Journal of Machine Learning Research 22 (2021) 1-50 Submitted 5/20; Revised 3/21; Published 9/21

Cooperative SGD: A Unified Framework for the Design and Analysis of Local-Update SGD Algorithms

Jianyu Wang JIANYUW1@ANDREW.CMU.EDU
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Gauri Joshi GAURIJ@ANDREW.CMU.EDU
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Mehryar Mohri

Abstract

When training machine learning models using stochastic gradient descent (SGD) with a large number of nodes or massive edge devices, the communication cost of synchronizing gradients at every iteration is a key bottleneck that limits the scalability of the system and hinders the benefit of parallel computation. Local-update SGD algorithms, where worker nodes perform local iterations of SGD and periodically synchronize their local models, can effectively reduce the communication frequency and save the communication delay. In this paper, we propose a powerful framework, named Cooperative SGD, that subsumes a variety of local-update SGD algorithms (such as local SGD, elastic averaging SGD, and decentralized parallel SGD) and provides a unified convergence analysis. Notably, special cases of the unified convergence analysis provided by the cooperative SGD framework yield 1) the first convergence analysis of elastic averaging SGD for general non-convex objectives, and 2) improvements upon previous analyses of local SGD and decentralized parallel SGD. Moreover, we design new algorithms such as elastic averaging SGD with overlapped computation and communication, and decentralized periodic averaging which are shown to be 4x or more faster than the baseline in reaching the same training loss.

Keywords: Communication-efficient training, distributed SGD with local updates, distributed optimization, federated learning, convergence analysis

1. Introduction

Stochastic gradient descent (SGD) is the backbone of most state-of-the-art machine learning algorithms. Due to its widespread applicability, speeding-up SGD is arguably the single most impactful and transformative problem in machine learning. Classical SGD was designed to be run on a single computing node, and its error-convergence has been extensively analyzed and improved in optimization and learning theory (Dekel et al., 2012; Ghadimi and Lan, 2013). However, due to the massive training datasets and deep neural network architectures used today, running SGD at a single node can be prohibitively slow. This calls for distributed implementations of SGD, where gradient computation and aggregation is parallelized across multiple worker nodes.

Limitations of Synchronous/Asynchronous Distributed SGD. A commonly used method to parallelize gradient computation and process more training data per iteration is the parameter server framework (Dean et al., 2012; Li et al., 2014; Cui et al., 2014). Each of the m worker nodes computes the gradients of one mini-batch of data, and a parameter server aggregates these gradients and updates the model parameters. Synchronization delays in waiting for slow workers can be alleviated via asynchronous gradient aggregation (Recht et al., 2011; Cui et al., 2014; Gupta et al., 2016; Mitliagkas et al., 2016; Dutta et al., 2018). However, by design, parameter server framework requires gradients to be communicated between the parameter server and workers after every iteration. Thus, it suffers from communication delays which are especially dominant in modern on-device training on resource-constrained computing nodes.

Local-Update Distributed SGD. To address the limitations of (a)synchronous distributed SGD, a promising idea is to allow workers to perform τ local updates to the model instead of just computing gradients, and then periodically averaging the local models. It can directly give τ-fold reduction in the communication delay per iteration, since communication only happens once every τ iterations. Local-update SGD is also attractive as far as the data locality and privacy is concerned, since users' private data is processed locally and only the trained model is communicated over the network. This covers the emerging topic of federated learning (McMahan et al., 2016; Konečnỳ et al., 2016; Mohri et al., 2019), where training tasks are performed on consumer devices (or IoT infrastructures) and only a random subset of local models are selected to communicate instead of all workers.

We refer existing local-update SGD methods that perform periodic averaging of local models as periodic simple averaging SGD (PSASGD) in the rest of the paper[1]. Extensive empirical results have validated the effectiveness of PSASGD (Moritz et al., 2015; Zhang et al., 2016; Povey et al., 2014; Su and Chen, 2015; Chaudhari et al., 2017; Smith et al., 2018; Lin et al., 2018) in reducing communication delays while maintaining similar accuracy levels. Typically, more local updates allows higher system throughput but incurs slightly higher error at convergence. Only a few recent works (Zhou and Cong, 2017; Yu et al., 2018; Stich, 2018) give a rigorous theoretical understanding of how the convergence of PSASGD depends on the number of local updates (or the communication period). These current theoretical results rely on assumptions, such as uniformly bounded gradient norm and strong convexity. We remove these assumptions and develop a more general analysis framework.

Elastic Averaging. Instead of simple averaging at each communication round, elastic-averaging SGD (EASGD) proposed in (Zhang et al., 2015) *adds a proximal term to the objective function in order to allow some slack between the models* – an idea that is drawn from the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Parikh and Boyd, 2014). Although the efficiency of EASGD and its asynchronous and periodic averaging variants has been empirically validated (Zhang et al., 2015; Chaudhari et al., 2017), its convergence analysis under non-convex objectives remains an open problem. The original paper (Zhang et al., 2015) only gives an analysis of EASGD with 1 local update and for quadratic objective functions. A similar idea like EASGD also appears in recent

1. Other names for PSASGD in previous literature (Stich, 2018; Yu et al., 2018) include 'local SGD' or 'parallel restarted SGD'

work (Hanzely and Richtárik, 2020), which focuses on personalized federated learning and strongly-convex settings.

Decentralized Averaging. Another approach to average local models is to perform gossip-type averaging in a sparse-connected network topology (a ring for instance), known as decentralized parallel SGD (D-PSGD). Each node only needs to average with its neighbors' models, thus reducing the communication complexity significantly (Blot et al., 2016; Jin et al., 2016; Lian et al., 2017a). D-PSGD has a rich history in the distributed and consensus optimization community (Tsitsiklis et al., 1986; Nedic and Ozdaglar, 2009; Duchi et al., 2012; Nedić et al., 2018). Recently, D-PSGD was successfully applied to non-convex functions in (Zeng and Yin, 2016; Jiang et al., 2017; Lian et al., 2017a). However, *their convergence analyses do not allow workers to make more than one local updates.* Analyzing decentralized averaging with multiple local updates before each consensus round is a non-trivial extension of vanilla decentralized SGD.

Main Contributions. A common thread in all the above communication-efficient SGD methods is that *they allow worker nodes to perform local model-updates and limit the synchronization/consensus between the local models.* In this paper, we propose a powerful framework named *cooperative SGD* that enables us to obtain a unified analysis and comparison of local-update distributed SGD algorithms with various model-averaging methods including periodic simple averaging, elastic averaging and decentralized averaging. This framework encompasses both temporal communication-reduction (by allowing multiple local updates at nodes and reducing communication frequency) and spatial communication-reduction (by allowing decentralized inter-node communication via a sparse network topology). More specifically, the main contributions of this paper are as follows:

(i) We provide a unified convergence analysis for the cooperative SGD class of algorithms (*i.e.*, distributed SGD algorithms with local updates). By varying the number of local updates and the model averaging protocol, the analysis can directly apply to existing algorithms. In the cases of PSASGD and D-PSGD, our analysis yields tighter and stronger convergence guarantee compared to previous results.

(ii) To the best of our knowledge, the unified analysis gives the first convergence guarantee for EASGD with non-convex objective functions. The analysis also provides new insights such as the best hyper-parameter choice, which can yield the lowest optimization error bound.

(iii) Moreover, we find that the elastic-averaging protocol can help to overlap communication and computation in distributed SGD and further improve the communication efficiency. Empirical results show that the improved version of EASGD can achieve the same training loss using $2\times$ less wall-clock time than its original counterpart.

(iv) The general framework greatly enlarges the design space of local-update SGD algorithms. We present several promising new communication-efficient variants, such as periodic decentralized averaging SGD and hierarchical averaging SGD. All of these new algorithms can be subsumed by the cooperative SGD framework and be analyzed under the same umbrella.

Relation to Gradient Quantization and Sparsification methods. In the context of communication-efficient SGD algorithms, many previous works focus on reducing the communication message size using sparsification (where only important components are transmitted), see (Wangni et al., 2017; Lin et al., 2017; Jiang and Agrawal, 2018; Stich et al., 2018) or quantization techniques (where only quantized gradients are transmitted), see (Wen et al., 2017; Wang et al., 2018; Bernstein et al., 2018; Alistarh et al., 2017; Sattler et al., 2018). These compression methods reduce the amount of information exchange while local-update SGD methods reduce the frequency. When the network latency (*e.g.*, time to establish handshakes) is high and the communication is not bandwidth-limited, local-update SGD methods will be more effective in reducing the total communication time; when the opposite is true, the compression methods will be more effective. In general, these two kinds of methods are orthogonal and can be combined together.

2. Preliminaries and Related Work

Notation. We use $\mathbf{1}$ to denote $[1, 1, \ldots, 1]^\top$ and define matrix $\mathbf{J} = \mathbf{1}\mathbf{1}^\top/(\mathbf{1}^\top\mathbf{1})$. Unless otherwise stated, $\mathbf{1}$ is a size m column vector, and the matrix $\mathbf{J}$ and identity matrix $\mathbf{I}$ are of size $m \times m$, where m is the number of workers. Let $\|\cdot\|$, $\|\cdot\|_{\mathrm{F}}$ and $\|\cdot\|_{\mathrm{op}}$ denote the ℓ_2 vector norm, Frobenius matrix norm and operator norm, respectively.

Problem Formulation. Suppose the model parameters are denoted by $\mathbf{x} \in \mathbb{R}^d$ and the local training data distribution at the i-th worker node is denoted by $\mathcal{S}_i$. Then, the problem of interest is to minimize the empirical risk as follows:

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) = \min_{\mathbf{x}\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} \underbrace{\mathbb{E}_{s\sim\mathcal{S}_i}\left[f_i(\mathbf{x}; s)\right]}_{F_i(\mathbf{x})} \tag{1}$$

where m is the number of worker nodes, $f_i(\cdot)$ is the loss function defined by the learning model and $F_i(\mathbf{x})$ denotes the local objective function at the i-th worker. The classic solution to solve (1) is parallel mini-batch SGD, where workers compute stochastic gradients of the local objectives in parallel and use the averaged gradient to update model parameters. The update rule is written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \left[\frac{1}{m}\sum_{i=1}^{m} g_i(\mathbf{x}_k; \xi_k^{(i)})\right] \tag{2}$$

where η is the learning rate, $\xi_k^{(i)} \sim \mathcal{S}_i$ are randomly sampled mini-batches from the local data distribution, and $g_i(\mathbf{x}; \xi) = \frac{1}{|\xi|}\sum_{s\in\xi} \nabla f_i(\mathbf{x}; s)$ denotes the stochastic gradient. For simplicity, we will use $g_i(\mathbf{x})$ instead of $g_i(\mathbf{x}; \xi)$ in the rest of the paper. From the update rule (2), one can observe that the model parameters at workers are always synchronized and exactly the same. Therefore, we refer to this method as *fully synchronous SGD*, the convergence analysis of which has been well presented in (Dekel et al., 2012; Bottou et al., 2018).

Periodic Simple-Averaging SGD (PSASGD). In this algorithm, workers are allowed to perform local updates so as to reduce the total communication round significantly. Locally

Table 1: Largest allowable communication period in PSASGD algorithm (larger means tighter bounds) in order to achieve a convergence rate of $1/\sqrt{Km}$ (or $1/Km$ in the strongly convex case), where K denotes the total iterations and m is the number of worker nodes. IID represents for the case where local objective functions (F_i's) are identical. (Cvx stands for Convex; Grad. B means the stochastic gradient is uniformly bounded by a constant.)

Papers	IID case	Non-IID case	Extra Asm.	Avg. Methods
(Stich, 2018)	$K^{1/2}m^{-1/2}$	$K^{1/2}m^{-1/2}$	Cvx.; Grad. B	Simple Avg.
(Jiang and Agrawal, 2018)	$K^{1/2}m^{-5/2}$	$K^{1/4}m^{-5/4}$	-	Simple Avg.
(Yu et al., 2018)	$K^{1/4}m^{-3/4}$	$K^{1/4}m^{-3/4}$	Grad. B	Simple Avg.
Ours	$K^{1/2}m^{-3/2}$	$K^{1/4}m^{-3/4}$	-	General

trained models are averaged after every τ iterations. Its update rule can be written as

$$\mathbf{x}_{k+1}^{(i)} = \begin{cases} \frac{1}{m}\sum_{j=1}^{m}\left[\mathbf{x}_k^{(j)} - \eta g_i(\mathbf{x}_k^{(j)})\right], & k \bmod \tau = 0 \\ \mathbf{x}_k^{(i)} - \eta g_i(\mathbf{x}_k^{(i)}), & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{x}_k^{(i)}$ denotes the model parameters in the i-th worker and τ is defined as the communication period (*i.e.*, the number of local updates). A simple illustration of the update rule is presented in Figure 1a.

The idea of periodic averaging can be at least traced back to the work of McDonald et al. (2010), but the convergence analysis only appears in very recent works. Stich (2018) studies the convergence of PSASGD under strongly convex objective functions. Yu et al. (2018) provides a convergence guarantee for non-convex objectives by assuming the stochastic gradients at workers are uniformly bounded. Jiang and Agrawal (2018) removes this assumption and analyzes PSASGD as a special case of gradient sparsification. Our unified analysis can yield tighter optimization error bound than the above mentioned works. Furthermore, it is not limited to simple-averaging but can be applied to other model averaging protocols. A comparison of the convergence results of PSASGD is presented in Table 1. Other follow-up works that further improve the convergence rate of PSASGD, *e.g.*, assuming convex functions (Woodworth et al., 2020b), sharing a subset of data (Haddadpour et al., 2019), using momentum acceleration (Yu et al., 2019; Wang et al., 2020b), using cross-client variance-reduction (Karimireddy et al., 2019; Liang et al., 2019), analyzing under various non-IID distributed data assumptions (Haddadpour and Mahdavi, 2019; Khaled et al., 2020) etc. are beyond the scope of this paper.

It is also worth noting that PSASGD is related to and can be considered as a key component of federated learning (FL) algorithms (McMahan et al., 2016). While both of them allow multiple local updates at workers (or clients), FL algorithms typically have additional mechanisms to address system or privacy concerns and improve the performance, such as randomly sampling workers (Li et al., 2020), using separate client and server learning rates (Karimireddy et al., 2019; Woodworth et al., 2020a; Wang et al., 2020b), etc. By combining with the techniques in these literature, our framework and analysis can also be compatible with the above mentioned additional mechanisms.

(a) Periodic simple-averaging SGD. (b) Elastic averaging SGD.

Figure 1: Illustration of PSASGD and EASGD in the model parameter space. Blue and black arrows denote local SGD iterations and the update of auxiliary variables, respectively. Red arrows represent averaging local models with each other or with the auxiliary variable. In this toy example, the number of local updates τ is set to 3.

Elastic Averaging SGD (EASGD). Instead of performing a simple average of the local models, EASGD proposed by Zhang et al. (2015) maintains an auxiliary variable $\mathbf{z}_k$ that serves as an an anchor while updating the local models $\mathbf{x}_k^{(i)}$. The update rule of vanilla EASGD[2] is given by

$$\mathbf{x}_{k+1}^{(i)} = \begin{cases} \mathbf{x}_k^{(i)} - \eta g_i(\mathbf{x}_k^{(i)}) - \alpha(\mathbf{x}_k^{(i)} - \mathbf{z}_k), & k \bmod \tau = 0 \\ \mathbf{x}_k^{(i)} - \eta g_i(\mathbf{x}_k^{(i)}), & \text{otherwise} \end{cases}, \tag{4}$$

$$\mathbf{z}_{k+1} = \begin{cases} (1 - m\alpha)\mathbf{z}_k + m\alpha\overline{\mathbf{x}}_k, & k \bmod \tau = 0 \\ \mathbf{z}_k, & \text{otherwise} \end{cases} \tag{5}$$

where $\overline{\mathbf{x}}_k = \sum_{i=1}^m \mathbf{x}_k^{(i)}/m$ and α is the elasticity parameter. From (5), we observe that the auxiliary variable $\mathbf{z}_k$ can be considered as a moving average of the averaged model $\overline{\mathbf{x}}_k$. A larger value of the parameter α forces more consensus between the locally trained models and improves stability, but it may reduce the convergence speed – a phenomenon that is not yet well-understood. While (Zhang et al., 2015) suggests that α should be smaller than $1/m$, later in Section 5.1, we will show that α can be selected in a broader range. In Figure 1b, we show how local models move in the model parameter space. The black dots show the movement of the anchor model, while the red arrows show the elastic force pulling the workers' models towards the anchor model.

Decentralized SGD (D-PSGD). In the decentralized SGD algorithm D-PSGD (also referred as consensus-based distributed SGD), nodes perform one local update and average

2. The paper (Zhang et al., 2015) also presents periodic averaging and momentum variants of EASGD. However, only vanilla EASGD has been theoretically analyzed, and only for quadratic loss functions.

their models only with neighboring nodes. The update rule is given as

$$\mathbf{x}_{k+1}^{(i)} = \sum_{j=1}^{m} w_{ji} \left[\mathbf{x}_k^{(j)} - \eta g_i(\mathbf{x}_k^{(i)}) \right] \tag{6}$$

where w_{ji} is the $(j,i)^{th}$ element of the mixing matrix $\mathbf{W}$, and it represents the contribution of node j in the averaged model at node i. The element w_{ji} is not zero if and only if node i and node j are neighbors to each other. One can design a sparse mixing topology so as to reduce the communication complexity. Although D-PSGD has been extensively studied in the last decade (Nedic and Ozdaglar, 2009; Duchi et al., 2012; Scaman et al., 2018; Nedić et al., 2018), it still remains open how to analyze the case when workers perform more than one local updates.

3. The Proposed Framework: Cooperative SGD

In this section, we will introduce a general SGD framework called Cooperative SGD that tracks the versions of the model at each worker in a distributed SGD system, where each worker performs τ local model updates. We show how Cooperative SGD subsumes several existing algorithms such as periodic simple-averaging, elastic averaging and decentralized averaging described above and several other variants including hierarchical averaging. We also highlight three different ways in which it reduces the communication delay per iteration, namely by 1) performing more local updates, 2) overlapping local computation with the communication time spent on averaging and broadcasting model updates, and 3) sparsifying the network topology used to average locally trained models. The main advantage of having a single local-update SGD update rule is that we can give a unified convergence analysis, as presented in Section 4.

3.1 Key Elements and Update Rule

The local-update SGD algorithm is denoted by $\mathcal{A}(\tau, \mathbf{W}, v)$, where τ is the number of local updates (or communication period), $\mathbf{W}$ is the mixing matrix used for model averaging at each communication round, and v is the number of auxiliary variables. These parameters feature in the update rule as follows.

1. **Model Versions at Workers.** At iteration k, the m workers have different versions $\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(m)} \in \mathbb{R}^d$ of the model. In addition, there are v auxiliary variables $\mathbf{z}_k^{(1)}, \ldots, \mathbf{z}_k^{(v)}$ that are either stored at v additional nodes or at one or more of the workers, depending upon implementation.

2. **Gradients and Local Updates.** In each iteration, the workers evaluate the gradient $g_i(\mathbf{x}_k^{(i)})$ for one mini-batch of data and update $\mathbf{x}_k^{(i)}$. The auxiliary variables are only updated by averaging a subset of the local models as described in point 3 below.

3. **Model-Averaging.** In iteration k, the local models and auxiliary variables decide whether to average with neighbors according to the synchronization matrix $\mathbf{S}_k \in$

$\mathbb{R}^{(m+v)\times(m+v)}$. To capture local updates, we use a time-varying $\mathbf{S}_k$ that varies as:

$$\mathbf{S}_k = \begin{cases} \mathbf{W}, & k \bmod \tau = 0 \\ \mathbf{I}_{(m+v)\times(m+v)}, & \text{otherwise}, \end{cases} \tag{7}$$

where the identity matrix $\mathbf{I}_{(m+v)\times(m+v)}$ means that there is no inter-node communication during the τ local updates, and mixing matrix $\mathbf{W}$ defines how local models are averaged together at the communication round.

We now present a general update rule that combines the above elements. Define matrices $\mathbf{X}_k, \mathbf{G}_k \in \mathbb{R}^{d\times(m+v)}$ that concatenate all local models and gradients:

$$\mathbf{X}_k = [\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(m)}, \mathbf{z}_k^{(1)}, \ldots, \mathbf{z}_k^{(v)}], \tag{8}$$

$$\mathbf{G}_k = [g_1(\mathbf{x}_k^{(1)}), \ldots, g_m(\mathbf{x}_k^{(m)}), \mathbf{0}, \ldots, \mathbf{0}]. \tag{9}$$

The update rule in terms of these matrices can be written as

$$\mathbf{X}_{k+1} = (\mathbf{X}_k - \eta \mathbf{G}_k)\, \mathbf{S}_k. \tag{10}$$

Remark 1 *Instead of using update* (10)*, one can use an alternative rule:* $\mathbf{X}_{k+1} = \mathbf{X}_k\mathbf{S}_k - \eta\mathbf{G}_k$*. The convergence analyses and insights in this paper can be extended to this update rule. We choose to study the update rule* (10) *for all existing algorithms (PSASGD, EASGD, D-PSGD) since fully synchronous SGD corresponds to the special case* $\mathbf{S}_k = \mathbf{J}$*.*

3.2 Existing Algorithms as Special Cases

We now show how existing communication-efficient algorithms are special cases of the general Cooperative SGD framework $\mathcal{A}(\tau, \mathbf{W}, v)$. While a detailed comparison of various algorithms is provided in Table 2, we would like to highlight the case of EASGD $\mathcal{A}(\tau, \mathbf{W}_\alpha, 1)$, in which there is an additional auxiliary variable and the mixing matrix $\mathbf{W}$ is controlled by a hyper-parameter α as follows

$$\mathbf{W}_\alpha = \begin{bmatrix} (1-\alpha)\mathbf{I} & \alpha\mathbf{1} \\ \alpha\mathbf{1}^\top & 1-m\alpha \end{bmatrix} \in \mathbb{R}^{(m+1)\times(m+1)}. \tag{11}$$

One can easily validate that the updates defined in (7), (10) and (11) are equivalent to (4) and (5) when using the alternative update rule $\mathbf{X}_{k+1} = \mathbf{X}_k\mathbf{S}_k - \eta\mathbf{G}_k$. We generalize EASGD by allowing multiple auxiliary variables and general model averaging protocols in the Cooperative SGD framework.

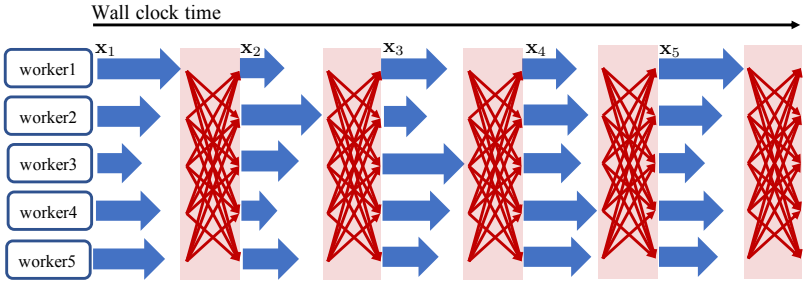
In addition to these special cases, the cooperative SGD framework allows us to design other communication-efficient SGD variants, as we further describe in Section 6.

3.3 Three Types of Communication-Efficiency Offered by Cooperative SGD

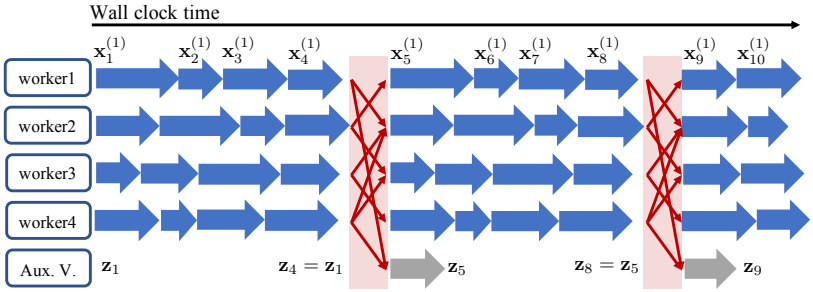
The Cooperative SGD framework improves the communication-efficiency of fully synchronous SGD in three different ways, as described below. We illustrate these in Figure 2, which compares the execution timeline of cooperative SGD with fully synchronous SGD.

Table 2: Cooperative SGD framework $\mathcal{A}(\tau, \mathbf{W}, v)$ can subsume previous algorithms as special cases by varying three key hyper-parameters: the number of local updates τ, the model-averaging protocol $\mathbf{W}$, and the number of additional auxiliary variables v.

Algorithms	Local Updates (τ)	Avg. Protocol ($\mathbf{W}$)	Addnl. Aux. Var. (v)
Fully sync. SGD	1	Simple Avg. ($\mathbf{W} = \mathbf{J}$)	0
PSASGD	Multiple	Simple Avg. ($\mathbf{W} = \mathbf{J}$)	0
EASGD	Multiple	Elastic Avg. ($\mathbf{W} = \mathbf{W}_\alpha$)	1
D-PSGD	1	Decentralized/Arbitrary	0
Cooperative SGD	Multiple	Decentralized/Arbitrary	Multiple



(a) Fully synchronous SGD.



(b) Cooperative SGD.

Figure 2: Illustration of communication-reduction strategies for $\tau = 4$. Blue and red arrows represent gradient computation and dependencies among workers, respectively. The grey arrows represent the update of the auxiliary variables, which involves communication and happens in parallel to workers gradient computation. Note that in fully synchronous SGD, although we illustrate communication via a fully-connected topology in this figure, in practice, the communication is typically implemented using a star topology or All-Reduce (Goyal et al., 2017). A comparison of the communication time of these implementations is shown in Table 3.

Periodic Averaging. By performing τ local updates at each workers and employing the periodic averaging strategy, the communication delay of Cooperative SGD is amortized over τ iterations and is τ times smaller than fully synchronous SGD. Moreover, periodic averaging evens out random variations in workers' computing time, and alleviates the synchronization

delay in waiting for slow workers. Observe in Figure 2 that the idle time of workers is significantly reduced. A quantitative justification can be found in a follow-up work (Wang and Joshi, 2018).

The Effect of Auxiliary Variables: Non-blocking Execution. The auxiliary variables $\mathbf{z}_k$ can be thought of as slightly stale versions of the average model, since they remain the same while worker nodes conduct local updates, that is, $\mathbf{z}_{j\tau} = \mathbf{z}_{j\tau-1} = \cdots = \mathbf{z}_{(j-1)\tau+1}$ for $j \geq 1$. Observe that according to the update rule (10), the worker nodes only need $\mathbf{z}_{(j-1)\tau+1}$ before the model-averaging step from $\mathbf{x}_{j\tau}$ to $\mathbf{x}_{j\tau+1}$. So, the auxiliary variables can perform averaging of the previous round of local updates and broadcast their new version while the workers perform the next set of local updates. Thus, *auxiliary variables allow the local computation to overlap with inter-node communication and enabling non-blocking execution*.

Later, in Figure 4c, we show experimental results demonstrating that this overlap of the the update of auxiliary variable and workers' local computation, directly reduces about 50% training time. Although the elastic averaging SGD algorithm proposed in (Zhang et al., 2015) allows this natural overlap between communication and computation, the authors did not take advantage of it in a non-blocking implementation to reduce the training time. To the best of our knowledge, this work is the first to identify and implement the non-blocking execution pipeline; its algorithmic variants and alternative choices of the mixing matrix have been subsequently studied in (Wang et al., 2019a, 2020a).

Sparse Averaging through a Decentralized Topology. Lastly, instead of synchronizing with all workers, a local model just needs to exchange information with its neighbors, where the mixing topology is captured by the mixing matrix $\mathbf{W}$. Thus, using a sparse mixing matrix $\mathbf{W}$ reduces the overall communication delay incurred per iteration. In Table 3, we provide a detailed comparison of the communication time between sparse (or decentralized) averaging and fully synchronization.

Table 3: Comparison between sparse averaging (*i.e.*, decentralized averaging) and fully synchronization (*i.e.*, exact averaging). When the latency to establish handshakes is dominant, sparse averaging can provide significant reduction in communication time.

Averaging protocol	# Handshakes	Transmitted data size
Decentralized	$\max_i \text{degree}_i$	$2d \cdot \max_i \text{degree}_i$
Fully synchronized (All-Reduce)	m	$2d$
Fully synchronized (Parameter Server)	m	$2dm$

As mentioned earlier, the three communication delay reduction avenues described above are orthogonal to and can be combined with gradient compression or quantization methods, which reduce the number of bits sent per inter-node communication rather than the frequency of communication.

4. Unified Convergence Analysis

In this section, we present the unified convergence analysis of algorithms in the cooperative SGD framework and study how the communication period τ and model-averaging protocol (captured by $\mathbf{W}$ and v) affect the error-convergence. The convergence analysis is conducted under the following common assumptions:

1. (Smoothness): $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \forall i \in \{1, 2, \ldots, m\}$;
2. (Lower bounded): $F(\mathbf{x}) \geq F_{\text{inf}}$;
3. (Unbiased gradients): $\mathbb{E}_{\xi|\mathbf{x}} [g_i(\mathbf{x})] = \nabla F(\mathbf{x})$;
4. (Bounded variance): $\mathbb{E}_{\xi|\mathbf{x}} \|g_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \beta \|\nabla F(\mathbf{x})\|^2 + \sigma^2$ where β and σ^2 are non-negative constants and in inverse proportion to the mini-batch size.
5. (Mixing Matrix): $\mathbf{W}\mathbf{1}_{m+v} = \mathbf{1}_{m+v}$, $\mathbf{W}^\top = \mathbf{W}$. Besides, the magnitudes of all eigenvalues except the largest one are strictly less than 1: $\max\{|\lambda_2(\mathbf{W})|, |\lambda_{m+v}(\mathbf{W})|\} < \lambda_1(\mathbf{W}) = 1$.

Assumptions 2 and 3 imply that all worker nodes have IID data distributions or the access to a same training set, which is common in large-scale data centers with shared or networked file system. For the brevity and interpretability of the results, we will first present the main results for this IID distributed case. *The analysis technique can be directly applied to the non-IID case with alternative assumptions. A generalized version of the main theorem for the non-IID data case is provided in Section 4.3.*

4.1 Update Rule for the Averaged Model

To facilitate the convergence analysis, we firstly introduce the quantities of interests. Multiplying $\mathbf{1}_{m+v}/(m+v)$ on both sides in (10), we get the vector-form update rule:

$$\mathbf{X}_{k+1}\frac{\mathbf{1}_{m+v}}{m+v} = \mathbf{X}_k\frac{\mathbf{1}_{m+v}}{m+v} - \eta\mathbf{G}_k\frac{\mathbf{1}_{m+v}}{m+v} \tag{12}$$

where $\mathbf{S}_k$ disappears due to the special property from Assumption 5: $\mathbf{W}\mathbf{1}_{m+v} = \mathbf{1}_{m+v}$ and $\mathbf{I}_{m+v}\mathbf{1}_{m+v} = \mathbf{1}_{m+v}$. Then, define the average model and effective learning rate as

$$\mathbf{u}_k = \mathbf{X}_k\frac{\mathbf{1}_{m+v}}{m+v}, \ \eta_{\text{eff}} = \frac{m}{m+v}\eta. \tag{13}$$

After rearranging Eqn. (12), one can obtain

$$\mathbf{u}_{k+1} = \mathbf{u}_k - \eta_{\text{eff}}\left[\frac{1}{m}\sum_{i=1}^{m} g_i(\mathbf{x}_k^{(i)})\right]. \tag{14}$$

Observe that the averaged model $\mathbf{u}_k$ is performing perturbed stochastic gradient descent. In the sequel, we will focus on the convergence of the averaged model $\mathbf{u}_k$, which is common practice in distributed optimization literature (Nedic and Ozdaglar, 2009; Duchi et al., 2012; Yuan et al., 2016; Lian et al., 2017a). While these previous works of decentralized

optimization shed light on the convergence analysis in this paper, our main contributions include unifying local-update SGD algorithms and bounding the derivations among local models in the presence of local updates.

Since the objective function $F(\mathbf{x})$ is non-convex, the expected gradient norm is used as an indicator of convergence (Bottou et al., 2018). We say the algorithm achieves an ϵ-suboptimal solution if:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \epsilon. \tag{15}$$

This guarantees convergence of the algorithm to a stationary point.

4.2 Main Results and Discussions

In deep learning, it is common to keep the learning rate as a constant and decay it only when the training procedure saturates (Goyal et al., 2017). Thus, we present the analysis for fixed learning rate case and study the error floor (upper bound) at convergence.

Theorem 1 (Convergence of Cooperative SGD, IID case) *For algorithm $\mathcal{A}(\tau, \mathbf{W}, v)$, suppose the total number of iterations K can be divided by the communication period τ. Under Assumptions 1–5 (with $\beta = 0$ [3]), if the learning rate satisfies*

$$\eta_{\mathit{eff}} L + 5\eta_{\mathit{eff}}^2 L^2 \left[\left(1+\frac{v}{m}\right)\frac{\tau}{1-\zeta}\right]^2 \leq 1 \tag{16}$$

where $\zeta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_{m+v}(\mathbf{W})|\}$, and all local models are initialized at a same point $\mathbf{u}_1$, then the average-squared gradient norm after K iterations is bounded as follows

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \underbrace{\frac{2\left[F(\mathbf{u}_1)-F_{\mathit{inf}}\right]}{\eta_{\mathit{eff}} K} + \frac{\eta_{\mathit{eff}} L\sigma^2}{m}}_{\textit{Fully Sync. SGD}} + \underbrace{\eta_{\mathit{eff}}^2 L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right)\left(1+\frac{v}{m}\right)^2}_{\textit{Additional Network Error}} \tag{17}$$

where $\mathbf{u}_k, \eta_{\mathit{eff}}$ are defined in (13).

Proof Check Appendix D. ■

Due to space limitations, we defer all proofs to the Appendix. If K is decided preemptively, then with a proper learning rate, we obtain the following corollary. A similar technique also appears in (Ghadimi and Lan, 2013; Lian et al., 2017a; Yu et al., 2018).

Corollary 1 (Optimized Learning Rate, IID case) *For algorithm $\mathcal{A}(\tau, \mathbf{W}, v)$, under Assumption 1–5, if the learning rate is $\eta = \frac{m+v}{Lm}\sqrt{\frac{m}{K}}$ and the hyper-parameters satisfy:*

3. Constant β in Assumption 4 only influences the constraint on the learning rate (16) and will not appear in the expression of gradient norm upper bound (17). In order to get neater results, β is set as 0 in the main paper. In the Appendix, we provide the proof for arbitrary β.

$10m[(1+\frac{v}{m})\frac{\tau}{1-\zeta}]^2 \leq K$, *the average-squared gradient norm after* K *iterations is bounded by*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \underbrace{\frac{2L\left[F(\mathbf{u}_1)-F_{inf}\right]+\sigma^2}{\sqrt{mK}}}_{\textit{Fully Sync. SGD}} + \underbrace{\frac{m}{K}\left(1+\frac{v}{m}\right)^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right)\sigma^2}_{\textit{Additional Network Error}} \quad (18)$$

$$= \mathcal{O}(\frac{1}{\sqrt{mK}}) + \mathcal{O}(\frac{m}{K}). \quad (19)$$

Furthermore, if $(m+v)^2 m[(1+\frac{v}{m})\frac{\tau}{1-\zeta}]^2 \leq K$, *then the mean square error will be dominated by the first term in* (18) *and bounded by* $2[L(F(\mathbf{u}_1)-F_{inf})+\sigma^2]/\sqrt{mK} = \mathcal{O}(1/\sqrt{mK})$.

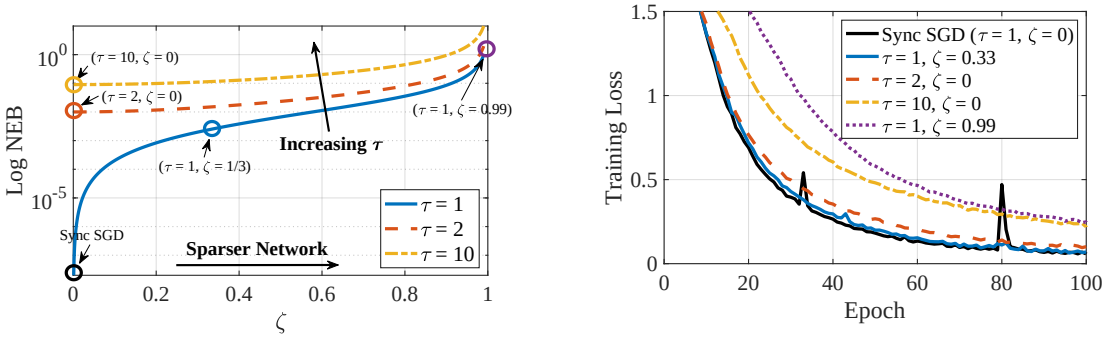
Proof Check Appendix E. ■

Error decomposition. It is worth noting that the upper bounds (17) and (18) are decomposed into two parts. The first two terms are same as the optimization error bound in fully synchronous SGD (Ghadimi and Lan, 2013), except the dependence on L, σ^2 due to different choices of learning rate[4]. The last term is *network error*, resulted from performing local updates and reducing inter-worker communication, *i.e.*, sparse model-averaging protocol. It directly increases the optimization error bound and is a measure of local models' discrepancies. When all local models are fully synchronized at each iteration ($\tau = 1, \zeta = 0, v = 0$), then the network error becomes zero.

Dependence on $\tau, \mathbf{W}$. Theorem 1 together with Corollary 1 state that the optimization error bound is determined by the communication period τ and the second largest absolute eigenvalue ζ of the mixing matrix $\mathbf{W}$. In particular, the bound will monotonically increase along with τ and ζ. The definition of ζ is common in random walks on graphs and reflects the mixing rates of different variables. When there is no communication among local workers, then $\mathbf{W} = \mathbf{I}_{m+v}$ and $\zeta = 1$; When local models are fully synchronized, then $\mathbf{W} = \mathbf{J}_{m+v}$ and $\zeta = 0$. Typically, a sparser $\mathbf{W}$ means a larger value of ζ. Besides, the network error bound is linear to τ but proportional to $(1+\zeta^2)/(1-\zeta^2)$, as shown in Figure 3a. It is more sensitive to the changes in τ.

Error-Runtime Trade-off In Figure 3b, we further evaluate various hyper-parameter settings for training VGGNet (Simonyan and Zisserman, 2014) for image classification on the CIFAR-10 dataset (Krizhevsky, 2009). More details about the experimental setting can be found in the Appendix. As predicted by (18), the empirical results show that a higher network error (larger τ or larger ζ) may lead to a slower convergence. However, the benefit of using local updates is that it can reduce the run time per iteration, resulting much less total training time to achieve a target accuracy. In different system settings, one can change the communication period τ or the sparsity of the mixing topology to achieve the best trade-off between error-convergence and communication-efficiency.

4. If we set $\eta = \frac{m+v}{m}\sqrt{m(F(\mathbf{u}_1)-F_{\text{inf}})/(\sigma^2 LK)}$ in Corollary 1, then the rate of the first two terms in (18) is exactly the same as (Ghadimi and Lan, 2013).



(a) Numerical plot of the additional network error bound (NEB) term in (17).

(b) Experimental results on neural networks.

Figure 3: (a) Illustration of how the additional network error term in (17) monotonically increases with τ and ζ; (b) Experiments on CIFAR-10 with VGG-16 and 8 worker nodes. For the same learning rate, larger τ or larger ζ lead to a higher error floor at convergence. Each line in (b) corresponds to a circled point in (a).

Linear Speedup. Corollary 1 also reveals that when the total iterations K is sufficiently large, the optimization error bound will be dominated by the first term in (19). That is, the algorithm has an asymptotic convergence rate of $1/\sqrt{mK}$, which matches the rate of fully synchronous SGD (Ghadimi and Lan, 2013). On the other hand, note that the total iterations to achieve an error of ϵ is $K = 1/(m\epsilon^2)$. The algorithm requires m times less iterations to reach the same level of error when using m times more workers. In this sense, the cooperative SGD class of algorithms can achieve a linear speedup in terms of number of workers.

Comparison to Previous Results. Using the unified analysis of local-update SGD presented in Theorem 1 and corollary 1, one can directly derive novel analyses of fully synchronous SGD, EASGD, PSASGD and D-PSGD. To be specific, by directly setting $\mathbf{W} = \mathbf{J}$ (i.e., $\zeta = 0$) and $v = 0$ in Corollary 1, one can obtain the result for PSASGD. As presented in Table 1, we get stronger convergence guarantee than previous works (Yu et al., 2018; Jiang and Agrawal, 2018). Moreover, compared to D-PSGD analysis in Lian et al. (2017a), our result (when $\tau = 1, v = 0, \zeta > 0$) leads to a better dependence on the number of workers m. In particular, they suggest the total iterations K should be greater than $\mathcal{O}(m^5)$. But we only require $K > \mathcal{O}(m^3)$ (see Corollary 1).

Remark 2 (Discussion on Lower Bounds) *We note that the rate $\mathcal{O}(1/\sqrt{mK})$ matches the lower bound of fully synchronous SGD (i.e.mini-batch SGD) for smooth non-convex functions as presented in Arjevani et al. (2019). There are many other classic literature discussing the lower bounds of distributed or decentralized optimization, such as (Arjevani and Shamir, 2015; Scaman et al., 2018). However, most of them focused on convex or strongly-convex loss functions, and hence cannot be directly applied to our non-convex setting.*

4.3 Extension to Non-IID Distributed Case.

Now we are going to extend the main results to the non-IID data distributed setting, where local objectives F_i are different and local data cannot be shuffled across workers. While this challenging setting has been studied in some recent works, such as (Li et al., 2019; Koloskova et al., 2020) and many others, the purpose of this subsection is to demonstrate that our main results and analysis techniques under the IID case are extendable to alternative assumptions.

In this case, the stochastic gradient at each worker is no longer an unbiased estimator of the global gradient. Specifically, we revise assumptions 3 and 4 as follows:

- (Unbiased gradients): $\mathbb{E}_{\xi|\mathbf{x}}\left[g_i(\mathbf{x})\right] = \nabla F_i(\mathbf{x})$;
- (Bounded variance): $\mathbb{E}_{\xi|\mathbf{x}}\left\|g_i(\mathbf{x}) - \nabla F_i(\mathbf{x})\right\|^2 \leq \sigma^2$ where σ^2 is a non-negative constant and in inverse proportion to the mini-batch size;
- (Bounded dissimilarities): $\frac{1}{m}\sum_{i=1}^{m}\left\|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\right\|^2 \leq \kappa^2$ where κ^2 is a non-negative constant.

One can also assume other kinds of assumptions to capture the dissimilarities among local functions. Our previous theorems in IID case lay the foundation of the analysis and can be easily adapted.

Theorem 2 (Convergence of Cooperative SGD, non-IID case) *For algorithm $\mathcal{A}(\tau, \mathbf{W}, v)$, under the new assumptions stated above, if the learning rate satisfies $\eta_{\mathit{eff}}L \leq 1$, and all local models are initialized at a same point $\mathbf{u}_1$, then the average-squared gradient norm after K iterations is bounded as follows*

$$\underbrace{2\left[\frac{2[F(\mathbf{u}_1) - F_{\mathit{inf}}]}{\eta_{\mathit{eff}}K} + \frac{\eta_{\mathit{eff}}L\sigma^2}{m} + \eta_{\mathit{eff}}^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right)\left(1+\frac{v}{m}\right)^2\right]}_{\textit{Error Bound in IID Case}} + C\kappa^2 \tag{20}$$

where $\mathbf{u}_k, \eta_{\mathit{eff}}$ are defined in (13), and constant C is defined as

$$C = \frac{6\eta^2L^2\tau^2}{1-\zeta}\left(\frac{2\zeta^2}{1+\zeta} + \frac{2\zeta}{1-\zeta} + \frac{\tau-1}{\tau}\right) \leq \frac{1}{2}. \tag{21}$$

Proof Check Appendix F. ■

Comparison to the IID Case. Theorem 2 shows that the first term in the error upper bound of non-IID case (20) is nearly identical to the bound of IID case (17), except a constant. Furthermore, there is an additional term ($C\kappa^2$) in (20) which depends on the dissimilarities among local objectives. Due to this additional term, now the new error bound (20) is in an order of τ^2 while the error bound (17) only linearly increases with the communication period τ. In other words, cooperative SGD is less robust to the choice of τ in the non-IID case than the IID case, since the algorithm requires a smaller τ (less communication reduction) in order to achieve the same error floor as the IID case.

5. Novel Analyses of Existing Algorithms and Insights

Using the unified analysis of Cooperative SGD presented in Section 4, one can directly derive novel analyses of EASGD, PSASGD, D-PSGD. In particular, as mentioned in Section 2, when $\mathbf{W} = \mathbf{W}_\alpha$ and $v = 1$, the local-update SGD algorithm reduces to EASGD (Zhang et al., 2015). We highlight this special case in this section, since the unified analysis not only provides the first convergence guarantee but also gives new insights to further improve the communication efficiency with the help of auxiliary variables.

5.1 Convergence Analysis of EASGD $\mathcal{A}(\tau, \mathbf{W}_\alpha, 1)$ and Optimal Choice of α

Recall that EASGD uses hyper-parameter α to control the eigenvalues of mixing matrix. For $\mathbf{W}_\alpha$ defined in (11), the second largest eigenvalue magnitude is

$$\zeta = \max\{|1-\alpha|, |1-(m+1)\alpha|\}. \tag{22}$$

In order to let $\mathbf{W}_\alpha$ satisfy the conditions in Assumption 5, it is required that $\zeta < 1$, namely $0 \leq \alpha < 2/(m+1)$. This condition suggests that α can be selected in a broader range than the original paper (Zhang et al., 2015) suggested ($0 \leq \alpha < 1/m$). Intuitively, a larger α forces more consensus between the locally trained models and improves stability. However, from equation (22), we observe that there exists a best α that minimizes the value of ζ.

Lemma 1 (Best Choice of α) *If $\alpha = 2/(m+2)$, then the second largest absolute eigenvalue of $\mathbf{W}_\alpha$, given in (22), achieves the minimal value $m/(m+2)$; If $0 < \alpha < 2/(m+1)$, then $\zeta < 1$.*

Accordingly, by choosing the best α, the optimization error upper bounds (17) and (18) can also be minimized. To be specific, we have the following theorem.

Theorem 3 (Convergence of EASGD with the best α, IID case) *When α is set to $2/(m+2)$ as suggested by Lemma 1, the error of EASGD $\mathcal{A}(\tau, \mathbf{W}_\alpha, 1)$ can be bounded as follows:*

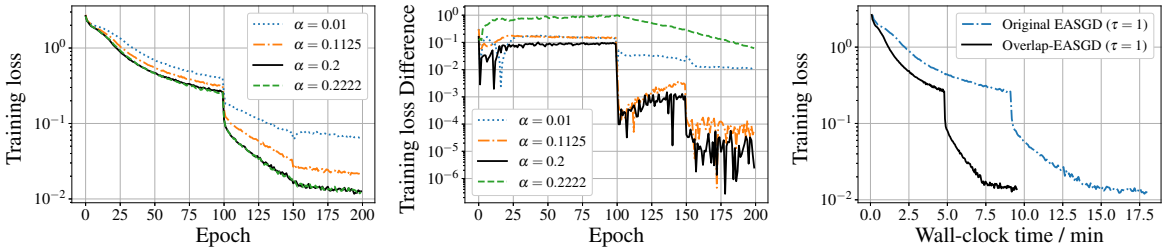
$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2\left[F(\mathbf{u}_1) - F_{inf}\right]}{\eta_{\mathit{eff}} K} + \frac{\eta_{\mathit{eff}} L \sigma^2}{m} + \frac{1}{2}\eta_{\mathit{eff}}^2 L^2 \sigma^2 (m+1)\left[1 + C(\tau-1)\right] \tag{23}$$

where $\mathbf{u}_k$ and η_{eff} are defined in (13), and $C = 1 + 2(1+1/m)^2$.

By setting $\eta_{\text{eff}} = \frac{1}{L}\sqrt{\frac{m}{K}}$, one can also obtain a finite horizon result as Corollary 1. To the best of our knowledge, this theorem is the first convergence guarantee for EASGD with general non-convex objectives and also the first theoretical justification for the best choice of elasticity parameter α.

Empirical Validation. Although the best α is obtained by minimizing an error upper bound, the empirical results indicate that this theoretical approximation works well in practice. As shown in Figures 4a and 4b, the best choice $\alpha = 2/(m+2) = 0.2$ yields the fastest convergence and the least discrepancies between workers and the auxiliary variable. When α is greater than $2/(m+1) \approx 0.2222$, we observe the algorithm cannot converge.

Overlapping Communication and Computation via Auxiliary Variables. From the update rule of EASGD, we observe that if each worker node maintain a local copy of the auxiliary variable $\mathbf{z}$, then local models can be updated without any communication in the model-averaging step (see Eqn. (4)). After averaging with the local copy of $\mathbf{z}$, each local model will directly start next round of local updates. Meanwhile, a communication thread on each worker node can use non-blocking (asynchronous) communication to obtain the averaged model and use it to update the local copy of auxiliary variable (see Eqn. (5)). As long as the number of local updates τ is large enough, the communication can be fully overlapped with the local computation. In Figure 4c, we show that by overlapping the update of auxiliary variable and workers computation, it directly reduces about 50% training time in EASGD, even though worker nodes only perform 1 local SGD iteration.



(a) Average training loss of local models.

(b) The difference of training loss between workers and the auxiliary variable.

(c) Benefit of using auxiliary variable: overlapping communication and computation.

Figure 4: EASGD training on CIFAR-10 with VGG-16. Since there are 8 worker nodes and 1 auxiliary variable, the best value of α given by Lemma 1 is $2/(m+2) = 0.2$, which performs better than the empirical choice $\alpha = 0.9/m = 0.1125$ suggested in Zhang et al. (2015). The best choice of α yields the lowest training loss and the least discrepancies between workers and auxiliary variable.

5.2 Convergence Analysis of PSASGD $\mathcal{A}(\tau, \mathbf{J}, 0)$

By directly setting $\mathbf{W} = \mathbf{J}$ (i.e., $\zeta = 0$) and $v = 0$ in Theorem 1, one can obtain the convergence guarantee for PSASGD.

Corollary 2 (Convergence of PSASGD) *For $\mathcal{A}(\tau, \mathbf{J}, 0)$, under the same assumptions as Theorem 1, if the learning rate satisfies $\eta L + \eta^2 L^2 \tau(\tau-1) \leq 1$, then we have*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla F(\overline{\mathbf{x}}_k)\right\|^2\right] \leq \frac{2\left[F(\mathbf{x}_1) - F_{inf}\right]}{\eta K} + \frac{\eta L \sigma^2}{m} + \eta^2 L^2 \sigma^2 (\tau - 1). \tag{24}$$

If we set the learning rate as $\eta = \frac{1}{L}\sqrt{\frac{m}{K}}$ and $10m\tau^2 \leq K$, then

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla F(\mathbf{u}_k)\right\|^2\right] \leq \frac{2L\left[F(\mathbf{u}_1) - F_{inf}\right] + \sigma^2}{\sqrt{mK}} + \frac{m(\tau-1)\sigma^2}{K} \tag{25}$$

$$= \mathcal{O}(\frac{1}{\sqrt{mK}}) + \mathcal{O}(\frac{m(\tau-1)}{K}). \tag{26}$$

Moreover, plugging the error of IID case (24) *back into* (20)*, one can directly obtain the error upper bound of PSASGD in the non-IID setting.*

The notable insight provided by Corollary 2 is that there exists a trade-off between the error-convergence and communication-efficiency. While a larger communication period leads to higher error at convergence, it directly reduces the communication delay by τ times and enables higher throughput. The primary advantage of PSASGD is that one can easily change the communication period and find the best one that has the fastest convergence rate with respect to wall-clock time. The best value of τ should depend on the bandwidth/latency of the communication network and vary in different environments.

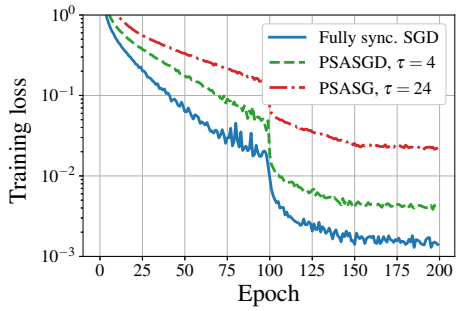
On the other hand, we note that when $K \geq m^3\tau^2$, the first term in (26) will dominate the error bound and hence, PSASGD achieves the same rate $1/\sqrt{mK}$ as fully synchronous SGD. This puts a constraint on the largest value of communication period. Furthermore, in the non-IID setting, the second term in (26) increases with τ^2. Therefore, in order to achieve the rate $1/\sqrt{mK}$, the communication period should satisfy $K \geq m^3\tau^4$. We summarize and compare our results with previous works in Table 1.

Extension to Federated Averaging. When worker nodes have IID data distribution, then Corollary 2 can be directly applied to the federated averaging (FedAvg) algorithm proposed by McMahan et al. (2016). At each round in FedAvg, only a fraction c of the m workers are selected at random. These selected worker nodes perform local updates and send the updates back to the central server which aggregates them and updates the global model. In the IID case, since the workers' local data are statistically identical, sampling a subset of cm out of m workers (c is the fraction of workers chosen per round) is equivalent to reducing the total number of workers. Therefore, changing m to cm in Corollary 2, where scalar c denotes the worker sampling ratio, we obtain the convergence guarantee for FedAvg in IID case. The analysis of FedAvg with both non-IID data distributions and random client sampling is a non-trivial extension and has been considered in some recent works, such as (Li et al., 2020; Karimireddy et al., 2019).

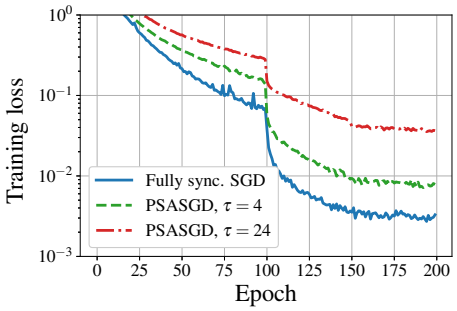
Empirical validation. In Figure 5, we show the trade-off in PSASGD with different learning rate choices. One can see that even though PASGD with $\tau = 24$ finishes the training first, it has the highest loss after the same number of iterations (or epochs). Comparing Figure 5 (a) and (b), observe that the small learning rate reduces the gap between different communication periods. This phenomenon has already been discussed in Theorem 1: small learning rate can alleviate the relative effect of the network error term. Besides, for completeness, we present the test accuracy of PSASGD in Figure 5 (c) and (d). An interesting observation is that PSASGD with large communication period has even better generalization performance than fully synchronous SGD before the learning rate is changed. A similar observation also appears in (Lin et al., 2018).

5.3 Convergence Analysis of D-PSGD $\mathcal{A}(1, \mathbf{W}, 0)$

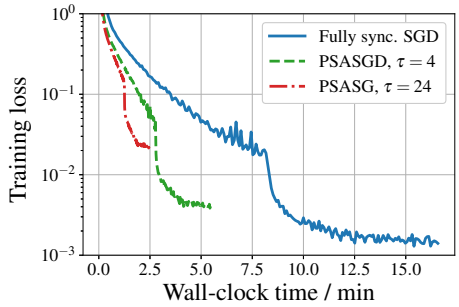
In decentralized parallel SGD (D-PSGD), local models are sparsely averaged according to a decentralized topology. Setting $\tau = 1$ and $v = 0$ in Theorem 1, we directly get the convergence guarantee for D-PSGD.



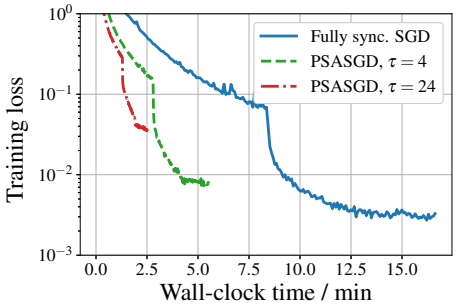
(a) Learning rate equals to 0.04.



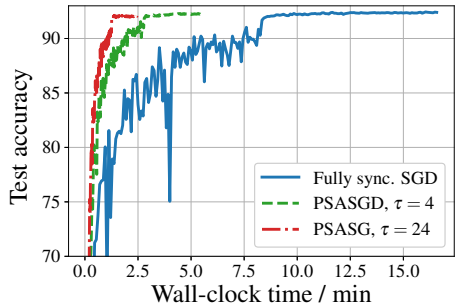
(b) Learning rate equals to 0.4.



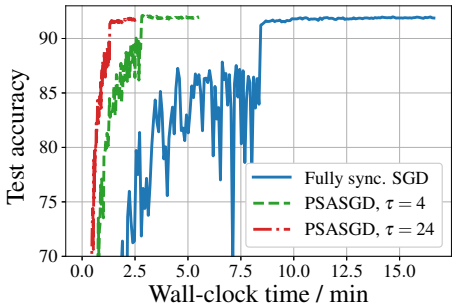
(c) Learning rate equals to 0.04.



(d) Learning rate equals to 0.4.



(e) Learning rate equals to 0.04.



(f) Learning rate equals to 0.4.

Figure 5: Illustration of error-convergence and communication-efficiency trade-off in PSASGD. We train a VGG-16 on CIFAR-10 with 8 worker nodes. Each line was trained for 200 epochs and the learning rate is decayed by 10 at epoch 100, 150. After the same number of epochs, a larger communication period leads to higher training loss but costs much less wall clock time.

Corollary 3 (Convergence of D-PSGD) *For $\mathcal{A}(1, \mathbf{W}, 0)$, under the same assumptions as Theorem 1, if the learning rate satisfies $\eta L + \eta^2 L^2 \frac{2\zeta}{1-\zeta}\left(\frac{\zeta}{1+\zeta} + \frac{1}{1-\zeta}\right) \leq 1$, where $\zeta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_m(\mathbf{W})|\}$, then we have*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla F(\overline{\mathbf{x}}_k)\right\|^2\right] \leq \frac{2\left[F(\mathbf{x}_1) - F_{inf}\right]}{\eta K} + \frac{\eta L \sigma^2}{m} + \eta^2 L^2 \sigma^2 \frac{2\zeta^2}{1-\zeta^2}. \tag{27}$$

If the learning rate is $\eta = \frac{1}{L}\sqrt{\frac{m}{K}}$ and the hyper-parameters satisfy: $10m(1-\zeta)^{-2} \leq K$, the average-squared gradient norm after K iterations is bounded by

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla F(\mathbf{u}_k)\right\|^2\right] \leq \frac{2L\left[F(\mathbf{u}_1) - F_{inf}\right] + \sigma^2}{\sqrt{mK}} + \frac{m\sigma^2}{K}\frac{2\zeta^2}{1-\zeta^2} \tag{28}$$

$$= \mathcal{O}(\frac{1}{\sqrt{mK}}) + \mathcal{O}(\frac{m}{K}\frac{\zeta^2}{1-\zeta^2}). \tag{29}$$

Moreover, plugging the error of IID case (27) *back into* (20)*, one can directly obtain the error upper bound of D-PSGD in the non-IID setting.*

From Eqn. (29), we conclude that if $K \geq \mathcal{O}(m^3)$, then the first term in (29) will dominate the error upper bound and D-PSGD can achieve the same rate $1/\sqrt{mK}$ as fully synchronous SGD. Compared to previous result in (Lian et al., 2017a) which suggests $K \geq \mathcal{O}(m^5)$, we slightly improve the dependence on the number of workers. Compared to literature in the decentralized optimization community (Duchi et al., 2012; Yuan et al., 2016; Zeng and Yin, 2016), we remove the assumption of uniformly bounded gradients and focus on non-convex objective functions and stochastic gradients.

Comparison of PSASGD and D-PSGD The general framework enables easy comparisons between different communication reduction strategies. Here, we compare periodic communication and sparse averaging (decentralized averaging) strategies. Note that when PSASGD $\mathcal{A}(\tau, \mathbf{J}, 0)$ and D-PSGD $\mathcal{A}(1, \mathbf{W}, 0)$ have the same error floor at convergence (*i.e.*, when (24) equals to (27)), we have

$$\frac{2\zeta_\tau^2}{1-\zeta_\tau^2} = \tau - 1 \Rightarrow \zeta_\tau = \sqrt{1 - \frac{2}{\tau+1}}. \tag{30}$$

Equation (30) provides a threshold for ζ. As long as we design a mixing matrix such that $\zeta \leq \zeta_\tau$, D-PSGD $\mathcal{A}(1, \mathbf{W}, 0)$ would perform better than PSASGD $\mathcal{A}(\tau, \mathbf{J}, 0)$ in terms of the worst-case final error at convergence. Along with the increase of τ, the value of threshold ζ_τ rapidly converges to 1. Therefore, when τ becomes large, D-PSGD has a lower error floor in a very broad range of ζ.

As for communication efficiency, the benefit of sparse or decentralized averaging relies on the number of workers. It at most reduces the communication overhead by m times, since at least one connection should be preserved for each worker. As the mixing matrix affects the communication delay implicitly, it is not trivial to design a good mixing matrix that not only has small eigenvalues but also enables efficient implementation. On the contrary, periodic averaging has higher flexibility without such limitations. If we set $\tau \geq m$, then PSASGD always has shorter training time than D-PSGD.

6. Designing New Local-Update SGD Algorithms

As shown in Sections 4 and 5, the Cooperative SGD framework enables us to analyze and compare existing communication-efficient SGD algorithms such as PSASGD, EASGD and D-PSGD. The Cooperative SGD framework can also be used to design new algorithms that combine the communication-efficiency strategies adopted by these algorithms.

6.1 Decentralized Periodic Averaging

For a fixed topology worker network where $\mathbf{W}$ is prescribed, increasing the number of local updates τ can be an effective way to speedup the training procedure. In Figure 6, one can observe that by using more local updates, *decentralized periodic averaging SGD* ($\tau = 10, \zeta = 0.75$) has significant speedup over the original D-PSGD algorithm ($\tau = 1, \zeta = 0.75$) in terms of wall-clock time to achieve the same training loss. The new algorithm can be useful in the setting of decentralized federated learning, where workers are connected in an arbitrarily connected topology and can only communicate local models with neighbors, see a follow-up work (Li et al., 2019).

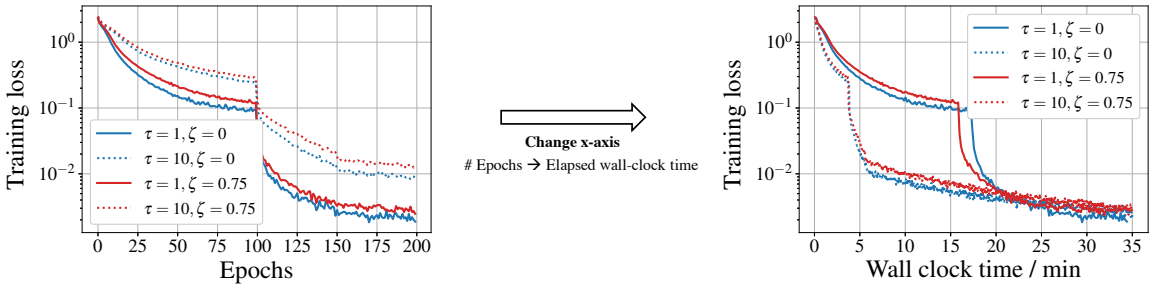


Figure 6: Decentralized periodic averaging on CIFAR-10 with VGG-16. Fully synchronous SGD corresponds to ($\tau = 1, \zeta = 0$). Allowing more local updates (higher τ) leads to slower convergence in terms of epochs. But it requires about 4x less wall-clock time to achieve a training loss of 0.1.

6.2 Generalized Elastic Averaging

In generalized elastic averaging $\mathcal{A}(1, \mathbf{W}', 1)$, we modify D-PSGD with mixing matrix $\mathbf{W}$ by adding an auxiliary variable (with elasticity parameter α) stored at a new node that is connected to all m worker nodes. Recall that a sparse mixing matrix $\mathbf{W}$ can reduce communication delay, but it may have large ζ that leads to inferior convergence. Introducing the auxiliary variable results in the mixing matrix $\mathbf{W}'$ shown in (31) below. The second largest eigenvalue of this matrix is $(1 - \alpha)$ lower than ζ as shown by Theorem 4.

Theorem 4 *Suppose there is a m-dimension symmetric matrix $\mathbf{W}$ such that $\mathbf{W}\mathbf{1} = \mathbf{1}$, and its eigen-values satisfy $-1 \leq \lambda_m(\mathbf{W}) \leq \cdots \leq \lambda_1(\mathbf{W}) \leq 1$. Let $\zeta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_m(\mathbf{W})|\}$. Then, for matrix $\mathbf{W}'$ which is defined as:*

$$\mathbf{W}' = \begin{bmatrix} (1-\alpha)\mathbf{W} & \alpha\mathbf{1} \\ \alpha\mathbf{1}^\top & 1 - m\alpha \end{bmatrix}, \tag{31}$$

we have

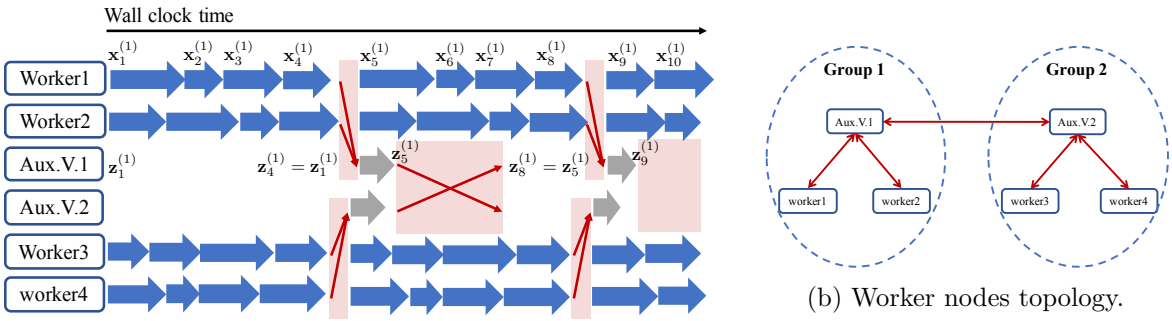
$$\zeta' = \max\{|\lambda_2(\mathbf{W}')|, |\lambda_{m+1}(\mathbf{W}')|\} \tag{32}$$
$$= \max\{(1-\alpha)\zeta, |1-(m+1)\alpha|\}. \tag{33}$$

Setting $\alpha = \frac{1+\zeta}{m+1+\zeta}$ *yields the minimum* $\zeta' = \frac{m\zeta}{m+1+\zeta}$.

The proof is given in the Appendix Appendix H. Theorem 4 implies that by setting $\alpha = \frac{1+\zeta}{m+1+\zeta}$, the new algorithm $\mathcal{A}(1, \mathbf{W}', 1)$ gives a lower error bound at convergence as compared to D-PSGD $\mathcal{A}(1, \mathbf{W}, 0)$ as $\zeta' < \zeta$. Furthermore, since the updates and broadcast of the auxiliary variable can overlap with the local computation at workers (as explained in Section 3.3), we do not expect an increase in the training time. Thus, adding an auxiliary variable is a highly effective method to increase the consensus between loosely connected workers.

6.3 Elastic Hierarchical Averaging SGD.

As mentioned in Section 5, auxiliary variables in the cooperative SGD framework can help to overlap communication and computation. This feature is beneficial when the inter-worker communication is expensive, since the costly communication can be totally hidden by using the elastic averaging protocol with fine-tuned number of local updates. In particular, consider that workers are divided into groups that cannot directly communicate with each other. Local models in each group will be averaged via an auxiliary node. On the other hand, expensive inter-auxiliary node communication can occur concurrently with local updates at workers. A brief illustration is provided in Figure 7. Our unified convergence analysis can be applied to this hierarchical averaging model and ongoing research includes finding the node structure that gives the best convergence.



(a) Execution pipeline.

(b) Worker nodes topology.

Figure 7: Illustration of a variant of local-update SGD: elastic hierarchical averaging. Blue, red, grey arrows represent gradient computation, communication among workers, and update of auxiliary variables respectively.

7. Concluding Remarks

In this paper, we propose a general framework named *Cooperative SGD*, in which worker nodes are allowed to have different model versions and these local models are synchronized infrequently via various model-averaging protocols (*e.g.*, simple averaging, elastic averaging and decentralized averaging). This formulation subsumes many existing communication-efficient distributed SGD variants, including periodic simple averaging SGD (*i.e.*, local SGD), elastic averaging SGD, and decentralized parallel SGD. Our general framework not only bridges these separate studied algorithms together but also greatly enlarges the design space of local-update SGD algorithms. We present several novel instances of the Cooperative SGD algorithm, such as periodic decentralized SGD, generalized elastic averaging SGD, and hierarchical averaging SGD.

By analyzing Cooperative SGD for general non-convex objectives in both IID and non-IID data partitions settings, we provide strong convergence guarantees for existing communication-efficient distributed SGD variants, and to the best of our knowledge, the first general analysis of elastic averaging SGD. The unified analysis reveals how the number of local updates and the averaging protocol influence the convergence rate and brings novel insights including comparisons of different model-averaging protocols, the best hyper-parameter choice in EASGD, and the communication overlapping via auxiliary variables.

Further exploration of the local-update SGD design space and analyses of new variants are ripe for future investigation. We list some promising directions below.

- **Extensions of Cooperative SGD framework**: In our proposed cooperative SGD algorithm, the mixing matrix $\mathbf{W}$ is a fixed matrix across iterations. However, much looser synchronization protocols are also possible. For example, Koloskova et al. (2020); Wang et al. (2019b) extended our framework by allowing random mixing matrices. Besides, by using the techniques in (Assran et al., 2018; Pu et al., 2020), it is easy to relax the constraint on $\mathbf{W}$ from doubly stochastic to row (or column) stochastic.

- **Asynchronous local-update SGD**: When the worker nodes have different computing speeds, waiting for the slowest nodes (the stragglers) to finish their local updates before performing model aggregation can result in a long tail latency. In order mitigate stragglers, we can use one of two possible asynchronous strategies: 1) fix a time window, allow the workers to perform a variable number of local updates within that time and then aggregate the updates by assigning appropriate weights to each worker, or 2) each worker performs the same number τ of local updates, but it averages with the central or anchor models in an asynchronous and lock-free manner, allowing certain workers to have stale versions of the central model. The convergence analysis of these asynchronous local-update SGD variants is a challenging problem – it requires additional assumptions bounding the degree of staleness in order to guarantee convergence, see (Dutta et al., 2018; Lian et al., 2015, 2017b).

- **Adding momentum or gradient tracking mechanism to accelerate local-update SGD**: Momentum has been shown to be effective in improving the training and generalization performance of stochastic optimization. Investigating variants of Cooperative SGD with momentum is a timely and interesting research direction. Two such momentum variants have recently been proposed in the follow-up works (Yu et al.,

2019; Wang et al., 2020b). On the other hand, gradient tracking (Shi et al., 2015; Xin et al., 2020) is a popular technique to accelerate the convergence of decentralized SGD. We would expect this technique can be directly apply to PSASGD (or federated averaging), because we have shown in this paper that decentralized SGD and PSASGD are just different facets of a general algorithm. Some algorithmic variants have already been investigated in Liang et al. (2019); Karimireddy et al. (2019).

- **Handling non-IID data**: In the unified analysis presented in this work, we show that Cooperative SGD class of algorithms may suffer from dissimilarities among local data distributions. It is important to design some new mechanisms to mitigate this additional error. Follow-up works along this direction includes Sahu et al. (2018); Haddadpour et al. (2019); Karimireddy et al. (2019); Liang et al. (2019). There also exists some very recent works focusing on the convergence analysis of PSASGD using other forms of dissimilarity assumptions, see Haddadpour and Mahdavi (2019); Khaled et al. (2020).

- **Overlapping communication and computation**: In this paper, we show that the auxiliary variables in elastic averaging protocol can help to overlap communication and computation delay. But the mixing matrix used in EASGD may not be optimal. In our follow-up work (Wang et al., 2020a), we propose *Overlap-Local-SGD* which uses a column-stochastic mixing matrix instead of the doubly-stochastic one in EASGD. Empirically, Overlap-Local-SGD outperforms EASGD in both IID and non-IID data settings.

- **Combination with gradient compression**: As mentioned in the introduction, gradient compression (*i.e.*, sparsification and quantization) is an orthogonal way to reduce the communication overhead – it reduces the amount of bits communicated per round whereas local-update SGD reduces the frequency of communication. Combining Cooperative SGD and gradient compression techniques may enjoy the advantages in both worlds. Following this idea, Reisizadeh et al. (2019); Basu et al. (2019) propose the quantized and sparsified version of PSASGD.

Acknowledgments

This work was partially supported by the NSF CRII Award CCF-1850029, CMU Dean's fellowship, Qualcomm Innovation fellowship, and an IBM Faculty Award. The experiments were conducted on the ORCA cluster provided by the Parallel Data Lab at CMU, and on Amazon AWS (supported by an AWS credit grant).

Appendix A. Experimental Setting

All experiments presented in the main paper is conducted on CIFAR-10 (Krizhevsky, 2009) dataset, which consists of $60,000$ color images ($50,000$ for training and $10,000$ for validation). Without special explanations, we train a deep neural network VGG-16 (Simonyan and Zisserman, 2014) for 200 epochs. The initial learning rate is 0.4 and decays by 10 after 100 and 150 epochs. The mini-batch size per worker is 128. Besides, we use a network of 8 machines, each of which has one NVIDIA Titan X GPU and a 40 Gbps/s Ethernet interface. The algorithms were implemented with Pytorch and MPI4Py.

Appendix B. Proof Preliminaries

For the ease of writing, we first define some notations. Let Ξ_k denote the set $\{\xi_k^{(1)}, \ldots, \xi_k^{(m)}\}$ of mini-batches at m workers in iteration k. We use notation $\mathbf{E}_k$ to denote the conditional expectation $\mathbb{E}_{\Xi_K|\mathbf{X}_k}$. Besides, define averaged stochastic gradient and averaged full batch gradient as follows:

$$\mathcal{G}_k = \frac{1}{m}\sum_{i=1}^{m} g(\mathbf{x}_k^{(i)}),\ \mathcal{H}_k = \frac{1}{m}\sum_{i=1}^{m} \nabla F(\mathbf{x}_k^{(i)}). \tag{34}$$

Similar to $\mathbf{X}_k$ and $\mathbf{G}_k$, we stack all full batch gradients in a $d \times (m+v)$ dimension matrix:

$$\nabla F(\mathbf{X}_k) = [\nabla F(\mathbf{x}_k^{(1)}), \ldots, \nabla F(\mathbf{x}_k^{(m)}), \mathbf{0}, \ldots, \mathbf{0}]. \tag{35}$$

Accordingly, the Frobenius norm of full batch gradients is $\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2 = \sum_{i=1}^{m} \left\|\nabla F(\mathbf{x}_k^{(i)})\right\|^2$. In order to facilitate reading, the definitions of matrix Frobenius norm and operator norm are also provided here.

Definition 5 (Horn and Johnson (1990)) *The Frobenius norm defined for $\mathbf{A} \in M_n$ by*

$$\|\mathbf{A}\|_F^2 = |\operatorname{Tr}(\mathbf{A}\mathbf{A}^\top)| = \sum_{i,j=1}^{n} |a_{ij}|^2. \tag{36}$$

Definition 6 (Horn and Johnson (1990)) *The operator norm defined for $\mathbf{A} \in M_n$ by*

$$\|\mathbf{A}\|_{op} = \max_{\|\mathbf{x}\|=1} \|\mathbf{A}\mathbf{x}\| = \sqrt{\lambda_{max}(\mathbf{A}^\top \mathbf{A})}. \tag{37}$$

All notations used in the proof are listed below.

Appendix C. A Supporting Lemma for Theorem 1

Before providing the proof of Theorem 1, we prefer to first present an important lemma that describes the basic intuition for the convergence of cooperative SGD: the discrepancies of local models have a negative impact on the convergence. The proof of Theorem 1 will be built upon this lemma.

Number of workers	m
Number of auxiliary variables	v
Total iterations	K
Communication period	τ
Mixing matrix	$\mathbf{W}$
Learning rate	η
Lipschitz constant	L
Variance bounds for stochastic gradients	β, σ^2

Table 4: List of notations.

Lemma 2 (Error decomposition) *For algorithm $\mathcal{A}(\tau, \mathbf{W}, v)$, under Assumption 1–5, if the learning rate satisfies $\eta_{\mathit{eff}} L(1+\beta/m) \leq 1$ and all local model parameters are initialized at the same point $\mathbf{x}_1$, then the average-squared gradient after K iterations is bounded as follows*

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla F(\mathbf{u}_k)\right\|^2\right] \leq \underbrace{\frac{2\left[F(\mathbf{x}_1)-F_{\mathit{inf}}\right]}{\eta_{\mathit{eff}} K}+\frac{\eta_{\mathit{eff}} L\sigma^2}{m}}_{\textit{fully sync SGD}} + \underbrace{\frac{L^2}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_F^2}{m}}_{\textit{network error}} \tag{38}$$

where $\mathbf{u}_k, \eta_{\mathit{eff}}$ are defined in (13) and both $\mathbf{I}$ and $\mathbf{J}$ are $(m+v)\times(m+v)$ matrices.

C.1 Proof of Lemma 2

C.1.1 Lemmas

Lemma 3 *Under Assumption 3 and 4, we have the following variance bound for the averaged stochastic gradient:*

$$\mathbb{E}_{\Xi_K|\mathbf{X}_k}\left[\left\|\mathcal{G}_k-\mathcal{H}_k\right\|^2\right] \leq \frac{\beta}{m^2}\left\|\nabla F(\mathbf{X}_k)\right\|_F^2+\frac{\sigma^2}{m}. \tag{39}$$

Proof According to the definition of $\mathcal{G}_k, \mathcal{H}_k$ (34), we have

$$\mathbb{E}_{\Xi_K|\mathbf{X}_k}\left[\left\|\mathcal{G}_k-\mathcal{H}_k\right\|^2\right] \tag{40}$$

$$=\mathbb{E}_{\Xi_K|\mathbf{X}_k}\left\|\frac{1}{m}\sum_{i=1}^{m}\left[g(\mathbf{x}_k^{(i)})-\nabla F(\mathbf{x}_k^{(i)})\right]\right\|^2 \tag{41}$$

$$=\frac{1}{m^2}\mathbb{E}_{\Xi_K|\mathbf{X}_k}\left[\sum_{i=1}^{m}\left\|g(\mathbf{x}_k^{(i)})-\nabla F(\mathbf{x}_k^{(i)})\right\|^2+\sum_{j\neq l}^{m}\left\langle g(\mathbf{x}_k^{(j)})-\nabla F(\mathbf{x}_k^{(j)}),\, g(\mathbf{x}_k^{(l)})-\nabla F(\mathbf{x}_k^{(l)})\right\rangle\right] \tag{42}$$

$$=\frac{1}{m^2}\sum_{i=1}^{m}\mathbb{E}_{\xi_k^{(i)}|\mathbf{X}_k}\left\|g(\mathbf{x}_k^{(i)})-\nabla F(\mathbf{x}_k^{(i)})\right\|^2+$$

$$\frac{1}{m^2}\sum_{j\neq l}^{m}\left\langle\mathbb{E}_{\xi_k^{(j)}|\mathbf{X}_k}\left[g(\mathbf{x}_k^{(j)})-\nabla F(\mathbf{x}_k^{(j)})\right],\,\mathbb{E}_{\xi_k^{(l)}|\mathbf{X}_k}\left[g(\mathbf{x}_k^{(l)})-\nabla F(\mathbf{x}_k^{(l)})\right]\right\rangle \tag{43}$$

where equation (43) is due to $\{\xi_k^{(i)}\}$ are independent random variables. Now, directly applying Assumption 3 and 4 to (43), one can observe that all cross terms are zero. Then, we have

$$\mathbb{E}_{\Xi_K|\mathbf{X}_k} \|\mathcal{G}_k - \mathcal{H}_k\|^2 \leq \frac{1}{m^2} \sum_{i=1}^{m} \left[\beta \left\| \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 + \sigma^2 \right] \tag{44}$$

$$= \frac{\beta}{m} \frac{\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2}{m} + \frac{\sigma^2}{m}. \tag{45}$$

■

Lemma 4 *Under Assumption 3, the expected inner product between stochastic gradient and full batch gradient can be expanded as*

$$\mathbf{E}_k \left[\langle \nabla F(\mathbf{u}_k), \mathcal{G}_k \rangle \right] = \frac{1}{2} \|\nabla F(\mathbf{u}_k)\|^2 + \frac{1}{2m} \sum_{i=1}^{m} \left\| \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 - \frac{1}{2m} \sum_{i=1}^{m} \left\| \nabla F(\mathbf{u}_k) - \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 \tag{46}$$

where $\mathbf{E}_k$ *denotes the conditional expectation* $\mathbb{E}_{\Xi_K|\mathbf{X}_k}$.

Proof

$$\mathbf{E}_k \left[\langle \nabla F(\mathbf{u}_k), \mathcal{G}_k \rangle \right] = \mathbf{E}_k \left[\left\langle \nabla F(\mathbf{u}_k), \frac{1}{m} \sum_{i=1}^{m} g(\mathbf{x}_k^{(i)}) \right\rangle \right] \tag{47}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left\langle \nabla F(\mathbf{u}_k), \nabla F(\mathbf{x}_k^{(i)}) \right\rangle \tag{48}$$

$$= \frac{1}{2m} \sum_{i=1}^{m} \left[\|\nabla F(\mathbf{u}_k)\|^2 + \left\| \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 - \left\| \nabla F(\mathbf{u}_k) - \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 \right] \tag{49}$$

$$= \frac{1}{2} \|\nabla F(\mathbf{u}_k)\|^2 + \frac{1}{2m} \sum_{i=1}^{m} \left\| \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 - \frac{1}{2m} \sum_{i=1}^{m} \left\| \nabla F(\mathbf{u}_k) - \nabla F(\mathbf{x}_k^{(i)}) \right\|^2 \tag{50}$$

where equation (49) comes from $2\mathbf{a}^\top \mathbf{b} = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a} - \mathbf{b}\|^2$. ■

Lemma 5 *Under Assumption 3 and 4, the squared norm of stochastic gradient can be bounded as*

$$\mathbf{E}_k \left[\|\mathcal{G}_k\|^2 \right] \leq \left(\frac{\beta}{m} + 1 \right) \frac{\|\nabla F(\mathbf{X}_k)\|_F^2}{m} + \frac{\sigma^2}{m}.$$

Proof Since $\mathbf{E}_k[\mathcal{G}_k] = \mathcal{H}_k$, then we have

$$\mathbf{E}_k\left[\|\mathcal{G}_k\|^2\right] = \mathbf{E}_k\left[\|\mathcal{G}_k - \mathbf{E}_k[\mathcal{G}_k]\|^2\right] + \|\mathbf{E}_k[\mathcal{G}_k]\|^2 \tag{51}$$

$$= \mathbf{E}_k\left[\|\mathcal{G}_k - \mathcal{H}_k\|^2\right] + \|\mathcal{H}_k\|^2 \tag{52}$$

$$\leq \frac{\beta}{m}\frac{\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2}{m} + \frac{\sigma^2}{m} + \|\mathcal{H}_k\|^2 \tag{53}$$

$$\leq \frac{\beta}{m}\frac{\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2}{m} + \frac{\sigma^2}{m} + \frac{1}{m}\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2 \tag{54}$$

$$= \left(\frac{\beta}{m} + 1\right)\frac{\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2}{m} + \frac{\sigma^2}{m}, \tag{55}$$

where (53) follows Lemma 3 and (54) comes from the convexity of vector norm and Jensen's inequality:

$$\|\mathcal{H}_k\|^2 = \left\|\frac{1}{m}\sum_{i=1}^{m}\nabla F(\mathbf{x}_k^{(i)})\right\|^2 \leq \frac{1}{m}\sum_{i=1}^{m}\left\|\nabla F(\mathbf{x}_k^{(i)})\right\|^2 = \frac{1}{m}\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2. \tag{56}$$

■

C.1.2 Proof of Lemma 2

According to Lipschitz continuous gradient assumption, we have

$$\mathbf{E}_k\left[F(\mathbf{u}_{k+1})\right] - F(\mathbf{u}_k) \leq -\eta_{\text{eff}}\mathbf{E}_k\left[\langle\nabla F(\mathbf{u}_k),\, \mathcal{G}_k\rangle\right] + \frac{\eta_{\text{eff}}^2 L}{2}\mathbf{E}_k\left[\|\mathcal{G}_k\|^2\right]. \tag{57}$$

Combining with Lemmas 4 and 5, we obtain

$$\mathbf{E}_k\left[F(\mathbf{u}_{k+1})\right] - F(\mathbf{u}_k) \leq -\frac{\eta_{\text{eff}}}{2}\|\nabla F(\mathbf{u}_k)\|^2 - \frac{\eta_{\text{eff}}}{2m}\sum_{i=1}^{m}\left\|\nabla F(\mathbf{x}_k^{(i)})\right\|^2 + \frac{\eta_{\text{eff}}}{2m}\sum_{i=1}^{m}\left\|\nabla F(\mathbf{u}_k) - \nabla F(\mathbf{x}_k^{(i)})\right\|^2 +$$
$$\frac{\eta_{\text{eff}}^2 L}{2m}\sum_{i=1}^{m}\left\|\nabla F(\mathbf{x}_k^{(i)})\right\|^2 \cdot \left(\frac{\beta}{m} + 1\right) + \frac{\eta_{\text{eff}}^2 L\sigma^2}{2m} \tag{58}$$

$$\leq -\frac{\eta_{\text{eff}}}{2}\|\nabla F(\mathbf{u}_k)\|^2 - \frac{\eta_{\text{eff}}}{2}\left[1 - \eta_{\text{eff}}L\left(\frac{\beta}{m} + 1\right)\right]\cdot\frac{1}{m}\sum_{i=1}^{m}\left\|\nabla F(\mathbf{x}_k^{(i)})\right\|^2 +$$
$$\frac{\eta_{\text{eff}}^2 L\sigma^2}{2m} + \frac{\eta_{\text{eff}}L^2}{2m}\sum_{i=1}^{m}\left\|\mathbf{u}_k - \mathbf{x}_k^{(i)}\right\|^2. \tag{59}$$

After minor rearranging and according to the definition of Frobenius norm, it is easy to show

$$\|\nabla F(\mathbf{u}_k)\|^2 \leq \frac{2\left[F(\mathbf{u}_k) - \mathbf{E}_k\left[F(\mathbf{u}_{k+1})\right]\right]}{\eta_{\text{eff}}} + \frac{\eta_{\text{eff}}L\sigma^2}{m} + \frac{L^2}{m}\sum_{i=1}^{m}\left\|\mathbf{u}_k - \mathbf{x}_k^{(i)}\right\|^2 -$$
$$\left[1 - \eta_{\text{eff}}L\left(\frac{\beta}{m} + 1\right)\right]\frac{1}{m}\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2. \tag{60}$$

Taking the total expectation and averaging over all iterates, we have

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2\left[F(\mathbf{u}_1)-F_{\inf}\right]}{\eta_{\text{eff}}K} + \frac{\eta_{\text{eff}}L\sigma^2}{m} + \frac{L^2}{Km}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left\|\mathbf{u}_k-\mathbf{x}_k^{(i)}\right\|^2 - \left[1-\eta_{\text{eff}}L\left(\frac{\beta}{m}+1\right)\right]\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\left\|\nabla F(\mathbf{X}_k)\right\|_{\text{F}}^2}{m}. \tag{61}$$

If the effective learning rate satisfies $\eta_{\text{eff}}L(\beta/m+1)\leq 1$, then

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2\left[F(\mathbf{u}_1)-F_{\inf}\right]}{\eta_{\text{eff}}K} + \frac{\eta_{\text{eff}}L\sigma^2}{m} + \frac{L^2}{Km}\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}\left\|\mathbf{u}_k-\mathbf{x}_k^{(i)}\right\|^2. \tag{62}$$

Recalling the definition $\mathbf{u}_k=\mathbf{X}_k\mathbf{1}_{m+v}/(m+v)$ and adding a positive term to the RHS, one can get

$$\sum_{i=1}^{m}\left\|\mathbf{u}_k-\mathbf{x}_k^{(i)}\right\|^2 \leq \sum_{i=1}^{m}\left\|\mathbf{u}_k-\mathbf{x}_k^{(i)}\right\|^2 + \sum_{j=1}^{v}\left\|\mathbf{u}_k-\mathbf{z}_k^{(j)}\right\|^2 \tag{63}$$

$$= \left\|\mathbf{u}\mathbf{1}_{m+v}^\top-\mathbf{X}_k\right\|_{\text{F}}^2 \tag{64}$$

$$= \left\|\mathbf{X}_k\frac{\mathbf{1}_{m+v}\mathbf{1}_{m+v}^\top}{m+v}-\mathbf{X}_k\right\|_{\text{F}}^2 = \|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\text{F}}^2 \tag{65}$$

where $\mathbf{I},\mathbf{J}$ are $(m+v)\times(m+v)$ matrices. Plugging the inequality (65) into (62), we complete the proof.

Appendix D. Proof of Theorem 1: Convergence of Cooperative SGD

D.1 Lemmas

Lemma 6 (Kahan (2013)) *Consider two real matrices $\mathbf{A}\in\mathbb{R}^{d\times m}$ and $\mathbf{B}\in\mathbb{R}^{m\times m}$. If $\mathbf{B}$ is symmetric, then we have*

$$\|\mathbf{AB}\|_F \leq \|\mathbf{B}\|_{op}\|\mathbf{A}\|_F. \tag{66}$$

Proof Assume the rows of matirx $\mathbf{A}$ are denoted by $\mathbf{a}_1^\top,\ldots,\mathbf{a}_d^\top$ and $\mathcal{I}=\{i\in[1,d]:\|\mathbf{a}_i\|\neq 0\}$. Then, we have

$$\|\mathbf{AB}\|_{\text{F}}^2 = \sum_{i=1}^{d}\left\|\mathbf{a}_i^\top\mathbf{B}\right\|^2 = \sum_{i\in\mathcal{I}}\|\mathbf{B}\mathbf{a}_i\|^2 \tag{67}$$

$$= \sum_{i\in\mathcal{I}}^{d}\frac{\|\mathbf{B}\mathbf{a}_i\|^2}{\|\mathbf{a}_i\|^2}\|\mathbf{a}_i\|^2 \tag{68}$$

$$\leq \sum_{i\in\mathcal{I}}^{d}\|\mathbf{B}\|_{\text{op}}^2\|\mathbf{a}_i\|^2 = \|\mathbf{B}\|_{\text{op}}^2\sum_{i\in\mathcal{I}}^{d}\|\mathbf{a}_i\|^2 = \|\mathbf{B}\|_{\text{op}}^2\|\mathbf{A}\|_{\text{F}}^2 \tag{69}$$

where the last inequality follows the definition of matrix operator norm. ■

Lemma 7 (Kahan (2013)) *Suppose there are two matrices* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times m}$. *Then, we have*

$$|\operatorname{Tr}(\mathbf{AB})| \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F. \tag{70}$$

Proof Assume $\mathbf{a}_i^\top \in \mathbb{R}^n$ is the i-th row of matrix $\mathbf{A}$ and $\mathbf{b}_i \in \mathbb{R}^n$ is the i-th column of matrix $\mathbf{B}$. According to the definition of matrix trace, we have

$$\operatorname{Tr}(\mathbf{AB}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij}\mathbf{B}_{ji} \tag{71}$$

$$= \sum_{i=1}^{m} \mathbf{a}_i^\top \mathbf{b}_i. \tag{72}$$

Then, Cauchy-Schwartz inequality yields

$$|\sum_{i=1}^{m} \mathbf{a}_i^\top \mathbf{b}_i|^2 \leq \left(\sum_{i=1}^{m} \|\mathbf{a}_i\|^2\right)\left(\sum_{i=1}^{m} \|\mathbf{b}_i\|^2\right) \tag{73}$$

$$= \|\mathbf{A}\|_{\mathrm{F}}^2 \|\mathbf{B}\|_{\mathrm{F}}^2. \tag{74}$$

■

Lemma 8 *Suppose there is a* $m \times m$ *matrix* $\mathbf{W}$ *that satisfies Assumption 5. Then*

$$\left\|\mathbf{W}^j - \mathbf{J}\right\|_{op} = \zeta^j \tag{75}$$

where $\zeta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_m(\mathbf{W})|\}$.

Proof Since $\mathbf{W}$ is a real symmetric matrix, then it can be decomposed as $\mathbf{W} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$, where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{\Lambda} = \operatorname{diag}\{\lambda_1(\mathbf{W}), \lambda_2(\mathbf{W}), \ldots, \lambda_m(\mathbf{W})\}$. In particular, since the largest eigenvalue of $\mathbf{W}$ is 1 and $\mathbf{W1} = \mathbf{1}$, the corresponding eigenvector (i.e., the first column of $\mathbf{Q}$) is $\frac{\mathbf{1}}{\sqrt{m}}$. Similarly, matrix $\mathbf{J}$ can be decomposed as $\mathbf{Q}\mathbf{\Lambda}_0\mathbf{Q}^\top$ where $\mathbf{\Lambda}_0 = \operatorname{diag}\{1, 0, \ldots, 0\}$. Then, we have

$$\mathbf{W}^j - \mathbf{J} = (\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top)^j - \mathbf{J} = \mathbf{Q}\left(\mathbf{\Lambda}^j - \mathbf{\Lambda}_0\right)\mathbf{Q}^\top. \tag{76}$$

According to the definition of matrix operator norm,

$$\left\|\mathbf{W}^j - \mathbf{J}\right\|_{\mathrm{op}} = \sqrt{\lambda_{\max}((\mathbf{W}^j - \mathbf{J})^\top(\mathbf{W}^j - \mathbf{J}))} = \sqrt{\lambda_{\max}(\mathbf{W}^{2j} - \mathbf{J})}. \tag{77}$$

Since $\mathbf{W}^{2j} - \mathbf{J} = \mathbf{Q}\left(\mathbf{\Lambda}^{2j} - \mathbf{\Lambda}_0\right)\mathbf{Q}^\top$, the maximal eigenvalue will be $\max\{0, \lambda_2(\mathbf{W})^{2j}, \ldots, \lambda_m(\mathbf{W})^{2j}\} = \zeta^{2j}$. As a consequence, we have $\left\|\mathbf{W}^j - \mathbf{J}\right\|_{\mathrm{op}} = \sqrt{\lambda_{\max}(\mathbf{W}^{2j} - \mathbf{J})} = \zeta^j$.

■

D.2 Proof of Theorem 1

Recall the intermediate result (61) in the proof of Lemma 2:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2\left[F(\mathbf{u}_1)-F_{\inf}\right]}{\eta_{\text{eff}}K} + \frac{\eta_{\text{eff}}L\sigma^2}{m} + \frac{L^2}{Km}\sum_{k=1}^{K}\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\text{F}}^2 - \left[1-\eta_{\text{eff}}L\left(\frac{\beta}{m}+1\right)\right]\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\|\nabla F(\mathbf{X}_k)\|_{\text{F}}^2}{m}. \tag{78}$$

Our goal is to provide an upper bound for the network error term $\frac{L^2}{Km}\sum_{k=1}^{K}\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\text{F}}^2$. First of all, let us derive a specific expression for $\mathbf{X}_k(\mathbf{I}-\mathbf{J})$.

D.2.1 Decomposition.

According to the update rule (10) in Section 3, one can observe that

$$\begin{aligned}\mathbf{X}_k(\mathbf{I}-\mathbf{J}) =& \left(\mathbf{X}_{k-1}-\eta\mathbf{G}_{k-1}\right)\mathbf{S}_{k-1}(\mathbf{I}-\mathbf{J}) &(79)\\ =&\mathbf{X}_{k-1}(\mathbf{I}-\mathbf{J})\mathbf{S}_{k-1}-\eta\mathbf{G}_{k-1}(\mathbf{S}_{k-1}-\mathbf{J}) &(80)\end{aligned}$$

where (80) follows the special property of doubly stochastic matrix: $\mathbf{S}_{k-1}\mathbf{J}=\mathbf{J}\mathbf{S}_{k-1}=\mathbf{J}$ and hence $(\mathbf{I}-\mathbf{J})\mathbf{S}_{k-1}=\mathbf{S}_{k-1}(\mathbf{I}-\mathbf{J})$. Then, expanding the expression of $\mathbf{X}_{k-1}$, we have

$$\begin{aligned}\mathbf{X}_k(\mathbf{I}-\mathbf{J}) =& \left[\mathbf{X}_{k-2}(\mathbf{I}-\mathbf{J})\mathbf{S}_{k-2}-\eta\mathbf{G}_{k-2}(\mathbf{S}_{k-2}-\mathbf{J})\right]\mathbf{S}_{k-1}-\eta\mathbf{G}_{k-1}(\mathbf{S}_{k-1}-\mathbf{J}) &(81)\\ =&\mathbf{X}_{k-2}(\mathbf{I}-\mathbf{J})\mathbf{S}_{k-2}\mathbf{S}_{k-1}-\eta\mathbf{G}_{k-2}(\mathbf{S}_{k-2}\mathbf{S}_{k-1}-\mathbf{J})-\eta\mathbf{G}_{k-1}(\mathbf{S}_{k-1}-\mathbf{J}) &(82)\end{aligned}$$

Repeating the same procedure for $\mathbf{X}_{k-2},\mathbf{X}_{k-3},\ldots,\mathbf{X}_2$, finally we get

$$\mathbf{X}_k(\mathbf{I}-\mathbf{J}) =\mathbf{X}_1(\mathbf{I}-\mathbf{J})\mathbf{\Phi}_{1,k-1}-\eta\sum_{s=1}^{k-1}\mathbf{G}_s(\mathbf{\Phi}_{s,k-1}-\mathbf{J}) \tag{83}$$

where $\mathbf{\Phi}_{s,k-1}=\prod_{l=s}^{k-1}\mathbf{S}_l$. Since all optimization variables are initialized at the same point $\mathbf{X}_1(\mathbf{I}-\mathbf{J})=0$, the squared norm of the network error term can be directly written as

$$\mathbb{E}\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\text{F}}^2 =\eta^2\mathbb{E}\left\|\sum_{s=1}^{k-1}\mathbf{G}_s(\mathbf{\Phi}_{s,k-1}-\mathbf{J})\right\|_{\text{F}}^2. \tag{84}$$

Then, let us take a closer look at the expression of $\mathbf{\Phi}_{s,k-1}$. Without loss of generality, assume $k=j\tau+i$, where j denotes the index of communication rounds and i denotes the index of local updates. As a result, matrix $\mathbf{\Phi}_{s,k-1}$ can be expressed as follows:

$$\mathbf{\Phi}_{s,k-1}=\begin{cases}\mathbf{I}, & j\tau<s<j\tau+i\\ \mathbf{W}, & (j-1)\tau<s\leq j\tau\\ \mathbf{W}^2, & (j-2)\tau<s\leq(j-1)\tau\\ \ldots & \\ \mathbf{W}^j, & 0<s\leq\tau\end{cases}. \tag{85}$$

For the ease of writing, define accumulated stochastic gradient within one local update period as $\mathbf{Y}_r = \sum_{s=r\tau+1}^{(r+1)\tau} \mathbf{G}_s$ for $0 \leq r < j$ and $\mathbf{Y}_j = \sum_{s=j\tau+1}^{j\tau+i-1} \mathbf{G}_s$. Similarly, define accumulated full batch gradient $\mathbf{Q}_r = \sum_{s=r\tau+1}^{(r+1)\tau} \nabla F(\mathbf{X}_s)$ for $0 \leq r < j$ and $\mathbf{Q}_j = \sum_{s=j\tau+1}^{j\tau+i-1} \nabla F(\mathbf{X}_s)$. Accordingly, we have

$$\sum_{s=1}^{\tau} \mathbf{G}_s(\mathbf{\Phi}_{s,k-1} - \mathbf{J}) = \mathbf{Y}_0(\mathbf{W}^j - \mathbf{J}), \tag{86}$$

$$\sum_{s=\tau+1}^{2\tau} \mathbf{G}_s(\mathbf{\Phi}_{s,k-1} - \mathbf{J}) = \mathbf{Y}_1(\mathbf{W}^{j-1} - \mathbf{J}), \tag{87}$$

$$\dots \tag{88}$$

$$\sum_{s=j\tau+1}^{j\tau+i-1} \mathbf{G}_s(\mathbf{\Phi}_{s,k-1} - \mathbf{J}) = \mathbf{Y}_j(\mathbf{I} - \mathbf{J}). \tag{89}$$

Thus, summing all these terms we get

$$\sum_{s=1}^{k-1} \mathbf{G}_s(\mathbf{\Phi}_{s,k-1} - \mathbf{J}) = \sum_{r=0}^{j} \mathbf{Y}_r(\mathbf{W}^{j-r} - \mathbf{J}). \tag{90}$$

Note that the network error term can be decomposed into two parts:

$$\mathbb{E}\left\|\mathbf{X}_k(\mathbf{I} - \mathbf{J})\right\|_{\mathrm{F}}^2 = \eta^2 \mathbb{E}\left\|\sum_{r=0}^{j} \mathbf{Y}_r(\mathbf{W}^{j-r} - \mathbf{J})\right\|_{\mathrm{F}}^2 \tag{91}$$

$$= \eta^2 \mathbb{E}\left\|\sum_{r=0}^{j} (\mathbf{Y}_r - \mathbf{Q}_r)(\mathbf{W}^{j-r} - \mathbf{J}) + \sum_{r=0}^{j} \mathbf{Q}_r(\mathbf{W}^{j-r} - \mathbf{J})\right\|_{\mathrm{F}}^2 \tag{92}$$

$$\leq \underbrace{2\eta^2 \mathbb{E}\left\|\sum_{r=0}^{j} (\mathbf{Y}_r - \mathbf{Q}_r)(\mathbf{W}^{j-r} - \mathbf{J})\right\|_{\mathrm{F}}^2}_{T_1} + \underbrace{2\eta^2 \mathbb{E}\left\|\sum_{r=0}^{j} \mathbf{Q}_r(\mathbf{W}^{j-r} - \mathbf{J})\right\|_{\mathrm{F}}^2}_{T_2} \tag{93}$$

where (93) follows $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$. Next, we are going to separately provide bounds for T_1 and T_2. Recall that we are interested in the average of all iterates $\frac{L^2}{Km}\sum_{k=1}^{K} \left\|\mathbf{X}_k(\mathbf{I} - \mathbf{J})\right\|_{\mathrm{F}}^2$. Accordingly, we will also derive the bounds for $\frac{L^2}{Km}\sum_{k=1}^{K} T_1$ and $\frac{L^2}{Km}\sum_{k=1}^{K} T_2$.

D.2.2 Bounding T_1.

For the first term T_1, we have

$$
\begin{aligned}
T_1 =& 2\eta^2 \sum_{r=0}^{j} \mathbb{E} \left\| (\mathbf{Y}_r - \mathbf{Q}_r)(\mathbf{W}^{j-r} - \mathbf{J}) \right\|_{\mathrm{F}}^2 && (94)\\
\leq& 2\eta^2 \sum_{r=0}^{j} \mathbb{E} \left\| \mathbf{Y}_r - \mathbf{Q}_r \right\|_{\mathrm{F}}^2 \left\| \mathbf{W}^{j-r} - \mathbf{J} \right\|_{\mathrm{op}}^2 && (95)\\
=& 2\eta^2 \sum_{r=0}^{j} \mathbb{E} \left\| \mathbf{Y}_r - \mathbf{Q}_r \right\|_{\mathrm{F}}^2 \zeta^{2(j-r)} && (96)\\
=& 2\eta^2 \sum_{r=0}^{j-1} \mathbb{E} \left\| \mathbf{Y}_r - \mathbf{Q}_r \right\|_{\mathrm{F}}^2 \zeta^{2(j-r)} + 2\eta^2 \mathbb{E} \left\| \mathbf{Y}_j - \mathbf{Q}_j \right\|_{\mathrm{F}}^2 && (97)
\end{aligned}
$$

where (95) follows Lemma 6, (96) comes from Lemma 8. Recall that $\zeta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_{m+v}(\mathbf{W})|\}$. Then for any $0 \leq r < j$,

$$
\begin{aligned}
\mathbb{E}\left[\left\| \mathbf{Y}_r - \mathbf{Q}_r \right\|_{\mathrm{F}}^2 \right] =& \mathbb{E}\left[\left\| \sum_{s=r\tau+1}^{(r+1)\tau} [\mathbf{G}_s - \nabla F(\mathbf{X}_s)] \right\|_{\mathrm{F}}^2 \right] && (98)\\
=& \sum_{i=1}^{m} \mathbb{E}\left[\left\| \sum_{s=r\tau+1}^{(r+1)\tau} \left[g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}) \right] \right\|^2 \right] && (99)\\
=& \sum_{i=1}^{m} \mathbb{E}\left[\sum_{s=r\tau+1}^{(r+1)\tau} \left\| g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}) \right\|^2 \right] + \\
& \mathbb{E}\left[\sum_{s \neq l} \left\langle g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}),\, g(\mathbf{x}_l^{(i)}) - \nabla F(\mathbf{x}_l^{(i)}) \right\rangle \right]. && (100)
\end{aligned}
$$

Now we show that the cross terms are zero. For any $s < l$, according to Assumption 4, one can obtain

$$
\begin{aligned}
& \mathbb{E}\left[\left\langle g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}),\, g(\mathbf{x}_l^{(i)}) - \nabla F(\mathbf{x}_l^{(i)}) \right\rangle \right] \\
=& \mathbb{E}_{\mathbf{x}_s^{(i)}, \xi_s^{(i)}, \mathbf{x}_l^{(i)}} \mathbb{E}_{\xi_l^{(i)} | \mathbf{x}_s^{(i)}, \xi_s^{(i)}, \mathbf{x}_l^{(i)}} \left[\left\langle g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}),\, g(\mathbf{x}_l^{(i)}) - \nabla F(\mathbf{x}_l^{(i)}) \right\rangle \right] && (101)\\
=& \mathbb{E}\left[\left\langle g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}),\, \mathbb{E}_{\xi_l^{(i)} | \mathbf{x}_s^{(i)}, \xi_s^{(i)}, \mathbf{x}_l^{(i)}} \left[g(\mathbf{x}_l^{(i)}) - \nabla F(\mathbf{x}_l^{(i)}) \right] \right\rangle \right] && (102)\\
=& \mathbb{E}\left[\left\langle g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)}),\, 0 \right\rangle \right] = 0. && (103)
\end{aligned}
$$

As a result, we have

$$\mathbb{E}\left[\|\mathbf{Y}_r - \mathbf{Q}_r\|_{\mathrm{F}}^2\right] = \mathbb{E}\left[\sum_{s=r\tau+1}^{(r+1)\tau}\sum_{i=1}^{m}\left\|g(\mathbf{x}_s^{(i)}) - \nabla F(\mathbf{x}_s^{(i)})\right\|^2\right] \tag{104}$$

$$\leq \beta \sum_{s=r\tau+1}^{(r+1)\tau}\sum_{i=1}^{m}\mathbb{E}\left[\left\|\nabla F(\mathbf{x}_s^{(i)})\right\|^2\right] + \tau m\sigma^2 \tag{105}$$

$$= \beta \sum_{s=r\tau+1}^{(r+1)\tau}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2 + \tau m\sigma^2 \tag{106}$$

where (105) is according to Assumption 4. Using the same technique, one can obtain that

$$\mathbb{E}\,\|\mathbf{Y}_j - \mathbf{Q}_j\|_{\mathrm{F}}^2 \leq \beta \sum_{s=j\tau+1}^{j\tau+i-1}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2 + (i-1)m\sigma^2. \tag{107}$$

Substituting (106) and (107) back into (97), we have

$$T_1 \leq 2\eta^2\sum_{r=0}^{j-1}\left[\zeta^{2(j-r)}\left(\beta\sum_{s=r\tau+1}^{(r+1)\tau}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2 + \tau m\sigma^2\right)\right] + 2\eta^2\beta\sum_{s=j\tau+1}^{j\tau+i-1}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2 + 2\eta^2(i-1)m\sigma^2 \tag{108}$$

$$\leq 2\eta^2 m\sigma^2\left[\frac{\zeta^2}{1-\zeta^2}\tau + i - 1\right] + 2\eta^2\beta\sum_{r=0}^{j-1}\left[\zeta^{2(j-r)}\left(\sum_{s=r\tau+1}^{(r+1)\tau}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2\right)\right] + 2\eta^2\beta\sum_{s=j\tau+1}^{j\tau+i-1}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2 \tag{109}$$

where (109) follows the summation formula of power series:

$$\sum_{r=0}^{j-1}\zeta^{2(j-r)} \leq \sum_{r=-\infty}^{j-1}\zeta^{2(j-r)} \leq \frac{\zeta^2}{1-\zeta^2}. \tag{110}$$

Next, summing over all iterates in the j-th local update period (from $i = 1$ to $i = \tau$):

$$\sum_{i=1}^{\tau}T_1 \leq \eta^2 m\sigma^2\left[\frac{2\zeta^2}{1-\zeta^2}\tau^2 + \tau(\tau-1)\right] + 2\eta^2\beta\tau\sum_{r=0}^{j-1}\left[\zeta^{2(j-r)}\left(\sum_{s=r\tau+1}^{(r+1)\tau}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2\right)\right] +$$
$$2\eta^2\beta\tau\sum_{s=j\tau+1}^{(j+1)\tau-1}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2 \tag{111}$$

$$\leq \eta^2 m\sigma^2\left[\frac{2\zeta^2}{1-\zeta^2}\tau^2 + \tau(\tau-1)\right] + 2\eta^2\beta\tau\sum_{r=0}^{j}\left[\zeta^{2(j-r)}\left(\sum_{s=r\tau+1}^{(r+1)\tau}\|\nabla F(\mathbf{X}_s)\|_{\mathrm{F}}^2\right)\right]. \tag{112}$$

Then, summing over all periods from $j=0$ to $j=K/\tau-1$, where K is the total iterations:

$$\sum_{j=0}^{K/\tau-1}\sum_{i=1}^{\tau} T_1 \leq \frac{K}{\tau}\eta^2 m\sigma^2\left[\frac{2\zeta^2}{1-\zeta^2}\tau^2+\tau(\tau-1)\right]+2\eta^2\beta\tau\sum_{j=0}^{K/\tau-1}\sum_{r=0}^{j}\left[\zeta^{2(j-r)}\left(\sum_{s=r\tau+1}^{(r+1)\tau}\left\|\nabla F(\mathbf{X}_s)\right\|_{\mathrm{F}}^2\right)\right] \tag{113}$$

$$= K\eta^2 m\sigma^2\left[\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right]+2\eta^2\beta\tau\sum_{j=0}^{K/\tau-1}\sum_{r=0}^{j}\left[\zeta^{2(j-r)}\left(\sum_{s=r\tau+1}^{(r+1)\tau}\left\|\nabla F(\mathbf{X}_s)\right\|_{\mathrm{F}}^2\right)\right]. \tag{114}$$

Expanding the summation in (114), we have

$$\sum_{j=0}^{K/\tau-1}\sum_{i=1}^{\tau} T_1 \leq K\eta^2 m\sigma^2\left[\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right]+2\eta^2\beta\tau\sum_{r=0}^{K/\tau-1}\left[\left(\sum_{s=r\tau+1}^{(r+1)\tau}\left\|\nabla F(\mathbf{X}_s)\right\|_{\mathrm{F}}^2\right)\left(\sum_{j=r}^{K/\tau-1}\zeta^{2(j-r)}\right)\right] \tag{115}$$

$$\leq K\eta^2 m\sigma^2\left[\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right]+2\eta^2\beta\tau\sum_{r=0}^{K/\tau-1}\left[\left(\sum_{s=r\tau+1}^{(r+1)\tau}\left\|\nabla F(\mathbf{X}_s)\right\|_{\mathrm{F}}^2\right)\left(\sum_{j=r}^{+\infty}\zeta^{2(j-r)}\right)\right] \tag{116}$$

$$\leq K\eta^2 m\sigma^2\left[\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right]+\frac{2\eta^2\beta\tau}{1-\zeta^2}\sum_{r=0}^{K/\tau-1}\left(\sum_{s=r\tau+1}^{(r+1)\tau}\left\|\nabla F(\mathbf{X}_s)\right\|_{\mathrm{F}}^2\right) \tag{117}$$

$$= K\eta^2 m\sigma^2\left[\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right]+\frac{2\eta^2\beta\tau}{1-\zeta^2}\sum_{k=1}^{K}\left\|\nabla F(\mathbf{X}_k)\right\|_{\mathrm{F}}^2. \tag{118}$$

Here, we complete the first part.

D.2.3 Bounding T_2.

For the second term in (93), since $\|\mathbf{A}\|_{\mathrm{F}}^2=\mathrm{Tr}(\mathbf{A}^\top\mathbf{A})$, we have

$$T_2 = 2\eta^2\sum_{r=0}^{j}\mathbb{E}\left\|\mathbf{Q}_r(\mathbf{W}^{j-r}-\mathbf{J})\right\|_{\mathrm{F}}^2+2\eta^2\sum_{n=0}^{j}\sum_{l=0,l\neq n}^{j}\mathbb{E}\left[\mathrm{Tr}\left((\mathbf{W}^{j-n}-\mathbf{J})\mathbf{Q}_n^\top\mathbf{Q}_l(\mathbf{W}^{j-l}-\mathbf{J})\right)\right]. \tag{119}$$

According to Lemma 7, the trace can be bounded as:

$$\left|\mathrm{Tr}\left((\mathbf{W}^{j-n}-\mathbf{J})\mathbf{Q}_n^\top\mathbf{Q}_l(\mathbf{W}^{j-l}-\mathbf{J})\right)\right| \leq \left\|(\mathbf{W}^{j-n}-\mathbf{J})\mathbf{Q}_n^\top\right\|_{\mathrm{F}}\left\|\mathbf{Q}_l(\mathbf{W}^{j-l}-\mathbf{J})\right\|_{\mathrm{F}} \tag{120}$$

$$\leq \left\|\mathbf{W}^{j-n}-\mathbf{J}\right\|_{\mathrm{op}}\left\|\mathbf{Q}_n\right\|_{\mathrm{F}}\left\|\mathbf{Q}_l\right\|_{\mathrm{F}}\left\|\mathbf{W}^{j-l}-\mathbf{J}\right\|_{\mathrm{op}} \tag{121}$$

$$\leq \frac{1}{2}\zeta^{2j-n-l}\left[\left\|\mathbf{Q}_n\right\|_{\mathrm{F}}^2+\left\|\mathbf{Q}_l\right\|_{\mathrm{F}}^2\right] \tag{122}$$

where (121) follows Lemma 6 and (122) is because of $2ab \leq a^2 + b^2$. Then, it follows that

$$T_2 \leq 2\eta^2 \sum_{r=0}^{j} \mathbb{E} \left\| \mathbf{Q}_r \right\|_{\mathrm{F}}^2 \left\| (\mathbf{W}^{j-r} - \mathbf{J}) \right\|_{\mathrm{op}}^2 + \eta^2 \sum_{n=0}^{j} \sum_{l=0, l\neq n}^{j} \zeta^{2j-n-l} \mathbb{E} \left[\left\| \mathbf{Q}_n \right\|_{\mathrm{F}}^2 + \left\| \mathbf{Q}_l \right\|_{\mathrm{F}}^2 \right] \tag{123}$$

$$= 2\eta^2 \sum_{r=0}^{j} \zeta^{2(j-r)} \mathbb{E} \left\| \mathbf{Q}_r \right\|_{\mathrm{F}}^2 + 2\eta^2 \sum_{n=0}^{j} \sum_{l=0, l\neq n}^{j} \zeta^{2j-n-l} \mathbb{E} \left\| \mathbf{Q}_n \right\|_{\mathrm{F}}^2 \tag{124}$$

$$= 2\eta^2 \sum_{r=0}^{j} \zeta^{2(j-r)} \mathbb{E} \left\| \mathbf{Q}_r \right\|_{\mathrm{F}}^2 + 2\eta^2 \sum_{n=0}^{j} \zeta^{j-n} \mathbb{E} \left\| \mathbf{Q}_n \right\|_{\mathrm{F}}^2 \sum_{l=0, l\neq n}^{j} \zeta^{j-l} \tag{125}$$

$$= 2\eta^2 \left[\sum_{r=0}^{j-1} \zeta^{2(j-r)} \mathbb{E} \left\| \mathbf{Q}_r \right\|_{\mathrm{F}}^2 + \sum_{n=0}^{j-1} \zeta^{j-n} \mathbb{E} \left\| \mathbf{Q}_n \right\|_{\mathrm{F}}^2 \sum_{l=0, l\neq n}^{j} \zeta^{j-l} + \mathbb{E} \left\| \mathbf{Q}_j \right\|_{\mathrm{F}}^2 + \mathbb{E} \left\| \mathbf{Q}_j \right\|_{\mathrm{F}}^2 \sum_{l=0}^{j-1} \zeta^{j-l} \right] \tag{126}$$

$$\leq 2\eta^2 \left[\sum_{r=0}^{j-1} \zeta^{2(j-r)} \mathbb{E} \left\| \mathbf{Q}_r \right\|_{\mathrm{F}}^2 + \sum_{n=0}^{j-1} \frac{\zeta^{j-n}}{1-\zeta} \mathbb{E} \left\| \mathbf{Q}_n \right\|_{\mathrm{F}}^2 + \mathbb{E} \left\| \mathbf{Q}_j \right\|_{\mathrm{F}}^2 + \mathbb{E} \left\| \mathbf{Q}_j \right\|_{\mathrm{F}}^2 \frac{\zeta}{1-\zeta} \right] \tag{127}$$

where (124) uses the fact that indices n and l are symmetric and (127) is according to the summation formula of power series:

$$\sum_{l=0, l\neq n}^{j} \zeta^{j-l} \leq \sum_{l=-\infty}^{j} \zeta^{j-l} \leq \frac{1}{1-\zeta}, \tag{128}$$

$$\sum_{l=0}^{j-1} \zeta^{j-l} \leq \sum_{l=-\infty}^{j-1} \zeta^{j-l} \leq \frac{\zeta}{1-\zeta}. \tag{129}$$

After minor rearranging, we have

$$T_2 \leq 2\eta^2 \sum_{r=0}^{j-1} \left[\left(\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta} \right) \mathbb{E} \left\| \mathbf{Q}_r \right\|_{\mathrm{F}}^2 \right] + \frac{2\eta^2}{1-\zeta} \mathbb{E} \left\| \mathbf{Q}_j \right\|_{\mathrm{F}}^2 \tag{130}$$

$$= 2\eta^2 \sum_{r=0}^{j-1} \left[\left(\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta} \right) \mathbb{E} \left\| \sum_{s=1}^{\tau} \nabla F(\mathbf{X}_{r\tau+s}) \right\|_{\mathrm{F}}^2 \right] + \frac{2\eta^2}{1-\zeta} \mathbb{E} \left\| \sum_{s=1}^{i-1} \nabla F(\mathbf{X}_{j\tau+s}) \right\|_{\mathrm{F}}^2 \tag{131}$$

$$\leq 2\eta^2 \tau \sum_{r=0}^{j-1} \left[\left(\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta} \right) \sum_{s=1}^{\tau} \mathbb{E} \left\| \nabla F(\mathbf{X}_{r\tau+s}) \right\|_{\mathrm{F}}^2 \right] + \frac{2\eta^2 (i-1)}{1-\zeta} \sum_{s=1}^{i-1} \mathbb{E} \left\| \nabla F(\mathbf{X}_{j\tau+s}) \right\|_{\mathrm{F}}^2. \tag{132}$$

where (132) follows the convexity of Frobenius norm and Jensen's inequality. Next, summing over all iterates in the j-th period, we can get

$$\sum_{i=1}^{\tau} T_2 \leq 2\eta^2\tau^2 \sum_{r=0}^{j-1} \left[\left(\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta} \right) \sum_{s=1}^{\tau} \mathbb{E} \left\| \nabla F(\mathbf{X}_{r\tau+s}) \right\|_{\mathrm{F}}^2 \right] + \eta^2\tau(\tau-1)\frac{1}{1-\zeta} \sum_{s=1}^{\tau-1} \mathbb{E} \left\| \nabla F(\mathbf{X}_{j\tau+s}) \right\|_{\mathrm{F}}^2 . \tag{133}$$

Now, we are going to provide a bound for the summation over all periods (from $j=0$ to $j=K/\tau-1$). For clarity, let us first focus on the r-th local update period ($r<j$). The coefficient of $\sum_{s=1}^{\tau} \mathbb{E} \left\| \nabla F(\mathbf{X}_{r\tau+s}) \right\|^2$ in (133) is

$$\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta}. \tag{134}$$

Accordingly, the coefficient of $\sum_{s=1}^{\tau} \mathbb{E} \left\| \nabla F(\mathbf{X}_{r\tau+s}) \right\|^2$ in $\sum_{j=0}^{K/\tau-1} \sum_{i=1}^{\tau} T_2$ can be written as:

$$\sum_{j=r+1}^{K/\tau-1} \left(\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta} \right) \leq \sum_{j=r+1}^{\infty} \left(\zeta^{2(j-r)} + \frac{\zeta^{j-r}}{1-\zeta} \right) \tag{135}$$

$$\leq \frac{\zeta^2}{1-\zeta^2} + \frac{\zeta}{(1-\zeta)^2}. \tag{136}$$

As a result, we have

$$\sum_{j=0}^{K/\tau-1} \sum_{i=1}^{\tau} T_2 \leq 2\eta^2\tau^2 \left(\frac{\zeta^2}{1-\zeta^2} + \frac{\zeta}{(1-\zeta)^2} \right) \sum_{j=1}^{K/\tau-1} \sum_{s=1}^{\tau} \mathbb{E} \left\| \nabla F(\mathbf{X}_{j\tau+s}) \right\|_{\mathrm{F}}^2 + \frac{\eta^2\tau(\tau-1)}{1-\zeta} \sum_{j=0}^{K/\tau-1} \sum_{s=1}^{\tau-1} \mathbb{E} \left\| \nabla F(\mathbf{X}_{j\tau+s}) \right\|_{\mathrm{F}}^2 \tag{137}$$

Replacing all indices by k,

$$\sum_{j=0}^{K/\tau-1} \sum_{i=1}^{\tau} T_2 \leq 2\eta^2\tau^2 \left(\frac{\zeta^2}{1-\zeta^2} + \frac{\zeta}{(1-\zeta)^2} \right) \sum_{k=1}^{K} \mathbb{E} \left\| \nabla F(\mathbf{X}_k) \right\|_{\mathrm{F}}^2 + \frac{\eta^2\tau(\tau-1)}{1-\zeta} \sum_{k=1}^{K} \mathbb{E} \left\| \nabla F(\mathbf{X}_k) \right\|_{\mathrm{F}}^2 \tag{138}$$

$$= \frac{\eta^2\tau^2}{1-\zeta} \left(\frac{2\zeta^2}{1+\zeta} + \frac{2\zeta}{1-\zeta} + \frac{\tau-1}{\tau} \right) \sum_{k=1}^{K} \mathbb{E} \left\| \nabla F(\mathbf{X}_k) \right\|_{\mathrm{F}}^2 . \tag{139}$$

We complete the second part.

D.2.4 Final result.

According to (93), (118) and (139), the network error can be bounded as

$$\frac{1}{Km}\sum_{k=1}^{K}\mathbb{E}\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2 \leq \frac{1}{Km}\sum_{j=0}^{K/\tau-1}\sum_{i=1}^{\tau}(T_1+T_2) \tag{140}$$

$$\leq \eta^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right)+\frac{2\eta^2\beta\tau}{1-\zeta^2}\frac{1}{K}\sum_{k=1}^{K}\frac{\left\|\nabla F(\mathbf{X}_k)\right\|_{\mathrm{F}}^2}{m}+$$

$$\frac{\eta^2\tau^2}{1-\zeta}\left(\frac{2\zeta^2}{1+\zeta}+\frac{2\zeta}{1-\zeta}+\frac{\tau-1}{\tau}\right)\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\left\|\nabla F(\mathbf{X}_k)\right\|_{\mathrm{F}}^2}{m}. \tag{141}$$

Substituting the expression of network error back to inequality (61), we obtain

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla F(\mathbf{u}_k)\right\|^2 \leq \frac{2(F(\mathbf{x}_1)-F_{\inf})}{\eta_{\mathrm{eff}}K}+\frac{\eta_{\mathrm{eff}}L\sigma^2}{m}+\eta^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right)-$$

$$\left[1-\eta_{\mathrm{eff}}L\left(\frac{\beta}{m}+1\right)-\frac{2\eta^2L^2\beta\tau}{1-\zeta^2}\right]\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\left\|\nabla F(\mathbf{X}_k)\right\|_{\mathrm{F}}^2}{m}+$$

$$\frac{\eta^2L^2\tau^2}{1-\zeta}\left(\frac{2\zeta^2}{1+\zeta}+\frac{2\zeta}{1-\zeta}+\frac{\tau-1}{\tau}\right)\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\left\|\nabla F(\mathbf{X}_k)\right\|_{\mathrm{F}}^2}{m}. \tag{142}$$

When the learning rate satisfies

$$\eta_{\mathrm{eff}}L\left(\frac{\beta}{m}+1\right)+\frac{2\eta^2L^2\beta\tau}{1-\zeta^2}+\frac{\eta^2L^2\tau^2}{1-\zeta}\left(\frac{2\zeta^2}{1+\zeta}+\frac{2\zeta}{1-\zeta}+\frac{\tau-1}{\tau}\right)\leq 1, \tag{143}$$

we have

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla F(\mathbf{u}_k)\right\|^2 \leq \frac{2(F(\mathbf{x}_1)-F_{\inf})}{\eta_{\mathrm{eff}}K}+\frac{\eta_{\mathrm{eff}}L\sigma^2}{m}+\eta^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right) \tag{144}$$

where $\eta_{\mathrm{eff}}=m\eta/(m+v)$ and $\zeta=\max\{|\lambda_2(\mathbf{W})|,|\lambda_{m+v}(\mathbf{W})|\}$. Setting $\beta=0$, the condition on learning rate (143) can be further simplified as follows:

$$\frac{m}{m+v}\eta L+\frac{\eta^2L^2\tau^2}{1-\zeta}\left(\frac{2\zeta^2}{1+\zeta}+\frac{2\zeta}{1-\zeta}+\frac{\tau-1}{\tau}\right) \tag{145}$$

$$=\frac{m}{m+v}\eta L+\frac{\eta^2L^2\tau^2}{(1-\zeta)^2}\left(\frac{2\zeta^2(1-\zeta)}{1+\zeta}+2\zeta+\frac{\tau-1}{\tau}(1-\zeta)\right) \tag{146}$$

$$\leq\frac{m}{m+v}\eta L+\frac{\eta^2L^2\tau^2}{(1-\zeta)^2}(2+2+1) \tag{147}$$

$$=\frac{m}{m+v}\eta L+\frac{5\eta^2L^2\tau^2}{(1-\zeta)^2}\leq 1. \tag{148}$$

Here, we complete the proof.

Appendix E. Proof of Corollary 1 (Optimized Learning Rate)

Directly substituting $\eta = \frac{m+v}{Lm}\sqrt{\frac{m}{K}}$ into (144), we have

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla F(\mathbf{u}_k)\right\|^2 \leq \frac{2L(F(\mathbf{x}_1)-F_{\text{inf}})}{\sqrt{mK}} + \frac{\sigma^2}{\sqrt{mK}} + \frac{m}{K}\left(1+\frac{v}{m}\right)^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right)\sigma^2. \tag{149}$$

Note that the learning rate should satisfy the condition in (148). That is, the total iterations should satisfy:

$$\sqrt{\frac{m}{K}} + \frac{5m}{K}\left[\left(1+\frac{v}{m}\right)\frac{\tau}{1-\zeta}\right]^2 \leq 1. \tag{150}$$

When K is sufficiently large, the first term can be arbitrarily small. In particular, when $K > 4m$, the first term will be smaller than $1/2$. Then, it is enough to show the second term is smaller than $1/2$ as well.

$$\frac{5m}{K}\left[\left(1+\frac{v}{m}\right)\frac{\tau}{1-\zeta}\right]^2 \leq \frac{1}{2} \tag{151}$$

$$\Rightarrow K \geq 10m\left[\left(1+\frac{v}{m}\right)\frac{\tau}{1-\zeta}\right]^2. \tag{152}$$

Here, we complete the proof of the first part. Furthermore, when the communication period and total iterations satisfy

$$\frac{1}{\sqrt{mK}} \geq \frac{(m+v)}{K}\left(1+\frac{v}{m}\right)\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right) \tag{153}$$

then the last term in (149) is smaller than the second term. As a result, we have

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla F(\mathbf{u}_k)\right\|^2 \leq \frac{2L(F(\mathbf{x}_1)-F_{\text{inf}})}{\sqrt{mK}} + \frac{2\sigma^2}{\sqrt{mK}}. \tag{154}$$

In order to get a lower bound on K from (153), it is enough to show

$$\frac{(m+v)}{K}\left(1+\frac{v}{m}\right)\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right) \tag{155}$$

$$\leq \frac{(m+v)}{K}\left(1+\frac{v}{m}\right)\frac{1+\zeta^2}{1+\zeta}\frac{\tau}{1-\zeta} \tag{156}$$

$$\leq \frac{(m+v)}{K}\left(1+\frac{v}{m}\right)\frac{\tau}{1-\zeta} \leq \frac{1}{\sqrt{mK}} \tag{157}$$

$$\Rightarrow K \geq (m+v)^2 m\left[\left(1+\frac{v}{m}\right)\frac{\tau}{1-\zeta}\right]^2. \tag{158}$$

Once $m+v \geq \sqrt{10} \approx 3.1$, (158) is more strict than (152).

Appendix F. Proof of Theorem 2: Analysis of Non-IID Case

Now, we are going to present the theorems for the non-i.i.d. distributed data case. To start with, we need to slightly revise the assumptions since the stochastic gradient is no longer unbiased estimator of the global objective's gradient. Recall that $F_i(\mathbf{x})$ denotes the local objective function and let $g_i(\mathbf{x})$ denote the stochastic gradient computed by the i-th worker. Then, we have the following assumptions:

1. (Smoothness): $\|\nabla F_i(\mathbf{x}) - \nabla F_i(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$;

2. (Lower bounded): $F(\mathbf{x}) \geq F_{\text{inf}}$;

3. (Unbiased gradients): $\mathbb{E}_{\xi|\mathbf{x}} [g_i(\mathbf{x})] = \nabla F_i(\mathbf{x})$;

4. (Bounded variance): $\mathbb{E}_{\xi|\mathbf{x}} \|g_i(\mathbf{x}) - \nabla F_i(\mathbf{x})\|^2 \leq \sigma^2$ where σ^2 is a non-negative constant and in inverse proportion to the mini-batch size.

5. (Bounded Dissimilarities): $\frac{1}{m}\sum_{i=1}^m \|\nabla F_i(\mathbf{x}) - \nabla F(\mathbf{x})\|^2 \leq \kappa^2$ where κ^2 is a non-negative constant.

6. (Mixing Matrix): $\mathbf{W}\mathbf{1}_{m+v} = \mathbf{1}_{m+v}$, $\mathbf{W}^\top = \mathbf{W}$. Besides, the magnitudes of all eigenvalues except the largest one are strictly less than 1: $\max\{|\lambda_2(\mathbf{W})|, |\lambda_{m+v}(\mathbf{W})|\} < \lambda_1(\mathbf{W}) = 1$.

The main parts of the proof just follow the proof of Theorem 1. We only need to re-prove Lemma 2 in the context of new assumptions.

Since the objective function is L-smooth, we have

$$\mathbf{E}_k [F(\mathbf{u}_{k+1})] - F(\mathbf{u}_k) \leq -\eta_{\text{eff}} \mathbf{E}_k [\langle \nabla F(\mathbf{u}_k), \mathcal{G}_k \rangle] + \frac{\eta_{\text{eff}}^2 L}{2} \mathbf{E}_k \left[\|\mathcal{G}_k\|^2 \right], \tag{159}$$

where $\mathcal{G}_k = \frac{1}{m}\sum_{i=1}^m g_i(\mathbf{x}_k^{(i)})$. For the first term on RHS,

$$\mathbf{E}_k [\langle \nabla F(\mathbf{u}_k), \mathcal{G}_k \rangle] = \left\langle \nabla F(\mathbf{u}_k), \frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_k^{(i)}) \right\rangle \tag{160}$$

$$= \frac{1}{2} \left[\|\nabla F(\mathbf{u}_k)\|^2 + \|\mathcal{H}_k\|^2 - \left\| \nabla F(\mathbf{u}_k) - \frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_k^{(i)}) \right\|^2 \right] \tag{161}$$

$$= \frac{1}{2} \left[\|\nabla F(\mathbf{u}_k)\|^2 + \|\mathcal{H}_k\|^2 - \left\| \frac{1}{m}\sum_{i=1}^m \left(\nabla F_i(\mathbf{u}_k) - \nabla F_i(\mathbf{x}_k^{(i)}) \right) \right\|^2 \right], \tag{162}$$

where $\mathcal{H}_k = \frac{1}{m}\sum_{i=1}^m \nabla F_i(\mathbf{x}_k^{(i)})$ and the last equality comes from the definition of the global objective function $F(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m F_i(\mathbf{x})$. Then, using Jensen's inequality, one can obtain

$$-\mathbf{E}_k\left[\langle \nabla F(\mathbf{u}_k), \mathcal{G}_k\rangle\right] \leq -\frac{1}{2}\left[\|\nabla F(\mathbf{u}_k)\|^2 + \|\mathcal{H}_k\|^2 - \frac{1}{m}\sum_{i=1}^m \left\|\nabla F_i(\mathbf{u}_k) - \nabla F_i(\mathbf{x}_k^{(i)})\right\|^2\right] \tag{163}$$

$$\leq -\frac{1}{2}\left[\|\nabla F(\mathbf{u}_k)\|^2 + \|\mathcal{H}_k\|^2 - \frac{L^2}{m}\sum_{i=1}^m \left\|\mathbf{u}_k - \mathbf{x}_k^{(i)}\right\|^2\right] \tag{164}$$

$$\leq -\frac{1}{2}\left[\|\nabla F(\mathbf{u}_k)\|^2 + \|\mathcal{H}_k\|^2 - \frac{L^2}{m}\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right]. \tag{165}$$

For the second term in (159), one can directly reuse the result in Lemma 5 as follows:

$$\mathbf{E}_k\left[\|\mathcal{G}_k\|^2\right] = \mathbf{E}_k\left[\|\mathcal{G}_k - \mathbf{E}_k[\mathcal{G}_k]\|^2\right] + \|\mathbf{E}_k[\mathcal{G}_k]\|^2 \tag{166}$$

$$= \mathbf{E}_k\left[\|\mathcal{G}_k - \mathcal{H}_k\|^2\right] + \|\mathcal{H}_k\|^2 \tag{167}$$

$$\leq \frac{\sigma^2}{m} + \|\mathcal{H}_k\|^2. \tag{168}$$

Plugging (165) and (168) back into (159), we have

$$\mathbf{E}_k\left[F(\mathbf{u}_{k+1})\right] - F(\mathbf{u}_k) \leq -\frac{\eta_{\text{eff}}}{2}\|\nabla F(\mathbf{u}_k)\|^2 - \frac{\eta_{\text{eff}}}{2}(1-\eta_{\text{eff}}L)\|\mathcal{H}_k\|^2 + \frac{\eta_{\text{eff}}^2 L\sigma^2}{m} + \frac{\eta_{\text{eff}}L^2}{m}\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2. \tag{169}$$

When $\eta_{\text{eff}}L \leq 1$,

$$\mathbf{E}_k\left[F(\mathbf{u}_{k+1})\right] - F(\mathbf{u}_k) \leq -\frac{\eta_{\text{eff}}}{2}\|\nabla F(\mathbf{u}_k)\|^2 + \frac{\eta_{\text{eff}}^2 L\sigma^2}{m} + \frac{\eta_{\text{eff}}L^2}{m}\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2. \tag{170}$$

Then, taking the total expectation and summing over all iterates,

$$\sum_{k=1}^K \mathbb{E}\left[F(\mathbf{u}_{k+1}) - F(\mathbf{u}_k)\right] \leq -\frac{\eta_{\text{eff}}}{2}\sum_{k=1}^K \mathbb{E}\|\nabla F(\mathbf{u}_k)\|^2 + \frac{\eta_{\text{eff}}^2 L\sigma^2}{m} + \frac{\eta_{\text{eff}}L^2}{m}\sum_{k=1}^K \mathbb{E}\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2. \tag{171}$$

After minor rearranging and taking the average over all iterates, we have

$$\frac{1}{K}\sum_{k=1}^K \mathbb{E}\left[\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2[F(\mathbf{u}_1) - F_{\text{inf}}]}{\eta_{\text{eff}}K} + \frac{\eta_{\text{eff}}L\sigma^2}{m} + \frac{L^2}{mK}\sum_{k=1}^K \mathbb{E}\left[\left\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\right\|_{\mathrm{F}}^2\right]. \tag{172}$$

The above bound is exactly the same as the i.i.d. case. Then, we're going to bound the last term: the discrepancies among local models. For this part of proof, we can directly reuse the result in Appendix D.2, since we do not use the identity distributed assumption there.

The only change needed to be made is the definition of matrix $\nabla F(\mathbf{X}_k)$. Now, we define it as follows:

$$\nabla F(\mathbf{X}_k) = [\nabla F_1(\mathbf{x}_k^{(1)}), \nabla F_2(\mathbf{x}_k^{(2)}), \ldots, \nabla F_m(\mathbf{x}_k^{(m)}), \mathbf{0}, \ldots, \mathbf{0}]. \tag{173}$$

Then, it directly follows that

$$\frac{1}{Km}\sum_{k=1}^{K}\mathbb{E}\left[\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\mathrm{F}}^2\right] \leq \eta^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right) + \frac{\eta^2\tau^2}{1-\zeta}\left(\frac{2\zeta^2}{1+\zeta} + \frac{2\zeta}{1-\zeta} + \frac{\tau-1}{\tau}\right)\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbb{E}\,\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2}{m}. \tag{174}$$

For the ease of writing, we define $C_1 = \frac{2\zeta^2}{1+\zeta} + \frac{2\zeta}{1-\zeta} + \frac{\tau-1}{\tau}$. For the last term in (174), we have

$$\|\nabla F(\mathbf{X}_k)\|_{\mathrm{F}}^2 = \sum_{i=1}^{m}\left\|\nabla F_i(\mathbf{x}_k^{(i)})\right\|^2 \tag{175}$$

$$\leq 3\sum_{i=1}^{m}\left[\left\|\nabla F_i(\mathbf{x}_k^{(i)}) - \nabla F_i(\mathbf{u}_k)\right\|^2 + \|\nabla F_i(\mathbf{u}_k) - \nabla F(\mathbf{u}_k)\|^2 + \|\nabla F(\mathbf{u}_k)\|^2\right] \tag{176}$$

$$\leq 3L^2\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\mathrm{F}}^2 + 3m\kappa^2 + 3m\|\nabla F(\mathbf{u}_k)\|^2. \tag{177}$$

where the last inequality (177) comes from Assumption 4. Plugging (177) into (174),

$$\frac{L^2}{Km}\sum_{k=1}^{K}\mathbb{E}\left[\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\mathrm{F}}^2\right] \leq \eta^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right) + \frac{3\eta^2L^2\tau^2C_1}{1-\zeta}\left[\frac{L^2}{Km}\sum_{k=1}^{K}\mathbb{E}\left[\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\mathrm{F}}^2\right] + \kappa^2 + \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla F(\mathbf{u}_k)\|^2\right]\right]. \tag{178}$$

After minor rearranging, we get

$$(1-C_2)\frac{L^2}{Km}\sum_{k=1}^{K}\mathbb{E}\left[\|\mathbf{X}_k(\mathbf{I}-\mathbf{J})\|_{\mathrm{F}}^2\right] \leq \eta^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right) + C_2\kappa^2 + \frac{C_2}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla F(\mathbf{u}_k)\|^2\right] \tag{179}$$

where $C_2 = \frac{3\eta^2L^2\tau^2C_1}{1-\zeta}$. Then, plugging (179) back into (172),

$$\frac{1-2C_2}{1-C_2}\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2[F(\mathbf{u}_1)-F_{\mathrm{inf}}]}{\eta_{\mathrm{eff}}K} + \frac{\eta_{\mathrm{eff}}L\sigma^2}{m} + \frac{\eta^2L^2\sigma^2}{1-C_2}\left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right) + \frac{C_2\kappa^2}{1-C_2}. \tag{180}$$

That is,

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\left\|\nabla F(\mathbf{u}_k)\right\|^2\right] \leq \left[\frac{2[F(\mathbf{u}_1)-F_{\text{inf}}]}{\eta_{\text{eff}}K}+\frac{\eta_{\text{eff}}L\sigma^2}{m}\right]\frac{1-C_2}{1-2C_2}+ \left[\eta^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right)+C_2\kappa^2\right]\frac{1}{1-2C_2} \tag{181}$$

$$\leq 2\left[\frac{2[F(\mathbf{u}_1)-F_{\text{inf}}]}{\eta_{\text{eff}}K}+\frac{\eta_{\text{eff}}L\sigma^2}{m}\right]+ 2\left[\eta^2L^2\sigma^2\left(\frac{1+\zeta^2}{1-\zeta^2}\tau-1\right)+C_2\kappa^2\right] \tag{182}$$

where the last inequality comes from the fact: $C = 2C_2 \leq 1/2$. By setting $\eta_{\text{eff}} = \frac{1}{L}\sqrt{\frac{m}{K}}$, we can get another version of Corollary 1.

Appendix G. Proof of Lemma 1 and Theorem 3: Best Choice of α in EASGD

Recall that in EASGD, $\zeta = \max\{|1-\alpha|, |1-(m+1)\alpha|\}$. It is straightforward to show that

$$\zeta = \begin{cases}(m+1)\alpha-1, & \frac{2}{m+2}<\alpha\leq\frac{2}{m+1}\\ 1-\alpha, & 0\leq\alpha\leq\frac{2}{m+2}\end{cases}. \tag{183}$$

When $\alpha = \frac{2}{m+2}$, one can get the minimal value of ζ, which equals to $1-\alpha = (m+1)\alpha-1 = \frac{m}{m+2}$. Then, substituting $\zeta = \frac{m}{m+2}, \tau = 1, v = 0$ into Theorem 1, we complete the proof of Theorem 3.

Appendix H. Proof of Theorem 4: Generalized Elastic Averaging

Theorem 4 is built upon a known result about the eigenvalues of block matrices.

Lemma 9 (Fiedler (1974)) *Let $\mathbf{A}$ be a symmetric $m\times m$ matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$, let $\mathbf{u}, \|\mathbf{u}\| = 1$, be a unit eigenvector corresponding to λ_1; let $\mathbf{B}$ be a symmetric $n\times n$ matrix with eigenvalues $\beta_1, \beta_2, \ldots, \beta_n$, let $\mathbf{v}, \|\mathbf{v}\| = 1$, be a unit eigenvector corresponding to β_1. Then for any ρ, the matrix*

$$\mathbf{C} = \begin{bmatrix}\mathbf{A} & \rho\mathbf{u}\mathbf{v}^\top\\ \rho\mathbf{v}\mathbf{u}^\top & \mathbf{B}\end{bmatrix} \tag{184}$$

has eigenvalues $\lambda_2, \ldots, \lambda_m, \beta_2, \beta_n, \gamma_1, \gamma_2$, where γ_1, γ_2 are eigenvalues of the matrix:

$$\hat{\mathbf{C}} = \begin{bmatrix}\lambda_1 & \rho\\ \rho & \beta_1\end{bmatrix}. \tag{185}$$

In our case, recall the definition of $\mathbf{W}'$:

$$\mathbf{W}' = \begin{bmatrix}(1-\alpha)\mathbf{W} & \alpha\mathbf{1}\\ \alpha\mathbf{1}^\top & 1-m\alpha\end{bmatrix}. \tag{186}$$

In order to apply Lemma 9, let us set $\mathbf{A} = (1-\alpha)\mathbf{W}$. Accordingly, the eigenvalues of $\mathbf{A}$ are $1-\alpha, (1-\alpha)\lambda_2, \ldots, (1-\alpha)\lambda_m$. The eigenvector corresponding to $1-\alpha$ is $\frac{\mathbf{1}}{\sqrt{m}}$. Moreover, set $B = 1 - m\alpha$. Then, it has only one eigenvalue $1-m\alpha$ and the corresponding eigenvector is scalar 1. Substituting $\mathbf{A}, B$ into $\mathbf{W}'$, we have

$$\mathbf{W}' = \begin{bmatrix} \mathbf{A} & \alpha\sqrt{m} \cdot \frac{\mathbf{1}}{\sqrt{m}} \\ \alpha\sqrt{m} \cdot \frac{\mathbf{1}^\top}{\sqrt{m}} & B \end{bmatrix}. \tag{187}$$

According to Lemma 9, the eigenvalues of $\mathbf{W}'$ are $(1-\alpha)\lambda_2, \ldots, (1-\alpha)\lambda_m, \gamma_1, \gamma_2$, where γ_1, γ_2 are eigenvalues of the matrix:

$$\hat{\mathbf{C}} = \begin{bmatrix} 1-\alpha & \alpha\sqrt{m} \\ \alpha\sqrt{m} & 1-m\alpha \end{bmatrix}. \tag{188}$$

For matrix $\hat{\mathbf{C}}$ we have

$$\gamma^2 - [2-(m+1)\alpha]\,\gamma + 1 - (m+1)\alpha = 0 \tag{189}$$

The above equation yields $\gamma_1 = 1, \gamma_2 = 1-(m+1)\alpha$.

Finally, we have $\zeta' = \max\{|(1-\alpha)\lambda_2|, |(1-\alpha)\lambda_m|, |1-(m+1)\alpha|\} = \max\{(1-\alpha)\zeta, |1-(m+1)\alpha|\}$. As a consequence, when $(1-\alpha)\zeta = (m+1)\alpha - 1$, i.e., $\alpha = \frac{1+\zeta}{m+1+\zeta}$, the value of ζ' is minimized.

References

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

Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. *arXiv preprint arXiv:1506.01900*, 2015.

Yossi Arjevani, Yair Carmon, John C Duchi, Dylan J Foster, Nathan Srebro, and Blake Woodworth. Lower bounds for non-convex stochastic optimization. *arXiv preprint arXiv:1912.02365*, 2019.

Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Michael Rabbat. Stochastic gradient push for distributed deep learning. *arXiv preprint arXiv:1811.10792*, 2018.

Debraj Basu, Deepesh Data, Can Karakus, and Suhas Diggavi. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations. In *Advances in Neural Information Processing Systems*, pages 14668–14679, 2019.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: compressed optimisation for non-convex problems. *arXiv preprint arXiv:1802.04434*, 2018.

Michael Blot, David Picard, Matthieu Cord, and Nicolas Thome. Gossip training for deep learning. *arXiv preprint arXiv:1611.09726*, 2016.

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

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, January 2011.

Pratik Chaudhari, Carlo Baldassi, Riccardo Zecchina, Stefano Soatto, Ameet Talwalkar, and Adam Oberman. Parle: parallelizing stochastic gradient descent. *arXiv preprint arXiv:1707.00424*, 2017.

Henggang Cui, James Cipar, Qirong Ho, Jin Kyu Kim, Seunghak Lee, Abhimanu Kumar, Jinliang Wei, Wei Dai, Gregory R Ganger, Phillip B Gibbons, et al. Exploiting bounded staleness to speed up big data analytics. In *2014 USENIX Annual Technical Conference (USENIX ATC 14)*, pages 37–48, 2014.

Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in neural information processing systems*, pages 1223–1231, 2012.

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

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3):592–606, 2012.

Sanghamitra Dutta, Gauri Joshi, Soumyadip Ghosh, Parijat Dube, and Priya Nagpurkar. Slow and stale gradients can win the race: Error-runtime trade-offs in distributed SGD. *arXiv preprint arXiv:1803.01113*, 2018.

Miroslav Fiedler. Eigenvalues of nonnegative symmetric matrices. *Linear Algebra and its Applications*, 9:119–142, 1974.

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

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Suyog Gupta, Wei Zhang, and Fei Wang. Model accuracy and runtime tradeoff in distributed deep learning: A systematic study. In *IEEE 16th International Conference on Data Mining (ICDM)*, pages 171–180. IEEE, 2016.

Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

Farzin Haddadpour, Mohammad Mahdi Kamani, Mehrdad Mahdavi, and Viveck Cadambe. Trading redundancy for communication: Speeding up distributed sgd for non-convex optimization. In *International Conference on Machine Learning*, pages 2545–2554, 2019.

Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *arXiv preprint arXiv:2002.05516*, 2020.

Roger A Horn and Charles R Johnson. *Matrix analysis*, chapter 5. Cambridge university press, 1990.

Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Advances in Neural Information Processing Systems*, pages 2525–2536, 2018.

Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*, pages 5906–5916, 2017.

Peter H Jin, Qiaochu Yuan, Forrest Iandola, and Kurt Keutzer. How to scale distributed deep learning? *arXiv preprint arXiv:1611.04581*, 2016.

W Kahan. A tutorial overview of vector and matrix norms. *University of California, Berkeley, CA, USA, Lecture notes*, page 19, 2013.

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

A Khaled, K Mishchenko, and P Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, 2020.

Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR, 2020.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *OSDI*, volume 14, pages 583–598, 2014.

Xiang Li, Wenhao Yang, Shusen Wang, and Zhihua Zhang. Communication-efficient local decentralized sgd methods. *arXiv preprint arXiv:1910.09126*, 2019.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of FedAvg on non-IID data. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=HJxNAnVtDS`.

Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 2737–2745, 2015.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5336–5346, 2017a.

Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. *arXiv preprint arXiv:1710.06952*, 2017b.

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

Tao Lin, Sebastian U Stich, and Martin Jaggi. Don't use large mini-batches, use local SGD. *arXiv preprint arXiv:1808.07217*, 2018.

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

Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 456–464. Association for Computational Linguistics, 2010.

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

Ioannis Mitliagkas, Ce Zhang, Stefan Hadjis, and Christopher Ré. Asynchrony begets momentum, with an application to deep learning. In *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 997–1004. IEEE, 2016.

Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In *ICML*, 2019.

Philipp Moritz, Robert Nishihara, Ion Stoica, and Michael I Jordan. SparkNet: Training deep networks in spark. *arXiv preprint arXiv:1511.06051*, 2015.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

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

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, January 2014.

Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur. Parallel training of dnns with natural gradient and parameter averaging. *arXiv preprint arXiv:1410.7455*, 2014.

Shi Pu, Wei Shi, Jinming Xu, and Angelia Nedic. Push-pull gradient methods for distributed optimization in networks. *IEEE Transactions on Automatic Control*, 2020.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in neural information processing systems*, pages 693–701, 2011.

Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. *arXiv preprint arXiv:1909.13014*, 2019.

Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Felix Sattler, Simon Wiedemann, Klaus-Robert Müller, and Wojciech Samek. Sparse binary compression: Towards distributed deep learning with minimal communication. *arXiv preprint arXiv:1805.08768*, 2018.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Laurent Massoulié, and Yin Tat Lee. Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2740–2749, 2018.

Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Virginia Smith, Simone Forte, Ma Chenxin, Martin Takáč, Michael I Jordan, and Martin Jaggi. Cocoa: A general framework for communication-efficient distributed optimization. *Journal of Machine Learning Research*, 18:230, 2018.

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

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

Hang Su and Haoyu Chen. Experiments on parallel training of deep neural network using model averaging. *arXiv preprint arXiv:1507.01239*, 2015.

John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control*, 31(9):803–812, 1986.

Haozhao Wang, Song Guo, and Ruixuan Li. Osp: Overlapping computation and communication in parameter server for fast machine learning. In *Proceedings of the 48th International Conference on Parallel Processing*, pages 1–10, 2019a.

Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, pages 9850–9861, 2018.

Jianyu Wang and Gauri Joshi. Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD. *CoRR*, abs/1810.08313, 2018.

Jianyu Wang, Anit Kumar Sahu, Zhouyi Yang, Gauri Joshi, and Soummya Kar. Matcha: Speeding up decentralized sgd via matching decomposition sampling. *arXiv preprint arXiv:1905.09435*, 2019b.

Jianyu Wang, Hao Liang, and Gauri Joshi. Overlap local-SGD: An algorithmic approach to hide communication delays in distributed SGD. *arXiv preprint arXiv:2002.09539*, 2020a.

Jianyu Wang, Vinayak Tantia, Nicolas Ballas, and Michael Rabbat. SlowMo: Improving communication-efficient distributed SGD with slow momentum. In *International Conference on Learning Representations*, 2020b.

Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *arXiv preprint arXiv:1710.09854*, 2017.

Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. TernGrad: Ternary gradients to reduce communication in distributed deep learning. *arXiv preprint arXiv:1705.07878*, 2017.

Blake Woodworth, Kumar Kshitij Patel, and Nathan Srebro. Minibatch vs local sgd for heterogeneous distributed learning. *arXiv preprint arXiv:2006.04735*, 2020a.

Blake Woodworth, Kumar Kshitij Patel, Sebastian Stich, Zhen Dai, Brian Bullins, Brendan Mcmahan, Ohad Shamir, and Nathan Srebro. Is local sgd better than minibatch sgd? In *International Conference on Machine Learning*, pages 10334–10343. PMLR, 2020b.

Ran Xin, Soummya Kar, and Usman A Khan. Gradient tracking and variance reduction for decentralized optimization and machine learning. *arXiv preprint arXiv:2002.05373*, 2020.

Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted SGD for non-convex optimization with faster convergence and less communication. *arXiv preprint arXiv:1807.06629*, 2018.

Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum SGD for distributed non-convex optimization. In *International Conference on Machine Learning*, 2019.

Kun Yuan, Qing Ling, and Wotao Yin. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization*, 26(3):1835–1854, 2016.

Jinshan Zeng and Wotao Yin. On nonconvex decentralized gradient descent. *arXiv preprint arXiv:1608.05766*, 2016.

Jian Zhang, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Parallel SGD: When does averaging help? *arXiv preprint arXiv:1606.07365*, 2016.

Sixin Zhang, Anna E Choromanska, and Yann LeCun. Deep learning with elastic averaging SGD. In *Advances in Neural Information Processing Systems*, pages 685–693, 2015.

Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *arXiv preprint arXiv:1708.01012*, 2017.