Computer Communications 203 (2023) 163–179

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

# Robust decentralized stochastic gradient descent over unstable networks

Yanwei Zheng [a,1], Liangxu Zhang [a,2], Shuzhen Chen [a,*,1], Xiao Zhang [a,*,1], Zhipeng Cai [b,3], Xiuzhen Cheng [a,4]

[a] School of Computer Science and Technology, Shandong University, Qingdao, PR China
[b] Department of Computing Science, Georgia State University, Atlanta, GA 30303, USA

## ARTICLE INFO

*Keywords:*
Decentralized learning
Stochastic gradient descent
Unstable networks

## ABSTRACT

Decentralized learning is essential for large-scale deep learning due to its great advantage in breaking the communication bottleneck. Most decentralized learning algorithms focus on reducing the communication overhead without taking into account the possibility of a shaky network connection, and existing analyses over unstable networks have various limitations such as centralized settings, strong unrealistic assumptions, etc. Hence, in this work, we study a non-convex optimization problem over unstable networks that fully consider unstable factors including unstable network connections, communication and artificially injected noise. Specifically, we focus on the most commonly used Stochastic Gradient Descent (SGD) algorithm in a mild decentralized setting and propose a robust algorithm to handle unstable networks. It is shown that our algorithm can attain a convergence rate which has the same order as decentralized algorithms over stable networks, and achieves linear speedup comparing with centralized ones. Moreover, the proposed algorithm also applies to the general case that the data are not independently and identically distributed. Extensive experiments on image classification demonstrate that the practical performance of our algorithm is comparable with the state-of-art decentralized algorithms in stable networks with only a little accuracy loss.

## 1. Introduction

Distributed machine learning [1–3], especially for large-scale deep learning tasks, has attracted a lot of attention both academically and industrially. A typical distributed learning system is the *Parameter Server* (PS) [4], where a server maintains and aggregates a global model for all client workers. By pulling the global model from the server, client workers compute the gradients or model updates and push them to the server. This architecture needs the server to communicate with all clients and transfer the whole model. Therefore, PS is not robust when the server fails or communication is constrained [5]. Different from using a central server, another architecture is the *shared memory* where all workers independently compute the local gradients and average the global model by a shared memory [6]. Although this line of work avoids the problem of server failure, it still has a high communication cost. Theoretical research shows that decentralized algorithms can solve the above problems and have the same utility as centralized algorithms [7–9]. Decentralized algorithms reduce communication complexity by calculating an approximate average gradient between a set of workers during aggregation. In each optimization iteration, workers just make a model aggregation with neighbors according to the communication topology rather than executing a global average. This would inject extra noise into the average gradients so there is a trade-off between training accuracy and communication overhead for decentralized methods.

One of the problems often encountered in large-scale decentralized systems is the underlying unreliability of local devices. Especially in edge computing and federated learning, the devices involved in training are typically edge-side devices or private computers, making it difficult to guarantee stable network connectivity and reliable performance. Most decentralized approaches, in turn, are based on the assumption that the network is stable to ensure that communication is always successful during training. Therefore, when the network connection is unstable, these algorithms that synchronously aggregate the model will block until the network connection is restored. This makes applying typical methods to this faulty scenario directly not feasible. Also, it is tough to present the convergence of the algorithm directly under relaxed assumptions when the network connections are unstable.

[*] Corresponding authors.
*E-mail addresses:* zhengyw@sdu.edu.cn (Y. Zheng), lxzhang@mail.sdu.edu.cn (L. Zhang), szchen@mail.sdu.edu.cn (S. Chen), xiaozhang@sdu.edu.cn (X. Zhang), zcai@gsu.edu (Z. Cai), xzcheng@sdu.edu.cn (X. Cheng).

[1] Member, IEEE.

[2] Student Member, IEEE.

[3] Senior Member, IEEE.

[4] Fellow, IEEE.

https://doi.org/10.1016/j.comcom.2023.02.025
Received 8 November 2022; Received in revised form 11 January 2023; Accepted 24 February 2023
Available online 2 March 2023

**Table 1**
Comparison of related results.

| Algorithms | Unstable network connections | Noise | Convergence rate |
|---|---|---|---|
| DPSGD [7] | ✗ | ✗ | $O(\frac{1}{\sqrt{nK}})$ |
| RPS[a] [17] | ✓ | ✗ | $O(\frac{1}{\sqrt{nK}})$ |
| Choco-SGD[a] [12,13] | ✗ | ✓ | $O(\frac{1}{\sqrt{nK}})$ |
| A(DP)$^2$SGD[a] [16] | ✗ | ✓ | $O(\frac{1}{\sqrt{K}})$ |
| **Our algorithm** | ✓ | ✓ | $\boldsymbol{O}(\frac{1}{\sqrt{nK}})$ |

[a] RPS algorithm considers unstable networks but only focuses on the centralized settings. Choco-SGD only considers compressed noise and A(DP)$^2$SGD only considers differential noise.

Another factor to consider in large-scale decentralized learning is noise. The noise expresses a lower bound on the desired generalization error that any learning algorithm can achieve on the current task, i.e., it portrays the difficulty of the learning problem itself. As a result, analyzing the influence of noise on algorithm convergence can help to develop more resilient algorithms. There are many reasons for noise and the most intuitive is the Gaussian white noise [10] in the communication process, which is very common in the wireless channel. In addition to the inevitable channel noise, there are usually artificially introduced noises in the process of device communication, such as compression noise introduced by gradient compression [11–13] and privacy noise introduced by differential privacy protection [14–16]. Most studies usually investigate the effect of noise only at the application level. And the existing theoretical analysis is usually specific to a particular scenario of the above-mentioned noise sources.

Facing these challenges, it is urgent to explore algorithms that can be well implemented in unstable networks featured by unstable network connections and noise. In this unstable network situation, the technical challenge is that the convergence of the decentralized learning algorithm should be theoretically guaranteed under some mild conditions. There have been some works to study unstable network connections. Consensus optimizations over unreliable networks are studied in [18,19], but their work is under strong assumptions such that the feasible domain is compact, the gradients are bounded and the instability level is bounded. In [17], Yu et al. investigated the case of unreliable network connectivity under a loose assumption but only focused on the centralized settings. For communication noise, the experimental results in [20] show that communication noise in gradients can generalize the model. Theoretical results in [13] show that the decentralized algorithms can still converge in a compression noise environment. Another direction of noise is to ensure data privacy by introducing differential noise. In [16], the convergence of the decentralized algorithm with differential noise is guaranteed. Existing work either only considers unstable network connections or only considers noise. Combining these two issues, we explore the decentralized algorithm over unstable networks and present the theoretical results under loose assumptions. Specifically, we focus on the most commonly used SGD algorithm [7] in a mild decentralized setting. SGD has evolved into a standard and efficient algorithm for solving large-scale distributed deep learning tasks. Compared with other optimization algorithms, SGD has many advantages such as high efficiency, good generalization, the capability of escaping from stagnation, etc. Formally, we aim to solve the following stochastic optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) = \mathbb{E}_{\xi\sim\mathcal{D}} F(x;\xi), \tag{1}$$

where $\mathcal{D}$ is denoted as data distribution and $\xi$ is a random data sample. $x$ denotes the parametric model we intend to train and $F(\cdot)$ denotes the predefined loss function. This formulation encapsulates a variety of well-known learning problems, such as machine learning, federated learning [21], and deep learning. Our main contributions are as follows.

**Main Contributions:**

- In the first part, we consider a non-convex decentralized optimization problem over unstable network connections and propose a decentralized SGD algorithm to accommodate this scenario. Except for some standard assumptions, we only assume a realistic and necessary bound on network instability for analyzing convergence of the proposed learning algorithm. By choosing an appropriate learning rate, our algorithm achieves a convergence rate of $O(\frac{1}{\sqrt{nK}})$, where $n$ denotes the number of workers and $K$ denotes the number of total iterations. Our results are consistent with decentralized Stochastic Gradient Descent (SGD) [7] over reliable network connections. Besides, our theoretical results indicate that our algorithm achieves linear speedup w.r.t. the number of workers. Moreover, our theoretical results also apply to the general case that the data are not independently and identically distributed.
- In the second part, we adapt our algorithm to the scenarios with noise. Specifically, we propose a general noise model covering different noise categories such as channel noise, compressed noise and differential noise. Based on this, we present the convergence analysis and show the influence of noise on our algorithm. Under some mild assumption on noise, our algorithm can attain the same order of convergence rate as that implemented without considering any noise.
- Experimentally, we apply our algorithm to an image classification task, illustrating the convergence of our algorithm is the same as DPSGD [7], the state-of-the-art decentralized SGD algorithm without considering unstable connections and noise, with only a little accuracy loss. The experimental results, on the CIFAR10 dataset, show that the convergence speed of our algorithm is comparable with DPSGD and this is in line with our theoretical analysis. In addition, via unstable network simulations, we observe that the training loss decreases as the network instability level decreases, but the convergence speed is not affected. Surprisingly, our algorithm is robust to noise, especially with a large number of workers and trained on a complex model.

The comparison results for the convergence rate of the relevant algorithms are shown in Table 1.

**Road Map:** This paper is organized as follows. After outlining the relevant work in Section 2, we give the formal problem settings and basic model of the unstable networks in Section 3. In Section 4, we propose our algorithm which is robust to unstable networks and show that our algorithm can achieve the same convergence rate as those in stable networks. In Section 5, we specify a general model of noise and adapt our algorithm to tolerate communication and artificial injected noise. The same convergence rate is shown to be retained. Finally, we report the experimental results in Section 6 and conclude the paper in Section 7.

Highlights

- We study the effect of unstable networks and noise on the convergence of decentralized algorithms.
- Theoretical works demonstrate that our algorithm can achieve the sub-linear convergence of $O(\frac{1}{\sqrt{nK}})$.
- Deep learning experiments verify that our algorithm can achieve the same convergence rate as the optimal algorithm.

## 2. Related work

**Decentralized training** Decentralized methods based on gossip averages can be good solutions to the case of some workers malfunctioning. In such methods, all workers are connected through a peer-to-peer network and each worker aggregates models with only a portion of the workers, i.e., the neighbors in the network. Thus a sparse network topology will significantly reduce the communication load. [7] first gave the theoretical proof showing that decentralized SGD has the same convergence rate with centralized SGD and experimental results show that decentralized SGD outperforms centralized SGD in the bandwidth-constrained case. [9] proposed the SGP algorithm for the situation where the network topology is not symmetric, i.e., the topology matrix is column stochastic. Later studies mostly tended to reduce the communication load by cutting the number of communications or communication compression. [8,22] presented communication-efficient methods which have a comparable convergence rate with centralized methods. [11] proposed the DCD and ECD algorithms based on unbiased compressions such as random quantization [23] and sparsification [24] with the identical convergence rate as the centralized algorithm. In [12,13], the authors investigated a more general approach, called CHOCO-SGD, with arbitrary communication compression in decentralized training. Another direction is to study random network topology [25–28], which implies that the weighted matrix is dynamic or time-varying.

**Distributed training with faulty models** Most of the existing algorithms require a stable network connection, i.e., the communication is guaranteed to be successful. However, some practical applications must be able to handle unstable network connections, and this is particularly relevant in federated learning [29]. There have been many related works to study unstable network connections but they all have some limitations. [30–32] investigated the case of delayed information exchange in distributed training but only focused on centralized scenarios. [17] analyzed another scenario of unstable network connectivity where communication would fail in a fixed probability, but only the centralized scenarios and the AllReduce method were considered. Additionally, there are several concerns with current research on unstable network connections in decentralized systems. [18,19] analyzed the decentralized algorithm with delay gradients and characterized the convergence rate under bounded delay but their work is under strong assumptions such that the feasible domain is compact, the gradients are bounded and instability level is bounded. In addition to the situation that messages may be delayed in reception due to unstable networks, messages can be interrupted by noise during the exchange process. On the one hand, the noise can be artificially injected. [33] show that introducing noise into neural network training could better generalize the training model but they only present experimental verification without theoretical proof. Numerous works [14–16] have provided theoretical results, however, these only apply to a certain noise environment. On the other hand, there inevitably is some noise in the communication channel. [34,35] take into account both unstable network connections and noise, but their work is limited to strongly convex scenarios. Thus we combine these two directions and fully consider a faulty model in which there is both unstable network connectivity and noise during communication under the relaxation assumptions.

## 3. Preliminaries

**Table 2**
Frequently used notations.

| Notations | Descriptions |
|---|---|
| $n$ | The number of workers |
| $d$ | Dimensions of the local model |
| $x_i^k$ | Local model of worker $i$ at iteration $k$ |
| $\hat{x}_i^k$ | The last successfully received model |
| $Q(x_i^k)$ | The perturbed model of worker $i$ at iteration $k$ |
| $\xi_i^k$ | The sample data of worker $i$ at iteration $k$ |
| $F_i(x;\xi)$ | Loss function of worker $i$ |
| $f_i(x)$ | Expectation of loss function of worker $i$ |
| $\gamma$ | Learning rate |
| $K$ | The total iterations |
| $W_k$ | The mixing matrix at iteration $k$ |
| $\nabla f(\cdot)$ | The gradient of the function $f(\cdot)$ |
| $\mathbf{1}_n$ | The full-one vector in $\mathbb{R}^n$ |
| $e_i$ | The $i$th element of the standard basis of $\mathbb{R}^n$ |
| $\lambda_i(\cdot)$ | The $i$th largest eigenvalue of a matrix |
| $\|\cdot\|$ | The vector $\ell_2$ norm or the matrix spectral norm |
| $X_k$ | $[x_1^k, x_2^k, \dots, x_n^k] \in \mathbb{R}^{d\times n}$ |
| $\partial F(X_k, \xi_k)$ | $[\nabla F_1(x_1^k, \xi_1^k), \dots, \nabla F_n(x_n^k, \xi_n^k)] \in \mathbb{R}^{d\times n}$ |
| $\partial f(X_k)$ | $[\nabla f_1(x_1^k), \dots, \nabla f_n(x_n^k)] \in \mathbb{R}^{d\times n}$ |

We consider a general decentralized distributed system consisting of $n$ workers and all workers are connected to cooperatively optimize a non-convex problem through the stochastic gradient descent method. Let graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes the communication network topology, where $\mathcal{V} := \{1, 2, \dots, n\}$ and $\mathcal{E}$ denotes the set of edges in the graph. Let $\mathcal{J}_i = \{j \in \mathcal{V} \mid (i,j) \in \mathcal{E}\}$ denotes the neighbor set of worker $i$ and worker $i$ can only communicate with workers who belong to $\mathcal{J}_i$. In particular, we let $\mathcal{J}_i$ also contain $i$. In the stochastic gradient descent method, worker $i$ sample data $\xi$ from a local data distribution $\mathcal{D}_i$ to optimize the local loss function $F_i(x;\xi)$ of model $x \in \mathbb{R}^n$. Based on these mathematical setups, by distributing the data to all workers we can rewrite Eq. (1) in the following form:

$$\min_{x\in\mathbb{R}^n} f(x) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\mathbb{E}_{\xi\sim D_i} F_i(x;\xi)}_{=:f_i(x)}. \tag{2}$$

Note that we do not assume the data distribution is i.i.d.. This is a more general setup in practical applications.

**Unstable Network Connections** Consider a scenario where the workers who perform calculations during distributed learning are some personal devices or edge devices, which are very easy to go offline or crash. These devices may be disconnected from other devices due to unstable network connections but these devices are highly robust and can reconnect back in time. Given a non-negative constant $\tau_i^k$ indicates that offline worker $i$ has been disconnected for $\tau_i^k$ iterations, in other words, neighbors of worker $i$ have not received information from worker $i$ since iteration $k - \tau_i^k$. Let $\tau_k = \max \tau_i^k$ to denote the worst case at iteration $k$. We make the following assumption which is necessary in our convergence analysis.

**Assumption 1** (*Bounded Instability*). We assume that the time $\tau_i^k$ at which the offline worker has been offline at the $k$th iteration is uniformly bounded, i.e., there exits $\tau > 0$ such that $\tau_k \leq \tau$ for all iteration $k$.

This is a practical assumption because most devices are robust and network protocols usually have a heartbeat mechanism to reconnect after disconnection. Assumption 1 illustrates that the time required to reconnect for any worker after it goes offline is bounded by $\tau$. Thus we use $\tau$ to describe the instability of the entire network.

Throughout this paper, some frequently used notations are summarized in Table 2.

## 4. Robust decentralized SGD

### 4.1. Algorithm

In this section, we present our algorithm **RDSGD**-*Robust Decentralized Stochastic Gradient Descent* to deal with the scenario of unstable network connections. The existing algorithms usually block themselves

```
Algorithm 1: RDSGD algorithm
Input: Initialize $x_i^0$ and $\hat{x}_i^0$, $\forall i \in [n]$ with the same value,
       mixing matrix $W$, learning rate $\gamma$ and number of total
       iterations $K$.
1 for $k = 0, 1, \dots, K-1$(all workers in parallel) do
2     Randomly sample $\xi_i^k$ from local data for worker $i \in [n]$.
3     Compute gradient $\nabla F(x_i^k, \xi_i^k)$.
4     Update model according to $x_i^{k+1} = x_i^k - \gamma \nabla F(x_i^k, \xi_i^k)$.
5     Send $x_i^{k+1}$ and receive models from neighbors.
6     if receive any $x_j^{k+1}$, $j \in \mathcal{J}_i$ then
7         $\hat{x}_j^{k+1} = x_j^{k+1}$.
8     Aggregate model by $x_i^{k+1} = \sum_{j \in \mathcal{J}_i} W_k^{[ij]} \hat{x}_j^{k+1}$.
```

when any worker fails to receive information from its neighbors and the whole algorithm fails to work. To address this restriction, we use additional memory space to deal with the loss of information due to that the network connections are disabled.

In the RDSGD algorithm, in addition to holding a local model, each worker maintains a buffer to store the received models of neighboring workers. At each iteration, each node randomly samples from the local data, computes a local gradient in parallel and updates the model along the negative gradient direction. After that, each worker communicates with its neighbors and exchanges models with each other. However, it is not feasible to use the shared information directly to perform the model aggregation, because the communication may fail due to the instability of the network connection. Thus each worker can use the buffer to record the shared information and update the buffer when communication is successful. Then, each worker uses its model and the information in the buffer to perform model aggregation.

In detail, at iteration $k$, we use $x_i^k$ to denote the local model of worker $i$. And worker $i$ maintains a local buffer $\hat{x}_j^k$ to record the model of neighbor $j$, where $j \in \mathcal{J}_i$. At iteration $k$, RDSGD performs the following steps:

- Gradient calculation: each worker randomly samples data $\xi_i^k$ from local data distribution and calculates the stochastic gradient $\nabla F(x_i^k, \xi_i^k)$.
- Model update: each worker updates the model by a regular SGD step $x_i^{k+1} = x_i^k - \gamma \nabla F(x_i^k, \xi_i^k)$ given the learning rate $\gamma$.
- Model aggregation: each worker sends the model to its neighbors and receives models from all of its neighbors. If worker $i$ successfully receives the model of neighbor $j$, then $\hat{x}_j^{k+1} = x_j^{k+1}$; else $\hat{x}_j^{k+1}$ remains unchanged. After that, each worker performs $x_i^{k+1} = \sum_{j \in \mathcal{J}_i} W_k^{[ij]} \hat{x}_j^{k+1}$, where $W_k$ is a mixing matrix and $W_k^{[ij]}$ is the $i$th row and $j$th column element of $W_k$.

Note that model update and model aggregation can be exchanged, which does not affect our theoretical analysis. The pseudo-code of our algorithm is given by Algorithm 1.

### 4.2. Theoretical results

In this part, we present the main theoretical results of the RDSGD algorithm over unstable networks. Theorem 1 shows our numerical results of convergence rate which maintains the same as standard algorithms. For the sake of proving the theorem, we need to make some settings. First, let

$$X_k = [x_1^k, \dots, x_n^k] \in \mathbb{R}^{d \times n}.$$

According to our model of unstable networks and Assumption 1, when worker $i$ is not offline, $\hat{x}_i^k = x_i^k$, and when worker $i$ is offline, $\hat{x}_i^k$ indicates the last successful message $x_i^{k-\tau_i^k}$. Thus in the worst case, we can obtain $\hat{x}_i^k = x_i^{k-\tau_k}$ and in the matrix form we have

$$\hat{X}_k = X_{k-\tau_k} = [\hat{x}_1^k, \dots, \hat{x}_n^k] \in \mathbb{R}^{d \times n}.$$

Besides, we define the following matrix forms:

$$\partial F(X_k, \xi_k)$$
$$= [\nabla F_1(x_1^k, \xi_1^k), \dots, \nabla F_n(x_n^k, \xi_n^k)] \in \mathbb{R}^{d \times n},$$

$$\partial f(X_k) = [\nabla f_1(x_1^k), \dots, \nabla f_n(x_n^k)] \in \mathbb{R}^{d \times n}.$$

Returning to our algorithm, in line 8, the aggregation process uses a model with delay information, and this delay model will be used in the next iteration to calculate the gradient. Consequently, we can simplify the problem to the fact that the gradient is calculated utilizing a model containing delay information at each iteration. Note that the update formula of the algorithm in matrix form can be written as

$$X_{k+1} = X_k W_k - \gamma \partial F(\hat{X}_k, \xi_k), \tag{3}$$

Next, we make some necessary assumptions:

**Assumption 2.** Throughout this paper, we make the following commonly used assumptions:

1. **Lipschitzian gradients (L-smooth):** All local functions $f_i(\cdot)$ and their gradients $\nabla f_i(\cdot)$ are L-Lipschitz continuous, i.e.,
$$\|f_i(x) - f_i(y)\|_2 \le L\|x - y\|,$$
$$\|\nabla f_i(x) - \nabla f_i(y)\|_2 \le L\|x - y\|,$$
for all $x, y \in \mathbb{R}^d$ and $i \in \mathcal{V}$, where $L$ is the Lipschitz constant.
2. **Bounded variance:** Assume the variance of the stochastic gradient is bounded for any $x$ on each worker.
$$\mathbb{E}_{\xi \sim D_i} \|\nabla F_i(x;\xi) - \nabla f_i(x)\|^2 \le \sigma^2, \forall i, \forall x,$$
$$\frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(x;\xi) - \nabla f(x)\|^2 \le \varsigma^2, \forall i, \forall x.$$
Note that if data is i.i.d. then $\varsigma = 0$.
3. **Spectral gap:** $W_k$ is a doubly stochastic matrix ($W_k \mathbf{1} = \mathbf{1}, \mathbf{1}^\top W_k = \mathbf{1}^\top$) and we define $\rho := |\lambda_2(\mathbb{E}[W_k^\top W_k])| \in [0, 1)$.
4. **Start from 0:** We assume all workers' models start at 0, in other words, $X_0 = [0, \dots, 0]$ for simplifying the proof w.l.o.g.

Before presenting Theorem 1, we define some variables:

$$U_1 = \left(1 - \frac{72}{(1-\sqrt{\rho})^2}\gamma^2 n L^2\right),$$

$$U_2 = \frac{\gamma - \gamma^2 L}{2} - \tau\gamma^3 L^2 - \frac{12 n \gamma^3 L^2}{(1-\sqrt{\rho})^2 U_1}.$$

**Theorem 1.** *Under Assumptions 1 and 2, if $U_1 > 0$ and $U_2 \ge 0$ are satisfied, then we have the following convergence rate*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\left\|\nabla f\left(\frac{X_k \mathbf{1}_n}{n}\right)\right\|^2 \le \frac{2(f_0 - f^\star)}{\gamma K} + \frac{2\tau^2\gamma^2\sigma^2 L^2}{n}$$
$$+ \frac{\gamma\sigma^2 L}{n} + \frac{4n\gamma^2 L^2(\sigma^2 + 6\varsigma^2)}{U_1}\left(\frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2}\right), \tag{4}$$

*where $f_0 = f(0)$ and $f^\star$ denotes the optimal solution.*

For non-convex functions, the point where the gradient is 0 is the local optimal solution. Theorem 1 guarantees convergence of the algorithm by giving an upper bound on the average gradient of all workers. Specifically, by choosing an appropriate learning rate, we can get the following corollary

**Corollary 1.** *Under Assumptions 1 and 2, by choosing* $\gamma = \frac{1}{L+\sqrt{\sigma^2+6\varsigma^2}\sqrt{\frac{K}{n}}}$, *we have the following convergence rate*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2 \le \frac{2(f_0-f^\star+\tau^2L)L}{K} + \frac{(2f_0-2f^\star+5L)\sqrt{\sigma^2+6\varsigma^2}}{\sqrt{nK}}, \tag{5}$$

*if $K$ is large enough to satisfy*

$$\begin{aligned} K &\ge \frac{4n^5L^2}{\sigma^2+6\varsigma^2}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)^2, \\ K &\ge \frac{n}{\sigma^2+6\varsigma^2}\max\left\{4L^2, (\sqrt{6\tau}-1)^2L^2, \frac{144nL^2}{(1-\sqrt{\rho})^2}\right\}, \end{aligned} \tag{6}$$

*where $f_0$ and $f^\star$ follow the definitions in Theorem 1.*

Corollary 1 indicates a general convergence rate $O(\frac{1}{K}+\frac{1}{\sqrt{nK}})$. We discuss some properties about our theoretical results below.

- **Converge to a ball.** Based on our theoretical results, our final average gradient of all workers is constrained to a ball of a critical point. It is because the learning rate we choose is a constant and this is a general choice for $\gamma$, just as some other theoretical analysis for SGD.
- **Comparing with SGD and DPSGD.** In Corollary 1, if $K$ is sufficiently large, the second term is the dominant term and the convergence rate is $O(\frac{1}{\sqrt{nK}})$. Also, if $\tau = 0$ and $n = 1$ our algorithm is centralized SGD and the convergence rate reduces to $O(\frac{1}{\sqrt{K}})$, which is consistent with the convergence rate of centralized SGD and decentralized SGD [7].
- **Linear Speedup.** When $K$ is large enough, the convergence of our algorithm is $O(\frac{1}{\sqrt{nK}})$. Note that RDSGD to achieve $\epsilon$ accuracy requires $K$ to satisfy $K \ge \frac{1}{n\epsilon^2}$, which indicates that the convergence efficiency increases at a linear rate with respect to the number of workers.

### 4.3. Analysis of Algorithm 1

In this part, we give theoretical support for Theorem 1 and Corollary 1 in detail. Before we present the proof of our main results, we give some necessary notions and lemmas. We let

$$M_k := \frac{1}{n}\sum_{i=1}^n\left\|\frac{X_k\mathbf{1}_n}{n} - X_ke_i\right\|^2, \quad \forall\, k > 0, \tag{7}$$

and we have $\hat{M}_k := M_{k-\tau_k}$. Note that $M_k$ is the average consistency error which declares the gap between the global average model and local model. To obtain the bound of average gradient of all workers, we need to bound the $M_k$ first.

**Lemma 1.** *Under Assumption 2 we have*

$$\left\|\frac{\mathbf{1}_n}{n} - W^ke_i\right\|^2 \le \rho^k, \quad \forall\, i \in \{1,2,\dots,n\}, k \in \mathbb{N}.$$

**Proof.** Let $W^\infty = \lim_{k\to\infty}W^k$, and from Assumption 2 we have $\frac{\mathbf{1}_n}{n} = W^\infty e_i$. Thus

$$\begin{aligned} \left\|\frac{\mathbf{1}_n}{n} - W^ke_i\right\|^2 &= \|(W^\infty - W^k)e_i\|^2 \\ &\le \|W^\infty - W^k\|^2\|e_i\|^2 \\ &= \|W^\infty - W^k\|^2 \\ &\le \rho^k. \quad \square \end{aligned}$$

Lemma 1 is a common property for doubly stochastic matrix. To give an upper bound on the average consistency error, we first establish the relationship between the local loss function and $M_k$.

**Lemma 2.** *Under Assumption 2 and $\forall j \ge 0$ we have*

$$\mathbb{E}\|\partial f(\hat{X}_j)\|^2 \le 12nL^2\mathbb{E}\hat{M}_j + 6n\varsigma^2 + 2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2.$$

**Proof.** We start from the norm of $\partial f(\hat{X}_j)$. Using the definition of $L_2$ norm of matrix and triangular inequality, we have

$$\begin{aligned} &\mathbb{E}\|\partial f(\hat{X}_j)\|^2 \\ &= \sum_{i=1}^n\mathbb{E}\|\nabla f_i(\hat{x}_i^j)\|^2 \\ &= \sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\hat{x}_i^j) - \frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n} + \frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \\ &\le 2\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\hat{x}_i^j) - \frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \\ &\quad + 2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2. \end{aligned} \tag{8}$$

To bound consistency error, we introduce two variables $\nabla f_i\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right)$ and $\frac{\partial f\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right)\mathbf{1}_n}{n}$, and then use the L-smooth property to get the form of $M_k$. For the first term of (8), we have

$$\begin{aligned} &\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\hat{x}_i^j) - \frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \\ &\le 3\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i(\hat{x}_i^j) - \nabla f_i\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right)\right\|^2 \\ &\quad + 3\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right) - \frac{\partial f\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right)\mathbf{1}_n}{n}\right\|^2 \\ &\quad + 3\sum_{i=1}^n\mathbb{E}\left\|\frac{\partial f\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right)\mathbf{1}_n}{n} - \frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \\ &\le 3nL^2\mathbb{E}\hat{M}_j + 3\sum_{i=1}^n\mathbb{E}\left\|\nabla f_i\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right) - \nabla f\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right)\right\|^2 \\ &\quad + 3n\mathbb{E}\left\|\frac{\sum_{i=1}^n\left(\nabla f_i\left(\frac{\hat{X}_j\mathbf{1}_n}{n}\right) - \nabla f_i(\hat{x}_i^j)\right)}{n}\right\|^2 \\ &\le 6nL^2\mathbb{E}\hat{M}_j + 3n\varsigma^2, \end{aligned}$$

where the last term of last step comes from L-smooth.

Combining the above inequalities, we derive Lemma 2:

$$\mathbb{E}\|\partial f(\hat{X}_j)\|^2 \le 12nL^2\mathbb{E}\hat{M}_j + 6n\varsigma^2 + 2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2. \quad \square$$

Lemma 2 is an important formula to prove Theorem 1, which gives the relation between the gradient with delay and the average consistency error. Using Lemma 2 we can get the following Lemmas.

**Lemma 3.** *According to Assumption 2, Lemmas 1 and 2 and for any $k > 0$, we have*

$$\begin{aligned}
&\mathbb{E}\left\|\frac{X_{k+1}\mathbf{1}_n}{n} - X_{k+1}e_i\right\|^2 \\
&\le 2n\gamma^2(\sigma^2 + 6\varsigma^2)\left(\frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2}\right) \\
&\quad + 4n\gamma^2\sum_{j=0}^{k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\left(\rho^{k-j} + \frac{2\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\right) \\
&\quad + 24\gamma^2 nL^2\sum_{j=0}^{k}\mathbb{E}\hat{M}_j\left(\rho^{k-j} + \frac{2\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\right).
\end{aligned}$$

**Proof.** For convenience, we let $\Delta(\hat{X}_j) = \partial F(\hat{X}_j;\xi_j) - \partial f(\hat{X}_j)$. According to the updating formula (3) and Assumption 2, we have

$$\begin{aligned}
&\mathbb{E}\left\|\frac{X_{k+1}\mathbf{1}_n}{n} - X_{k+1}e_i\right\|^2 \\
&= \mathbb{E}\left\|\begin{array}{l}\frac{X_k\mathbf{1}_n - \gamma\partial F(\hat{X}_k;\xi_k)\mathbf{1}_n}{n} - \\ (X_kW_ke_i - \gamma\partial F(\hat{X}_k;\xi_k)e_i)\end{array}\right\|^2 \\
&= \mathbb{E}\left\|\begin{array}{l}\frac{X_0\mathbf{1}_n - \sum_{j=0}^{k}\gamma\partial F(\hat{X}_j;\xi_j)\mathbf{1}_n}{n} - X_0\prod_{j=0}^{k}W_je_i \\ + \sum_{j=0}^{k}\gamma\partial F(\hat{X}_j;\xi_j)\prod_{q=j+1}^{k}W_qe_i\end{array}\right\|^2 \\
&= \gamma^2\mathbb{E}\left\|\sum_{j=0}^{k}\partial F(\hat{X}_j;\xi_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2 \qquad (9) \\
&\le 2\gamma^2\underbrace{\mathbb{E}\left\|\sum_{j=0}^{k}\Delta(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2}_{T_1} \\
&\quad + 2\gamma^2\underbrace{\mathbb{E}\left\|\sum_{j=0}^{k}\partial f(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2}_{T_2}
\end{aligned}$$

The form of $T_1$ is the square of the $L_2$ norm of the sum of the $k+1$ terms and each term is the product of two factors. Therefore, we can expand $T_1$ with sum square formula to take the sum to the outside of the norm squared. Next, for the square of the $L_2$ norm of the product of two factors, we can use Holder's inequality to get product of squares of the $L_2$ norm of two factors.

$$\begin{aligned}
T_1 &= \mathbb{E}\left\|\sum_{j=0}^{k}\Delta(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2 \\
&= \underbrace{\sum_{j=0}^{k}\mathbb{E}\left\|\Delta(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2}_{T_3} \\
&\quad + \underbrace{2\sum_{j\neq j'}\mathbb{E}\left\langle\begin{array}{l}\Delta(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right), \\ \Delta(\hat{X}'_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j'+1}^{k}W_qe_i\right)\end{array}\right\rangle}_{T_4}
\end{aligned}$$

For $T_3$, using Holder's inequality and from Lemma 1, we have:

$$\begin{aligned}
T_3 &= \sum_{j=0}^{k}\mathbb{E}\left\|(\Delta(\hat{X}_j))\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2 \\
&\le \sum_{j=0}^{k}\mathbb{E}\left\|\Delta(\hat{X}_j)\right\|^2\left\|\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right\|^2 \\
&\le \sum_{j=0}^{k}\mathbb{E}\left\|\Delta(\hat{X}_j)\right\|_F^2\left\|\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right\|^2 \\
&\le n\sigma^2\sum_{j=0}^{k}\rho^{(k-j)} \le \frac{n\sigma^2}{1-\rho}
\end{aligned}$$

For $T_4$, using Cauchy–Schwarz's inequality:

$$\begin{aligned}
T_4 &= \sum_{j\neq j'}\mathbb{E}\left\langle\begin{array}{l}\Delta(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right), \\ \Delta(\hat{X}_{j'})\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j'+1}^{k}W_qe_i\right)\end{array}\right\rangle \\
&\le \sum_{j\neq j'}\mathbb{E}\left(\begin{array}{l}\left\|\Delta(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\| \\ \cdot\left\|\Delta(\hat{X}_{j'})\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right) \\
&\le \sum_{j\neq j'}\mathbb{E}\left(\begin{array}{l}\left\|\Delta(\hat{X}_j)\right\|\left\|\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\| \\ \cdot\left\|\Delta(\hat{X}_{j'})\right\|\left\|\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right) \\
&\le \mathbb{E}\sum_{j\neq j'}\left(\begin{array}{l}n\sigma^2\left\|\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\| \\ \cdot\left\|\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right) \\
&\le \mathbb{E}\sum_{j\neq j'}n\sigma^2\rho^{k-\frac{j+j'}{2}},
\end{aligned}$$

where the last second inequality comes from the assumption that the variance of the stochastic gradient is bounded and the last inequality comes from Lemma 1. Next we can continue to bound $T_4$ with $\sigma$ and $\rho$:

$$\begin{aligned}
T_4 &= 2n\sigma^2\sum_{j>j'}\rho^{k-\frac{j+j'}{2}} \\
&= 2n\sigma^2\rho\frac{(\rho^{k/2}-1)(\rho^{k/2}-\sqrt{\rho})}{(\sqrt{\rho}-1)^2(\sqrt{\rho}+1)} \le 2n\sigma^2\frac{1}{(1-\sqrt{\rho})^2}.
\end{aligned}$$

Putting $T_3$ and $T_4$ back to $T_1$ we obtain:

$$T_1 \le n\sigma^2\left(\frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2}\right). \qquad (10)$$

Next, we then start to bounding $T_2$ in the same way. At first, we expend the $T_2$ with sum square formula to take the sum to the outside of the norm squared.

$$\begin{aligned}
T_2 &= \mathbb{E}\left\|\sum_{j=0}^{k}\partial f(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2 \\
&= \underbrace{\sum_{j=0}^{k}\mathbb{E}\left\|\partial f(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2}_{T_5} \\
&\quad + \underbrace{\sum_{j\neq j'}\mathbb{E}\left\langle\begin{array}{l}\partial f(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right), \\ \partial f(\hat{X}_{j'})\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j'+1}^{k}W_qe_i\right)\end{array}\right\rangle}_{T_6}.
\end{aligned}$$

For $T_5$, we have:

$$\begin{aligned}
T_5 &= \sum_{j=0}^{k}\mathbb{E}\left\|\partial f(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right)\right\|^2 \\
&\le \sum_{j=0}^{k}\mathbb{E}\|\partial f(\hat{X}_j)\|^2\left\|\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right\|^2 \\
&\le 12nL^2\sum_{j=0}^{k}\mathbb{E}\hat{M}_j\left\|\frac{\mathbf{1}_n}{n} - \prod_{q=j+1}^{k}W_qe_i\right\|^2 + 6n\varsigma^2\frac{1}{1-\rho}
\end{aligned}$$

$$
\begin{aligned}
&+2n\sum_{j=0}^{k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\left\|\frac{\mathbf{1}_n}{n}-\prod_{q=j+1}^{k}W_qe_i\right\|^2\\
&\le 12nL^2\sum_{j=0}^{k}\mathbb{E}\hat{M}_j\rho^{k-j}+6n\varsigma^2\frac{1}{1-\rho}\\
&+2n\sum_{j=0}^{k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\rho^{k-j}.
\end{aligned}
$$

We derive $T_6$ in the same way as we deduced $T_3$ :

$$
\begin{aligned}
&\sum_{j\neq j'}\mathbb{E}\left\langle\begin{array}{l}(\partial f(\hat{X}_j))\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j+1}^{k}W_qe_i\right),\\(\partial f(\hat{X}_{j'}))\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j'+1}^{k}W_qe_i\right)\end{array}\right\rangle\\
&\le\sum_{j\neq j'}\mathbb{E}\left(\begin{array}{l}\left\|\partial f(\hat{X}_j)\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j+1}^{k}W_qe_i\right)\right\|\cdot\\\left\|\partial f(\hat{X}_{j'})\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right)\\
&\le\sum_{j\neq j'}\mathbb{E}\left(\begin{array}{l}\|\partial f(\hat{X}_j)\|\cdot\left\|\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j+1}^{k}W_qe_i\right)\right\|\cdot\\\|\partial f(\hat{X}_{j'})\|\cdot\left\|\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right)\\
&\le\sum_{j\neq j'}\mathbb{E}\left(\begin{array}{l}\frac{\|\partial f(\hat{X}_j)\|^2}{2}\cdot\left\|\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j+1}^{k}W_qe_i\right)\right\|\cdot\\\left\|\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right)\\
&\quad+\sum_{j\neq j'}\mathbb{E}\left(\begin{array}{l}\frac{\|\partial f(\hat{X}_{j'})\|^2}{2}\cdot\left\|\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j+1}^{k}W_qe_i\right)\right\|\cdot\\\left\|\left(\frac{\mathbf{1}_n}{n}-\prod_{q=j'+1}^{k}W_qe_i\right)\right\|\end{array}\right)\\
&\le\sum_{j\neq j'}\mathbb{E}\left(\frac{\|\partial f(\hat{X}_j)\|^2}{2}+\frac{\|\partial f(\hat{X}_{j'})\|^2}{2}\right)\rho^{k-\frac{j+j'}{2}}\\
&=\sum_{j\neq j'}\mathbb{E}(\|\partial f(\hat{X}_j)\|^2)\rho^{k-\frac{j+j'}{2}}\\
&\le\sum_{j\neq j'}^{k}\left(12nL^2\mathbb{E}\hat{M}_j+2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\right)\rho^{k-\frac{j+j'}{2}}\\
&\quad+\sum_{j\neq j'}6n\varsigma^2\rho^{k-\frac{j+j'}{2}},
\end{aligned}
$$

where the second term can be bounded with $\varsigma$ and $\rho$:

$$
\sum_{j\neq j'}6n\varsigma^2\rho^{k-\frac{j+j'}{2}}=12n\varsigma^2\sum_{j>j'}\rho^{k-\frac{j+j'}{2}}\le\frac{12n\varsigma^2}{(1-\sqrt{\rho})^2},
$$

So $T_6$ can be written as follow:

$$
\begin{aligned}
T_6&\le\sum_{j\neq j'}^{k}\left(12nL^2\mathbb{E}\hat{M}_j+2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\right)\rho^{k-\frac{j+j'}{2}}\\
&\quad+\sum_{j\neq j'}6n\varsigma^2\rho^{k-\frac{j+j'}{2}}\\
&\le 2\sum_{j=0}^{k}\left(12nL^2\mathbb{E}\hat{M}_j\sum_{j'=j+1}^{k}\sqrt{\rho}^{2k-j-j'}\right)+\frac{12n\varsigma^2}{(1-\sqrt{\rho})^2}\\
&\quad+2\sum_{j=0}^{k}\left(2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\sum_{j'=j+1}^{k}\sqrt{\rho}^{2k-j-j'}\right)\\
&\le 2\sum_{j=0}^{k}\left(12nL^2\mathbb{E}\hat{M}_j+2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\right)\frac{\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\\
&\quad+\frac{12n\varsigma^2}{(1-\sqrt{\rho})^2}
\end{aligned}
$$

Plugging $T_5$ and $T_6$ into $T_2$, we have the upper bound for $T_2$:

$$
\begin{aligned}
T_2&\le 12nL^2\sum_{j=0}^{k}\mathbb{E}\hat{M}_j\rho^{k-j}+6n\varsigma^2\frac{1}{1-\rho}\\
&\quad+2n\sum_{j=0}^{k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\rho^{k-j}+\frac{12n\varsigma^2}{(1-\sqrt{\rho})^2}\\
&\quad+2\sum_{j=0}^{k}\left(12nL^2\mathbb{E}\hat{M}_j+2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\right)\frac{\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\\
&\le 12nL^2\sum_{j=0}^{k}\mathbb{E}\hat{M}_j\rho^{k-j}+2n\sum_{j=0}^{k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\rho^{k-j}\\
&\quad+2\sum_{j=0}^{k}\left(12nL^2\mathbb{E}\hat{M}_j+2n\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\right)\frac{\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\\
&\quad+6n\varsigma^2\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)
\end{aligned}
\tag{11}
$$

Finally substitute (10) and (11) into (9), we complete the proof. □

Lemma 3 uses the definition of consistency error to give a relationship between consistency error and gradients in the presence of delays. Using Lemmas 2 and 3 we can easily obtain the following lemma:

**Lemma 4.** *Let $U_1=\left(1-\frac{72}{(1-\sqrt{\rho})^2}\gamma^2nL^2\right)$ and if $U_1>0$, we have*

$$
\begin{aligned}
\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k&\le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\frac{K}{U_1}\\
&\quad+\frac{12}{(1-\sqrt{\rho})^2U_1}n\gamma^2\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

**Proof.** According to the definition of $\hat{X}_k=X_{k-\tau_k}$, $\hat{X}_k$ is a model parameter with delay information. Substituting $\hat{X}_k$ into Lemma 3 we obtain the following formula:

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{\hat{X}_k\mathbf{1}_n}{n}-\hat{X}_ke_i\right\|^2\\
&=\mathbb{E}\left\|\frac{X_{k-\tau_k}\mathbf{1}_n}{n}-X_{k-\tau_k}e_i\right\|^2\\
&\le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\\
&\quad+4n\gamma^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\rho^{k-\tau_k-1-j}\right)\\
&\quad+4n\gamma^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\frac{2\sqrt{\rho}^{k-\tau_k-1-j}}{1-\sqrt{\rho}}\right)\\
&\quad+24\gamma^2nL^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\hat{M}_j\left(\rho^{k-\tau_k-1-j}+\frac{2\sqrt{\rho}^{k-\tau_k-1-j}}{1-\sqrt{\rho}}\right)
\end{aligned}
$$

Observing the above inequality, the last term on the right-hand side of the inequality is the term related to $\hat{M}$. By averaging the left-hand side of the inequality over $n$ nodes we can establish the relationship between average consensus error with gradient with delay. We continue by bounding its average $\hat{M}_k$ on all nodes, which is defined by:

$$
\begin{aligned}
\mathbb{E}\hat{M}_k&:=\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{\hat{X}_k\mathbf{1}_n}{n}-\hat{X}_ke_i\right\|^2\\
&\le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)
\end{aligned}
$$

$$
\begin{aligned}
&+4n\gamma^2 \sum_{j=0}^{k-\tau_k-1} \mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \rho^{k-\tau_k-1-j}\right)\\
&+4n\gamma^2 \sum_{j=0}^{k-\tau_k-1} \mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \frac{2\sqrt{\rho}^{k-\tau_k-1-j}}{1-\sqrt{\rho}}\right)\\
&+24\gamma^2 nL^2 \sum_{j=0}^{k-\tau_k-1} \mathbb{E}\hat{M}_j \left(\rho^{k-\tau_k-1-j} + \frac{2\sqrt{\rho}^{k-\tau_k-1-j}}{1-\sqrt{\rho}}\right)
\end{aligned}
$$

Summing from $k=0$ to $K-1$ we obtain:

$$
\begin{aligned}
&\sum_{k=0}^{K-1} \mathbb{E}\hat{M}_k \le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&+4n\gamma^2 \sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1} \mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \rho^{k-\tau_k-1-j}\right)\\
&+4n\gamma^2 \sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1} \mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \frac{2\sqrt{\rho}^{k-\tau_k-1-j}}{1-\sqrt{\rho}}\right)\\
&+24\gamma^2 nL^2 \sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1} \mathbb{E}\hat{M}_j \left(\begin{array}{c}\frac{2\sqrt{\rho}^{k-\tau_k-1-j}}{1-\sqrt{\rho}}\\ +\rho^{k-\tau_k-1-j}\end{array}\right)
\end{aligned}
$$

Summing the terms containing $\rho$ and reducing them using the sum of infinite series, we obtain:

$$
\begin{aligned}
&\sum_{k=0}^{K-1} \mathbb{E}\hat{M}_k \le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&+4n\gamma^2 \sum_{k=0}^{K-1} \mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2 \left(\sum_{i=0}^{\infty}\rho^i + \frac{2\sum_{i=0}^{\infty}\sqrt{\rho}^i}{1-\sqrt{\rho}}\right)\\
&+24\gamma^2 nL^2 \sum_{k=0}^{K-1} \mathbb{E}\hat{M}_k \left(\sum_{i=0}^{\infty}\rho^i + \frac{2\sum_{i=0}^{\infty}\sqrt{\rho}^i}{1-\sqrt{\rho}}\right)\\
&\le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&+\frac{12}{(1-\sqrt{\rho})^2} n\gamma^2 \sum_{k=0}^{K-1} \mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\\
&+\frac{72}{(1-\sqrt{\rho})^2}\gamma^2 nL^2 \sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k
\end{aligned}
$$

By rearranging the terms we obtain

$$
\begin{aligned}
&\underbrace{\left(1-\frac{72}{(1-\sqrt{\rho})^2}\gamma^2 nL^2\right)}_{U_1}\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k\\
&\le 2n\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&+\frac{12}{(1-\sqrt{\rho})^2} n\gamma^2 \sum_{k=0}^{K-1} \mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

When $U_1$ is greater than 0, we divide both sides of this equation by $U_1$ and get Lemma 4. □

Employing the above lemmas, we now present the details of the proof of Theorem 1.

**Proof.** First, by expanding $X_{k+1}$ with the model update formula, we can establish the relationship between models and the gradients. According to Assumption 2 $f(\cdot)$ is L-smooth and Assumption 2 $W_k$ is a doubly stochastic matrix, we have

$$
\begin{aligned}
&\mathbb{E}f\left(\frac{X_{k+1}\mathbf{1}_n}{n}\right)\\
&=\mathbb{E}f\left(\frac{X_k W_k\mathbf{1}_n}{n}-\gamma\frac{\partial F(\hat{X}_k;\xi_k)\mathbf{1}_n}{n}\right)\\
&=\mathbb{E}f\left(\frac{X_k\mathbf{1}_n}{n}-\gamma\frac{\partial F(\hat{X}_k;\xi_k)\mathbf{1}_n}{n}\right)\\
&\le \mathbb{E}f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\gamma\mathbb{E}\left\langle\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right),\frac{\partial F(\hat{X}_k;\xi_k)\mathbf{1}_n}{n}\right\rangle\\
&+\frac{\gamma^2 L}{2}\mathbb{E}\left\|\frac{\partial F(\hat{X}_k;\xi_k)\mathbf{1}_n}{n}\right\|^2\\
&=\mathbb{E}f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\gamma\mathbb{E}\left\langle\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right),\frac{\partial F(\hat{X}_k;\xi_k)\mathbf{1}_n}{n}\right\rangle\\
&+\frac{\gamma^2 L}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^n\nabla F_i(\hat{x}_i^k,\xi_i^k)}{n}\right\|^2.
\end{aligned}
\tag{12}
$$

We establish the relationship between the gradients of the loss functions and the gradients of the expected loss functions. For the last term, according to Assumption 2, we have

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{\sum_{i=1}^n\nabla F_i(\hat{x}_i^k,\xi_i^k)}{n}\right\|^2\\
&=\mathbb{E}\left\|\frac{\sum_{i=1}^n\nabla F_i(\hat{x}_i^k,\xi_i^k)-\sum_{i=1}^n\nabla f_i(\hat{x}_i^k,\xi_i^k)}{n}\right\|^2\\
&+\mathbb{E}\left\|\frac{\sum_{i=1}^n\nabla f_i(\hat{x}_i^k,\xi_i^k)}{n}\right\|^2\\
&\le\frac{\sigma^2}{n}+\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
\tag{13}
$$

Based on the fact that $2\langle a,b\rangle=\|a\|^2+\|b\|^2-\|a-b\|^2$ and putting (13) back to (12) we have

$$
\begin{aligned}
&\mathbb{E}f\left(\frac{X_{k+1}\mathbf{1}_n}{n}\right)\\
&\le\mathbb{E}f\left(\frac{X_k\mathbf{1}_n}{n}\right)+\frac{\gamma}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&-\frac{\gamma-\gamma^2 L}{2}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2-\frac{\gamma}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2\\
&+\frac{\gamma^2\sigma^2 L}{2n}.
\end{aligned}
\tag{14}
$$

For the second term on the right side of (14), we have

$$
\begin{aligned}
&\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\le 2\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\nabla f\left(\frac{\hat{X}_k\mathbf{1}_n}{n}\right)\right\|^2\\
&+2\mathbb{E}\left\|\nabla f\left(\frac{\hat{X}_k\mathbf{1}_n}{n}\right)-\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\le 2\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\nabla f\left(\frac{\hat{X}_k\mathbf{1}_n}{n}\right)\right\|^2\\
&+2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n(\nabla f_i\left(\frac{\hat{X}_k\mathbf{1}_n}{n}\right)-\nabla f_i(\hat{x}_i^k))\right\|^2\\
&\le 2L^2\mathbb{E}\left\|\frac{(X_k-\hat{X}_k)\mathbf{1}_n}{n}\right\|^2+2L^2\mathbb{E}\hat{M}_k.
\end{aligned}
\tag{15}
$$

For the first term on the right of (15), we have

$$\begin{aligned}
&\mathbb{E}\left\|\frac{(X_k-\hat{X}_k)\mathbf{1}_n}{n}\right\|^2 = \mathbb{E}\left\|\frac{\sum_{t=1}^{\tau_k}\gamma\partial F(\hat{X}_{k-t};\xi_{k-t})\mathbf{1}_n}{n}\right\|^2\\
&\leq \tau_k\sum_{t=1}^{\tau_k}\gamma^2\mathbb{E}\left\|\frac{\partial F(\hat{X}_{k-t};\xi_{k-t})\mathbf{1}_n}{n}\right\|^2\\
&\leq \tau_k\sum_{t=1}^{\tau_k}\gamma^2\left(\frac{\sigma^2}{n}+\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2\right)\\
&= \frac{\tau_k^2\gamma^2\sigma^2}{n}+\tau_k\gamma^2\sum_{t=1}^{\tau_k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2.
\end{aligned} \tag{16}$$

From (14), (15) and (16) we have

$$\begin{aligned}
&\mathbb{E}f\left(\frac{X_{k+1}\mathbf{1}_n}{n}\right)\\
&\leq \mathbb{E}f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\frac{\gamma}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2+\gamma L^2\mathbb{E}\hat{M}_k\\
&\quad -\frac{\gamma-\gamma^2L}{2}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2+\frac{\gamma^2\sigma^2L}{2n}+\frac{\tau^2\gamma^3\sigma^2L^2}{n}\\
&\quad +\tau\gamma^3L^2\sum_{t=1}^{\tau_k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2.
\end{aligned}$$

Summing $k$ from 0 to $K-1$ we obtain

$$\begin{aligned}
&\mathbb{E}f\left(\frac{X_K\mathbf{1}_n}{n}\right)\\
&\leq \mathbb{E}f\left(\frac{X_0\mathbf{1}_n}{n}\right)-\frac{\gamma}{2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2\\
&\quad -\left(\frac{\gamma-\gamma^2L}{2}-\tau\gamma^3L^2\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\quad +\gamma L^2\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k+\frac{\gamma^2\sigma^2LK}{2n}+\frac{\tau^2\gamma^3\sigma^2L^2K}{n}.
\end{aligned}$$

We can use Lemma 4 to replace the term with $\hat{M}_k$. Let $U_2=\frac{\gamma-\gamma^2L}{2}-\tau\gamma^3L^2-\frac{12n\gamma^3L^2}{(1-\sqrt{\rho})^2U_1}$ and we obtain

$$\begin{aligned}
&\mathbb{E}f\left(\frac{X_K\mathbf{1}_n}{n}\right)\leq \mathbb{E}f\left(\frac{X_0\mathbf{1}_n}{n}\right)+\frac{\gamma^2\sigma^2LK}{2n}+\frac{\tau^2\gamma^3\sigma^2L^2K}{n}\\
&\quad -\frac{\gamma}{2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f(\frac{X_k\mathbf{1}_n}{n})\right\|^2-U_2\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\quad +2n\gamma^3L^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\frac{K}{U_1}.
\end{aligned}$$

Note that $\mathbb{E}f(x_0)=f(0)=f_0$ because we assume that all workers start from 0, and $\mathbb{E}f(\frac{X_K\mathbf{1}_n}{n})\geq f^\star$ where $f^\star$ is the optimal solution. So by arranging the terms and dividing both sides of the inequality by $K$, while $U_1>0$ and $U_2\leq 0$ are satisfied Theorem 1 is proved. □

Next, we give the proof of Corollary 1.

**Proof.** The conditions of Theorem 1 are $U_1>0$ and $U_2\leq 0$, and we can choose appropriate $\gamma$ to satisfy these conditions. At first, we can let $U_1\geq 1/2$ which is a stronger restriction on $U_1$ and we can obtain

$$U_1=(1-\frac{72}{(1-\sqrt{\rho})^2}\gamma^2nL^2)\geq\frac{1}{2}\Rightarrow\gamma\leq\frac{1-\sqrt{\rho}}{12L\sqrt{n}}.$$

So when $U_1\geq\frac{1}{2}$ and $U_2\geq 0$, we can imply the following result:

$$\begin{aligned}
&U_2\geq\frac{\gamma-\gamma^2L}{2}-\tau\gamma^3L^2-\frac{24n\gamma^3L^2}{(1-\sqrt{\rho})^2}\geq 0\\
&\overset{\gamma>0}{\Rightarrow}\frac{\gamma L}{2}+\tau\gamma^2L^2+\frac{24n\gamma^2L^2}{(1-\sqrt{\rho})^2}\leq\frac{1}{2}.
\end{aligned} \tag{17}$$

To satisfy (17), we can make every term on the left side of the inequality smaller than $\frac{1}{6}$:

$$\begin{aligned}
\frac{\gamma L}{2}&\leq\frac{1}{6}\Rightarrow\gamma\leq\frac{1}{3L},\\
\tau\gamma^2L^2&\leq\frac{1}{6}\Rightarrow\gamma\leq\frac{1}{\sqrt{6\tau}L},\\
\frac{24n\gamma^2L^2}{(1-\sqrt{\rho})^2}&\leq\frac{1}{6}\Rightarrow\gamma\leq\frac{1-\sqrt{\rho}}{12L\sqrt{n}}.
\end{aligned}$$

Combining all the constraints on $\gamma$ we have

$$\gamma\leq\min\left\{\frac{1}{3L},\frac{1}{\sqrt{6\tau}L},\frac{1-\sqrt{\rho}}{12L\sqrt{n}}\right\}.$$

Let $\gamma=\frac{1}{L+\sqrt{\sigma^2+6\varsigma^2}\sqrt{\frac{K}{n}}}$ and from Theorem 1 we have

$$\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f(\frac{X_k\mathbf{1}_n}{n})\right\|^2\\
&\leq\frac{2(f_0-f^*)L}{K}+\frac{2(f_0-f^*)\sqrt{\sigma^2+6\varsigma^2}}{\sqrt{nK}}+\frac{\sigma L}{\sqrt{nK}}\\
&\quad+\frac{2\tau^2L^2}{K}+\frac{4n^2L^2}{KU_1}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\\
&=\frac{2(f_0-f^*+\tau^2L)L}{K}+\frac{(2f_0-2f^*+L)\sqrt{\sigma^2+6\varsigma^2}}{\sqrt{nK}}\\
&\quad+\frac{4\sqrt{\sigma^2+6\varsigma^2}L}{\sqrt{nK}}\underbrace{\frac{2n^{\frac{5}{2}}L}{\sqrt{\sigma^2+6\varsigma^2}\sqrt{K}}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)}_{U_3}.
\end{aligned}$$

Now if $U_3\leq 1$ is satisfied we conclude Corollary 1. When $U_3\leq 1$ we have

$$\frac{2n^{\frac{5}{2}}L}{\sqrt{\sigma^2+6\varsigma^2}\sqrt{K}}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\leq 1.$$

Converting to the constraints on $K$, we can get

$$K\geq\frac{4n^5L^2}{\sigma^2+6\varsigma^2}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)^2.$$

In the end, converting all other constraints on $\gamma$ to constraints on $K$:

$$K\geq\frac{n}{\sigma^2+6\varsigma^2}\max\left\{4L^2,(\sqrt{6\tau}-1)^2L^2,\frac{144nL^2}{(1-\sqrt{\rho})^2}\right\}.$$

Now we have concluded Corollary 1. □

## 5. Noise robust decentralized SGD

Algorithm 1 is designed to deal with the issue of unstable network connections. In addition to this case, the communication between workers usually is perturbed by noise. During model aggregation, noise is often introduced via wireless channel noise, gradient compression, or purposefully imposed privacy protection mechanisms. Let $Q(x_i^k)$ denote the perturbed model of worker $i$ at iteration $k$. At first, we give the following definition.

**Definition 1.** For any $x,y\in\mathbb{R}^d$, we define that $x\leq y\iff x_i\leq y_i$ for $i\in[1,\ldots,d]$, where $x_i$ and $y_i$ denote the $i$th dimension of $x$ and $y$.

To better bound the gap between the perturbed model and the original model, we give the following assumption.

**Assumption 3** (*Bounded Noise*). We assume that the perturbed model is bounded and for all $x\in\mathbb{R}^d$, they satisfy

$$\mathbb{E}Q(x)\leq cx \tag{18}$$

where $c$ is a constant and $c>0$.

**Algorithm 2:** NRDSGD algorithm

```
Input: Initialize $x_i^0$, $\hat{x}_i^0$ and $Q(x_i^0)$, $\forall i \in [n]$ with the same value,
       mixing matrix $W$, learning rate $\gamma$, variance of Gaussian
       distribution $\sigma_g$ and number of total iterations $K$.
for $k = 0, 1, \dots, K-1$ (all workers in parallel) do
    Randomly sample $\xi_i^k$ from local data for worker $i \in [n]$.
    Compute gradient $\nabla F(x_i^k, \xi_i^k)$.
    Update model according to $x_i^{k+1} = x_i^k - \gamma \nabla F(x_i^k, \xi_i^k)$.
    Sample noise $\eta_i^{k+1}$ from Gaussian distribution $\mathcal{N}(0, \sigma_g^2)$ and
      compute the perturbed model $Q(x_i^{k+1})$.
    Send $Q(x_i^{k+1})$ and receive models from neighbors.
    if receive any $Q(x_j^{k+1})$, $j \in \mathcal{J}_i$ then
        $\hat{x}_j^{k+1} = Q(x_j^{k+1})$.
    Aggregate model by $x_i^{k+1} = \sum_{j \in \mathcal{J}_i} W_{ij} \hat{x}_j^{k+1}$.
```

Assumption 3 requires that the value at any dimension of the perturbed model will not deviate the original model too far in expectation. In a high level, Assumption 3 gives a bound to guarantee that the perturbed model may only oscillate on the basis of the original model and the degree of oscillation is bounded by parameter $c$.

In fact, the presence of noise is a common phenomenon in the process of model exchange. In the problem we are studying, the sources of noise can be divided into two categories. One is artificially introduced noise caused by communication compression mechanisms or differential privacy(DP) mechanisms.

- Communication compression. In this case, $Q(x)$ can be considered as the compressed model. If we do not pay attention to the specific communication compression method, then the compression operators can be considered as artificial noise added to the model [11,13]. This kind of communication compression noise usually makes the perturbed model smaller than the original model, that is, the parameter $c$ is less than 1 in this case.
- Differential privacy (DP). DP mechanisms usually add noise to the data to prevent privacy leakage caused by queries on adjacent data sets. Commonly used DP mechanisms are Gaussian mechanism and Laplace mechanism [16,36]. DP mechanisms bound the noise based on the privacy precise so the amount of added noise is limited by the level of privacy protection.

In addition to the artificially introduced noise, there will inevitably be some noise in the communication process. In the real channel, a common type of noise is Gaussian noise which follows the Gaussian distribution with a mean value of 0. Therefore, we add Gaussian noise in the model exchange process to simulate the real scene: $Q(x) = x + \eta$, where $\eta \in \mathbb{R}^d$ and $\eta_i \sim \mathcal{N}(0, \sigma_g^2), \forall i \in [0, \dots .d]$ is the Gaussian distribution. Definitely, this Gaussian noise meets Assumption 3 because the expectation of noise is 0.

### 5.1. Algorithm

To imitate genuine data interchange, we introduce noise to the model to be exchanged:

$$Q(x_i^{k+1}) = x_i^{k+1} + \eta_i^{k+1}$$

where $\eta_i^{k+1} \sim \mathcal{N}(0, \sigma_g^2)$ is the Gaussian distribution and $Q(x_i^{k+1})$ is the perturbed model. Thus we obtain the Algorithm 2 **NRDSGD**–*Noise Robust Decentralized Stochastic Gradient Descent*.

### 5.2. Theoretical analysis

Based on our Assumption 3, the data to be exchanged has been perturbed by noise, thus $\hat{x}_i^k$ would become $Q(x_i^{k-\tau_k})$. Written in matrix form we have:

$$\hat{X}_k = Q(X_{k-\tau_k}) = [\hat{x}_1^k, \dots, \hat{x}_n^k] \in \mathbb{R}^{d \times n}.$$

Before presenting Theorem 2, we define the following variables:

$$\overline{\rho} = \left( \frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2} \right),$$

$$V_1 = \left(1 - 24nc^2\gamma^2L^2\right) \overline{\rho},$$

$$V_2 = \left( \frac{\gamma - \gamma^2 L}{2} - \tilde{c}^2 \tau \gamma^3 L^2 - \frac{4nc^2\gamma^3L^2\overline{\rho}}{V_1} \right).$$

These variables are constants, which are defined to simplify the final expression.

**Theorem 2.** *Under Assumptions 1, 2 and 3, if $V_1 > 0$ and $V_2 \geq 0$ are satisfied, then we have the following convergence rate*

$$\sum_{k=0}^{K-1} \mathbb{E} \left\| \nabla f \left( \frac{X_k \mathbf{1}_n}{n} \right) \right\|^2 \leq \frac{2(f_0 - f^\star)}{\gamma K} + \frac{\gamma \sigma^2 L}{n} + \frac{2\tilde{c}^2 \tau^2 \gamma^2 \sigma^2 L^2}{n} + \frac{4nc^2\gamma^2L^2(\sigma^2 + 6\varsigma^2)\overline{\rho}}{V_1},$$

*where $\tilde{c} = \max\{1, c\}$, $f_0 = f(0)$ and $f^\star$ denotes the optimal solution.*

Choosing an appropriate learning rate, we can get the following corollary

**Corollary 2.** *Under Assumptions 1, 2 and 3, by choosing $\gamma = \frac{1}{L+\sqrt{\sigma^2+6\varsigma^2}\sqrt{\frac{K}{n}}}$, we have the following convergence rate*

$$\frac{1}{K} \sum_{k=0}^{K-1} \mathbb{E} \left\| \nabla f \left( \frac{X_k \mathbf{1}_n}{n} \right) \right\|^2 \leq \frac{2(f_0 - f^\star + \tilde{c}^2 \tau^2 L)L}{K} + \frac{(2f_0 - 2f^\star + (4c^2+1)L)\sqrt{\sigma^2 + 6\varsigma^2}}{\sqrt{nK}},$$

*if $K$ is large enough to satisfy*

$$K \geq \frac{nL^2}{\sigma^2 + 6\varsigma^2} \max\left\{4, (\tilde{c}\sqrt{6\tau} - 1)^2, 4n^4\overline{\rho}^2, 48c^2\overline{\rho}\right\},$$

*where $\tilde{c}$, $f_0$ and $f^\star$ follows the definition in Theorem 2.*

Theorem 2 and Corollary 2 demonstrate that NRDSGD achieves the same convergence rate and maintains the same properties comparing to RDSGD. Moreover, the effect of communication noise is reflected on the constant $c$. It shows that communication noise have the similar effect with the unstable workers, which may lower down the precise.

### 5.3. Analysis of Algorithm 2

In this part, we give theoretical support for Theorem 2 in detail. Similarly to our proof of Theorem 1, we have the same definition of $M_k$. But due to that $\hat{X}_k$ becomes $Q(X_{k-\tau_k})$, so we have

$$\hat{M}_k := \frac{1}{n} \sum_{i=1}^{n} \left\| \frac{Q(X_{k-\tau_k})\mathbf{1}_n}{n} - Q(X_{k-\tau_k}) e_i \right\|^2.$$

For Lemmas 1–3, we do not use the definition of $\hat{X}_k$, so these lemmas are still worked in this case.

**Lemma 5.** *Let $V_1 = \left(1 - 24nc^2\gamma^2L^2 \left(\frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2}\right)\right)$ and for any $K \geq 1$ if $V_1 > 0$*

$$\mathbb{E}\hat{M} \leq \frac{2nc^2\gamma^2K}{V_1}(\sigma^2 + 6\varsigma^2)\left( \frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2} \right) K + \frac{4nc^2\gamma^2}{V_1} \left( \frac{1}{1-\rho} + \frac{2}{(1-\sqrt{\rho})^2} \right) \sum_{k=0}^{K-1} \mathbb{E} \left\| \frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n} \right\|^2.$$

**Proof.** Under the Assumption 3 and Lemma 3, we have

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{Q(X_{k+1})\mathbf{1}_n}{n} - Q(X_{k+1})e_i\right\|^2\\
&\leq c^2\mathbb{E}\left\|\frac{X_{k+1}\mathbf{1}_n}{n} - X_{k+1}e_i\right\|^2\\
&\leq 2nc^2\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\\
&\quad + 4nc^2\gamma^2\sum_{j=0}^{k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\left(\rho^{k-j}+\frac{2\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\right)\\
&\quad + 24nc^2\gamma^2L^2\sum_{j=0}^{k}\mathbb{E}\hat{M}_j\left(\rho^{k-j}+\frac{2\sqrt{\rho}^{k-j}}{1-\sqrt{\rho}}\right).
\end{aligned}
$$

Note the definition of $\hat{M}_k$, we have

$$
\begin{aligned}
&\mathbb{E}\hat{M}_k = \mathbb{E}\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{Q(X_{k-\tau_k})\mathbf{1}_n}{n} - Q(X_{k-\tau_k})e_i\right\|^2\\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{Q(X_{k-\tau_k})\mathbf{1}_n}{n} - Q(X_{k-\tau_k})e_i\right\|^2\\
&\leq 2nc^2\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\\
&\quad + 4nc^2\gamma^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\rho^{k-j-\tau_k-1}\\
&\quad + 4nc^2\gamma^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\frac{2\sqrt{\rho}^{k-j-\tau_k-1}}{1-\sqrt{\rho}}\\
&\quad + 24nc^2\gamma^2L^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\hat{M}_j\rho^{k-j-\tau_k-1}\right)\\
&\quad + 24nc^2\gamma^2L^2\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\hat{M}_j\frac{2\sqrt{\rho}^{k-j-\tau_k-1}}{1-\sqrt{\rho}}\right).
\end{aligned}
$$

Summing from $k=0$ to $K-1$, we have

$$
\begin{aligned}
&\sum_{k=0}^{K-}\mathbb{E}\hat{M}_k \leq 2nc^2\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&\quad + 4nc^2\gamma^2\sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\rho^{k-j-\tau_k-1}\right)\\
&\quad + 4nc^2\gamma^2\sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\left\|\frac{\partial f(\hat{X}_j)\mathbf{1}_n}{n}\right\|^2\frac{2\sqrt{\rho}^{k-j-\tau_k-1}}{1-\sqrt{\rho}}\right)\\
&\quad + 24nc^2\gamma^2L^2\sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\hat{M}_j\rho^{k-j-\tau_k-1}\right)\\
&\quad + 24nc^2\gamma^2L^2\sum_{k=0}^{K-1}\sum_{j=0}^{k-\tau_k-1}\mathbb{E}\left(\hat{M}_j\frac{2\sqrt{\rho}^{k-j-\tau_k-1}}{1-\sqrt{\rho}}\right)
\end{aligned}
$$

Summing the terms containing $\rho$ and reducing them using the sum of infinite series, we obtain

$$
\begin{aligned}
&\sum_{k=0}^{K-}\mathbb{E}\hat{M}_k \leq 2nc^2\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&\quad + 4nc^2\gamma^2\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\left(\sum_{j=0}^{\infty}\rho^j+\frac{2\sum_{j=0}^{\infty}\sqrt{\rho}^j}{1-\sqrt{\rho}}\right)\\
&\quad + 24nc^2\gamma^2L^2\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_j\left(\sum_{j=0}^{\infty}\rho^j+\frac{2\sum_{j=0}^{\infty}\sqrt{\rho}^j}{1-\sqrt{\rho}}\right)\\
&\leq 2nc^2\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&\quad + 4nc^2\gamma^2\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\quad + 24nc^2\gamma^2L^2\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k.
\end{aligned}
$$

By rearranging the terms we obtain

$$
\begin{aligned}
&\left(1-24nc^2\gamma^2L^2\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\right)\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k\\
&\leq 2nc^2\gamma^2(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&\quad + 4nc^2\gamma^2\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

Let $V_1=\left(1-24nc^2\gamma^2L^2\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\right)$ and if $V_1>0$ we have

$$
\begin{aligned}
&\sum_{k=0}^{K-1}\mathbb{E}\hat{M}_k \leq \frac{2nc^2\gamma^2K}{V_1}(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&\quad + \frac{4nc^2\gamma^2}{V_1}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2. \quad \square
\end{aligned}
$$

Now we can start to prove Theorem 2.

**Proof.** The proof is the same as Theorem 1, the only difference is the definition of $\hat{X}_k$, so we can directly start from (15) to prove. According to Assumption 3, for the first term on the right side of (15), we have

$$
\begin{aligned}
\mathbb{E}\left\|\frac{(X_k-\hat{X}_k)\mathbf{1}_n}{n}\right\|^2 &= \mathbb{E}\left\|\frac{(Q(X_{k-\tau_k})-X_k)\mathbf{1}_n}{n}\right\|^2\\
&\leq \mathbb{E}\left\|\frac{(cX_{k-\tau_k}-X_k)\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

Next, we analyze two cases: (1) $0<c\leq 1$, and (2) $c>1$.

**Case 1:** if $0<c\leq 1$, then $cX_{k-\tau_k}-X_k \leq X_{k-\tau_k}-X_k$. Thus we have

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{(X_k-\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\leq \mathbb{E}\left\|\frac{(X_{k-\tau_k}-X_k)\mathbf{1}_n}{n}\right\|^2\\
&= \mathbb{E}\left\|\frac{\sum_{t=1}^{\tau_k}\gamma\partial F(\hat{X}_{k-t};\xi_{k-t})\mathbf{1}_n}{n}\right\|^2\\
&\leq \tau_k\sum_{t=1}^{\tau_k}\gamma^2\mathbb{E}\left\|\frac{\partial F(\hat{X}_{k-t};\xi_{k-t})\mathbf{1}_n}{n}\right\|^2\\
&\leq \tau_k\sum_{t=1}^{\tau_k}\gamma^2\left(\frac{\sigma^2}{n}+\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2\right)\\
&= \frac{\tau_k^2\gamma^2\sigma^2}{n}+\tau_k\gamma^2\sum_{t=1}^{\tau_k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

**Case 2:** if $c>1$, then $cX_{k-\tau_k}-X_k \leq c(X_{k-\tau_k}-X_k)$. Thus we have

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{(X_k-\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\leq c^2\mathbb{E}\left\|\frac{(X_{k-\tau_k}-X_k)\mathbf{1}_n}{n}\right\|^2
\end{aligned}
$$

$$
\begin{aligned}
&= c^2\mathbb{E}\left\|\frac{\sum_{t=1}^{\tau_k}\gamma\partial F(\hat{X}_{k-t};\xi_{k-t})\mathbf{1}_n}{n}\right\|^2\\
&\le c^2\tau_k\sum_{t=1}^{\tau_k}\gamma^2\mathbb{E}\left\|\frac{\partial F(\hat{X}_{k-t};\xi_{k-t})\mathbf{1}_n}{n}\right\|^2\\
&\le c^2\tau_k\sum_{t=1}^{\tau_k}\gamma^2\left(\frac{\sigma^2}{n}+\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2\right)\\
&= \frac{c^2\tau_k^2\gamma^2\sigma^2}{n}+c^2\tau_k\gamma^2\sum_{t=1}^{\tau_k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

Combining the two cases and let $\tilde{c}=\max\{1,c\}$, we have

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{(X_k-\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\le \frac{\tilde{c}^2\tau_k^2\gamma^2\sigma^2}{n}+\tilde{c}^2\tau_k\gamma^2\sum_{t=1}^{\tau_k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
\tag{19}
$$

From (14) (15) and (19) we obtain

$$
\begin{aligned}
&\mathbb{E}f\left(\frac{X_{k+1}\mathbf{1}_n}{n}\right)\\
&\le \mathbb{E}f\left(\frac{X_k\mathbf{1}_n}{n}\right)-\frac{\gamma}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2+\gamma L^2\mathbb{E}\hat{M}_k\\
&\quad-\frac{\gamma-\gamma^2L}{2}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2+\frac{\tilde{c}^2\tau^2\gamma^3\sigma^2L^2}{n}\\
&\quad+\frac{\gamma^2\sigma^2L}{2n}+\tilde{c}^2\tau\gamma^3L^2\sum_{t=1}^{\tau_k}\mathbb{E}\left\|\frac{\partial f(\hat{X}_{k-t})\mathbf{1}_n}{n}\right\|^2.
\end{aligned}
$$

Summing from $k=0$ to $K-1$ and substitute the inequality of Lemma 5 into the above inequality, we obtain

$$
\begin{aligned}
&\mathbb{E}f\left(\frac{X_K\mathbf{1}_n}{n}\right)\\
&\le \mathbb{E}f\left(\frac{X_0\mathbf{1}_n}{n}\right)-\frac{\gamma}{2}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2\\
&\quad-\left(\frac{\gamma-\gamma^2L}{2}-\tilde{c}^2\tau\gamma^3L^2\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\quad+\frac{2nc^2\gamma^3L^2K}{V_1}(\sigma^2+6\varsigma^2)\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)K\\
&\quad+\frac{4nc^2\gamma^3L^2}{V_1}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\sum_{k=0}^{K-1}\mathbb{E}\left\|\frac{\partial f(\hat{X}_k)\mathbf{1}_n}{n}\right\|^2\\
&\quad+\frac{\gamma^2\sigma^2LK}{2n}+\frac{\tilde{c}^2\tau^2\gamma^3\sigma^2L^2K}{n}.
\end{aligned}
$$

Note that all models start from 0, thus let $f_0$ denote $f(0)$ and $f^\star$ denote the optimal solution. Let $V_2=\left(\frac{\gamma-\gamma^2L}{2}-\tilde{c}^2\tau\gamma^3L^2-\frac{4nc^2\gamma^3L^2}{V_1}\left(\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}\right)\right)$ and rearrange the terms then we complete the proof of Theorem 2. □

The proof of Corollary 2 is similar to Corollary 1, and we give the proof of Corollary 2 in the following part.

**Proof.** We first guarantee $V_1\ge 0$. We make $V_1\ge\frac{1}{2}$ which is a stronger restriction on $V_1$ and we can obtain

$$
V_1=\left(1-24nc^2\gamma^2L^2\left(\underbrace{\frac{1}{1-\rho}+\frac{2}{(1-\sqrt{\rho})^2}}_{\overline{\rho}}\right)\right)\ge\frac{1}{2}
$$

Converting to constraints on $\gamma$ we get $\gamma^2\le\frac{1}{48nc^2L^2\overline{\rho}}$. So when $V_1\ge\frac{1}{2}$ and $V_2\ge 0$, we can imply the following result:

$$
\begin{aligned}
&V_2=\frac{\gamma-\gamma^2L}{2}-\tilde{c}^2\tau\gamma^3L^2-8nc^2\gamma^3L^2\overline{\rho}\ge 0\\
&\overset{\gamma>0}{\Rightarrow}\frac{1}{2}-\frac{\gamma L}{2}-\tilde{c}^2\tau\gamma^2L^2-8nc^2\gamma^2L^2\overline{\rho}\ge 0\\
&\Rightarrow\frac{\gamma L}{2}+\tilde{c}^2\tau\gamma^2L^2+8nc^2\gamma^2L^2\overline{\rho}\le\frac{1}{2}.
\end{aligned}
\tag{20}
$$

To satisfy Eq. (20), we can make every term on the left side of the inequality smaller than $\frac{1}{6}$:

$$
\begin{aligned}
\frac{\gamma L}{2}&\le\frac{1}{6}\Rightarrow\gamma\le\frac{1}{3L},\\
\tilde{c}^2\tau\gamma^2L^2&\le\frac{1}{6}\Rightarrow\gamma\le\frac{1}{\tilde{c}L\sqrt{6\tau}},\\
8nc^2\gamma^2L^2\overline{\rho}&\le\frac{1}{6}\Rightarrow\gamma\le\frac{1}{4cL\sqrt{3n\overline{\rho}}}.
\end{aligned}
$$

Combining all the constraints on $\gamma$ we have

$$
\gamma\le\min\left\{\frac{1}{3L},\frac{1}{\tilde{c}L\sqrt{6\tau}},\frac{1}{4cL\sqrt{3n\overline{\rho}}}\right\}.
$$

Let $\gamma=\frac{1}{L+\sqrt{\sigma^2+6\varsigma^2}\sqrt{\frac{K}{n}}}$ and from Theorem 2 we have

$$
\begin{aligned}
&\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f\left(\frac{X_k\mathbf{1}_n}{n}\right)\right\|^2\\
&\le\frac{2(f_0-f^*)}{\gamma K}+\frac{\gamma\sigma^2L}{n}+\frac{2\tilde{c}^2\tau^2\gamma^2\sigma^2L^2}{n}\\
&\quad+\frac{4nc^2\gamma^2L^2(\sigma^2+6\varsigma^2)\overline{\rho}}{V_1}\\
&\le\frac{2(f_0-f^*)L}{K}+\frac{2(f_0-f^*)\sqrt{\sigma^2+6\varsigma^2}}{\sqrt{nK}}+\frac{\sigma L}{\sqrt{nK}}\\
&\quad+\frac{2\tilde{c}^2\tau^2L^2}{K}+\frac{4n^2c^2L^2\overline{\rho}}{KV_1}\\
&=\frac{2(f_0-f^*+\tilde{c}^2\tau^2L)L}{K}+\frac{(2f_0-2f^*+L)\sqrt{\sigma^2+6\varsigma^2}}{\sqrt{nK}}\\
&\quad+\frac{4c^2\sqrt{\sigma^2+6\varsigma^2}L}{\sqrt{nK}}\underbrace{\frac{2n^{\frac{5}{2}}L\overline{\rho}}{\sqrt{\sigma^2+6\varsigma^2}\sqrt{K}}}_{V_3}.
\end{aligned}
$$

While $V_3\le 1$, we complete the proof of Corollary 2. Thus we have

$$
\frac{2n^{\frac{5}{2}}L\overline{\rho}}{\sqrt{\sigma^2+6\varsigma^2}\sqrt{K}}\le 1\Rightarrow K\ge\frac{4n^5L^2\overline{\rho}^2}{\sigma^2+6\varsigma^2}.
$$

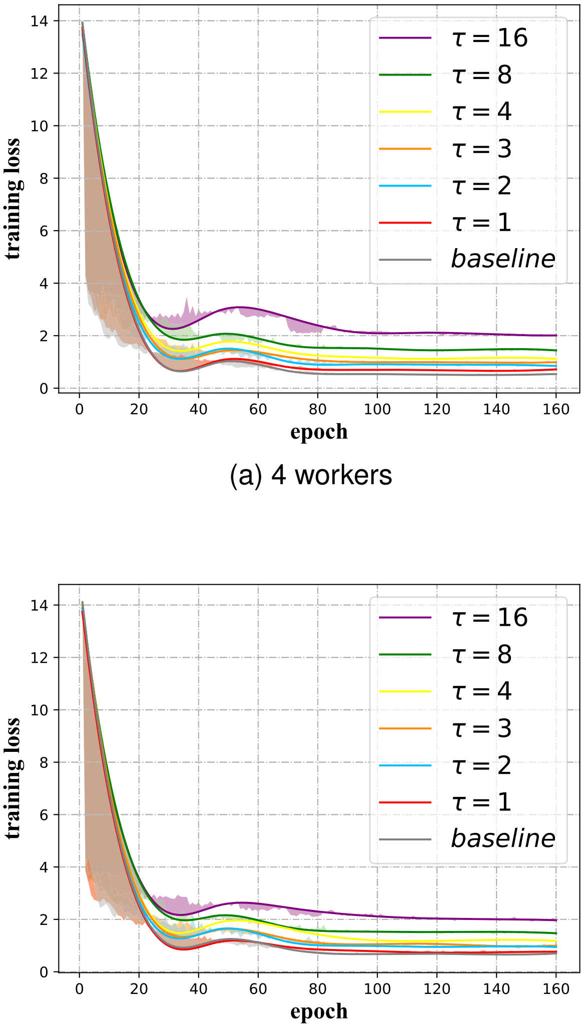
In the end, converting the constraints on $\gamma$ to constraints on $K$, we have

$$
\begin{aligned}
&\gamma=\frac{1}{L+\sqrt{\sigma^2+6\varsigma^2}\sqrt{\frac{K}{n}}}\le\min\left\{\frac{1}{3L},\frac{1}{\tilde{c}L\sqrt{6\tau}},\frac{1}{4cL\sqrt{3n\overline{\rho}}}\right\}\\
&\Leftrightarrow L+\sqrt{\sigma^2+6\varsigma^2}\sqrt{\frac{K}{n}}\ge\max\left\{3L,\tilde{c}L\sqrt{6\tau},4cL\sqrt{3n\overline{\rho}}\right\}\\
&\Rightarrow K\ge\frac{nL^2}{\sigma^2+6\varsigma^2}\max\left\{4,(\tilde{c}\sqrt{6\tau}-1)^2,48nc^2\overline{\rho}\right\}.
\end{aligned}
$$

Combining all the conditions, we complete the proof of Corollary 2. □

## 6. Experiments

We evaluate our decentralized learning algorithms experimentally and compare our algorithms with DPSGD [7] over stable networks



(a) 4 workers

(b) 8 workers

**Fig. 1.** Effect of network instability on ResNet18.

to show the influence of unstable workers and communication noise on decentralized SGD. We mainly explore the influence of network instability $\tau$ and the level of noise $c$ on the algorithm performance.

### 6.1. Experimental setup

**Dataset and Models** In our experiments, we validate our algorithm on the image classification task. We choose CIFAR10 [37] as our dataset. CIFAR10 contains a total of 10 categories of RGB color images, and each category has 6,000 images. We use ResNet [38] with different layers as our model and PyTorch as our distributed learning framework. Specifically, Python 3.6 and Pytorch 1.9.0 are adopted in our experiments.

**Implementation** We train the model on Intel(R) Xeon(R) CPU E5-2699C v4 @ 2.20 GHz and use CPU multiprocess to simulate different workers. To implement multiprocess parallelism, our communication scheme is the *torch.distributed* module, which implements CPU communication through MPI and GPU communication through NCCL. We mainly explore the effect of $\tau$ and $c$ on the algorithm performance. Here are some settings that are the same for both cases. We set epoch is 160 and weight decay is 0.0001. For learning rate, adopting the *linear scaling rule* as described in [2], we set learning rate decay of 10 percent at the 80th and the 120th epoch, respectively, to achieve a more precise convergence. Based on our theoretical analysis, the value of $\gamma$ is related to $n$ and $K$. Thus, we would discuss the initial value of $\gamma$ in detail later. Also, we do not apply any momentum and regularization to be consistent with our algorithm and theoretical analysis.

**Unstable Setting** As declared in Section 3, we use $\tau$ to represent the degree of network instability. A larger $\tau$ means that offline workers need more time to reconnect, which leads to longer time to get a global consensus. Note that $\tau = 0$ means that the network is stable and the algorithm we compared is DPSGD under a stable network.

**Noise Mechanism** Under Assumption 3 and our noise mechanism, we use parameter $c$ to measure the gap between the perturbed model and the original model so we mainly compare the effect of parameter $c$. A larger $c$ means that more noise is introduced and the perturbed model is further away from the original model.

**Network Topology** During our experiment, we choose the ring network, which is commonly used in distributed learning. The structure of the ring network is all workers are connected end to end in a loop. Therefore each worker only has and can communicate with two neighbors. Based on the ring topology, we use the *Metropolis–Hastings*(MH) algorithm proposed by [39] to construct the doubly stochastic matrix $W$. MH algorithm is described as follows:

$$W^{[ij]} = \begin{cases} \max(d_i, d_j) & \text{if } i \neq j \text{ and } j \in \mathcal{J}_i \\ 1 - \sum_{j \in \mathcal{J}_i} & \text{if } i = j \\ 0 & \text{otherwise,} \end{cases}$$

where $d_i$ and $d_j$ are the degrees of worker $i$ and worker $j$. Note that we let degree of a worker denotes the number of neighbors and the worker itself which indicates that each worker aggregates the models with all neighbors and itself. Thus, when the network topology is a ring, we obtain the following mixing matrix:

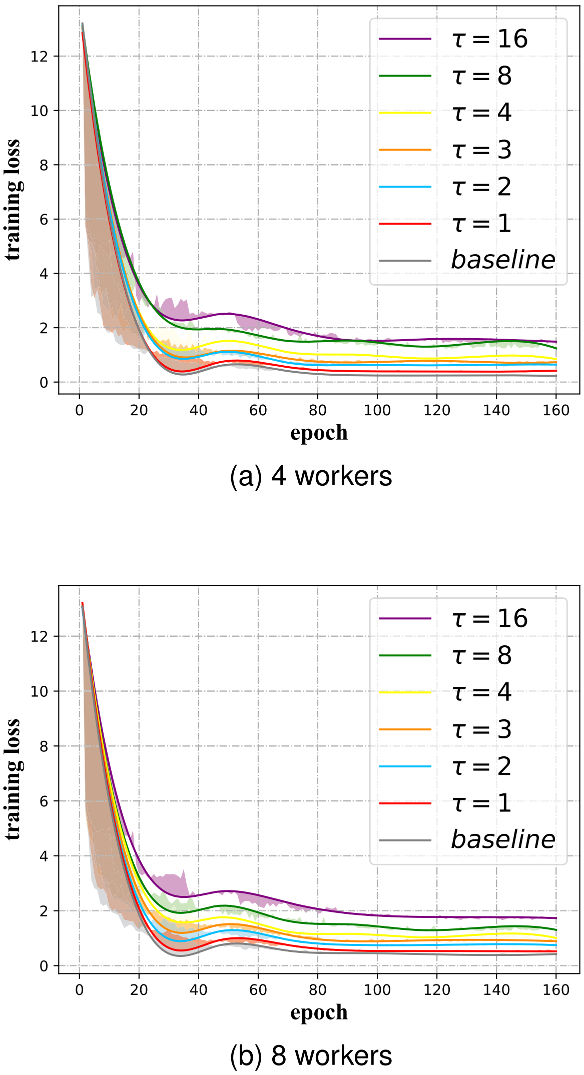
$$W = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & 0 & \cdots & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & & & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \frac{1}{3} & 0 \\ 0 & & & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \cdots & 0 & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

In the last, we summarize all the hyperparameter settings used in the experiments in the following

- Batch size: 256 per worker for ResNet.
- Epoch: 160.
- Number of workers: 4 and 8.
- Learning rate: 0.15 for 8 workers and 0.1 for 4 workers. For ResNet decay by a factor of 10 at the 81st epoch and the 121nd epoch.

### 6.2. Experimental results

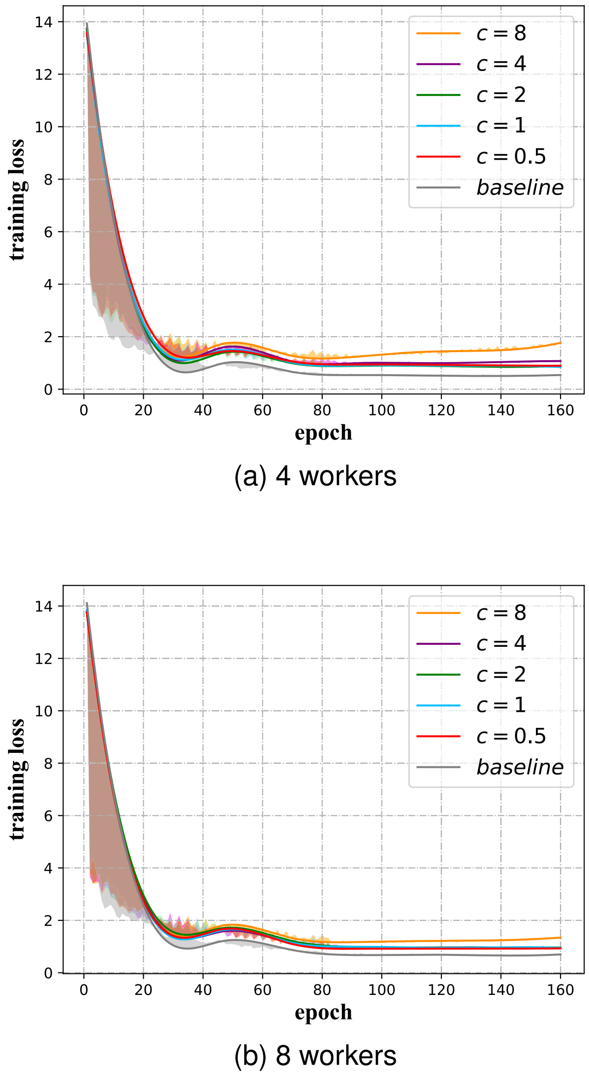
**Effect of $\tau$** In this part, we study the cases where $\tau$ is 1, 2, 3, 4, 8 and 16 respectively. The baseline algorithm we compare is DPSGD over a stable network connection. We study the effect of $\tau$ under the condition that $n$ is 4 and $n$ is 8 respectively. We set $\gamma$ is 0.1 when $n$ is 4 and $\gamma$ is 0.15 when $n$ is 8. Figs. 1 and 2 show the convergence of RDSGD algorithm when the network is unstable. It can be observed that DPSGD can achieve the best accuracy and the training loss becomes larger as the increase of $\tau$ whenever the number of workers is 4 or 8. When $\tau$ is less than 4, the loss of accuracy is not clearly reflected and when $\tau$ is between 8 and 16, the accuracy loss varies more. This can be explained by the experiment's lower number of iterations and the fact that only the worst case was considered. In the worst-case scenario, when $\tau$ is 16, all workers exchange information with their neighbors only 10 times in 160 epochs. But no matter the value of $\tau$ and $n$, the convergence speed of our algorithm is almost the same as DPSGD. We can find that when



(a) 4 workers

(b) 8 workers

**Fig. 2.** Effect of network instability on ResNet50.



(a) 4 workers

(b) 8 workers

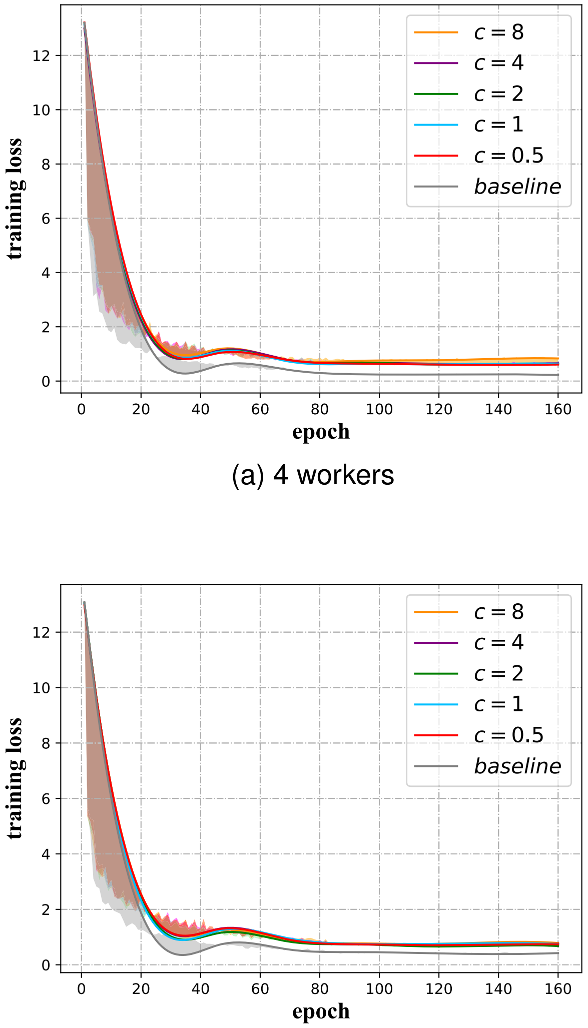
**Fig. 3.** Effect of noise on ResNet18.

the iteration reaches 80 epochs, the convergence speed of our algorithm slows down and when the iteration reaches 120 epochs, training loss hardly drops. The impact of $\tau$ on the performance of our algorithm just follows our theoretical analysis in Section 4.

**Effect of noise** For this case, we study the cases where $c$ is 0.5, 1, 2, 4 and 8 respectively under the condition that $\tau = 2$. Our baseline is DPSGD under a stable network connection and without injecting noise. Specifically we set $c = 1$ to represent RDSGD under the condition that $\tau = 2$ and without injecting noise. Figs. 3 and 4 show the performance of DPSGD and our algorithm trained on ResNet18 and ResNet50 when $n$ is 4 and 8. It can be observed that our algorithm is robust to communication noise and the effect of noise degrades as the number of workers is increased. Moreover, it also diminishes the effect of noise when training with a complex model.

**Effect of network topology** The network topology affects the efficiency and complexity of communication. When the network topology is more complex, the more workers communicate with each other and the higher the complexity of communication. We mainly explore the convergence of the algorithm under three network topologies: ring, complete and general topology. Each worker can only communicate with two neighbors in a ring network, and communicate with all other workers in a complete network. In a general network, any two workers can form an edge with a probability of 0.5, which means that any worker can communicate with half of the workers on expectation. We explore the convergence of the algorithm for three different network topologies with two models and different numbers of workers, and the experimental results are shown in Figs. 5 and 6. It can be observed that our algorithm has the same convergence rate in all experimental settings for all three network topologies. However, our algorithm can converge to the smallest training loss value in the general network topology, and the algorithm has the worst training loss value in the full topology. This implies that our algorithm achieves better convergence even when communication complexity is reduced.

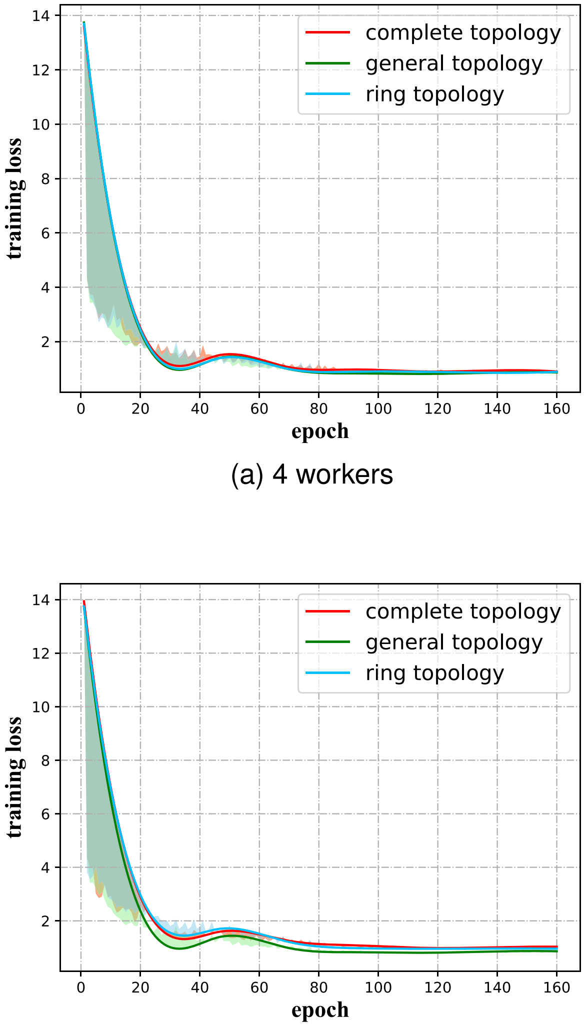
**Extend Discussion** We have constructed a network connection instability situation where a worker has to reconnect within $\tau$ steps when it goes offline. But for easier programming, we just validate the worst scenario where all workers reconnect after $\tau$ step when it goes offline. The exciting thing is that our algorithm has a satisfactory performance with little loss of accuracy even in the worst case. Therefore, if considering that the offline of the worker is an accidental event and each worker has high robustness and can be quickly reconnected, there is reason to believe that our algorithm can maintain the same convergence rate and achieve better accuracy.



(a) 4 workers

(b) 8 workers

Fig. 4. Effect of noise on ResNet50.



(a) 4 workers

(b) 8 workers

Fig. 5. Effect of network topology on ResNet18.

## 7. Conclusion

In this paper, we study a non-convex distributed optimization problem over unstable networks. We mainly considered unstable network connections and noise, and characterized unstable network connections as message delays and noise as data dimension deviations. We proposed a decentralized SGD algorithm capable of tolerating these unstable factors including unstable connectivity networks, communication noise and artificially injected noise. Through theoretical analysis, we demonstrate that our algorithm achieves the same convergence rate as traditional decentralized algorithms under relaxed assumptions. We apply our algorithm to an image classification task showing that our algorithm achieves comparable training accuracy with standard algorithms under stable networks.

Our work has shed some light on devising decentralized learning tolerating unstable network factors. It deserves more efforts to study whether there are still efficient decentralized learning algorithms under some other faults, such as Byzantine faults and network change caused by unstable nodes. In a wider consideration, any synchronous algorithm model can be introduced information delays and bias to study the convergence performance of the algorithm. Furthermore, it is urgently necessary to explore methods with improved accuracy in such unstable networks.

## CRediT authorship contribution statement

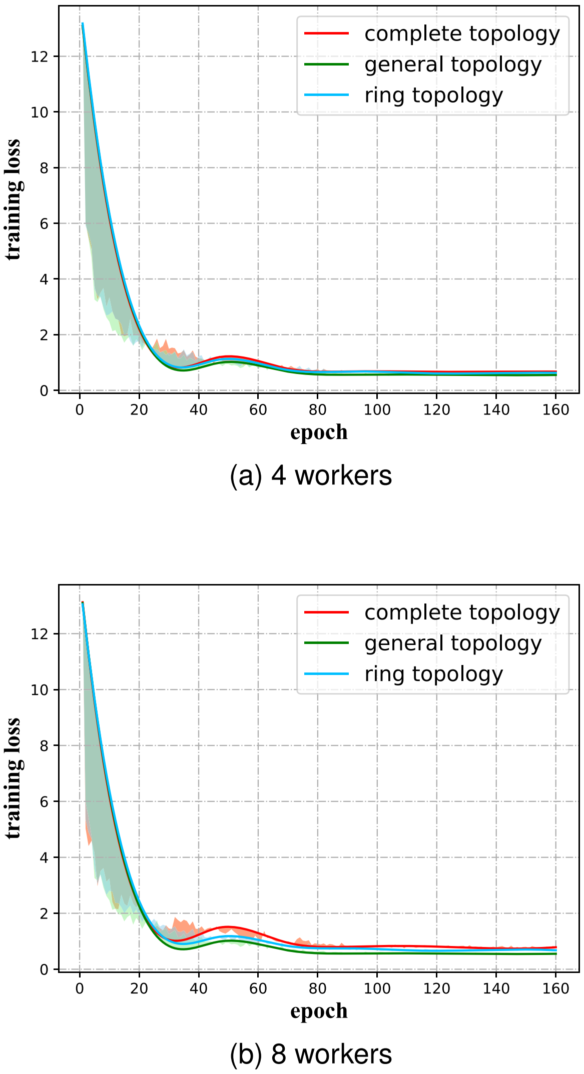
**Yanwei Zheng:** Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – original draft. **Liangxu Zhang:** Data curation, Writing – original draft, Software. **Shuzhen Chen:** Visualization, Investigation. **Xiao Zhang:** Resources, Supervision. **Zhipeng Cai:** Software, Validation. **Xiuzhen Cheng:** Supervision, Writing – review & editing.

## Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Xiao Zhang reports financial support was provided by the National Natural Science Foundation of China.

## Data availability

The dataset is public available.



(a) 4 workers

(b) 8 workers

**Fig. 6.** Effect of network topology on ResNet50.

## Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant No. 2022YFF0712100, in part by the National Natural Science Foundation of China under Grant 62202273, in part by Shandong Provincial Natural Science Foundation of China under Grant ZR2021QF044.

## References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, X. Zhang, TensorFlow: A System for Large-Scale Machine Learning, USENIX Association, 2016.
[2] P. Goyal, P. Dollár, R.B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, Accurate, large minibatch SGD: training ImageNet in 1 hour, 2017, CoRR, arXiv:1706.02677.
[3] Y. You, J. Li, S.J. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C. Hsieh, Large batch optimization for deep learning: Training BERT in 76 minutes, in: 8th International Conference on Learning Representations, 2020.
[4] M. Li, D.G. Andersen, A.J. Smola, K. Yu, Communication efficient distributed machine learning with the parameter server, in: Advances in Neural Information Processing Systems, vol. 27, 2014.
[5] S. Alqahtani, M. Demirbas, Performance analysis and comparison of distributed machine learning systems, 2019, CoRR, arXiv:1909.02061.
[6] B. Recht, C. Ré, S.J. Wright, F. Niu, Hogwild: A lock-free approach to parallelizing stochastic gradient descent, in: Advances in Neural Information Processing Systems, 2011, pp. 693–701.
[7] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, J. Liu, Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent, in: Advances in Neural Information Processing Systems, vol. 30, 2017.
[8] A. Nedic, A. Olshevsky, M.G. Rabbat, Network topology and communication-computation tradeoffs in decentralized optimization, Proc. IEEE 106 (5) (2018) 953–976.
[9] M. Assran, N. Loizou, N. Ballas, M.G. Rabbat, Stochastic gradient push for distributed deep learning, in: Proceedings of the 36th International Conference on Machine Learning, vol. 97, 2019, pp. 344–353.
[10] M. Seif, W.-T. Chang, R. Tandon, Privacy amplification for federated learning via user sampling and wireless aggregation, in: 2021 IEEE International Symposium on Information Theory, ISIT, 2021, pp. 2732–2737.
[11] H. Tang, S. Gan, C. Zhang, T. Zhang, J. Liu, Communication compression for decentralized training, in: Advances in Neural Information Processing System, 2018, pp. 7663–7673.
[12] A. Koloskova, S. Stich, M. Jaggi, Decentralized stochastic optimization and gossip algorithms with compressed communication, in: Proceedings of the 36th International Conference on Machine Learning, vol. 97, 2019, pp. 3478–3487.
[13] A. Koloskova, T. Lin, S.U. Stich, M. Jaggi, Decentralized deep learning with arbitrary communication compression, in: 8th International Conference on Learning Representations, 2020.
[14] H. Cheng, P. Yu, H. Hu, F. Yan, S. Li, H. Li, Y. Chen, LEASGD: an efficient and privacy-preserving decentralized algorithm for distributed learning, 2018, CoRR, arXiv:1811.11124.
[15] H. Cheng, P. Yu, H. Hu, S. Zawad, F. Yan, S. Li, H.H. Li, Y. Chen, Towards decentralized deep learning with differential privacy, in: Cloud Computing, vol. 11513, 2019, pp. 130–145.
[16] J. Xu, W. Zhang, F. Wang, A $(DP)^2$SGD: Asynchronous decentralized parallel stochastic gradient descent with differential privacy, IEEE Trans. Pattern Anal. Mach. Intell. (2021) 1.
[17] C. Yu, H. Tang, C. Renggli, S. Kassing, A. Singla, D. Alistarh, C. Zhang, J. Liu, Distributed learning over unreliable networks, in: Proceedings of the 36th International Conference on Machine Learning, vol. 97, 2019, pp. 7202–7212.
[18] L. Su, On the convergence rate of average consensus and distributed optimization over unreliable networks, in: Asilomar Conference on Signals, Systems, and Computers, 2018, pp. 43–47.
[19] B. Sirb, X. Ye, Decentralized consensus algorithm with delayed and stochastic gradients, SIAM J. Optim. 28 (2) (2018) 1232–1254.
[20] S.L. Smith, E. Elsen, S. De, On the generalization benefit of noise in stochastic gradient descent, in: Proceedings of the 37th International Conference on Machine Learning, vol. 119, 2020, pp. 9058–9067.
[21] H. Yu, Z. Chen, X. Zhang, X. Chen, F. Zhuang, H. Xiong, X. Cheng, FedHAR: Semi-supervised online learning for personalized federated human activity recognition, IEEE Trans. Mob. Comput. (2021).
[22] J. George, P. Gurram, Distributed stochastic gradient descent with event-triggered communication, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, 2020, pp. 7169–7178.
[23] A.T. Suresh, F.X. Yu, S. Kumar, H.B. McMahan, Distributed mean estimation with limited communication, in: Proceedings of the 34th International Conference on Machine Learning, vol. 70, 2017, pp. 3329–3337.
[24] J. Wangni, J. Wang, J. Liu, T. Zhang, Gradient sparsification for communication-efficient distributed optimization, in: Advances in Neural Information Processing Systems, 2018, pp. 1306–1316.
[25] S.P. Boyd, A. Ghosh, B. Prabhakar, D. Shah, Randomized gossip algorithms, IEEE Trans. Inf. Theory 52 (6) (2006) 2508–2530.
[26] A. Nedic, A. Olshevsky, Distributed optimization over time-varying directed graphs, IEEE Trans. Automat. Control 60 (3) (2015) 601–615.
[27] Z. He, J. He, C. Chen, X. Guan, Constrained distributed nonconvex optimization over time-varying directed graphs, in: IEEE Conference on Decision and Control, IEEE, 2020, pp. 378–383.
[28] Z. Chen, D. Chen, X. Zhang, Z. Yuan, X. Cheng, Learning graph structures with transformer for multivariate time series anomaly detection in iot, IEEE Internet Things J. (2021).
[29] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, vol. 54, 2017, pp. 1273–1282.
[30] A. Agarwal, J.C. Duchi, Distributed delayed stochastic optimization, in: Advances in Neural Information Processing Systems, 2011, pp. 873–881.
[31] S. Sra, A.W. Yu, M. Li, A. Smola, AdaDelay: Delay adaptive distributed stochastic convex optimization, in: Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, vol. 51, 2016, pp. 957–965.
[32] W. Zhang, S. Gupta, X. Lian, J. Liu, Staleness-aware async-SGD for distributed deep learning, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 2350–2356.
[33] L. Adilova, N. Paul, P. Schlicht, Introducing noise in decentralized training of neural networks, in: ECML PKDD 2018 Workshops, 2019, pp. 37–48.
[34] A. Spiridonoff, A. Olshevsky, I.C. Paschalidis, Robust asynchronous stochastic gradient-push: Asymptotically optimal and network-independent performance for strongly convex functions, J. Mach. Learn. Res. 21 (1) (2022).

[35] A. Fallah, M. Gurbuzbalaban, A.E. Ozdaglar, U. Simsekli, L. Zhu, Robust distributed accelerated stochastic gradient methods for multi-agent networks, 2019, ArXiv, arXiv:1910.08701.

[36] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Found. Trends Theor. Comput. Sci. 9 (3–4) (2014) 211–407.

[37] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Handb. Syst. Autoimmune Dis. 1 (4) (2009).

[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[39] A. Awan, R.A. Ferreira, S. Jagannathan, A. Grama, Distributed uniform sampling in unstructured peer-to-peer networks, in: 39th Hawaii International International Conference on Systems Science, 2006.

**Yanwei Zheng** received the Ph.D. degree in 2019 from the School of Computer Science and Engineering, Beihang University, Beijing. He is currently an Associate Professor with the School of Computer Science and Technology, Shandong University, Qingdao. His research interests include visual navigation and computer vision.

**Liangxu Zhang** received the B.S. degree in 2020 from the School of Computer Science and Technology, Shandong University. He is currently pursuing the M.S. degree in the school of Computer Science and Technology, Shandong University. His research interests include distributed computing, wireless and mobile security.

**Shuzhen Chen** received the B.S. degree in 2019 from the School of Computer Science and Technology, Shandong University. She is currently pursuing the Ph.D. degree in the school of Computer Science and Technology, Shandong University. Her research interests include distributed computing, wireless and mobile security.

**Xiao Zhang** received his B.S. and Ph.D degree from Central South University and Nanjing University, China, respectively. He is now an assistant professor in the School of Computer Science and Technology, Shandong University. Dr. Zhang's research interests include data mining, intelligent sensing, multi-task learning and federated learning.

**Zhipeng Cai** is currently an Associate Professor in the Department of Computer Science at Georgia State University, USA. He received his PhD and M.S. degrees in the Department of Computing Science at University of Alberta, and B.S. degree from Beijing Institute of Technology. Dr. Cai's research areas focus on Wireless Networking, Internet of Things, Machine Learning, Cyber-Security, Privacy and Big data. Dr. Cai is the recipient of an NSF CAREER Award. He served as a Steering Committee Co-Chair and a Steering Committee Member for WASA and IPCCC. Dr. Cai also served as a Technical Program Committee Member for more than 20 conferences, including INFOCOM, MOBIHOC, ICDE, and ICDCS. Dr. Cai has been serving as an Associate Editor-in-Chief for Elsevier High-Confidence Computing Journal (HCC), and an Associate Editor for several international journals, such as IEEE Internet of Things Journal (IoT-J), IEEE Transactions on Knowledge and Data Engineering (TKDE), and IEEE Transactions on Vehicular Technology (TVT). He has published more than 70 papers in prestigious journals with more than 40 papers published in IEEE/ACM Transactions.

**Xiuzhen Cheng** received her M.S. and Ph.D. degrees in computer science from the University of Minnesota—Twin Cities in 2000 and 2002, respectively. She is a professor of Computer Science at Shandong University, PR China. Her current research focuses on Blockchain computing, privacy-aware computing, and wireless and mobile security. She served/is serving on the editorial boards of several technical journals and the technical program committees of various professional conferences/workshops. She was a faculty member in the Department of Computer Science at The George Washington University from September 2002 to August 2020, and worked as a program director for the US National Science Foundation (NSF) from April to October in 2006 (full time) and from April 2008 to May 2010 (part time). She received the NSF CAREER Award in 2004. She is a member of ACM, and a Fellow of IEEE.