) = (1+\frac{4}{3}\eta)x^0$. Then after t rounds the iterate $x^t = (1+\frac{4}{3}\eta)^t x^0$ diverges exponentially.

Error feedback. The error feedback (EF) (Seide et al. 2014; Stich, Cordonnier, and Jaggi 2018; Karimireddy et al. 2019), also known as error compensation, is a popular compression framework to fix the divergence issues. In EF, each client maintains a term recording the compression error, and in each round, instead of directly compressing its local gradient, an error-compensated one is compressed and sent to the server. More concretely, for the naive direct compression framework:

$$c_i^t = \mathcal{C}(\nabla f_i(x^t)) \qquad \text{(direct compression)} \qquad (2)$$

$$x^{t+1} = x^t - \eta\frac{1}{n}\sum_{i=1}^{n} c_i^t \qquad \text{(model update)} \qquad (3)$$

and for the EF compression framework:

$$c_i^t = \mathcal{C}(e_i^t + \eta\nabla f_i(x^t)) \qquad \text{(error compensation)} \qquad (4)$$

$$x^{t+1} = x^t - \frac{1}{n}\sum_{i=1}^{n} c_i^t \qquad \text{(model update)} \qquad (5)$$

$$e_i^{t+1} = e_i^t + \eta\nabla f_i(x^t) - c_i^t. \qquad \text{(compute the error)} \qquad (6)$$

1.2 Data heterogeneity and linear speedup

Although the EF compression framework can fix the divergence of direct compression, previous works are not able to deal well with the data heterogeneity in federated learning. To obtain theoretical results, they typically require some strong assumptions on data heterogeneity (see Tables 1–2 for algorithms using EF framework). Two widely-used assumptions are *bounded gradient assumption* $\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\|\nabla f_i(x;\xi_i)\|^2 \le G^2$, for all clients $i \in [n]$ and $\forall x \in \mathbb{R}^d$, and *bounded dissimilarity assumption* $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x;\mathcal{D}_i) - \nabla f(x;\mathcal{D})\|^2 \le \zeta^2, \forall x \in \mathbb{R}^d$. [1]

Richtárik, Sokolov, and Fatkhullin (2021) proposed a new EF21 framework to remove these bounded assumptions and thus *can allow arbitrary data heterogeneity* among the clients (see Tables 1–4 for algorithms using EF21). However, EF21 *cannot enjoy linear speedup* in the number of clients n unlike EF, and thus EF21 leads to a worse computation complexity, i.e., $O(\frac{\sigma^2}{\delta^3\epsilon^4})$ vs. the speedup term $O(\frac{\sigma^2}{n\epsilon^4})$ in EF. Several EF21 variants have also been proposed recently (Huang, Li, and Li 2024; Gao, Islamov, and Stich 2024).

2 Contributions

In order to remove these restrictive assumptions (for allowing arbitrary data heterogeneity) and enjoy the linear speedup (for reducing the computation complexity linearly using more clients) simultaneously, we propose a new EFSkip framework, which indeed achieve both of the design objectives (see the last row of Tables 1–4), i.e., EFSkip *can deal with arbitrary data heterogeneity and can enjoy the linear speedup in the number of clients simultaneously, making it a superior compression framework for communication- and computation-efficient distributed and decentralized learning.* Tables 1–4 present an overview of the comparison of EFSkip with previous works. We would like to highlight the following results:

Computation complexity. Although EF21 (Richtárik, Sokolov, and Fatkhullin 2021; Fatkhullin et al. 2021; Zhao et al. 2022) removes the bounded gradient/dissimilarity assumptions on data heterogeneity required by EF (i.e., removing the term $O(\frac{G}{\epsilon^3})$ or $O(\frac{\zeta}{\epsilon^3})$), EF21 cannot enjoy the linear speedup and thus leads to a worse computation complexity compared with EF (i.e., $O(\frac{\sigma^2}{\delta^3\epsilon^4})$ vs. $O(\frac{\sigma^2}{n\epsilon^4})$). However, EFSkip can achieve both goals simultaneously, i.e., removing the heterogeneity term $O(\frac{G}{\epsilon^3})$ or $O(\frac{\zeta}{\epsilon^3})$ and enjoying linear speedup $O(\frac{\sigma^2}{n\epsilon^4})$ in terms of the number of clients n. Moreover, compared with the complexity $O(\frac{\sigma^2}{\delta^3\epsilon^4} + \frac{1}{\delta\epsilon^2})$ of EF21, our $O(\frac{\sigma^2}{n\epsilon^4} + \frac{1}{\epsilon^2})$ of EFSkip also removes the dependency on the compression factor δ, and thus *matching the result without compression.* Note that $\delta \in (0,1]$ is usually equal to the *compression ratio* and no compression implies $\delta = 1$ (see Definition 1 and Example 1 in Section 3).

For nonconvex problems with PL condition (Tables 3–4), we also show that EFSkip can enjoy linear speedup in the number of clients n and removes the compression factor δ, i.e., $O(\frac{\sigma^2}{n\mu^2\epsilon}\log\frac{1}{\epsilon})$ of EFSkip vs. $O(\frac{\sigma^2}{\delta^3\mu^2\epsilon}\log\frac{1}{\epsilon})$ of EF21. Also, for PL setting in Tables 3–4 (note that there is no result for EF in this PL setting), both EF21 and EFSkip obtain better results compared with that without PL condition in Tables 1-2, and the results in PL setting can directly apply to strongly convex problems.

Communication complexity. Similar to computation complexity, EFSkip removes the terms G or ζ that depend on the gradient and data dissimilarity bound, and improves the order $O(\frac{G}{\delta\epsilon^3})$ or $O(\frac{\zeta}{\delta\epsilon^3})$ of methods using EF to $O(\frac{s}{\epsilon^2})$ (see Table 1). Here s stands for the skipsize, and is set to be $s = \log_{\frac{1}{1-\delta}}(n+2)$ in EFSkip. Note that $s = 1$ if no compression (i.e., $\delta = 1$) is applied. In practice, a small constant s is enough, e.g., $s = 4$.

2.1 EFSkip vs. the previous EF21 framework

Similar to the comparison between direct compression and EF provided at the end of Section 1.1, i.e., (2)–(6), in this section we compare the EF21 framework (Richtárik, Sokolov, and Fatkhullin 2021) with our EFSkip framework (see Figure 1).

[1] These bounded assumptions are quite strong and even do not hold for simple quadratic functions, e.g., linear regression. Let us just consider a 1-dimensional linear regression $f_i(x) = (a_i x - b_i)^2$ and $f(x) = \frac{1}{n}\sum_{i=1}^{n}(a_i x - b_i)^2$, then the gradient $\|\nabla f_i(x)\|^2 = 4(a_i^2 x - a_i b_i)^2$ cannot be bounded by a constant G^2 for $\forall x \in \mathbb{R}$, and the dissimilarity $\|\nabla f_i(x) - \nabla f(x)\|^2 = 4((a_i^2 - \frac{1}{n}\sum_{i=1}^{n} a_i^2)x^2 - a_i b_i + \frac{1}{n}\sum_{i=1}^{n} a_i b_i)^2$ also cannot be bounded by a constant ζ^2 for $\forall x \in \mathbb{R}$.

For the previous EF21 compression framework (Richtárik, Sokolov, and Fatkhullin 2021):

$$x^{t+1} = x^t - \eta(g^{t-1} + \frac{1}{n}\sum_{i=1}^{n} c_i^{t-1}) \qquad \text{(model update)} \tag{7}$$

$$g_i^t = g_i^{t-1} + c_i^{t-1} \qquad \text{(update local shift)} \tag{8}$$

$$\Delta_i^0 = \tilde{\nabla}_b f_i(x^{t+1}) - g_i^t \qquad \text{(compute shifted local gradient information)} \tag{9}$$

$$c_i^t = \mathcal{C}(\Delta_i^0), \qquad \text{(communication compression)} \tag{10}$$

where $\tilde{\nabla}_b f_i(x^{t+1}) := \frac{1}{b}\sum_{j=1}^{b} \nabla f_i(x^{t+1}; \xi_{i,j})$ denotes the stochastic gradient computed by a minibatch with size b drawn from local data distribution $\mathcal{D}_i$.

For our new EFSkip compression framework:

if $t \bmod s = 0$ **then** (compute stochastic gradient once every s rounds) (11)

$$x^{t+1} = x^{t+1-s} - \eta(g^{t-s} + \frac{1}{n}\sum_{i=1}^{n} c_i^{t-1}) \qquad \text{(model update)} \tag{12}$$

$$g_i^t = g_i^{t-1} + c_i^{t-1} \qquad \text{(update local shift)} \tag{13}$$

$$\Delta_i^0 = \tilde{\nabla}_b f_i(x^{t+1}) - g_i^t \qquad \text{(compute shifted local gradient information)} \tag{14}$$

$$c_i^t = \mathcal{C}(\Delta_i^0) \qquad \text{(communication compression)} \tag{15}$$

else (skip gradient computation via reusing Δ_i^0) (16)

$$c_i^t = c_i^{t-1} + \mathcal{C}(\Delta_i^0 - c_i^{t-1}). \qquad \text{(communication compression)} \tag{17}$$

Figure 1: EF21 Framework vs. Our EFSkip Framework

Compression framework	**Algorithm**	**Communication complexity** [1] (#communication rounds)	**Computation complexity** (#stochastic gradients)	**Stong assumption** [2] **on data heterogeneity**	**Linear speedup in #clients n**
Error Feedback (EF)	Qsparse-SGD (Basu et al. 2019)	$O\Big(\frac{\sigma^2}{nb\epsilon^4} + \frac{nbG^2}{\delta^2\epsilon^2}\Big)$	$O\Big(\frac{\sigma^2}{n\epsilon^4} + \frac{nb^2G^2}{\delta^2\epsilon^2}\Big)$	bounded gradient	✓ if $n \le \frac{\delta\sigma}{bG\epsilon}$
	CSER (Xie et al. 2020)	$O\Big(\frac{\sigma^2}{n\epsilon^4} + \frac{G}{\delta\epsilon^3} + \frac{1}{\epsilon^2}\Big)$	$O\Big(\frac{\sigma^2}{n\epsilon^4} + \frac{G}{\delta\epsilon^3} + \frac{1}{\epsilon^2}\Big)$	bounded gradient	✓ if $n \le \frac{\delta\sigma^2}{G\epsilon}$
	NEOLITHIC (Huang et al. 2022)	$O\Big(\frac{\sigma^2}{n\epsilon^4} + \frac{\zeta\theta R}{\delta\epsilon^3} + \frac{R}{\epsilon^2}\Big)$	$O\Big(\frac{\sigma^2}{n\epsilon^4} + \frac{\zeta\theta R}{\delta\epsilon^3} + \frac{R}{\epsilon^2}\Big)$ [3]	bounded dissimilarity	✓ if $n \le \frac{\delta\sigma^2}{\zeta\theta R\epsilon}$
EF21	EF21-SGD (Richtárik, Sokolov, and Fatkhullin 2021; Fatkhullin et al. 2021)	$O\Big(\frac{1}{\delta\epsilon^2}\Big)$	$O\Big(\frac{\sigma^2}{\delta^3\epsilon^4} + \frac{1}{\delta\epsilon^2}\Big)$	No	✗
EFSkip (this paper)	EFSkip-SGD (Theorem 1)	$O\Big(\frac{s}{\epsilon^2}\Big)$ [4]	$O\Big(\frac{\sigma^2}{n\epsilon^4} + \frac{1}{\epsilon^2}\Big)$ [5]	No	✓

Table 1: Communication and computation complexity results of algorithms for finding an ϵ-solution $\mathbb{E}[\|\nabla f(\widehat{x})\|^2] \le \epsilon^2$ of nonconvex problem (1) in *distributed setting*.

[1] In this column of communication complexity, we list the number of communicaiton rounds since all algorithms use the same compression operator in Definition 1, i.e., the communication bits of the compressed message for each round are the same. Note that communication complexity = communication rounds × communication bits per round.

[2] Here bounded gradient assumption is $\mathbb{E}_{\xi_i\sim\mathcal{D}_i}\|\nabla f_i(x;\xi_i)\|^2 \le G^2$, for all clients $i \in [n]$ and $\forall x \in \mathbb{R}^d$, and bounded disimilarity assumption is $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x;\mathcal{D}_i) - \nabla f(x;\mathcal{D})\|^2 \le \zeta^2$, for $\forall x \in \mathbb{R}^d$, i.e., the local gradient of loss function on clients are close to the global gradient. No means that no additional assumption is required, i.e., allowing *arbitrary data heterogeneity* among the clients.

[3] The θ and $R \ge 1$ are parameters such that $\theta := 4(1-\delta)^R$.

[4] Here the skipsize $s = \log_{\frac{1}{1-\delta}}(n+2)$ for EFSkip-SGD (Algorithm 1) in distributed setting. Note that $s = 1$ if no compression (i.e., $\delta = 1$) is applied. In practice, a small constant s is enough, e.g., $s = 4$.

[5] **Main result:** EFSkip-SGD enjoys the *linear speedup* in the number of clients n (reducing the computation complexity linearly using more clients) and also removes the compression factor δ (matching the result without compression). Note that $\delta \in (0, 1]$ is usually equal to the *compression ratio* and no compression implies $\delta = 1$ (see Definition 1). In particular, top-k or random-k sparsification satisfies Definition 1 with $\delta = \frac{k}{d}$, where d is the dimension of model (i.e., total number of parameters) in problem (1).

Compression framework	**Algorithm**	**Communication complexity** (#communication rounds)	**Computation complexity** (#stochastic gradients)	**Stong assumption on data heterogeneity**	**Linear speedup in #clients n**
Error Feedback (EF)	SQuARM-SGD (Singh et al. 2021)	$O\left(\frac{\sigma^4}{n\epsilon^4} + \frac{nG^2}{\delta^2\epsilon^2}\right)$	$O\left(\frac{\sigma^4}{n\epsilon^4} + \frac{nG^2}{\delta^2\epsilon^2}\right)$	bounded gradient	✓ if $n \leq \frac{\delta\sigma^2}{G\epsilon}$
	DeepSqueeze (Tang et al. 2019)	$O\left(\frac{\sigma^2}{n\epsilon^4} + \frac{\zeta}{\delta^{3/2}\epsilon^3} + \frac{1}{\epsilon^2}\right)$	$O\left(\frac{\sigma^2}{n\epsilon^4} + \frac{\zeta}{\delta^{3/2}\epsilon^3} + \frac{1}{\epsilon^2}\right)$	bounded dissimilarity	✓ if $n \leq \frac{\delta^{3/2}\sigma^2}{\zeta\epsilon}$
	CHOCO-SGD (Koloskova et al. 2020)	$O\left(\frac{\sigma^2}{n\epsilon^4} + \frac{G}{\delta\epsilon^3} + \frac{1}{\epsilon^2}\right)$	$O\left(\frac{\sigma^2}{n\epsilon^4} + \frac{G}{\delta\epsilon^3} + \frac{1}{\epsilon^2}\right)$	bounded gradient	✓ if $n \leq \frac{\delta\sigma^2}{G\epsilon}$
EF21	BEER (Zhao et al. 2022)	$O\left(\frac{1}{\delta\epsilon^2}\right)$	$O\left(\frac{\sigma^2}{\delta^2\epsilon^4} + \frac{1}{\delta\epsilon^2}\right)$	No	✗
EFSkip (this paper)	EFSkip-BEER (Theorem 2)	$O\left(\frac{s}{\delta\epsilon^2}\right)$ [1]	$O\left(\frac{\sigma^2}{n\delta\epsilon^4} + \frac{1}{\delta\epsilon^2}\right)$ [2]	No	✓

Table 2: Communication and computation complexity results of algorithms for finding an ϵ-solution $\mathbb{E}[\|\nabla f(\widehat{x})\|^2] \leq \epsilon^2$ of nonconvex problem (1) in *decentralized setting*.

[1] Here the skipsize $s = \log_{\frac{1}{1-\delta}} c_s$, where c_s is an absolute constant, for EFSkip-BEER (Algorithm 2) in decentralized setting.

[2] **Main result:** EFSkip-BEER also enjoys the linear speedup in the number of clients n for this decentralized setting. Recall that $\delta \in (0, 1]$ is usually equal to the *compression ratio* and no compression implies $\delta = 1$.

Compression framework	**Algorithm**	**Communication complexity** (#communication rounds)	**Computation complexity** (#stochastic gradients)	**Linear speedup in #clients n**
EF21	EF21-SGD (Richtárik, Sokolov, and Fatkhullin 2021; Fatkhullin et al. 2021)	$O\left(\frac{1}{\delta\mu} \log \frac{1}{\epsilon}\right)$	$O\left((\frac{\sigma^2}{\delta^3\mu^2\epsilon} + \frac{1}{\delta\mu}) \log \frac{1}{\epsilon}\right)$	✗
EFSkip (this paper)	EFSkip-SGD (Theorem 3)	$O\left(\frac{s}{\mu} \log \frac{1}{\epsilon}\right)$	$O\left((\frac{\sigma^2}{n\mu^2\epsilon} + \frac{1}{\mu}) \log \frac{1}{\epsilon}\right)$	✓

Table 3: Communication and computation complexity results of algorithms for finding an ϵ-solution $\mathbb{E}[f(\widehat{x}) - f^*] \leq \epsilon$ of nonconvex problem (1) in *distributed setting under PL condition* (i.e., the global function $f(x)$ satisfies PL condition (10)). Note that μ-strong convexity implies μ-PL. As a result, all results obtained under PL condition directly hold for strongly convex problems.

[1] Here the skipsize $s = \log_{\frac{1}{1-\delta}}(2n + 4)$ for EFSkip-SGD in the distributed PL setting.

[2] **Main result:** EFSkip-SGD also enjoys the linear speedup in the number of clients n for the distributed PL setting. Recall that $\delta \in (0, 1]$ is usually equal to the *compression ratio* and no compression implies $\delta = 1$.

Compression framework	**Algorithm**	**Communication complexity** (#communication rounds)	**Computation complexity** (#stochastic gradients)	**Linear speedup in #clients n**
EF21	BEER (Zhao et al. 2022)	$O\left(\frac{1}{\delta\mu} \log \frac{1}{\epsilon}\right)$	$O\left((\frac{\sigma^2}{\delta^2\mu^2\epsilon} + \frac{1}{\delta\mu}) \log \frac{1}{\epsilon}\right)$	✗
EFSkip (this paper)	EFSkip-BEER (Theorem 4)	$O\left(\frac{s}{\delta\mu} \log \frac{1}{\epsilon}\right)$	$O\left((\frac{\sigma^2}{n\delta\mu^2\epsilon} + \frac{1}{\delta\mu}) \log \frac{1}{\epsilon}\right)$	✓

Table 4: Communication and computation complexity results of algorithms for finding an ϵ-solution $\mathbb{E}[f(\widehat{x}) - f^*] \leq \epsilon$ of nonconvex problem (1) in *decentralized setting under PL condition*. Similarly, all results obtained under PL condition directly hold for strongly convex problems.

[1] Here the skipsize $s = \log_{\frac{1}{1-\delta}} c_s$, where c_s is an absolute constant, for EFSkip-BEER in the decentralized PL setting.

[2] **Main result:** EFSkip-BEER also enjoys the linear speedup in the number of clients n for this decentralized PL setting.

In particular, EFSkip can reduce to EF21 with skipsize $s = 1$ since $t \bmod 1$ is always equal to 0. With our new EFSkip compression framework, we propose two algorithms i) EFSkip-SGD (Algorithm 1) for the nonconvex distributed setting in Section 4; ii) EFSkip-BEER (Algorithm 2) for the nonconvex decentralized setting in Section 5.

3 Preliminaries

Let $[n]$ denote the set $\{1, 2, \cdots, n\}$ and $\|\cdot\|$ denote the Euclidean norm of a vector. Let $\mathbf{1}$ denote the all-ones vector. Let $\langle u, v\rangle$ denote the inner product of vectors u and v, and $u \odot v$ denote the element-wise product. Let $a \mod b$ denote the remainder of a divided by b. Let $f^* := \min_{x\in\mathbb{R}^d} f(x) > -\infty$ denote the optimal value of the objective function in (1). We use $O(\cdot)$ to hide the absolute constants.

Compression, in the form of sparsification or quantization, can be used to reduce the communication cost. We now introduce the notion of a general *biased compression operator* widely used in many distributed and federated learning algorithms, e.g., (Stich, Cordonnier, and Jaggi 2018; Koloskova et al. 2020; Richtárik, Sokolov, and Fatkhullin 2021; Fatkhullin et al. 2021; Richtárik et al. 2022).

Definition 1 (Compression operator) *A (randomized) map $\mathcal{C} : \mathbb{R}^d \mapsto \mathbb{R}^d$ is a biased compression operator if there exists a $0 < \delta \le 1$, such that for all $x \in \mathbb{R}^d$,*

$$\mathbb{E}\left[\|\mathcal{C}(x) - x\|^2\right] \le (1-\delta)\|x\|^2. \tag{7}$$

In particular, no compression ($\mathcal{C}(x) \equiv x$) implies $\delta = 1$.

Compared with an *unbiased compression operator* used in, e.g., (Alistarh et al. 2017; Khirirat, Feyzmahdavian, and Johansson 2018; Mishchenko et al. 2019; Li and Richtárik 2020; Li and Richtárik 2021; Zhao et al. 2021), the general compression operator in Definition 1 does not impose the additional constraint such that $\mathbb{E}[\mathcal{C}(x)] = x$. Moreover, the unbiased compression operator can be converted into a biased one satisfying Definition 1, i.e., for any unbiased compression operator $\mathcal{C}' : \mathbb{R}^d \mapsto \mathbb{R}^d$ that satisfies $\mathbb{E}[\mathcal{C}'(x)] = x$ and $\mathbb{E}[\|\mathcal{C}'(x) - x\|^2] \le \omega\|x\|^2$, we can construct a *biased* compression operator $\mathcal{C} : \mathcal{C}(x) = \frac{\mathcal{C}'(x)}{1+\omega}$ and the new compression operator satisfies Definition 1 with $\delta = \frac{1}{1+\omega}$. Note that ω can be larger than 1 for unbiased compressors. Thus, Definition 1 is a generalization of the unbiased compression.

Example 1 *The top-k sparsification keeps the coordinates with the top-k largest absolute values, i.e., $\mathrm{top}_k(x) := x \odot u_x$, where $u_x \in \{0,1\}^d$ satisfying $\|u_x\|_1 = k$ and $u_x(i) = 1$ iff $|x_i| \ge |x_j|$ for all j with $u_x(j) = 0$. In particular, top_k is a δ-compression operator with $\delta = \frac{k}{d}$, i.e., satifies (7) as*

$$\mathbb{E}\left[\|\mathrm{top}_k(x) - x\|^2\right] \le \left(1 - \frac{k}{d}\right)\|x\|^2. \tag{8}$$

Besides, random-k that randomly keeps k coordinates is also a δ-compression operator with $\delta = \frac{k}{d}$.

4 EFSkip for Distributed Setting

For the nonconvex problem (1) in the distributed setting where a central server exists and communicates with all n clients, we propose the EFSkip-SGD (Algorithm 1) and provide its theoretical results.

4.1 EFSkip-SGD algorithm

Now we formally describe the distributed SGD with our EFSkip compression framework as EFSkip-SGD in Algorithm 1. We would like to highlight that the clients only compute their local stochastic gradients once every s rounds.

4.2 Theoretical results of EFSkip-SGD

Before providing the theoretical results of EFSkip-SGD, we first state the following standard assumptions (Nesterov 2004; Ghadimi and Lan 2013; Li et al. 2021; Li, Hanzely, and Richtárik 2021; Li and Li 2022).

Assumption 1 (Smoothness) *The local function f_i is L-smooth, i.e., $\forall x, y \in \mathbb{R}^d$,*

$$\|\nabla f_i(x) - \nabla f_i(y)\| \le L\|x - y\|.$$

We point out that Assumption 1 can be relaxed to the average smoothness assumption $\mathbb{E}_i[\|\nabla f_i(x) - \nabla f_i(y)\|^2] \le L^2\|x - y\|^2$ which does not affect the results obtained in this work.

During the training, the clients are allowed to compute stochastic gradients sampled from their local data distributions (see Line 9 of Algorithm 1). We make the following standard assumption for the stochastic gradients.

Assumption 2 (Stochastic gradient) *Let $\tilde{\nabla} f_i(x) := \nabla f_i(x; \xi_i)$ denote a stochastic gradient computed by client i via a sample ξ_i drawn i.i.d. from its local data distribution $\mathcal{D}_i$, we have*

$$\mathbb{E}\|\tilde{\nabla} f_i(x) - \nabla f_i(x)\|^2 \le \sigma^2. \tag{9}$$

Let $\tilde{\nabla}_b f_i(x) := \frac{1}{b}\sum_{j=1}^b \nabla f_i(x; \xi_{i,j})$ denote the stochastic gradient computed by a minibatch with size b drawn i.i.d. from $\mathcal{D}_i$, it is not hard to see $\mathbb{E}\|\tilde{\nabla}_b f_i(x) - \nabla f_i(x)\|^2 \le \frac{\sigma^2}{b}$.

Now we provide the theoretical results of EFSkip-SGD for solving the distributed nonconvex problem (1).

Theorem 1 *Suppose that Assumptions 1 and 2 hold. Let stepsize $\eta \le \frac{1}{(1+\sqrt{4/n})L}$, minibatch size $b = \frac{20\sigma^2}{n\epsilon^2}$, and skipsize $s = \log_{\frac{1}{1-\delta}}(n+2)$, then the communication complexity and computation complexity for EFSkip-SGD to find an ϵ-solution (i.e., $\mathbb{E}[\|\nabla f(\widehat{x}^T)\|^2] \le \epsilon^2$) of distributed nonconvex problem (1) are as follows:*

i) Communication complexity is computed as the total number of communication rounds (denoted as $\#\mathrm{rounds}$) times the communicated bits per round (denoted as d_δ):

$$\#\mathrm{rounds} = O\left(\frac{s}{\epsilon^2}\right), \text{ where skipsize } s = \log_{\frac{1}{1-\delta}}(n+2),[2]$$

and the communicated bits per round d_δ depends on the compression operator (Definition 1) used by Algorithm 1, e.g., for top-k sparsification (see Example 1) $d_\delta = O(k)$. Note that no compression $\mathcal{C}(x) \equiv x$ implies $d_\delta = O(d)$.

ii) Computation complexity is computed as the total number of stochastic gradient computations (denoted as $\#\mathrm{gradients}$):

$$\#\mathrm{gradients} = O\left(\frac{\sigma^2}{n\epsilon^4} + \frac{1}{\epsilon^2}\right).$$

[2] Note that we can simply set $s = 1$ if no compression (i.e., $\delta = 1$) is applied.

Algorithm 1: EFSkip-SGD

```
Input: initial point $x^0$, stepsize $\eta$, minibatch size $b$, skipsize $s$
Initialize $c_i^{-1} = 0$, $g_i^{-s} = \mathcal{C}(\tilde{g}_i^0)$, $\tilde{g}_i^0 = \frac{1}{b}\sum_{j=1}^{b} \nabla f_i(x^0; \xi_{i,j})$, $c^{-1} = \frac{1}{n}\sum_{i=1}^{n} c_i^{-1}$, $g^{-s} = \frac{1}{n}\sum_{i=1}^{n} g_i^{-s}$, $x^{-s+1} = x^0$
for $t = 0, 1, 2, \ldots, T-1$ do
  if $t \bmod s = 0$ then
    $g^t = g^{t-s} + c^{t-1}$
    Server updates $x^{t+1} = x^{t+1-s} - \eta g^t$ and broadcasts $x^{t+1}$ to all clients
  end if
  for each client $i \in [n]$ do in parallel
    if $t \bmod s = 0$ then
      Compute stochastic gradients $\tilde{g}_i^{t+1} = \frac{1}{b}\sum_{j=1}^{b} \nabla f_i(x^{t+1}; \xi_{i,j})$
      $\Delta_i^0 = \tilde{g}_i^{t+1} - g_i^t$, where $g_i^t = g_i^{t-s} + c_i^{t-1}$        //this gradient information $\Delta_i^0$ will be reused for $s$ rounds
      Compress $c_i^t = \mathcal{C}(\Delta_i^0)$ and send it to the server
    else                                                       //skip gradient computations via reusing $\Delta_i^0$
      Compress $\mathcal{C}(\Delta_i^0 - c_i^{t-1})$ and send it to the server
      Update $c_i^t = c_i^{t-1} + \mathcal{C}(\Delta_i^0 - c_i^{t-1})$
    end if
  end for
  Server aggregates compressed info $c^t = \begin{cases} \frac{1}{n}\sum_{i=1}^{n} \mathcal{C}(\Delta_i^0) & \text{if } t \bmod s = 0 \\ c^{t-1} + \frac{1}{n}\sum_{i=1}^{n} \mathcal{C}(\Delta_i^0 - c_i^{t-1}) & \text{if } t \bmod s \neq 0 \end{cases}$
end for
```

5 EFSkip for Decentralized Setting

For the nonconvex problem (1) in decentralized setting where there is no server and clients can only communicate with their neighbors over a prescribed network topology, we propose the EFSkip-BEER (Algorithm 2) and provide its theoretical results.

5.1 EFSkip-BEER algorithm

In the decentralized setting, the clients can only communicate with their local neighbors over a prescribed communication network, modeled by an undirected graph $\mathcal{G}([n], E)$. Here, each node $i \in [n]$ represents a client, and $(i, j) \in E$ if there is a communication link between client i and j.

Since clients cannot synchronize in one shot, we let each client i hold a local replicate of the parameter x, denoted as x_i and use $X := [x_1, x_2, ..., x_n] \in \mathbb{R}^{d\times n}$ to denote the parameters of all clients. Similarly, the collection of local gradients and stochastic minibatches computed by the clients are respectively given by $\nabla F(X) := [\nabla f_1(x_1), \nabla f_2(x_2), ..., \nabla f_n(x_n)] \in \mathbb{R}^{d\times n}$ and $\tilde{\nabla}_b F(X) := [\tilde{\nabla}_b f_1(x_1), \tilde{\nabla}_b f_2(x_2), ..., \tilde{\nabla}_b f_n(x_n)] \in \mathbb{R}^{d\times n}$. Information sharing across the clients over the network is implemented mathematically by the use of a mixing matrix W, defined in accordance with the network topology with $w_{ij} \geq 0$ for any $(i, j) \in E$ and $w_{ij} = 0$ for all $(i, j) \notin E$.

With these notations, we formally introduce EFSkip-BEER in Algorithm 2. At each round t, the algorithm maintains X^t and an estimator of the global gradients V^t, i.e., each i-th column of V^t estimates $\nabla f(x_i^t)$ that is not computable by client i. To facilitate the compression operation, EFSkip-BEER creates two extra variables H^t and G^t that serve respectively as surrogates of X^{t+1-s} and V^{t+1-s}. We start describing the algorithm from a computation round t (i.e., $t \bmod s = 0$).

Model and gradient update. The clients update their local models using a perturbed average consensus mechanism (Line 5 of Algorithm 2). Since $H^t \approx X^{t+1-s}$, the first two terms correspond to averaging the local parameters using the graph Laplacian $I - W$. The third term is a descent step with gradient estimator $V^{t+1-s} \approx \nabla F(X^{t+1-s})$. With the newly computed minibatch gradient at X^{t+1}, the estimator V is updated in Line 7 of Algorithm 2 using a gradient tracking scheme (Zhu and Martínez 2010; Di Lorenzo and Scutari 2016; Nedić, Olshevsky, and Shi 2017; Qu and Li 2017), with $G^t \approx V^{t+1-s}$ and the innovation term being the difference of the two last computed gradients.

Compression. Right after computing the new X^{t+1}, we invoke the EFSkip compression and compute the increment Δ_h^0 in Line 10 of Algorithm 2. To better illustrate the subsequent communication steps, we combine Line 15 and 4 of Algorithm 2 and rewrite $H^{t+s}W$, which will be used in the next computation round $t + s$, as:

$$H^{t+s}W = H^tW + \underbrace{\Big(C_h^t + \sum_{\tau=1}^{s-1} \mathcal{C}(\Delta_h^0 - C_h^{t-1+\tau})\Big)W}_{\text{network communication}}.$$

This shows in the communication rounds, clients are sending to their neighbors the compressed quantities C_h^t and $\{\mathcal{C}(\Delta_h^0 - C_h^{t-1+\tau})\}_{\tau=1}^{s-1}$ using mixing matrix W, with C_h recording the part of Δ_h^0 that has been sent. Clearly, the process is implementable over a network.

5.2 Theoretical results of EFSkip-BEER

We first state the assumptions and then provide the theoretical results of EFSkip-BEER in Theorem 2. The local functions f_is are nonconvex satisfying the following average smoothness assumption.

Algorithm 2: EFSkip-BEER

Input: initial point x^0, stepsize η, mixing stepsize γ, minibatch size b, skipsize s

```
Initialize $X^0 = x^0\mathbf{1}^\top$, $V^0 = \nabla F(X^0)$, $H^{-s} = 0$, $G^{-s} = 0$, $C_h^{-1} = 0$, $C_g^{-1} = 0$, $X^{-s+1} = X^0$, $V^{-s+1} = V^0$
for $t = 0, 1, 2, ..., T-1$ do
    if $t$ mod $s = 0$ then
        $H^t = H^{t-s} + C_h^{t-1}$
        $X^{t+1} = X^{t+1-s} + \gamma H^t(W - I) - \eta V^{t+1-s}$
        $G^t = G^{t-s} + C_g^{t-1}$
        $V^{t+1} = V^{t+1-s} + \gamma G^t(W - I) + \widetilde{\nabla}_b F(X^{t+1}) - \widetilde{\nabla}_b F(X^{t+1-s})$
    end if
    if $t$ mod $s = 0$ then
        $\Delta_h^0 = X^{t+1} - H^t$, where $H^t = H^{t-s} + C_h^{t-1}$
        $C_h^t = \mathcal{C}(\Delta_h^0)$
        $\Delta_g^0 = V^{t+1} - G^t$, where $G^t = G^{t-s} + C_g^{t-1}$
        $C_g^t = \mathcal{C}(\Delta_g^0)$
    else
        $C_h^t = C_h^{t-1} + \mathcal{C}(\Delta_h^0 - C_h^{t-1})$
        $C_g^t = C_g^{t-1} + \mathcal{C}(\Delta_g^0 - C_g^{t-1})$
    end if
end for
```

Assumption 3 (Average smoothness) *The local function f_i is average L-smooth, i.e., $\forall x, y \in \mathbb{R}^d$,*

$$\mathbb{E}\|\tilde{\nabla} f_i(x) - \tilde{\nabla} f_i(y)\|^2 \le L^2\|x - y\|^2.$$

Also, we make the following standard assumption on the mixing matrix (Nedić, Olshevsky, and Rabbat 2018).

Assumption 4 (Mixing matrix) *The mixing matrix $W = [w_{ij}] \in [0,1]^{n\times n}$ is symmetric ($W^\top = W$) and doubly stochastic($W\mathbf{1} = \mathbf{1}, \mathbf{1}^\top W = \mathbf{1}^\top$). Let its eigenvalues be $1 = |\lambda_1(W)| \ge |\lambda_2(W)| \ge \cdots \ge |\lambda_n(W)|$. The spectral gap satisfies*

$$\rho := 1 - |\lambda_2(W)| \in (0, 1].$$

The theoretical results of EFSkip-BEER for solving the decentralized nonconvex problem (1) in Theorem 2.

Theorem 2 *Suppose that Assumptions 2–4 hold. Let stepsize $\eta = c_\eta \delta\rho^2/L$, mixing stepsize $\gamma = c_\gamma \delta\rho$, minibatch size $b = \frac{2\sigma^2}{n\epsilon^2}$, skipsize $s = \log_{\frac{1}{1-\delta}} c_s$, where c_s, c_γ, c_η are some absolute constants, then the communication complexity and computation complexity for* EFSkip-BEER *to find an ϵ-solution (i.e., $\mathbb{E}[\|\nabla f(\widehat{x}^T)\|^2] \le \epsilon^2$) of decentralized nonconvex problem* (1) *are as follows:*

i) Communication complexity:

$$\#\text{rounds} = O\left(\frac{s}{\rho^2\delta\epsilon^2}\right), \textit{where } s = \log_{\frac{1}{1-\delta}} c_s.^3$$

ii) Computation complexity:

$$\#\text{gradients} = O\left(\frac{\sigma^2}{n\rho^2\delta\epsilon^4} + \frac{1}{\rho^2\delta\epsilon^2}\right).$$

6 Faster Convergence Under PL Condition

In this section, we prove faster convergence results of EFSkip-SGD (Algorithm 1) and EFSkip-BEER (Algorithm 2) for solving both distributed and decentralized nonconvex problem (1), when the global nonconvex function f in problem (1) satisfies the following Polyak-Łojasiewicz (PL) condition (Polyak 1963).

Assumption 5 (PL condition) *There exists some constant $\mu > 0$ such that for any $x \in \mathbb{R}^d$,*

$$\|\nabla f(x)\|^2 \ge 2\mu(f(x) - f^*). \tag{10}$$

It is worth noting that PL condition does not imply (strong) convexity of $f(x)$. For example, $f(x) = x^2 + 3\sin^2 x$ is a nonconvex function but it satisfies PL condition with $\mu = 1/32$. However, μ-strong convexity implies μ-PL. As a result, all results obtained under PL condition directly hold for strongly convex problems.

Theorem 3 *Suppose that Assumptions 1, 2 and 5 hold. Let stepsize $\eta \le \min\left\{\frac{1}{(1+\sqrt{4/n})L}, \frac{1}{2\mu}\right\}$, minibatch size $b = \frac{12\sigma^2}{n\mu\epsilon}$, and skipsize $s = \log_{\frac{1}{1-\delta}}(2n+4)$, then the communication complexity and computation complexity for* EFSkip-SGD *to find an ϵ-solution (i.e., $\mathbb{E}[f(x^T) - f^*] \le \epsilon$) of distributed nonconvex problem* (1) *under PL condition are as follows:*

i) Communication complexity:

$$\#\text{rounds} = O\left(\frac{s}{\mu}\log\frac{1}{\epsilon}\right), \textit{where } s = \log_{\frac{1}{1-\delta}}(2n+4),$$

ii) Computation complexity:

$$\#\text{gradients} = O\left(\left(\frac{\sigma^2}{n\mu^2\epsilon} + \frac{1}{\mu}\right)\log\frac{1}{\epsilon}\right).$$

Theorem 4 *Suppose that Assumptions 2–5 hold. Let stepsize $\eta = c_\eta\delta\rho^2/L$, mixing stepsize $\gamma = c_\gamma\delta\rho$, minibatch size $b = \frac{\sigma^2}{n\mu\epsilon}$, skipsize $s = \log_{\frac{1}{1-\delta}} c_s$, where c_s, c_γ, c_η are some absolute constants, then the communication complexity and computation complexity for* EFSkip-BEER *to find an ϵ-solution (i.e., $\mathbb{E}[f(x^T) - f^*] \le \epsilon$) of decentralized nonconvex problem* (1) *under PL condition are as follows:*

i) Communication complexity:

$$\#\text{rounds} = O\left(\frac{s}{\rho^2\delta\mu}\log\frac{1}{\epsilon}\right), \textit{where } s = \log_{\frac{1}{1-\delta}} c_s$$

ii) Computation complexity:

$$\#\text{gradients} = O\left(\left(\frac{\sigma^2}{n\rho^2\delta\mu^2\epsilon} + \frac{1}{\rho^2\delta\mu}\right)\log\frac{1}{\epsilon}\right).$$

Similar to Theorems 1–2, we can also simply set the skipsize $s = 1$ if no compression is applied for Theorems 3–4.

[3] We can also simply set the skipsize $s = 1$ if no compression (i.e., $\delta = 1$) is applied.

Acknowledgments

This research was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

Alistarh, D.; Grubic, D.; Li, J.; Tomioka, R.; and Vojnovic, M. 2017. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, 1709–1720.

Arora, S.; Cohen, N.; and Hazan, E. 2018. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning*, 244–253. PMLR.

Basu, D.; Data, D.; Karakus, C.; and Diggavi, S. 2019. Qsparse-local-SGD: Distributed SGD with quantization, sparsification and local computations. *Advances in Neural Information Processing Systems*, 32.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Di Lorenzo, P.; and Scutari, G. 2016. Next: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2): 120–136.

Fatkhullin, I.; Sokolov, I.; Gorbunov, E.; Li, Z.; and Richtárik, P. 2021. EF21 with bells & whistles: Practical algorithmic extensions of modern error feedback. *arXiv preprint arXiv:2110.03294*.

Gao, Y.; Islamov, R.; and Stich, S. 2024. EControl: Fast Distributed Optimization with Compression and Error Control. In *International Conference on Learning Representations*.

Ghadimi, S.; and Lan, G. 2013. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4): 2341–2368.

Huang, X.; Chen, Y.; Yin, W.; and Yuan, K. 2022. Lower bounds and nearly optimal algorithms in distributed learning with communication compression. *Advances in Neural Information Processing Systems*, 35: 18955–18969.

Huang, X.; Li, P.; and Li, X. 2024. Stochastic controlled averaging for federated learning with communication compression. In *International Conference on Learning Representations*.

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

Karimireddy, S. P.; Rebjock, Q.; Stich, S.; and Jaggi, M. 2019. Error feedback fixes signSGD and other gradient compression schemes. In *International Conference on Machine Learning*, 3252–3261. PMLR.

Khirirat, S.; Feyzmahdavian, H. R.; and Johansson, M. 2018. Distributed learning with compressed gradients. *arXiv preprint arXiv:1806.06573*.

Koloskova, A.; Lin, T.; Stich, S. U.; and Jaggi, M. 2020. Decentralized Deep Learning with Arbitrary Communication Compression. In *International Conference on Learning Representations*.

Konečný, J.; McMahan, H. B.; Ramage, D.; and Richtárik, P. 2016a. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*.

Konečný, J.; McMahan, H. B.; Yu, F. X.; Richtárik, P.; Suresh, A. T.; and Bacon, D. 2016b. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*.

Li, Z.; Bao, H.; Zhang, X.; and Richtárik, P. 2021. PAGE: A simple and optimal probabilistic gradient estimator for nonconvex optimization. In *International Conference on Machine Learning*, 6286–6295. PMLR.

Li, Z.; Hanzely, S.; and Richtárik, P. 2021. ZeroSARAH: Efficient Nonconvex Finite-Sum Optimization with Zero Full Gradient Computation. *arXiv preprint arXiv:2103.01447*.

Li, Z.; Kovalev, D.; Qian, X.; and Richtárik, P. 2020. Acceleration for Compressed Gradient Descent in Distributed and Federated Optimization. In *International Conference on Machine Learning*, 5895–5904. PMLR.

Li, Z.; and Li, J. 2022. Simple and Optimal Stochastic Gradient Methods for Nonsmooth Nonconvex Optimization. *Journal of Machine Learning Research*, 23(239): 1–61.

Li, Z.; and Richtárik, P. 2020. A Unified Analysis of Stochastic Gradient Methods for Nonconvex Federated Optimization. *arXiv preprint arXiv:2006.07013*.

Li, Z.; and Richtárik, P. 2021. CANITA: Faster rates for distributed convex optimization with communication compression. In *Advances in Neural Information Processing Systems*, 13770–13781.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, 1273–1282. PMLR.

Mishchenko, K.; Gorbunov, E.; Takáč, M.; and Richtárik, P. 2019. Distributed Learning with Compressed Gradient Differences. *arXiv preprint arXiv:1901.09269*.

Nedić, A.; Olshevsky, A.; and Rabbat, M. G. 2018. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5): 953–976.

Nedić, A.; Olshevsky, A.; and Shi, W. 2017. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4): 2597–2633.

Nesterov, Y. 2004. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer.

Polyak, B. T. 1963. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4): 864–878.

Qu, G.; and Li, N. 2017. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3): 1245–1260.

Richtárik, P.; Sokolov, I.; and Fatkhullin, I. 2021. EF21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34.

Richtárik, P.; Sokolov, I.; Gasanov, E.; Fatkhullin, I.; Li, Z.; and Gorbunov, E. 2022. 3PC: Three point compressors for communication-efficient distributed training and a better theory for lazy aggregation. In *International Conference on Machine Learning*, 18596–18648. PMLR.

Seide, F.; Fu, H.; Droppo, J.; Li, G.; and Yu, D. 2014. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth annual conference of the international speech communication association*.

Singh, N.; Data, D.; George, J.; and Diggavi, S. 2021. SQuARM-SGD: Communication-efficient momentum SGD for decentralized optimization. *IEEE Journal on Selected Areas in Information Theory*, 2(3): 954–969.

Stich, S. U.; Cordonnier, J.-B.; and Jaggi, M. 2018. Sparsified SGD with Memory. *Advances in Neural Information Processing Systems*, 31.

Tang, H.; Lian, X.; Qiu, S.; Yuan, L.; Zhang, C.; Zhang, T.; and Liu, J. 2019. DeepSqueeze: Decentralization meets error-compensated compression. *arXiv preprint arXiv:1907.07346*.

Wang, J.; Charles, Z.; Xu, Z.; Joshi, G.; McMahan, H. B.; y Arcas, B. A.; Al-Shedivat, M.; Andrew, G.; Avestimehr, S.; Daly, K.; et al. 2021. A field guide to federated optimization. *arXiv preprint arXiv:2107.06917*.

Xie, C.; Zheng, S.; Koyejo, S.; Gupta, I.; Li, M.; and Lin, H. 2020. CSER: Communication-efficient SGD with error reset. *Advances in Neural Information Processing Systems*, 33: 12593–12603.

Zhao, H.; Burlachenko, K.; Li, Z.; and Richtárik, P. 2021. Faster Rates for Compressed Federated Learning with Client-Variance Reduction. *arXiv preprint arXiv:2112.13097*.

Zhao, H.; Li, B.; Li, Z.; Richtárik, P.; and Chi, Y. 2022. BEER: Fast $O(1/T)$ Rate for Decentralized Nonconvex Optimization with Communication Compression. In *Advances in Neural Information Processing Systems*, 31653–31667.

Zhao, H.; Li, Z.; and Richtárik, P. 2021. FedPAGE: A Fast Local Stochastic Gradient Method for Communication-Efficient Federated Learning. *arXiv preprint arXiv:2108.04755*.

Zhu, M.; and Martínez, S. 2010. Discrete-time dynamic average consensus. *Automatica*, 46(2): 322–329.