Private and Communication-Efficient Edge Learning: A Sparse Differential Gaussian-Masking Distributed SGD Approach

Xin Zhang[*+] Minghong Fang[*] Jia Liu[*] Zhengyuan Zhu[+]
[*]Department of Computer Science, Iowa State University
[+]Department of Statistics, Iowa State University
Ames, IA 50011, U.S.A.

ABSTRACT

With rise of machine learning (ML) and the proliferation of smart mobile devices, recent years have witnessed a surge of interest in performing ML in wireless edge networks. In this paper, we consider the problem of jointly improving data privacy and communication efficiency of distributed edge learning, both of which are critical performance metrics in wireless edge network computing. Toward this end, we propose a new decentralized stochastic gradient method with sparse differential Gaussian-masked stochastic gradients (SDM-DSGD) for non-convex distributed edge learning. Our main contributions are three-fold: i) We propose a generalized differential-coded DSGD update, which enable a much lower transmit probability for gradient sparsification, and provide an $\tilde{O}(1/\sqrt{NT})$ convergence rate; ii) We theoretically establish the privacy and communication efficiency performance guarantee for our SDM-DSGD method, which outperforms all existing works; and iii) We reveal theoretical insights and offer practical design guidelines for the interactions between privacy preservation and communication efficiency, two conflicting performance goals. We conduct extensive experiments with a variety of learning models on MNIST and CIFAR-10 datasets to verify our theoretical findings. Collectively, our results contribute to the theory and algorithm design for distributed edge learning.

CCS CONCEPTS

• **Computing methodologies → Distributed algorithms**; *Machine learning*; • **Security and privacy**; • **Networks** → *Network performance analysis*;

KEYWORDS

Edge network computing, distributed learning optimization, communication efficiency, differential privacy.

ACM Reference Format:
Xin Zhang[*+] Minghong Fang[*] Jia Liu[*] Zhengyuan Zhu[+]. 2018. Private and Communication-Efficient Edge Learning: A Sparse Differential Gaussian-Masking Distributed SGD Approach. In *MobiHoc '20: ACM Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, June 30 – July 03, 2020, Shanghai, China.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

In recent years, advances in machine learning (ML) have enabled many new and emerging applications that transform human lives. Traditionally, the training of of ML applications often rely on cloud-based data-centers to collect and process vast amount of data. With the proliferation of smart mobile devices and IoT (Internet-of-Things), data and requests for ML are increasingly being generated by devices from wireless edge networks. Due to high latency, low bandwidth, and privacy concerns[1], collecting all data to the cloud for processing may no longer be feasible or desirable. Therefore, the hope of "ML at the wireless edge" ("edge ML" for short) is to retain data in wireless edge networks and perform ML training *distributively* across end-user devices and edge servers (or called edge clouds). By doing so, one could potentially improve edge ML training performance while ensuring user privacy.

However, the successful deployment of edge ML faces significant technical hurdles. During the execution of distributed edge ML algorithms, each node in the network needs to exchange information with its local neighbors, which often injects intensive communication load into the network. This problem is further exacerbated by the inherent capacity constraints of wireless channels (due to channel fading, interference, etc.) and edge devices (due to limits in transmitter power, receiver sensitivity, etc.). Moreover, merely keeping data at edge devices does *not* ensure privacy: the released local messages that are exchanged over the air during each iteration of the algorithm still allow adversaries to infer the local sensitive data [1, 3, 9, 28]. Indeed, existing ML algorithms have not been designed for the edge environments. Hence, there is a pressing need for a fundamental understanding on how to design distributed algorithms to ensure *both* communication-efficient and privacy-preserving edge ML under severe communication and privacy constraints.

Unfortunately, the design of private and communication-efficient edge ML training faces two inherently *conflicting* challenges: i) On one hand, to preserve users' privacy while ensuring training convergence, one often needs to inject certain *unbiased i.i.d.* (independent and identically distributed) random noise into the exchanged information. However, the privacy guarantee with i.i.d. noise typically degrades with the number of training iterations T [1, 9, 22, 29]. Hence, for a given privacy constraint, there exists a maximum

[1] Data privacy concerns in designing decentralized learning algorithms have long been raised in the literature (see, e.g., [3, 9, 28]). This is because in many applications (e.g., healthcare, finance, recommendation systems, etc.), ML models are usually trained by data over distributed systems that contain users' privacy information.

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

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

value of T that a distributed edge ML training algorithm can run; ii) On the other hand, to design communication-efficient distributed edge ML algorithms, it is necessary to perform gradient sparsification and/or compression (see, e.g., [18, 20, 21, 25]). However, as we show later, these operations induce a *different* type of random noise, which increases the the value of T for achieving some desired training loss while providing *no* privacy guarantee. It is highly challenging to reconcile these two conflicting types of randomness in distributed edge ML algorithmic design.

To date, results on edge ML algorithmic designs that are both private and communication-efficient remain rather limited. Most of the existing work focus on either communication efficiency [17, 21, 30] or data privacy [4, 13]. For the limited amount of work that considered both, they are either only restricted to the server/worker architecture or having unsatisfactory performances and high implementation complexity (see Section 2 for more in-depth discussion). The above limitations of the existing work motivate us to propose a new decentralized stochastic gradient descent (DSGD) method with sparse differential Gaussian-masked stochastic gradients. For convenience, we refer to this method as SDM-DSGD. Our SDM-DSGD method addresses the aforementioned technical challenges and offer significantly improved privacy and communication efficiency performances. Our main results and their significance are summarized as follows:

- We propose a SDM-DSGD method for non-convex distributed edge ML training, which is differentially-private, communication-efficient, and applicable for general network topologies. We show that, with the properly chosen parameters, our SDM-DSGD algorithm is (ϵ, δ)-differentially private (DP) and enjoys an $\tilde{O}(1/\sqrt{NT})$ convergence rate, where N is the number of nodes in the network and T is the final iteration index of the algorithm. Moreover, we show that the maximum value of T scales as $O(m^4)$, where m is the size of local dataset at each node[2].
- It is also worth pointing out that our SDM-DSGD is a *generalized* differential-coded DSGD approach in the sense that: *All* existing differential-coded DSGD algorithms can be seen as a special case of our SDM-DSGD (e.g., [21]). Specifically, our key updating step in SDM-DSGD is a linear combination of the current state and the standard DSGD update. Remarkably, with this generalized updating step, one can perform gradient sparsification with a much lower transmit probability p, which implies significantly improved privacy. This also relaxes the restricted constraint in [21] on finding a "valid" p. We also note that, thanks to this new algorithmic structure, the non-private version of our algorithm (i.e., no Gaussian-masking) may be of independent interest.
- Based on our theoretical results from SDM-DSGD, we go one-step further to investigate the interactions between i) gradient differential sparsification and ii) Gaussian masking, which are the two key components responsible for communication efficiency and privacy in SDM-DSGD, respectively. Toward this end, we compare an alternative design that also has the same two components but with their order being reversed. Our analysis shows that the proposed SDM-DSGD scheme is superior and can reduce the privacy budget by a p^2-fraction. This insight deepens our understanding on these two components and offers algorithmic design guidelines in practice.
- Lastly, we conduct extensive experiments to examine the performance of our SDM-DSGD algorithm with a variety of deep learning models on MNIST and CIFAR-10 datasets. Our experiments show that the accuracy of SDM-DSGD outperforms two state-of-the-art decentralized learning algorithms [12, 21] under the same communication cost and privacy budget. These experiments corroborate our theoretical results.

Collectively, our results in this paper contribute to the state of the art of theories and algorithm design for communication-efficient and privacy-preserving decentralized learning. The rest of the paper is organized as follows. In Section 3, we will review necessary background for our algorithm design and analysis. In Section 4, we introduce our SDM-DSGD algorithm and then analyze its performances in privacy and convergence. Numerical results are provided in Section 5. In Section 6, we provide concluding remarks.

2 RELATED WORK

As mentioned in Section 1, results on private and communication-efficient distributed learning algorithms remain quite limited in the literature. For example, to achieve both communication efficiency and differential privacy, Agarwal *et al.* [2] proposed the cpSGD algorithm based on the randomized quantization and Binomial masking. It is shown both theoretically and experimentally that Binomial masking achieves nearly the same utility as Gaussian masking, while the communication cost is significantly reduced. However, this work mainly focused on the distributed mean estimation (DME) problem under the server/worker architecture. It remains unclear how to implement the cpSGD algorithm to train general deep learning models in networks with general communication topologies. Another related work is [3], where Cheng *et al.* proposed a new decentralized algorithm named leader-follower elastic averaging stochastic gradient descent (LEASGD). In the LEASGD algorithm, the computation nodes are dynamically categorized into two pools: leader pool with nodes of lower loss values and follower pool with nodes of higher loss. In each iteration, the leader nodes will pair with followers to guide the followers in the right direction. Gaussian masking was adopted in the communication step to protect the data privacy. Although this work numerically showed LEASGD's communication efficiency, the implementation complexity of LEASGD is high due to the categorization and lead-follower pairing in each iteration. Also, the theoretical performance of LEASGD is unclear under the non-convex cases. In contrast to these existing work, in this paper we propose a communication-efficient and privacy-preserving distributed training algorithm named SDM-DSGD, for distributed nonconvex learning. Our SDM-DSGD algorithm can be viewed as a variant of state-of-the-art decentralized learning algorithm DSGD [12] by introducing the randomized sparification and the Gaussian mechanism.

3 DIFFERENTIAL PRIVACY AND GRADIENT SPARSIFICATION: A PRIMER

To facilitate subsequent technical discussions on privacy and communication efficiency, in this section, we provide the necessary background on differential privacy and gradient sparsification.

[2] Our algorithms can straightfowardly be extended to cases with datasets having unbalanced sizes, i.e., $m_{n_1} \neq m_{n_2}$ for $n_1 \neq n_2$.

1) Differential Privacy: Differential privacy (DP) [5, 7] is a canonical privacy metric for the privacy-preserving data analysis. Under the DP framework, privacy is defined and measured by how noticeable the distribution of the outcome of some query mechanism changes when only one sample in the dataset is changed:

Definition 1 ((ϵ, δ)-Differential Privacy [5]). *Two datasets $\mathcal{D}$ and $\mathcal{D}'$ are said to be adjacent if and only if they differ by only one element. Given two adjacent N-element datasets $\mathcal{D}$ and $\mathcal{D}' \in \mathscr{D}^N$ and $\epsilon, \delta > 0$, a randomized query mechanism $\mathcal{M} : \mathscr{D}^N \rightarrow \mathbb{R}^d$ is called (ϵ, δ)-differentially-private ((ϵ, δ)-DP) if and only if for any measurable set $E \in \mathbb{R}^d$, the output of $\mathcal{M}$ satisfies $\mathbb{P}(\mathcal{M}(\mathcal{D}) \in E) \leq e^{\epsilon}\mathbb{P}(\mathcal{M}(\mathcal{D}') \in E) + \delta$.*

In the literature, two popular approaches to achieve DP are the so-called Gaussian and Laplacian masking mechanisms, both of which share the same form: $\mathcal{M}(\mathcal{D}) = q(\mathcal{D}) + \eta$, where $q(\cdot)$ is a query function and η is the injected Gaussian or Laplacian masking noise. In our work, we focus on the Gaussian masking mechanism.

2) Sparsification: In the literature, sparsification, also known as sparse compression, is a commonly used compression technique for compressing gradients to design communication-efficient distributed learning algorithms [20, 21, 25]. The key idea of sparsification is to apply the the following Bernoulli randomized operation to sparsify a high-dimensional vector:

Definition 2 (Sparsifier [25]). *For any vector $\mathbf{x} = [x_1, \cdots, x_d]^\top \in \mathbb{R}^d$ and a constant $p \in [0, 1)$, $S(\mathbf{x})$ outputs a sparse vector with the i-th element $[S(\mathbf{x})]_i$ following the Bernoulli(p) distribution:*

$$\begin{cases} \Pr([S(\mathbf{x})]_i = \frac{\mathbf{x}_i}{p}) = p, \\ \Pr([S(\mathbf{x})]_i = 0) = 1 - p. \end{cases}$$

We can see that the sparsifier operation randomly selects some coordinates and sets the information in these coordinates to zero. Also, it follows immediately from Definition 2 that the sparsification operation is *unbiased* and the introduced variance depends on the magnitude of input vector (we omit the proof due to its simplicity):

Lemma 1. *For any $\mathbf{x} \in \mathbb{R}^d$, the random output $S(\mathbf{x})$ satisfies: 1) unbiased expectation: $\mathbb{E}\big(S(\mathbf{x})\big) = \mathbf{x}$; and 2) input-dependent variance: $Var\big(S(\mathbf{x})\big) = (1/p - 1)\|\mathbf{x}\|_2$.*

It is worth noting that, although sparsification is a random mechanism, it does *not* provide any privacy guarantee in terms of DP: Given a dataset $\mathcal{D}$ and a query function $q(\cdot)$, there exists an adjacent dataset $\mathcal{D}'$ such that $q(\mathcal{D}) \neq q(\mathcal{D}')$. Under the sparsification with probability p, consider the event $E = \{q(\mathcal{D})/p\}$. Then, for dataset $\mathcal{D}$, to have the output $q(\mathcal{D})/p$, it requires that all the coordinates need to be selected. Thus, it holds that $\Pr(S(q(\mathcal{D}) \in E) = p^d > 0$, where d is the dimension. However, for dataset $\mathcal{D}'$, because $q(\mathcal{D}) \neq q(\mathcal{D}')$, it is impossible to have an output as $q(\mathcal{D})/p$, which implies $\Pr(S(q(\mathcal{D}') \in E) = 0$. Thus, it is impossible to find valid ϵ and $\delta \in [0, 1)$ to satisfy the DP inequality in Definition 1. Interestingly, although sparsification does not offer DP, we will show it later that by sparsifying part of the original information, the sparsifier operation does help to improve the privacy protection performance.

4 A SPARSE DIFFERENTIAL GAUSSIAN-MASKING SGD APPROACH

In this section, we first present the problem formulation of edge ML training in Section 4.1. Then, we will present our SDG-DSGD algorithm in Section 4.2 and its main theoretical results in Section 4.3.

4.1 Problem Formulation of Edge ML Training

In this paper, we use an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ to represent a wireless edge network with *general* network topology, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links, respectively, with number of nodes as $|\mathcal{N}| = n$. We let $\mathbf{x} \in \mathbb{R}^d$ denote a global decision vector to be learned or estimated. In edge ML training, we want to distributively solve an unconstrained optimization problem: $\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}; \mathcal{D})$, and $f(\mathbf{x}; \mathcal{D})$ can be decomposed node-wise as follows[3]:

$$\min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}; \mathcal{D}) = \min_{\mathbf{x} \in \mathbb{R}^d} \sum_{i=1}^{n} f(\mathbf{x}; \mathcal{D}_i), \tag{1}$$

where $f(\mathbf{x}; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{z} \in \mathcal{D}} f(\mathbf{x}; \mathbf{z})$ for any dataset $\mathcal{D}$. Here, each local objective function $f(\mathbf{x}; \mathcal{D}_i)$ is only observable to node i. It is easy to see that Problem (1) can be equivalently reformulated as the following *consensus form*:

$$\min \sum_{i=1}^{n} f(\mathbf{x}_i; \mathcal{D}_i), \;\; s.t. \mathbf{x}_i = \mathbf{x}_j, \;\; \forall (i, j) \in \mathcal{L}. \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^d$ is the local copy of $\mathbf{x}$ at node i. The constraints in Problem (2) guarantee that the all local copies are equal to each other, hence the name consensus form.

4.2 The SDM-DSGD Algorithm

As the name suggests, our proposed SDM-DSGD method is inspired by the classical decentralized gradient descent (DGD) algorithm [12, 16, 26], which is one of the most effective approaches for distributively solving network consensus optimization problems. The DGD framework is built upon the notion of *consensus matrix*, which is denoted as $\mathbf{W} \in \mathbb{R}^{n \times n}$ in this paper. Specifically, in each iteration of DGD, each node in the network performs an update that integrates a local (stochastic) gradient step and a weighted average from its neighbors' parameters based on $\mathbf{W}$. Mathematically, $\mathbf{W}$ satisfies the following properties:

1) *Doubly Stochastic:* $\sum_{i=1}^{n} [\mathbf{W}]_{ij} = \sum_{j=1}^{N} [\mathbf{W}]_{ij} = 1$.
2) *Symmetric:* $[\mathbf{W}]_{ij} = [\mathbf{W}]_{ji}, \forall i, j \in \mathcal{N}$.
3) *Network-Defined Sparsity Pattern:* $[\mathbf{W}]_{ij} > 0$ if $(i, j) \in \mathcal{L}$ and $[\mathbf{W}]_{ij} = 0$ otherwise, $\forall i, j \in \mathcal{N}$.

Properties 1)–3) imply that the spectrum of $\mathbf{W}$ (i.e., the set of all eigenvalues) lies in the interval $(-1, 1]$ with exactly one eigenvalue being equal to 1. Also, all eigenvalues being real implies that they can be sorted as $-1 < \lambda_n(\mathbf{W}) \leq \cdots \leq \lambda_1(\mathbf{W}) = 1$. To facilitate later discussions, we let $\beta \triangleq \max\{|\lambda_2(\mathbf{W})|, |\lambda_n(\mathbf{W})|\} \in (0, 1)$, i.e., the second-largest eigenvalue of $\mathbf{W}$ in magnitude. The use of the consensus matrix is due to the fact that $(\mathbf{W} \otimes \mathbf{I}_P)\mathbf{x} = \mathbf{x}$ *if and only* if $\mathbf{x}_i = \mathbf{x}_j$, $(i, j) \in \mathcal{L}$[16], where $\mathbf{x} = [\mathbf{x}_1^\top, \ldots, \mathbf{x}_n^\top]^\top$ and $\otimes$ represents the Kronecker product. Therefore, Problem (2) can be reformulated as $\min_{\mathbf{x} \in \mathbb{R}^D} \sum_{i=1}^{N} f_i(\mathbf{x}_i)$, s.t. $(\mathbf{W} \otimes \mathbf{I}_P)\mathbf{x} = \mathbf{x}$, which further leads to

[3] Here we assume the datasets are balanced

the original DGD algorithmic design[16]. With the notion of $\mathbf{W}$, our SDM-DSGD algorithm can be stated as follows:

Algorithm 1: A Sparse Differential Gaussian-Masking Distributed Stochastic Gradient Descent (SDM-DSGD) Algorithm.

```
Initialization:
  1. Set the initial state $\mathbf{x}_{i,0} = \mathbf{y}_{i,0} = \mathbf{d}_{i,0} = \mathbf{0}$, $\forall i$, and $t = 1$.
Main Loop:
  2. In the $t$-th iteration, each node sends the sparsified differential $S(\mathbf{d}_{i,t})$ to its neighbors, where $S(\cdot)$ is the sparsifier operation. Also, upon collecting all neighbors' information, each node $i \in \mathcal{N}$ updates the following local values:
     a) Reconstruct node $i$'s neighbors inexact copies: $\mathbf{x}_{j,t} = \mathbf{x}_{j,t-1} + S(\mathbf{d}_{j,t-1})$, $\forall j \in \mathcal{N}_i$;
     b) Update local copy: $\mathbf{y}_{i,t} = (1-\theta)\mathbf{x}_{i,t} + \theta\big(\sum_{j \in \mathcal{N}_i} [\mathbf{W}]_{ij}\mathbf{x}_{j,t} - \gamma\big(\nabla f(\mathbf{x}_{i,t};\zeta_{i,t}) + \eta_{i,t}\big)\big)$, where $\nabla f(\cdot;\zeta_{i,t})$ is the stochastic gradient, and $\eta_{i,t} \sim N(0, \sigma^2 \mathbf{I}_d)$ is a Gaussian random noise.
     c) Compute the local differential: $\mathbf{d}_{i,t} = \mathbf{y}_{i,t} - \mathbf{x}_{i,t}$.
  3. Stop if some convergence criterion is met; otherwise, let $t \leftarrow t + 1$ and go to Step 2.
```

Remark 1. Algorithm 1 is motivated by and bears some similarity with the DGD-type communication-efficient distributed learning in the literature [21, 30]. In these existing work, rather than exchanging the states directly, the compressed differentials between two successive iterations of the variables are communicated to reduce the communication load. By contrast, Algorithm 1 differs from these existing work in the following key aspects: i) As noted in Section 1, the update in Step 2.b) in SDM-DSGD generalizes the existing work by using a *linear combination* of the current state and the DSGD update. It was shown that when using the sparsification in the proposed algorithm in [21], the transmit probability p in Definition 2 is required to be greater than $4(1-\lambda_n)^2/(4(1-\lambda_n)^2 + (1-|\lambda_n|)^2)$, where λ_n is the smallest eigenvalue of $\mathbf{W}$. In contrast, our generalized framework allows a much smaller p in the sparsification, i.e., significantly better communication-efficiency. ii) In addition to performance gains in terms of communication-efficiency, as will be shown later, our generalized algorithm also improves the convergence speed from $O(T^{-1/3})$ to $\tilde{O}(T^{-1/2})$ as long as $p = \Omega(1/\log(T))$.

Before we state our main theoretical results, it is insightful to offer some intuitions on how our SDM-DSGD method is derived. Toward this end, we rewrite the update rule of in SDM-DSGD algorithm in the following vector form:

$$\begin{cases} \mathbf{x}_t = \mathbf{x}_{t-1} + S(\mathbf{d}_{t-1}), \\ \mathbf{y}_t = (1-\theta)\mathbf{x}_t + \theta\Big(\tilde{\mathbf{W}}\mathbf{x}_t - \gamma\big(\nabla \mathbf{f}(\mathbf{x}_t;\zeta_t) + \boldsymbol{\eta}_t\big)\Big), \\ \mathbf{d}_t = \mathbf{y}_t - \mathbf{x}_t, \end{cases} \tag{3}$$

where $\tilde{\mathbf{W}} \triangleq \mathbf{W} \otimes \mathbf{I}_n$ and $\mathbf{f}(\mathbf{x}_t;\zeta_t) = \sum_{i=1}^n f(\mathbf{x}_i;\zeta_{i,t})$. Define a Lyapunov function $V_\gamma(\mathbf{x};\mathcal{D}) \triangleq \frac{1}{2}\mathbf{x}^\top(\mathbf{I} - \tilde{\mathbf{W}})\mathbf{x} + \sum_{i=1}^n f(\mathbf{x}_i;\mathcal{D}_i)$, and its stochastic version $V_\gamma(\mathbf{x};\zeta) \triangleq \frac{1}{2}\mathbf{x}^\top(\mathbf{I} - \tilde{\mathbf{W}})\mathbf{x} + \sum_{i=1}^n f(\mathbf{x}_i;\zeta_i)$, where $\mathbf{x} = [\mathbf{x}_1^\top, \cdots, \mathbf{x}_n^\top]^\top \in \mathbb{R}^{nd}$ and the random sample batch $\zeta = \{\zeta_i\}_{i=1}^n$. It can be readily verified that:

$$\begin{cases} \mathbf{y}_t = \mathbf{x}_t - \theta(\nabla V_\gamma(\mathbf{x}_t;\zeta_t) + \gamma\boldsymbol{\eta}_t), \\ \mathbf{d}_t = -\theta(\nabla V_\gamma(\mathbf{x}_t;\zeta_t) + \gamma\boldsymbol{\eta}_t), \end{cases} \tag{4}$$

which implies that the iterates update can be written as:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + S(\mathbf{d}_t) = \mathbf{x}_t - \theta\nabla V_\gamma(\mathbf{x}_t;\zeta_t) - \theta\gamma\boldsymbol{\eta}_t + \boldsymbol{\epsilon}_t, \tag{5}$$

where $\boldsymbol{\epsilon}_t$ represents the noise from the sparsifier. Therefore, the iterative update rule in SDM-DSGD can be viewed as applying the stochastic gradient descent algorithm on the Lyapunov function $V_\gamma(\mathbf{x};\mathcal{D})$ with two additional noises $\theta\gamma\boldsymbol{\eta}_t$ and $\boldsymbol{\epsilon}_t$, one from the privacy protection and the other one from the sparse compression.

4.3 Main Theoretical Results

In this subsection, we will establish the privacy and convergence properties of the proposed SDM-DSGD method. For better readability, we state the main theorems and their key insights in this subsection and relegate the proofs of the main theorems to the appendices. We start with stating the following assumptions:

Assumption 1. *The global objective function $f(\cdot)$ satisfies:*

1. *Given dataset $\mathcal{D} = \{\mathcal{D}_i\}_{i=1}^n$, $f(\mathbf{x};\mathcal{D})$ is bounded from below, i.e., $\exists \mathbf{x}^*_{\mathcal{D}} \in \mathbb{R}^d$, such that $f(\mathbf{x};\mathcal{D}) \geq f(\mathbf{x}^*_{\mathcal{D}};\mathcal{D})$, $\forall \mathbf{x} \in \mathbb{R}^d$;*
2. *The function $f(\mathbf{x};z)$ is continuously differentiable and has L-Lipschitz continuous gradient, i.e., there exists a constant $L > 0$ such that $|\nabla f(\mathbf{x};z) - \nabla f(\mathbf{y};z)| \leq L\|\mathbf{x} - \mathbf{y}\|_2$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d$;*
3. *The stochastic gradient is unbiased and has bounded variance with respective to the local dataset, i.e. $\mathbb{E}_{z\sim\mathcal{D}_i}[\nabla f(\mathbf{x};z)] = f(\mathbf{x};\mathcal{D}_i)$ and $Var_{z\sim\mathcal{D}_i}[\nabla f(\mathbf{x};z)] \leq \tilde{\sigma}^2$;*
4. *The function $f(\mathbf{x};z)$ is coordinate-wise $G/\sqrt{d}$-smooth, i.e., for all coordinates k, $|[\nabla f(\mathbf{x};z)]_k| \leq G/\sqrt{d}$.*

The first three assumptions are standard for the convergence analysis of stochastic algorithms [8, 11, 12]. The last assumption characterizes the sensitivity of objective function with respect to $\mathbf{x}$ in each coordinate. Note that it also implies the ℓ_2-sensitivity bound $\|\nabla f(\mathbf{x};z)\| \leq G$, which is useful for differential privacy (see Defintion 3 in Appendix 1 and [9, 22]).

1) Privacy Analysis: In our SDM-DSGD algorithm, in the t-th iteration, each node releases the local information $S(\mathbf{d}_t)$, which is generated by first applying Gaussian masking on the local stochastic gradient $g(x_{i,t};\zeta_{i,t})$ and then the sparsifier operation on the local differential d_t. Consider the latent vectors $\omega_{i,t} \sim \text{Bin}(d,p)$ and $\omega_t = [\omega_{1,t}^\top, \cdots, \omega_{n,t}^\top]^\top$, where $\text{Bin}(d,p)$ is the binomial distribution with d trials and success probability p. Each coordinate of $\omega_{i,t}$ denotes whether the information is transmitted or not. Note that the generation of ω_t does not depend on data. Define the active set $C_{1,i,t} = \{k : [\omega_{i,t}]_k = 1\}$ and inactive set $C_{0,i,t} = \{k : [\omega_{i,t}]_k = 0\}$, and $C_{1,t} = \cup_i C_{1,i,t}$, $C_{0,t} = \cup_i C_{0,i,t}$. Then, only active coordinates are released. We perform this coordinate decomposition on $S(\mathbf{d}_t)$ since the adversaries can only infer sensitive data from the coordinates in $C_{1,t}$, while the coordinates in $C_{0,t}$ are private. In what follows, we compare our mechanism with two existing privacy protection techniques:

1) Difference from randomized response mechanism: In our sparsifier, the binomial vector $\mathbf{z}$ is used to determine the active and inactive sets. A similar mechanism is the randomized response (RR) mechanism [7, 24]. Our mechanism differs from RR as follows: i) The RR mechanism is a binary-response query, e.g., {Yes, No}, while our query is sampling from $\mathbb{R}^d$; ii) The RR mechanism is designed for the proportion estimation over several queries. However, in each iteration, only one query is available for the sparsifier.

2) Difference from sparse vector technique: In our privacy protection mechanism, due to the sparsity of the released vectors, part of information is protected. This is similar to the idea of the sparse vector technique (SVT) [6, 14, 19]. In SVT, a threshold value is chosen so that only the first c queries above the threshold will be released. For privacy protection, two randomized procedures are used in SVT by adding noises on the threshold and queries. The key difference between our method and SVT are: i) SVT is designed to select c important queries (i.e. coordinates). However, in our algorithm, the sparsity of the released vector is random based on transmitted probability p. ii) In our method, the transmitted coordinates are amplified by a $(1/p)$-factor to ensure that the released vector is unbiased. In contrast, it can be verified that the released vector in SVT is biased due to the lack of such an amplifying operation.

Theorem 1 (Privacy Guarantee). *Choose the variance of added Gaussian noise* $\boldsymbol{\eta}$ *as* $\sigma^2 \geq 1/1.25$. *Under Assumption 1, for any* $\delta \in (0,1)$, *the execution of SDM-DSGD algorithm with* T *iterations is* $(4\alpha \sum_{t=1}^{T} |\overline{C_{1,t}}|(\tau G/\sqrt{d}m\sigma)^2 + \epsilon/2, \delta)$*-differentially-private, where* $\alpha = 2\log(1/\delta)/\epsilon + 1$, $|\overline{C_{1,t}}| = \max_i\{|C_{1,i,t}|\}$ *and* τ *is the subsampling rate for SGD.*

Remark 2. The lower bound $\sigma^2 \geq 1/1.25$ follows from [23] to guarantee the privacy amplification under the subsampling. Theorem 1 shows that the sparsifier improves the differential privacy guarantee by a factor $\sum_{t=1}^{T} |\overline{C_{1,t}}|/dT \leq 1$, which depends on p. Hence, the smaller the value of p, the less information will be communicated, and the better privacy protection. Meanwhile, it can be seen that the privacy loss increases as the iteration number T gets large, which is expected. This is because, with fewer iterations, less information will be released, which implies a better privacy protection. However, fewer iterations cause a larger training loss in edge ML. This leads to a *training-privacy trade-off*, which we will further analyze later. Also, by inverting Theorem 1, we have the following result:

Corollary 2. *Under the same conditions in Theorem 1 and let the subsampling rate be* $1/m$ *(i.e., each node subsamples one out of* m *data), if the variance of added Gaussian noise* $\boldsymbol{\eta}$ *is chosen as*

$$\sigma^2 \geq \max\{\frac{8\sum_{t=1}^{T} |\overline{C_{1,t}}| G^2(2\log(1/\delta)+\epsilon)}{d\epsilon^2 m^4}, \frac{1}{1.25}\}, \qquad (6)$$

then given the total iteration number T, *the SDM-DSGD algorithm is* (ϵ,δ)*-DP for any* $\delta \in (0,1)$.

2) Convergence Analysis for Training Loss: As shown in Eq. (5), instead of directly optimizing $f(x;\mathcal{D})$, our SDM-DSGD algorithm can be viewed as applying stochastic gradient descent on the Lyapunov function $V_\gamma(\mathbf{x};\mathcal{D})$. However, besides the random sampling noise, we also have the noises from the Gaussian masking and sparsification. Note that the compression noise $\boldsymbol{\epsilon}$ is *dependent* on the sampling and added Gaussian masking noises, which significantly complicates our convergence analysis. In what follows, we first quantify the optimization error of the sum output $\bar{x}_T = \sum_{i=1}^{n} x_{i,T}$.

Lemma 1 (Convergence). *Under Assumption 1, fixing the variance of added Gaussian noise* σ^2 *to be a constant, starting from* $x_{i,0} = 0\ \forall i$, *and setting* $\theta < 2p/(1-\lambda_n+\gamma L) \in (0,1)$, *the iterates* $x_{i,t}$ *generated by Eq. (5) satisfy:*

$$\min_{t\in\{0,\cdots,T-1\}} \|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 \leq \text{(I)} + \text{(II)} + \text{(III)} + \text{(IV)}, \qquad (7)$$

where $\text{(I)} = \frac{2C_1}{\theta\gamma T}$, $\text{(II)} = \frac{2LC_3}{n}(\frac{\gamma}{1-\beta})^2$, $\text{(III)} = \frac{2\theta\gamma^2 LC_2}{n(1-\beta)}(\frac{1}{p}-1) + \frac{L\theta\gamma C_2}{n^2 p}$, *and* $\text{(IV)} = (\frac{2\gamma L}{n(1-\beta)} + \frac{L}{n^2})(\frac{1}{p}-1)\Big[\frac{2pnC_1}{\big(2p-(1-\lambda_n+\gamma L)\theta\big)T} + \frac{(1-\lambda_n+\gamma L)\theta^2\gamma C_2}{2p-(1-\lambda_n+\gamma L)\theta}\Big]$. *In the above terms,* $C_1 \triangleq f(0;\mathcal{D}) - f(x^*_{\mathcal{D}};\mathcal{D})$, $C_2 \triangleq n\tilde{\sigma}^2/m\tau + nd\sigma^2$ *and* $C_3 \triangleq (nG)^2 + (nd\sigma)^2$ *are constants, and* $\tilde{\sigma}^2$ *is the variance of the stochastic gradients.*

Remark 3. There are four terms in the convergence error of SDM-DSGD in Eq. (7): (I) is the common convergence error that goes to zero as T and step-size $\theta\gamma$ increase; (II) is the approximation error between the Lyapunov function $V_\gamma(\mathbf{x};\mathcal{D})$ and $\mathbf{f}(\mathbf{x};\mathcal{D})$, which decreases with γ. These two terms are similar to those in the convergence of DGD-based algorithms [27, 30]; (III) and (IV) are the error terms introduced by the compression, random sampling, as well as the Gaussian masking noises. The following simplified convergence rate result follows immediately from Lemma 1.

Corollary 3. *Fixing the variance of Gaussian masking noise* σ^2 *to be a constant, setting* $\theta = \min\{p/(1-\lambda_n+\gamma L), p/2\}$, $\gamma = c\sqrt{n\log(T)/T}$, *and* $p \gg 1/\log(T)$, *where* c *is a constant, if the number of iterations satisfies* $T > n^5/(1-\beta)^4$, *then it holds that:*

$$\min_{t\in\{0,\cdots,T-1\}} \|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 = O\Big(\sqrt{\frac{\log(T)}{nT}}\Big). \qquad (8)$$

Remark 4. Several remarks for Corollary 3 are in order: 1) The parameter θ is used to adjust the variance introduced by the sparsifier and could be set to a constant over the iterations, while the step-size γ is required to be diminishing to control the sampling and Gaussian masking noises, as well as the approximation error; 2) The convergence rate is $O(\sqrt{\log(T)/nT}) = \tilde{O}(1/\sqrt{nT})$, which is approximately the same as the result in [12] (ignoring the logarithm factor); 3) In the standard DSGD algorithm [12], to reach ε-accuracy, the communication complexity is $O(d/\varepsilon^2)$. In contrast, the communication complexity of our algorithm is $O(1/\varepsilon^2 \log(1/\varepsilon^3))$ by letting $p = 1/d$. Thus, our algorithm outperforms DSGD in overparameterized regime (i.e., large d); 4) The lower bound of p is $1/\log(T)$. For example, with 10^4 training iterations (a common setting for many deep learning training), the lower bound is approximately 0.1, which is a small value; 5) The result in Corollary 3 is based on two conditions: i) σ^2 is fixed over all iterations; and ii) the number of iterations T is sufficiently large.

Finally, by putting all aforementioned theoretical results together, we have the following key result for training-privacy trade-off:

Theorem 4 (Training-Privacy Trade-off). *Under Assumption 1, let* $\sigma^2 = 8TG^2(2\log(1/\delta)+\epsilon)/m^4\epsilon^2$, $\theta = \min\{p/(1-\lambda_n+\gamma L), p/2\}$, *and* $\gamma = c\sqrt{n\log(T)/T}$, *where* c *is a constant. If* $T = m^4\epsilon^2/20G^2\log(1/\delta) = O(m^4)$, *then the SDM-DSGD algorithm is*

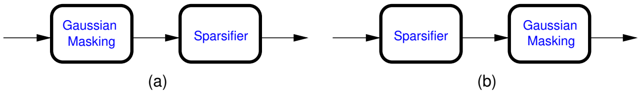


Figure 1: (a) SDM-DSGD scheme; (b) The alternative design.

(ϵ,δ)-DP and the convergence rate is

$$\min_{t\in\{0,\cdots,T-1\}} \|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 = \tilde{O}\Big(\frac{\sqrt{20G^2\log(1/\delta)}}{\sqrt{n}m^2\epsilon}\Big). \tag{9}$$

Remark 5. Note that by letting $T = m^4\epsilon^2/20G^2\log(1/\delta) = O(m^4)$, we have that $\sigma^2 > 1/1.25$ over all iterations. Note also that the local sample size m is usually much larger than the number of nodes, i.e., $m \gg n$, which implies that $m^4\epsilon^2/20G^2\log(1/\delta) > n^5/(1-\beta)^4$. Hence, the convergence speed improvement still holds with $T = m^4\epsilon^2/20G^2\log(1/\delta)$.

3) Insights and Guidelines for Privacy and Communication Efficiency Co-Design: Note that in our algorithm, the Gaussian masking is applied before the sparse compression (see Figure 1 (a). Thus, a fundamental and interesting question arises: *Is this a good design?* To answer this question, consider the alternative design that *reverses* the Gaussian masking and sparsifier operations: we first perform sparsify operation on the local differential, and then apply Gaussian masking on those non-zero coordinates of the compressed local differential (see Figure 1 (b)). This alternative design can be mathematically written as:

$$\begin{cases} \mathbf{x}_t = \mathbf{x}_{t-1} + \underbrace{\big(S(\mathbf{d}_{t-1}) + \theta\gamma\tilde{\boldsymbol{\eta}}_t\big)}_{\text{released message}}, \\ \mathbf{d}_t = (1-\theta)\mathbf{x}_t + \theta\Big(\tilde{\mathbf{W}}\mathbf{x}_t - \gamma\big(\mathbf{g}(x_t;\zeta_t)\big)\Big) - \mathbf{x}_t, \end{cases} \tag{10}$$

where $[\tilde{\boldsymbol{\eta}}_t]_{C_{1,t}} \sim N(0,\sigma^2\mathbf{I})$ and $[\tilde{\boldsymbol{\eta}}_t]_{C_{0,t}} = \mathbf{0}$; and the factor $\theta\gamma$ before $\tilde{\boldsymbol{\eta}}_t$ is to make the result comparable. For this alternative design, we have the following result:

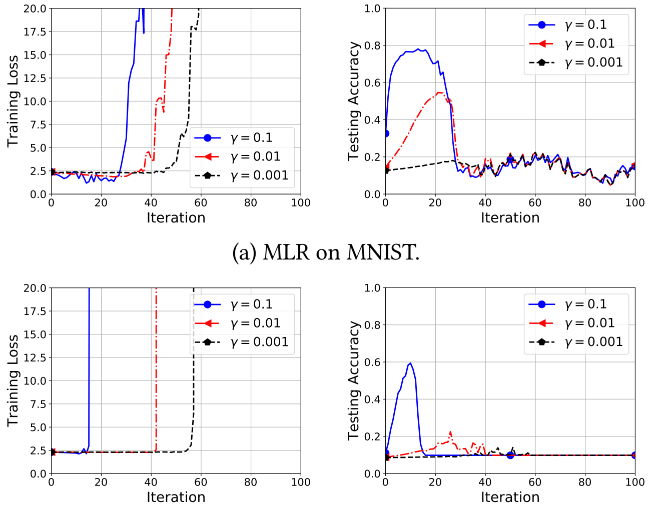
Proposition 5. *Let the variance of Gaussian masking noise $\boldsymbol{\eta}$ be chosen as $\sigma^2 \geq 1/1.25$. Under Assumption 1, for any $\delta \in (0,1)$, the alternative design in (10) is $\big(4\alpha\sum_{t=1}^T|\overline{C_{1,t}}|(\tau G)^2/dm^2\sigma^2p^2+\epsilon/2,\delta\big)$-DP, where $\alpha \triangleq 2\log(1/\delta)/\epsilon - 1$, p is the transmit probability of the sparsifier, and τ is the subsampling rate.*

We can see that, in the "ϵ-part," the DP performance of our SDM-DSGD is smaller than that of the alternative design by a $(1/p^2)$-factor, and so our SDM-DSGD design is superior. This difference is because in the alternative design, the sparse compression amplifies the ℓ_2-sensitivity of the query by $1/p^2$.

5 EXPERIMENTAL EVALUATION

In this section, we present experimental results of several nonconvex machine learning problems to evaluate the performance of our method. In particular, we compare the communication cost and privacy-accuracy trade-off with two state-of-art algorithms:

- Decentralized SGD (DSGD)[10, 16, 26]: Each node updates its local parameter as $x_{i,t+1} = \sum_{j\in\mathcal{N}_i}[\mathbf{W}]_{ij}x_{j,t} - \gamma g(x_{i,t};\zeta_{i,t})$ with stochastic gradient $g(x_{i,t};\zeta_{i,t})$ of random sample $\zeta_{i,t}$, and exchange the uncompressed local parameter x_i with its neighbors.



(a) MLR on MNIST.

(b) CNN on MNIST.

Figure 2: Two examples where DC-DSGD diverges: $p = 0.2$ and the step-size γ is chosen from $\{0.1, 0.01, 0.001\}$.

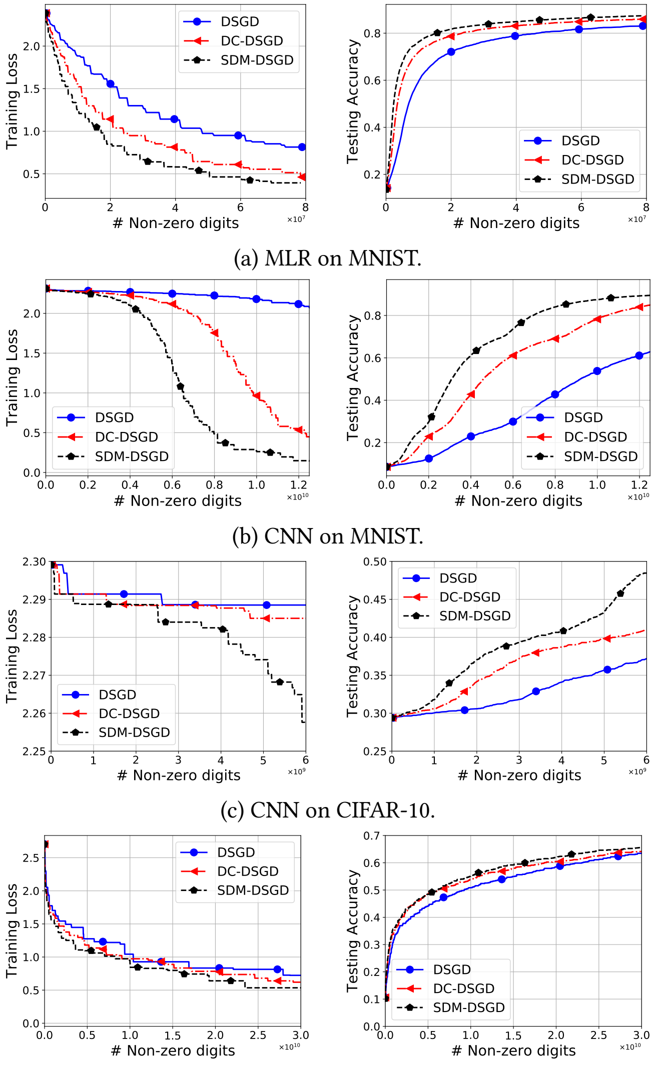
- Differential Compressed Decentralized SGD (DC-DSGD) [21]: This algorithm also communicates compressed local differentials and estimating neighbors' copies. However, in the local copy updating step, DC-DSDG does not have tuning parameter θ (or can be viewed as fixing $\theta = 1$ in our SDM-DSGD).

In our experiment, we choose the transmit probability p from $\{1, 0.5, 0.2\}$. We set $\theta = 1$ for $p = 1$ and 0.5, which is corresponding to DSGD and DC-DSGD, respectively. However, for $p = 0.2$, the algorithm does not converge if we choose $\theta = 1$, i.e., DC-DSGD fails under $p = 0.2$ (See Figure 2). Thus, we set $\theta = 0.6$ for the case when $p = 0.2$, which is corresponding to our algorithm.

Dataset and Learning Models: We adopt all three algorithms to solve a variety of nonconvex learning problems over MNIST and CIFAR-10 datasets. For MNIST, we apply the multi-class logistic regression (MLR) and convolutional neural network (CNN) classifiers. The adopted CNN model has two convolutional layers (size $3\times3\times16$), each is followed by a max-pooling layer with size 2×2, and then a fully connected layer. The ReLU activation is used for the two convolutional layers and the "softmax" activation is applied at the output layer. For CIFAR-10, we apply the above CNN model and the ResNet20 model. The batch size is 64 for both MLR and the CNN classifiers on MNIST. The batch size is 128 and 32 for the CNN and ResNet20 classifiers on CIFAR-10, respectively.

Network Model: We use a network with 50 nodes. Similar to [17, 18], the communication graph $\mathcal{G}$ is generated by the Erdös-Rènyi graph with edge connectivity $p_c = 0.35$. The network concensus matrix is chosen as $\mathbf{W} = \mathbf{I} - \frac{2}{3\lambda_{\max}(\mathbf{L})}\mathbf{L}$, where $\mathbf{L}$ is the Laplacian matrix of $\mathcal{G}$, and $\lambda_{\max}(\mathbf{L})$ denotes the largest eigenvalue of $\mathbf{L}$.

Procedure for Privacy: Note that privacy protection is not considered in the original DSGD and DC-DSGD methods. To have a fair comparison, we add the same Gaussian noise $N(\mathbf{0},\mathbf{I})$ to the stochastic gradients, so that all algorithms are privacy-preserving. To control the object function's ℓ_2-sensitivity to $\mathbf{x}$, we adopt a modified gradient clipping technique [1]: $\text{Clip}([g]_i) = \text{sign}([g]_i)\max\{|[g]_i|, C\}$,



(a) MLR on MNIST.

(b) CNN on MNIST.

(c) CNN on CIFAR-10.

(d) ResNet20 on CIFAR-10.

Figure 3: Results of objective loss (left) and testing accuracy (right) with models trained by different algorithms.

$\forall g \in \mathbb{R}^d$. With this clipping, each coordinate of the gradient is bounded by C in magnitude. Here, we set $C = 5$. In our experiment, we keep track of the privacy loss based on Theorem 1.

Numerical Results: We illustrate the results of training loss and testing accuracy with respect to communication costs in Figure 3. We compute the total non-zero digits (i.e. the non-sparsified digits) communicated in the each iteration, which is used to measure the communication cost in the training. In the left-hand-side figures in Figure 3, we show the training loss vs. the amount of the non-zero digits. In the right-hand-side figures in Figure 3, we show the testing accuracy vs the amount of the non-zero digits. We can see that under the same amount of non-zero digits, our SDM-DSGD has the fastest convergence speed and the best testing accuracy: in the case of training CNN on MNIST, with 6×10^9 non-zero digits, SDM-DSGD's testing accuracy is at 80%, while those of DC-DSGD and DSGD are 60% and less than 40%, respectively.

Table 1: The results of testing accuracy with models trained by different algorithms with $(\epsilon, \delta = 10^{-5})$-DP guarantee.

MLR on MNIST			
$\epsilon(\times 10^{-3})$	1.0	2.0	5.0
DSGD	0.1422	0.1956	0.6324
DC-DSGD	0.1621	0.2959	0.7408
SDM-DSGD	**0.1880**	**0.4296**	**0.7810**
CNN on MNIST			
$\epsilon(\times 10^{-2})$	2.0	5.0	10.0
DSGD	0.0886	0.1059	0.2570
DC-DSGD	0.0917	0.1442	0.5265
SDM-DSGD	**0.0960**	**0.2150**	**0.6728**
CNN on CIFAR-10			
$\epsilon(\times 10^{-2})$	5.0	10.0	20.0
DSGD	0.2960	0.3036	0.3544
DC-DSGD	0.2991	0.3292	0.3964
SDM-DSGD	**0.3013**	**0.3570**	**0.4296**
ResNet20 on CIFAR-10			
$\epsilon(\times 10^{-2})$	2.0	5.0	10.0
DSGD	0.3265	0.4631	0.5735
DC-DSGD	0.3324	0.4922	0.5957
SDM-DSGD	**0.3470**	**0.5079**	**0.6099**

For privacy loss, we summarize the results in Table 1. We can see that under the same δ, the testing accuracy is increasing as the privacy budget gets large (i.e., large ϵ): in the case of MLR on MNIST, with ϵ increasing from 1×10^3 to 5×10^3, testing accuracy is increasing from 19% to 78%. This is because with larger privacy budget, more information is allowed to be released, which leads to a better training. Meanwhile, under the same privacy budget ϵ, our SDM-DSGD consistently has a higher accuracy compared with the other two algorithms: our algorithm improves the accuracy at least 4% with MLR and 15% with CNN on MNIST.

6 CONCLUSION

In this paper, we proposed the SDM-DSGD algorithm to improve both data privacy and communication efficiency in distributed edge learning. In our SDM-DSGD algorithm, we proposed to exchange the sparsified differentials between the computation nodes and develop a "generalized" computing scheme for local updates. We theoretically and numerically showed that by doing so, the proposed algorithm converges with a small sparsification probability p compared with the state-of-the-art DC-DSGD method. Moreover, we considered the protection data privacy by injecting a Gaussian masking noise. We studied the interaction between the sparsification and the Gaussian masking mechanism and showed that the "randomize-then-sparisify" is the preferred approach. Finally, we established the privacy-accuracy tradeoff theoretically. Through extensive experiments, we showed that SDM-DSGD outperforms existing algorithms. Our results advance the state-of-the-art of communication efficiency and data privacy in distributed edge learning.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications*

Security, pages 308–318. ACM, 2016.

[2] Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pages 7564–7575, 2018.

[3] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. Leasgd: an efficient and privacy-preserving decentralized algorithm for distributed learning. *arXiv preprint arXiv:1811.11124*, 2018.

[4] Tie Ding, Shanying Zhu, Jianping He, Cailian Chen, and Xinping Guan. Consensus-based distributed optimization in multi-agent systems: Convergence and differential privacy. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3409–3414. IEEE, 2018.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[6] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 381–390. ACM, 2009.

[7] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

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

[9] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, pages 6343–6354, 2018.

[10] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*, pages 5904–5914, 2017.

[11] Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 2737–2745, 2015.

[12] Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

[13] Yang Liu, Ji Liu, and Tamer Basar. Differentially private gossip gradient descent. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 2777–2782. IEEE, 2018.

[14] Min Lyu, Dong Su, and Ninghui Li. Understanding the sparse vector technique for differential privacy. *Proceedings of the VLDB Endowment*, 10(6):637–648, 2017.

[15] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

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

[17] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. An exact quantized decentralized gradient descent algorithm. *IEEE Transactions on Signal Processing*, 67(19):4934–4947, 2019.

[18] Amirhossein Reisizadeh, Hossein Taheri, Aryan Mokhtari, Hamed Hassani, and Ramtin Pedarsani. Robust and communication-efficient collaborative learning. *arXiv preprint arXiv:1907.10595*, 2019.

[19] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774. ACM, 2010.

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

[21] Hanlin Tang, Shaoduo Gan, Ce Zhang, Tong Zhang, and Ji Liu. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*, pages 7652–7662, 2018.

[22] Bao Wang, Quanquan Gu, March Boedihardjo, Farzin Barekat, and Stanley J Osher. Dp-lssgd: A stochastic optimization method to lift the utility in privacy-preserving erm. 2019.

[23] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. *arXiv preprint arXiv:1808.00087*, 2018.

[24] Yue Wang, Xintao Wu, and Donghui Hu. Using randomized response for differential privacy preserving data collection. In *EDBT/ICDT Workshops*, volume 1558, 2016.

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

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

[27] Jinshan Zeng and Wotao Yin. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11):2834–2848, 2018.

[28] Chunlei Zhang, Muaz Ahmad, and Yongqiang Wang. Admm based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, 14(3):565–580, 2018.

[29] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. *arXiv preprint arXiv:1703.09947*, 2017.

[30] Xin Zhang, Jia Liu, Zhengyuan Zhu, and Elizabeth S Bentley. Compressed distributed gradient descent: Communication-efficient consensus over networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2431–2439. IEEE, 2019.

A PROOF OF MAIN RESULTS

A.1 Proof of Theorem 1

Proof. To prove the privacy guarantee, we first states the following definitions and related lemmas.

Definition 3 (ℓ_2-Sensitivity). *The ℓ_2-Sensitivity is defined as the maximum change in the ℓ_2-Norm of the function value $f(\cdot)$ on two adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$:*

$$\Delta(f) = \max_{\mathcal{D},\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2, \tag{11}$$

where $\mathcal{D}$ and $\mathcal{D}'$ are two adjacent datasets if only one element are different.

Definition 4 (Rényi Differential Privacy (RDP) [15]). *A randomized mechanism $\mathcal{M} : \mathscr{D}^N \to \mathbb{R}^d$ is (α, ρ)-Rényi Differential Privacy $\big((\alpha, \rho)$-RDP$\big)$, if for any two adjacent dataset $\mathcal{D}$, $\mathcal{D}' \in \mathscr{D}^N$, it holds that*

$$D_\alpha(\mathcal{M}(\mathcal{D})\|\mathcal{M}(\mathcal{D}')) = \frac{1}{\alpha - 1} \log \mathbb{E}\Big(\frac{\mathcal{M}(\mathcal{D})}{\mathcal{M}(\mathcal{D}')}\Big)^{\alpha} \le \rho, \tag{12}$$

where the expectation is taken over $\mathcal{M}(\mathcal{D}')$. $D_\alpha(\cdot\|\cdot)$ is also known as Rényi divergence.

Lemma 2 (RDP of Gaussian Mechanism [23]). *Consider the mechanism $\mathcal{M} = f(\mathcal{D}) + \boldsymbol{\eta}$, with the function $f : \mathscr{D} \to \mathbf{R}^d$ and Gaussian noise $\boldsymbol{\eta} \sim N(0, \sigma^2\mathbf{I}_d)$.*

i). The mechanism $\mathcal{M}$ is $(\alpha, \alpha\Delta^2(f)/(2\sigma^2))$-RDP, where $\Delta(f)$ is the ℓ_2-Sensitivity of f;

ii). If the mechanism $\mathcal{M}$ is applied to a subset of samples using uniform sampling without replacement and $\sigma^2 \ge 1.1/25$, then $\mathcal{M}$ is $(\alpha, \alpha\tau^2\Delta^2(f)/\sigma^2)$-RDP, where τ is the sampling rate.

Lemma 3 (Sequential Composibility of RDP [15]). *Consider $f : \mathscr{D}^N \to \mathbb{R}^{d_1}$ and $g : \mathscr{D}^N \times \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$. If f is (α, ρ_1)-RDP and g is (α, ρ_2)-RDP, then the mechanism (X, Y) satisfies $(\alpha, \rho_1 + \rho_2)$-RDP, where $X \sim f(\mathcal{D})$ and $Y \sim g(X, \mathcal{D})$.*

Lemma 4 (From RDP to (ϵ, δ)-DP [15]). *If f is an (α, ρ)-RDP mechanism, then it also satisfies $\big(\rho + \log(1/\delta)/(\alpha - 1), \delta\big)$-DP for any $\delta \in (0, 1)$.*

With the above definition and lemmas, we provide the following privacy guarantee for our algorithm in the following. Our proof is inspired by [22]. Given the active set at tth iteration $C_{1,t}$, and $C_{1,i,t}$ respective to the ith node, the updating equation (5) can be rewritten as:

$$\begin{aligned}
[\mathbf{x}_{t+1}]_{C_{1,t}} &= [\mathbf{x}_t]_{C_{1,t}} - [(\theta\nabla V_\gamma(\mathbf{x}_t;\zeta_t) + \theta\gamma\boldsymbol{\eta}_t)/p]_{C_{1,t}} &(13)\\
&= [\mathbf{x}_t]_{C_{1,t}} - \theta\big([\nabla V_\gamma(\mathbf{x}_t;\zeta_t)]_{C_{1,t}} + \gamma[\boldsymbol{\eta}_t]_{C_{1,t}}\big)/p &(14)
\end{aligned}$$

Thus, we need to analyze the privacy gaurantee of the above SGD updating with noise $[\boldsymbol{\eta}_t]_{C_{1,t}}$, of which each coordinate is from

$N(0, \sigma^2)$. Given dataset $\mathcal{D}$, consider the following mechanism $\hat{\mathcal{M}}_t = [\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})]_{C_{1,t}} + \gamma[\boldsymbol{\eta}_t]_{C_{1,t}}$ with the query $\mathbf{q}_t = [\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})]_{C_{1,t}}$. With the adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, the ℓ_2-sensitivity of $\mathbf{q}_t$ is

$$\begin{aligned}
\Delta(\mathbf{q}_t) &= \|[\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})]_{C_{1,t}} - [\nabla V_\gamma(\mathbf{x}_t; \mathcal{D}')]_{C_{1,t}}\|_2 \\
&= \|[(\tilde{\mathbf{W}}\mathbf{x}_t - \gamma \nabla \mathbf{f}(\mathbf{x}_t; \mathcal{D}_i)) - (\tilde{\mathbf{W}}\mathbf{x}_t - \gamma \nabla \mathbf{f}(\mathbf{x}_t; \mathcal{D}_i'))]_{C_{1,t}}\|_2 \\
&= \gamma \sum_{i=1}^{n} \|[\nabla f(x_{i,t}; \mathcal{D}_i) - \nabla f(x_{i,t}; \mathcal{D}_i')]_{C_{1,i,t}}\|_2 \\
&\overset{(a)}{=} \frac{\gamma}{m} \|[\nabla f(x_{i,t}|\zeta_{i,j}) - \nabla f(x_{i,t}; \zeta_{i,j}')]_{C_{1,i,t}}\|_2 \\
&= \frac{\gamma}{m} \sqrt{\sum_{k \in C_{1,i,t}} ([\nabla f(x_{i,t}; \zeta_{i,j})]_k - [\nabla f(x_{i,t}; \zeta_{i,j}')]_k)^2} \\
&\overset{(b)}{\leq} \frac{2\gamma\sqrt{|C_{1,i,t}|}G}{\sqrt{d}m} \leq \frac{2\gamma\sqrt{|\overline{C_{1,t}}|}G}{\sqrt{d}m} \qquad (15)
\end{aligned}$$

where (a) by assuming that the only different data is $\zeta_{i,j}$ and $\zeta_{i,j}'$ in the ith node; (b) by the coordinate-wise $G/\sqrt{d}$-Lipschitz of the function $f(\cdot)$, and $|\overline{C_{1,t}}| = \max_i\{|C_{1,i,t}|\} \leq d$. Thus, based on Lemma 2 i), with $\boldsymbol{\eta}_t \sim N(0, \sigma^2 \mathbf{I}_{nd})$, the mechanism $\hat{\mathcal{M}}_t$ satisfies $\big(\alpha, 2\alpha|\overline{C_{1,t}}|(G/\sqrt{d}m\sigma)^2\big)$-RDP[4]. Then for the mechanism $\mathcal{M}_t = [\nabla V_\gamma(\mathbf{x}_t; \zeta_t)]_{C_{1,t}} + \gamma[\boldsymbol{\eta}_t]_{C_{1,t}}$, which is equivalent to applying $\hat{\mathcal{M}}_t$ to a subset of random sample $\boldsymbol{\zeta}_t$, according to Lemma 2 ii), $\mathcal{M}_t$ satisfies $\big(\alpha, 4\alpha|\overline{C_{1,t}}|(\tau G/\sqrt{d}m\sigma)^2\big)$-RDP with $\sigma^2 \geq 1/1.25$. So set $\alpha = 2\log(1/\delta)/\epsilon + 1$ and with Lemma 4, we have $\mathcal{M}_t$ satisfies $(4\alpha|\overline{C_{1,t}}|(\tau G/\sqrt{d}m\sigma)^2 + \epsilon/2, \delta)$-DP with $\sigma^2 \geq 1/1.25$.

Next, we derive the privacy guarantee over T iterations. By Lemma 3, with T iterations, i.e. sequentially composition of $\{\mathcal{M}_t\}_{t=1}^T$, the algorithm output $\mathbf{x}_T$ satisifies $\big(\alpha, \sum_{t=1}^T 4\alpha|\overline{C_{1,t}}|(\tau G/\sqrt{d}m\sigma)^2\big)$-RDP. With Lemma 4, $\mathbf{x}_T$ satisfies $(4\alpha\sum_{t=1}^T |\overline{C_{1,t}}|(\tau G/\sqrt{d}m\sigma)^2 + \epsilon/2, \delta)$-DP when $\sigma^2 \geq 1/1.25$. □

A.2 Proof of Proposition 5

Proof. Under this design, we have the updating:

$$[\mathbf{x}_{t+1}]_{C_{1,t}} = [\mathbf{x}_t]_{C_{1,t}} - [\theta\nabla V_\gamma(\mathbf{x}_t; \zeta_t)/p + \theta\gamma\tilde{\boldsymbol{\eta}}_t]_{C_{1,t}}. \qquad (16)$$

Consider the mechanism $\hat{\mathcal{M}}_t = [\theta\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})/p]_{C_{1,t}} + \theta\gamma[\tilde{\boldsymbol{\eta}}_t]_{C_{1,t}}$ with the query $\mathbf{q}_t = [\theta\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})/p]_{C_{1,t}}$. With the adjacent datasets $\mathcal{D}$ and $\mathcal{D}'$, we have the ℓ_2-sensitivity of $\mathbf{q}_t$ is

$$\begin{aligned}
\Delta(\mathbf{q}_t) &= \frac{\theta}{p}\|[\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})]_{C_{1,t}} - [\nabla V_\gamma(\mathbf{x}_t; \mathcal{D}')]_{C_{1,t}}\|_2 \\
&= \frac{\theta}{p}\|[(\tilde{\mathbf{W}}\mathbf{x}_t - \gamma \nabla \mathbf{f}(\mathbf{x}_t; \mathcal{D}_i)) - (\tilde{\mathbf{W}}\mathbf{x}_t - \gamma \nabla \mathbf{f}(\mathbf{x}_t; \mathcal{D}_i'))]_{C_{1,t}}\|_2 \\
&= \frac{\theta\gamma}{p}\| \sum_{i=1}^{n} [\nabla f(x_{i,t}; \mathcal{D}_i) - \nabla f(x_{i,t}; \mathcal{D}_i')]_{C_{1,i,t}}\|_2 \\
&\overset{(a)}{=} \frac{\theta\gamma}{mp} \|[\nabla f(x_{i,t}; \zeta_{i,j}) - \nabla f(x_{i,t}; \zeta_{i,j}')]_{C_{1,i,t}}\|_2 \\
&= \frac{\theta\gamma}{mp} \sqrt{\sum_{k \in C_{1,i,t}} ([\nabla f(x_{i,t}; \zeta_{i,j})]_k - [\nabla f(x_{i,t}; \zeta_{i,j}')]_k)^2} \\
&\overset{(b)}{\leq} \frac{2\theta\gamma\sqrt{|C_{1,i,t}|}G}{\sqrt{d}mp} \leq \frac{2\theta\gamma\sqrt{|\overline{C_{1,t}}|}G}{\sqrt{d}mp} \qquad (17)
\end{aligned}$$

where (a) by assuming that the only different data is $\zeta_{i,j}$ and $\zeta_{i,j}'$ in the ith node; (b) by the coordinate-wise $G/\sqrt{d}$-Lipschitz of the function $f(\cdot)$. Thus, based on Lemma 2 i), with $[\tilde{\boldsymbol{\eta}}_t]_{C_{1,t}} \sim N(0, \sigma^2\mathbf{I})$, the mechanism $\hat{\mathcal{M}}_t$ satisfies $\big(\alpha, 2\alpha|\overline{C_{1,t}}|(G/\sqrt{d}m\sigma p)^2\big)$-RDP. Define the mechanism $\mathcal{M}_t = [\theta\nabla V_\gamma(\mathbf{x}_t; \zeta_t)/p]_{C_{1,t}} + [\tilde{\boldsymbol{\eta}}_t]_{C_{1,t}}$. With the similar derivation, we have $\mathcal{M}_t$ satisfies $(4\alpha|\overline{C_{1,t}}|(\tau G)^2/dm^2\sigma^2p^2 + \epsilon/2, \delta)$-DP with $\sigma^2 \geq 1/1.25$. Hence with T iterations, the algorithm satisfies $(4\alpha\sum_{t=1}^T |\overline{C_{1,t}}|(\tau G)^2/dm^2\sigma^2p^2 + \epsilon/2, \delta)$-DP with $\sigma^2 \geq 1/1.25$. □

A.3 Proof of Theorem 1

Proof. First, we give the following useful lemma.

Lemma 5. *Under the same conditions in Lemma 1, at tth iteration, the random sparsified output $S(\mathbf{d}_t)$ has:*

i). First Moment: $\mathbb{E}[S(\mathbf{d}_t)|\mathbf{x}_t] = -\theta\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})$;

ii). Second Moment: $\mathbb{E}[\|S(\mathbf{d}_t)\|_2^2|\mathbf{x}_t] \leq \frac{\theta^2}{p}\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + \frac{(\theta\gamma)^2}{p}(\frac{n\tilde{\sigma}^2}{m\tau} + nd\sigma^2)$.

Proof. i). For the first moment,

$$\begin{aligned}
\mathbb{E}[S(\mathbf{d}_t)|\mathbf{x}_t] &= \mathbb{E}[\mathbb{E}[S(\mathbf{d}_t)|\mathbf{d}_t]|\mathbf{x}_t] = \mathbb{E}[\mathbf{d}_t|\mathbf{x}_t] \\
&= \mathbb{E}[-\theta(\nabla V_\gamma(\mathbf{x}_t; \zeta_t) + \gamma\boldsymbol{\eta}_t)|\mathbf{x}_t] \\
&= -\theta\nabla V_\gamma(\mathbf{x}_t; \mathcal{D}) \qquad (18)
\end{aligned}$$

ii). For the second moment,

$$\begin{aligned}
&\mathbb{E}[\|S(\mathbf{d}_t)\|_2^2|\mathbf{x}_t] = \|\mathbb{E}[S(\mathbf{d}_t)]|\mathbf{x}_t\|_2^2 + \mathrm{Var}[S(\mathbf{d}_t)|\mathbf{x}_t] \\
&\overset{(a)}{=} \theta^2\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + \mathbb{E}[\mathrm{Var}[S(\mathbf{d}_t)|\mathbf{d}_t]|\mathbf{x}_t] + \mathrm{Var}[\mathbb{E}[S(\mathbf{d}_t)|\mathbf{d}_t]|\mathbf{x}_t] \\
&\overset{(b)}{=} \theta^2\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + (\frac{1}{p} - 1)\mathbb{E}[\|\mathbf{d}_t\|_2^2|\mathbf{x}_t] + \mathrm{Var}[\mathbf{d}_t|\mathbf{x}_t] \\
&\overset{(c)}{\leq} \theta^2\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + (\frac{1}{p} - 1)\theta^2\mathbb{E}[\|\nabla V_\gamma(\mathbf{x}_t; \zeta_t) + \gamma\boldsymbol{\eta}_t\|_2^2|\mathbf{x}_t] \\
&+ (\theta\gamma)^2(\frac{n\tilde{\sigma}^2}{m\tau} + nd\sigma^2) \\
&\overset{(d)}{=} \theta^2\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + (\frac{1}{p} - 1)\theta^2\mathbb{E}[\|\nabla V_\gamma(\mathbf{x}_t; \zeta_t)\|_2^2 + \|\gamma\boldsymbol{\eta}_t\|_2^2|\mathbf{x}_t] \\
&+ (\theta\gamma)^2(n\frac{\tilde{\sigma}^2}{m\tau} + nd\sigma^2) \\
&\leq \theta^2\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + (\frac{1}{p} - 1)\theta^2[\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 \\
&+ \gamma^2(\frac{n\tilde{\sigma}^2}{mn\tau} + d\sigma^2)] + (\theta\gamma)^2(\frac{\tilde{\sigma}^2}{m\tau} + nd\sigma^2) \\
&= \frac{\theta^2}{p}\|\nabla V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + \frac{(\theta\gamma)^2}{p}(\frac{n\tilde{\sigma}^2}{m\tau} + nd\sigma^2) \qquad (19)
\end{aligned}$$

where (a) is by the Eve's law; (b) is from the properies of the sparsifier in Section 3; (c) is by $\mathbf{d}_t = -\theta(\nabla V_\gamma(\mathbf{x}_t; \zeta_t) + \gamma\boldsymbol{\eta}_t)$, at each node the subsampling rate is τ; (d) is because the randomness of the subsampling and the Gaussian mechanism are independent. □

Step 1: Define the filtration $\mathcal{F}_t = \sigma\langle\mathbf{x}_1, \cdots, \mathbf{x}_t\rangle$. Note that the Lyapunov function $V_\gamma(\mathbf{x}; \mathcal{D})$ has $(1 - \lambda_n + \gamma L)$-Lipschitz gradient.

[4] Note that the mechanism $\hat{\mathcal{M}}_t$ adds the Gaussian noise $\gamma[\boldsymbol{\eta}_t]_{C_{1,t}} \sim N(0, \gamma^2\sigma^2\mathbf{I}_{C_{1,i,t}})$.

Thus, we have

$$\begin{aligned}
&V_\gamma(\mathbf{x}_{t+1};\mathcal{D})\\
&\leq V_\gamma(\mathbf{x}_t;\mathcal{D}) + \langle \nabla V_\gamma(\mathbf{x}_t;\mathcal{D}), \mathbf{x}_{t+1}-\mathbf{x}_t\rangle + \frac{(1-\lambda_n+\gamma L)}{2}\|\mathbf{x}_{t+1}-\mathbf{x}_t\|^2\\
&= V_\gamma(\mathbf{x}_t;\mathcal{D}) + \langle \nabla V_\gamma(\mathbf{x}_t;\mathcal{D}), S(\mathbf{d}_t)\rangle + \frac{(1-\lambda_n+\gamma L)}{2}\|S(\mathbf{d}_t)\|^2
\end{aligned} \tag{20}$$

Take conditional expectation at both sides:

$$\begin{aligned}
&\mathbb{E}[V_\gamma(\mathbf{x}_{t+1})|\mathcal{F}_t]\\
&\leq V_\gamma(\mathbf{x}_t;\mathcal{D}) + \langle \nabla V_\gamma(\mathbf{x}_t;\mathcal{D}), \mathbb{E}[S(\mathbf{d}_t)|\mathcal{F}_t]\rangle + \frac{(1-\lambda_n+\gamma L)}{2}\mathbb{E}[\|S(\mathbf{d}_t)\|^2|\mathcal{F}_t]\\
&= V_\gamma(\mathbf{x}_t;\mathcal{D}) - \theta\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{(1-\lambda_n+\gamma L)}{2}[\frac{\theta^2}{p}\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 +\\
&\frac{(\theta\gamma)^2}{p}(\frac{n\tilde{\sigma}^2}{m\tau}+d\sigma^2)]\\
&\leq V_\gamma(\mathbf{x}_t) + \big(\frac{(1-\lambda_n+\gamma L)\theta^2}{2p}-\theta\big)\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{(1-\lambda_n+\gamma L)(\theta\gamma)^2}{2p}(\frac{n\tilde{\sigma}^2}{m\tau}+nd\sigma^2)
\end{aligned} \tag{21}$$

Thus, by setting $2p\theta-(1-\lambda_n+\gamma L)\theta^2>0$, i.e. $\theta<2p/(1-\lambda_n+\gamma L)$, we have the following descent inequality:

$$\begin{aligned}
&\big(2p\theta-(1-\lambda_n+\gamma L)\theta^2\big)\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2\\
&\leq 2p(V_\gamma(\mathbf{x}_t)-\mathbb{E}[V_\gamma(\mathbf{x}_{t+1})|\mathcal{F}_t]) + (1-\lambda_n+\gamma L)(\theta\gamma)^2(\frac{n\tilde{\sigma}^2}{m\tau}+nd\sigma^2).
\end{aligned} \tag{22}$$

Telescope the inequalities from $t=0$ to T, it holds that:

$$\begin{aligned}
&\big(2p\theta-(1-\lambda_n+\gamma L)\theta^2\big)\sum\nolimits_{t=0}^{T}\mathbb{E}\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2\\
&\leq 2p(V_\gamma(\mathbf{x}_0;\mathcal{D})-\mathbb{E}[V_\gamma(\mathbf{x}_{T+1};\mathcal{D})])\\
&+ (1-\lambda_n+\gamma L)(\theta\gamma)^2(\frac{n\tilde{\sigma}^2}{m\tau}+nd\sigma^2)(T+1)
\end{aligned} \tag{23}$$

Because of the two facts that $V_\gamma(\mathbf{x}_{T+1};\mathcal{D}) \geq \gamma\sum_{i=1}^n f(x_{i,T+1};\mathcal{D}_i) \geq \gamma\sum_{i=1}^n f(x^*_{\mathcal{D}};\mathcal{D}_i)$ and $V_\gamma(\mathbf{x}_0;\mathcal{D}) = \gamma\sum_{i=1}^n f(\mathbf{0};\mathcal{D}_i)$, it holds that:

$$\begin{aligned}
&\big(2p\theta-(1-\lambda_n+\gamma L)\theta^2\big)\sum\nolimits_{t=0}^{T}\mathbb{E}\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2\\
&\leq 2p\gamma\big(\sum\nolimits_{i=1}^n f(\mathbf{0};\mathcal{D}_i)-\sum\nolimits_{i=1}^n f(x^*_{\mathcal{D}};\mathcal{D}_i)\big)\\
&+ (1-\lambda_n+\gamma L)(\theta\gamma)^2(\frac{n\tilde{\sigma}^2}{m\tau}+nd\sigma^2)(T+1).
\end{aligned} \tag{24}$$

Thus, we have

$$\begin{aligned}
\sum\nolimits_{t=0}^{T}\mathbb{E}\|\nabla V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 \leq{}& \frac{2pn\gamma C_1}{\big(2p\theta-(1-\lambda_n+\gamma L)\theta^2\big)}\\
&+\frac{(1-\lambda_n+\gamma L)\theta\gamma^2(T+1)C_2}{2p-(1-\lambda_n+\gamma L)\theta}.
\end{aligned} \tag{25}$$

where $C_1 = f(\mathbf{0};\mathcal{D}) - f(x^*_{\mathcal{D}};\mathcal{D})$ and $C_2 = n\tilde{\sigma}^2/m\tau + nd\sigma^2$ are two constants.

Step 2: In the following, we provide the bound for $\|\mathbf{x}_t-\bar{\mathbf{x}}_t\| = \|\big(\mathbf{I}_{nd}-(\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\mathbf{x}_t\|$, where $\bar{\mathbf{x}}_t = \big((\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\mathbf{x}_t$. For notation convenience, we define $\mathbf{Q} = \mathbf{I}_{nd} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d$, $\mathbf{f}(\mathbf{x};\zeta) = \sum_{i=1}^n f(x_i;\zeta_i)$. From the updating (5), it holds that:

$$\begin{aligned}
\mathbf{x}_t &= \mathbf{x}_{t-1} - \theta(\nabla V_\gamma(\mathbf{x}_{t-1};\zeta_{t-1}) + \gamma\boldsymbol{\eta}_{t-1}) + \boldsymbol{\epsilon}_{t-1}\\
&= \mathbf{x}_{t-1} - \theta\big((\mathbf{I}-\tilde{\mathbf{W}})\mathbf{x}_{t-1} + \gamma\nabla\mathbf{f}(\mathbf{x}_{t-1};\zeta_{t-1}) + \gamma\boldsymbol{\eta}_{t-1}\big) + \boldsymbol{\epsilon}_{t-1}\\
&= \big((1-\theta)\mathbf{I}+\theta\tilde{\mathbf{W}}\big)\mathbf{x}_{t-1} - \theta\gamma\nabla\mathbf{f}(\mathbf{x}_{t-1};\zeta_{t-1}) - \theta\gamma\boldsymbol{\eta}_{t-1} + \boldsymbol{\epsilon}_{t-1}.
\end{aligned} \tag{26}$$

It can be seen that the updating is the stochastic DGD updating with a mixed concensus matrix $\tilde{\mathbf{W}}_\theta = (1-\theta)\mathbf{I}+\theta\tilde{\mathbf{W}}$, which is also doubly stochastic with $\theta\in(0,1)$. Thus, it holds that starting $\mathbf{x}_0 = \mathbf{0}$,

$$\begin{aligned}
\mathbf{x}_t &= \tilde{\mathbf{W}}_\theta\mathbf{x}_{t-1} - \theta\gamma\nabla\mathbf{f}(\mathbf{x}_{t-1};\zeta_{t-1}) - \theta\gamma\boldsymbol{\eta}_{t-1} + \boldsymbol{\epsilon}_{t-1}\\
&= \sum\nolimits_{s=0}^{t-1}\tilde{\mathbf{W}}_\theta^{t-1-s}\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s) - \theta\gamma\boldsymbol{\eta}_s + \boldsymbol{\epsilon}_s\big),
\end{aligned} \tag{27}$$

then due to the rows sums and columns sum of $\tilde{\mathbf{W}}_\theta$ are 1, it holds that

$$\begin{aligned}
&\mathbf{Q}\mathbf{x}_t = \mathbf{Q}\sum\nolimits_{s=0}^{t-1}\tilde{\mathbf{W}}_\theta^{t-1-s}\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s) - \theta\gamma\boldsymbol{\eta}_s + \boldsymbol{\epsilon}_s\big)\\
&= \sum\nolimits_{s=0}^{t-1}\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s) - \theta\gamma\boldsymbol{\eta}_s + \boldsymbol{\epsilon}_s\big),
\end{aligned} \tag{28}$$

which results to

$$\begin{aligned}
&\|\mathbf{Q}\mathbf{x}_t\|_2^2\\
&= 2\|\sum\nolimits_{s=0}^{t-1}\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s) - \theta\gamma\boldsymbol{\eta}_s\big)\|_2^2\\
&+ 2\|\sum\nolimits_{s=0}^{t-1}\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\boldsymbol{\epsilon}_s\|_2^2\\
&= 2\|\sum\nolimits_{s=0}^{t-1}\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s) - \theta\gamma\boldsymbol{\eta}_s\big)\|_2^2\\
&+ \sum\nolimits_{s=0}^{t-1}2\|\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\boldsymbol{\epsilon}_s\|_2^2\\
&+ \sum\nolimits_{s,s'=0,s\neq s'}^{t-1}2\langle\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\boldsymbol{\epsilon}_s,\\
&\big(\tilde{\mathbf{W}}_\theta^{t-1-s'} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\boldsymbol{\epsilon}_{s'}\rangle
\end{aligned} \tag{29}$$

Take the expectation at the both sides,

$$\begin{aligned}
&\frac{1}{2}\mathbb{E}[\|\mathbf{Q}\mathbf{x}_t\|_2^2]\\
&\overset{(a)}{=} \mathbb{E}[\|\sum\nolimits_{s=0}^{t-1}\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s) - \theta\gamma\boldsymbol{\eta}_s\big)\|_2^2]\\
&+ \sum\nolimits_{s=0}^{t-1}\mathbb{E}[\|\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\boldsymbol{\epsilon}_s\|_2^2]\\
&\leq \mathbb{E}[\|\sum\nolimits_{s=0}^{t-1}\big(\tilde{\mathbf{W}}_\theta^{t-1-s} - (\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top)\otimes\mathbf{I}_d\big)\big(-\theta\gamma\nabla\mathbf{f}(\mathbf{x}_s|\zeta_s) - \theta\gamma\boldsymbol{\eta}_s\big)\|_2^2]\\
&+ \sum\nolimits_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}(\frac{1}{p}-1)\mathbb{E}\|\theta(\nabla V(\mathbf{x}_s|\zeta_s)+\gamma\boldsymbol{\eta}_s)\|_2^2\\
&\leq (\theta\gamma)^2\sum\nolimits_{s=0}^{t-1}\beta_\theta^{t-1-s}\sum\nolimits_{s'=0}^{t-1}\beta_\theta^{t-1-s'}\mathbb{E}[\|\nabla\mathbf{f}(\mathbf{x}_s;\zeta_s)+\boldsymbol{\eta}_s\|_2^2]^{\frac{1}{2}}\\
&\mathbb{E}[\|\nabla\mathbf{f}(\mathbf{x}_{s'};\zeta_{s'})+\boldsymbol{\eta}_{s'}\|_2^2]^{\frac{1}{2}} + (\frac{1}{p}-1)\mathbb{E}\|\theta(\nabla V(\mathbf{x}_s;\zeta_s)+\gamma\boldsymbol{\eta}_s)\|_2^2\\
&\overset{(b)}{\leq} (\theta\gamma)^2\sum\nolimits_{s=0}^{t-1}\beta_\theta^{t-1-s}\sum\nolimits_{s'=0}^{t-1}\beta_\theta^{t-1-s'}((nG)^2+(nd\sigma)^2)\\
&+ (\frac{1}{p}-1)\mathbb{E}\|\theta(\nabla V(\mathbf{x}_s;\zeta_s)+\gamma\boldsymbol{\eta}_s)\|_2^2
\end{aligned}$$

$$
\begin{aligned}
&\le \frac{(\theta\gamma)^2((nG)^2+(nd\sigma)^2)}{(1-\beta_\theta)^2} + \sum\nolimits_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}\theta^2(\frac{1}{p}-1)\mathbb{E}[\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2 \\
&+ \gamma^2(\frac{n\tilde{\sigma}^2}{m\tau}+nd\sigma^2)] \\
&\le \frac{(\theta\gamma)^2((nG)^2+(nd\sigma)^2)}{(1-\beta_\theta)^2} + \frac{\theta^2\gamma^2C_2}{1-\beta_\theta^2}(\frac{1}{p}-1) \\
&+ \sum\nolimits_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}\theta^2(\frac{1}{p}-1)\mathbb{E}[\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2] \\
&\overset{(c)}{\le} \Big(\frac{\theta\gamma}{1-\beta_\theta}\Big)^2C_3 + \frac{\theta^2\gamma^2C_2}{1-\beta_\theta}(\frac{1}{p}-1)] \\
&+ \sum\nolimits_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}\theta^2(\frac{1}{p}-1)\mathbb{E}\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2 \\
&\overset{(d)}{\le} \Big(\frac{\gamma}{1-\beta}\Big)^2C_3 + \frac{\theta\gamma^2C_2}{1-\beta}(\frac{1}{p}-1) \\
&+ \sum\nolimits_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}\theta^2(\frac{1}{p}-1)\mathbb{E}[\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2] \qquad (30)
\end{aligned}
$$

where $\beta_\theta = \max\{|\lambda_2(\mathbf{W}_\theta)|, |\lambda_n(\mathbf{W}_\theta)|\}$ with $\mathbf{W} = (1-\theta)\mathbf{I}+\theta\mathbf{W}$ and (a) is because of $\mathbb{E}[\boldsymbol{\epsilon}_t] = \mathbf{0}$; (b) is because the function $\mathbf{f}(\mathbf{x};\zeta)$ is coordinately $G/\sqrt{d}$-Lipschitz and hence $\mathbb{E}[\|\nabla\mathbf{f}(\mathbf{x};\zeta)+\eta^2\|_2^2] = \mathbb{E}[\|\nabla\mathbf{f}(\mathbf{x};\zeta)\|_2^2+\|\eta^2\|_2^2] \le (nG)^2+(nd\sigma)^2$; (c) is because of $\beta_\theta \in (0,1)$ and $C_3 = (nG)^2+(nd\sigma)^2$; (d) is from Lemma 6.

Lemma 6. *Given $\theta \in (0,1)$, it holds*

$$
\frac{1}{1-\beta_\theta} \le \frac{1}{\theta(1-\beta)} \qquad (31)
$$

with $\beta_\theta = \max\{|\lambda_2(\mathbf{W}_\theta)|, |\lambda_n(\mathbf{W}_\theta)|\}$ and $\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_n(\mathbf{W})|\}$.

Proof. First, for $1/(1-\beta)$, according to the definition:

$$
\begin{aligned}
&\beta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_n(\mathbf{W})|\} \\
\Rightarrow\ &1-\beta = \min\{1-|\lambda_2(\mathbf{W})|, 1-|\lambda_n(\mathbf{W})|\} \\
\Rightarrow\ &1/(1-\beta) = \max\{1/(1-|\lambda_2(\mathbf{W})|), 1/(1-|\lambda_n(\mathbf{W})|)\}. \qquad (32)
\end{aligned}
$$

Then for $1/(1-\beta_\theta)$, note that $\mathbf{W}_\theta = (1-\theta)\mathbf{I}+\theta\mathbf{W}$, which implies $\lambda_i(\mathbf{W}_\theta) = (1-\theta)+\theta\lambda_i(\mathbf{W})$. So $\beta_\theta = \max\{|(1-\theta)+\theta\lambda_2(\mathbf{W})|, |(1-\theta)+\theta\lambda_n(\mathbf{W})|\}$. Note that for any $\lambda \in (-1,1]$ and $\theta \in (0,1)$ it holds that $|(1-\theta)+\theta\lambda| \le (1-\theta)+\theta|\lambda|$, thus,

$$
\begin{aligned}
&\beta_\theta \le \max\{(1-\theta)+\theta|\lambda_2(\mathbf{W})|, (1-\theta)+\theta|\lambda_n(\mathbf{W})|\} \\
\Rightarrow\ &1-\beta \ge \min\{\theta-\theta|\lambda_2(\mathbf{W})|, \theta-\theta|\lambda_n(\mathbf{W})|\} \\
\Rightarrow\ &1/(1-\beta_\theta) \le \max\{1/\theta(1-|\lambda_2(\mathbf{W})|), 1/\theta(1-|\lambda_n(\mathbf{W})|)\} \\
\Rightarrow\ &1/(1-\beta_\theta) \le 1/\theta(1-\beta). \qquad (33)
\end{aligned}
$$

□

Step 3: Note that $\mathbf{x}_t = \tilde{\mathbf{W}}_\theta\mathbf{x}_{t-1} - \theta\gamma\nabla\mathbf{f}(\mathbf{x}_{t-1}|\zeta_{t-1}) - \theta\gamma\boldsymbol{\eta}_{t-1} + \boldsymbol{\epsilon}_{t-1}$, which implies that

$$
\begin{aligned}
\bar{x}_t &= \bar{x}_{t-1} - \frac{1}{n}\sum\nolimits_{i=1}^n[\theta\gamma\nabla f(x_{i,t-1};\zeta_{i,t-1}) - \theta\gamma\eta_{i,t-1} + \epsilon_{i,t-1}] \\
&= \bar{x}_{t-1} - \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n[\nabla f(x_{i,t-1}|\zeta_{i,t-1}) - \eta_{i,t-1} + \epsilon_{i,t-1}/\theta\gamma] \\
&= \bar{x}_{t-1} - \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t-1}|\mathcal{B}_{i,t-1}), \qquad (34)
\end{aligned}
$$

where $\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) = \nabla f(x_{i,t};\zeta_{i,t}) - \eta_{i,t} + \epsilon_{i,t}/\theta\gamma$, and $\mathcal{B}_{i,t} = \sigma\langle\zeta_{i,t}, \eta_{i,t}, \epsilon_{i,t}/\theta\gamma\rangle$.

Consider $f(\bar{x}_t;\mathcal{D}) = \frac{1}{n}\sum_{i=1}^n f(\bar{x}_t;\mathcal{D}_i)$, by the L-Lipschitz continuous gradient, it holds:

$$
\begin{aligned}
&f(\bar{x}_{t+1};\mathcal{D}) \le f(\bar{x}_t;\mathcal{D}) + \langle\nabla f(\bar{x}_t;\mathcal{D}), \bar{x}_{t+1}-\bar{x}_t\rangle + \frac{L}{2}\|\bar{x}_{t+1}-\bar{x}_t\|_2^2 \\
&\le f(\bar{x}_t;\mathcal{D}) - \langle\nabla f(\bar{x}_t;\mathcal{D}), \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t})\rangle \\
&+ \frac{L}{2}\|\frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t})\|_2^2 \\
&\le f(\bar{x}_t;\mathcal{D}) - \langle\nabla f(\bar{x}_t;\mathcal{D}), \theta\gamma\nabla f(\bar{x}_t;\mathcal{D}) + \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\rangle + \frac{L}{2}\|\theta\gamma\nabla f(\bar{x}_t;\mathcal{D}) + \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 \\
&\le f(\bar{x}_t;\mathcal{D}) - \theta\gamma\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 - \langle\nabla f(\bar{x}_t;\mathcal{D}), \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\rangle + \frac{L}{2}[\|\theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + \|\frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + 2\langle\theta\gamma\nabla f(\bar{x}_t;\mathcal{D}), \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\rangle] \qquad (35)
\end{aligned}
$$

Take conditional expectation, it holds:

$$
\begin{aligned}
&\mathbb{E}[f(\bar{x}_{t+1};\mathcal{D})|\mathcal{F}_t] \\
&\overset{(a)}{\le} f(\bar{x}_t;\mathcal{D}) - \theta\gamma\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 - \langle\nabla f(\bar{x}_t;\mathcal{D}), \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla f(x_{i,t};\mathcal{D}_i) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\rangle + \frac{L}{2}[\|\theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + \mathbb{E}[\|\frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t] + 2\langle\theta\gamma\nabla f(\bar{x}_t;\mathcal{D}), \frac{\theta\gamma}{n}\sum\nolimits_{i=1}^n\nabla f(x_{i,t};\mathcal{D}_i) \\
&- \theta\gamma\nabla f(\bar{x}_t;\mathcal{D})\rangle] \\
&= f(\bar{x}_t;\mathcal{D}) - (\theta\gamma - \frac{L(\theta\gamma)^2}{2})\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 \\
&+ (\theta\gamma - L(\theta\gamma)^2)\langle\nabla f(\bar{x}_t;\mathcal{D}), \nabla f(\bar{x}_t;\mathcal{D}) - \frac{1}{n}\sum\nolimits_{i=1}^n\nabla f(x_{i,t};\mathcal{D}_i)\rangle \\
&+ \frac{L(\theta\gamma)^2}{2}\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t] \\
&\overset{(b)}{\le} f(\bar{x}_t;\mathcal{D}) - (\theta\gamma - \frac{L(\theta\gamma)^2}{2})\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + \frac{\theta\gamma - L(\theta\gamma)^2}{2}\times \\
&[\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + \|\frac{1}{n}\sum\nolimits_{i=1}^n\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2] \\
&+ \frac{L(\theta\gamma)^2}{2}\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t] \\
&= f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 \\
&+ \frac{\theta\gamma - L(\theta\gamma)^2}{2}\|\frac{1}{n}\sum\nolimits_{i=1}^n\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2 \\
&+ \frac{L(\theta\gamma)^2}{2}\underbrace{\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^n\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t]}_{(A)} \qquad (36)
\end{aligned}
$$

where (a) is because $\mathbb{E}[\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t})|\mathcal{F}_t] = \nabla f(x_{i,t};\mathcal{D}_i)$; (b) is by $2\langle \mathbf{a},\mathbf{b}\rangle \le \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2$. Now we give the bound for (A) : Note that

$$\begin{aligned}
&\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&= \mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D})\\
&\quad + \frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&\overset{(a)}{=} \mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&\quad + \|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2. \qquad (37)
\end{aligned}$$

where (a) is because $\mathbb{E}[\frac{1}{n}\sum_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t})|\mathcal{F}_t] = \frac{1}{n}\sum_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D})$. Recall the definition that $\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) = \nabla f(x_{i,t};\zeta_{i,t}) - \eta_{i,t} + \epsilon_{i,t}/\theta\gamma$, hence,

$$\begin{aligned}
&\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&\overset{(a)}{=} \frac{1}{n^2}\sum\nolimits_{i=1}^{n}\mathbb{E}[\|\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \nabla f(x_{i,t};\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&= \frac{1}{n^2}\sum\nolimits_{i=1}^{n}\mathbb{E}[\|\nabla f(x_{i,t};\zeta_{i,t}) - \eta_{i,t} + \epsilon_{i,t}/\theta\gamma - \nabla f(x_{i,t};\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&\overset{(b)}{=} \frac{1}{n^2}\sum\nolimits_{i=1}^{n}\mathbb{E}[\|\nabla f(x_{i,t};\zeta_{i,t}) - \nabla f(x_{i,t};\mathcal{D})\|_2^2\\
&+ \|\eta_{i,t}\|_2^2 + \|\epsilon_{i,t}/\theta\gamma\|_2^2|\mathcal{F}_t]\\
&\le \frac{1}{n}[\frac{\tilde{\sigma}^2}{m\tau} + d\sigma^2] + (\frac{1}{n\theta\gamma})^2\mathbb{E}[\|\boldsymbol{\epsilon}_t\|_2^2|\mathcal{F}_t]\\
&\le \frac{1}{n}[\frac{\tilde{\sigma}^2}{m\tau} + d\sigma^2] + (\frac{1}{n\theta\gamma})^2\mathbb{E}[(\frac{1}{p}-1)\|\theta V_\gamma(\mathbf{x}_t|\zeta_t) + \theta\gamma\boldsymbol{\eta}_t\|_2^2|\mathcal{F}_t]\\
&\le \frac{1}{n}[\frac{\tilde{\sigma}^2}{m\tau} + d\sigma^2] + (\frac{1}{n\gamma})^2(\frac{1}{p}-1)[\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{\gamma^2\tilde{\sigma}^2 n}{m\tau} + \gamma^2\sigma^2 nd]\\
&= (\frac{1}{n\gamma})^2(\frac{1}{p}-1)\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{1}{np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) \qquad (38)
\end{aligned}$$

where (a) is because the noise is independent across i, and (b) is because the expectation of the three terms are zero. Thus, we have

$$\begin{aligned}
&\mathbb{E}[f(\bar{x}_{t+1};\mathcal{D})|\mathcal{F}_t]\\
&\le f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{\theta\gamma - L(\theta\gamma)^2}{2}\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{L(\theta\gamma)^2}{2}\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&\le f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{\theta\gamma - L(\theta\gamma)^2}{2}\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{L(\theta\gamma)^2}{2}\mathbb{E}[\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla\tilde{f}(x_{i,t}|\mathcal{B}_{i,t}) - \frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D})\|_2^2|\mathcal{F}_t]\\
&+ \frac{L(\theta\gamma)^2}{2}\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&\le f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{\theta\gamma}{2}\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{L(\theta\gamma)^2}{2}[(\frac{1}{n\gamma})^2(\frac{1}{p}-1)\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{1}{np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d)]\\
&\le f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{\theta\gamma}{2}\|\frac{1}{n}\sum\nolimits_{i=1}^{n}\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{L\theta^2}{2n^2}(\frac{1}{p}-1)\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{L(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d)\\
&\overset{(a)}{\le} f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{\theta\gamma}{2n}\sum\nolimits_{i=1}^{n}\|\nabla f(x_{i,t};\mathcal{D}_i) - \nabla f(\bar{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{L\theta^2}{2n^2}(\frac{1}{p}-1)\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{L(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d)\\
&\overset{(b)}{\le} f(\bar{x}_t;\mathcal{D}) - \frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + \frac{\theta\gamma L}{2n}\|\mathbf{x}_t - \bar{\mathbf{x}}_t\|_2^2\\
&+ \frac{L\theta^2}{2n^2}(\frac{1}{p}-1)\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{L(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) \qquad (39)
\end{aligned}$$

where (a) is by the Jensen's inequality, (b) is by the L-Lipschitz continuous gradient $\nabla f(\mathbf{x};\zeta)$.

Taking full expectation and plugging the result in step 2, we have:

$$\begin{aligned}
&\mathbb{E}[f(\bar{x}_{t+1};\mathcal{D}) - f(\bar{x}_t;\mathcal{D})]\\
&\le \mathbb{E}[-\frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2 + \frac{\theta^3\gamma L}{n}\sum_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}(\frac{1}{p}-1)\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2\\
&+ \frac{\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p}-1) + \frac{L\theta^2}{2n^2}(\frac{1}{p}-1)\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2\\
&+ \frac{L(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d)] \qquad (40)
\end{aligned}$$

Telescope the above inequality from 1 to T:

$$\begin{aligned}
&\sum_{t=0}^{T-1}\mathbb{E}[\frac{\theta\gamma}{2}\|\nabla f(\bar{x}_t;\mathcal{D})\|_2^2]\\
&\le f(\bar{x}_0;\mathcal{D}) - \mathbb{E}[f(\bar{x}_T;\mathcal{D})] + \frac{\theta^3\gamma L}{n}\sum_{t=1}^{T-1}\sum_{s=0}^{t-1}\beta_\theta^{2(t-1-s)}(\frac{1}{p}-1)\times\\
&\mathbb{E}[\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2] + \frac{T\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{T\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p}-1)\\
&+ \sum_{t=0}^{T-1}\frac{L\theta^2}{2n^2}(\frac{1}{p}-1)\mathbb{E}\|V_\gamma(\mathbf{x}_t;\mathcal{D})\|_2^2 + \frac{LT(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d)\\
&\overset{(a)}{\le} f(0;\mathcal{D}) - f(x^*_{\mathcal{D}};\mathcal{D}) + \frac{\theta^3\gamma L}{n}\sum_{s=0}^{T-2}\sum_{t=s+1}^{T-1}\beta_\theta^{2(t-1-s)}(\frac{1}{p}-1)\times\\
&\mathbb{E}[\|\nabla V(\mathbf{x}_s;\mathcal{D})\|^2] + \frac{T\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{T\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p}-1)
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{t=0}^{T-1} \frac{L\theta^2}{2n^2}(\frac{1}{p} - 1)\mathbb{E}\|V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + \frac{LT(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) \\
&\leq f(0; \mathcal{D}) - f(x^*_{\mathcal{D}}; \mathcal{D}) + \frac{\theta^3\gamma L}{n(1-\beta_\theta^2)} \sum_{s=0}^{T-2}(\frac{1}{p} - 1)\|\nabla V(\mathbf{x}_s; \mathcal{D})\|^2 \\
&+ \frac{T\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{T\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p} - 1) + \sum_{t=0}^{T-1} \frac{L\theta^2}{2n^2}(\frac{1}{p} - 1)\times \\
&\mathbb{E}\|V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + \frac{LT(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) \\
&\overset{(b)}{\leq} f(0; \mathcal{D}) - f(x^*_{\mathcal{D}}; \mathcal{D}) + \frac{\theta^2\gamma L}{n(1-\beta)} \sum_{t=0}^{T-1}(\frac{1}{p} - 1)\mathbb{E}[\|\nabla V(\mathbf{x}_t; \mathcal{D})\|^2] \\
&+ \frac{T\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{T\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p} - 1) \\
&+ \sum_{t=0}^{T-1} \frac{L\theta^2}{2n^2}(\frac{1}{p} - 1)\mathbb{E}\|V_\gamma(\mathbf{x}_t; \mathcal{D})\|_2^2 + \frac{LT(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) \\
&\leq f(0; \mathcal{D}) - f(x^*_{\mathcal{D}}; \mathcal{D}) + (\frac{\theta^2\gamma L}{n(1-\beta)} + \frac{L\theta^2}{2n^2})(\frac{1}{p} - 1)\times \\
&\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla V(\mathbf{x}_t; \mathcal{D})\|^2] + \frac{T\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 \\
&+ \frac{T\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p} - 1) + \frac{LT(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) \\
&\overset{(c)}{\leq} f(0; \mathcal{D}) - f(x^*_{\mathcal{D}}; \mathcal{D}) + \frac{T\theta\gamma LC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{T\theta^2\gamma^3 LC_2}{n(1-\beta)}(\frac{1}{p} - 1) \\
&+ \frac{LT(\theta\gamma)^2}{2np}(\frac{\tilde{\sigma}^2}{m\tau} + \sigma^2 d) + (\frac{\theta^2\gamma L}{n(1-\beta)} + \frac{L\theta^2}{2n^2})(\frac{1}{p} - 1)\times \\
&\Big(\frac{2pn\gamma C_1}{\big(2p\theta - (1-\lambda_n+\gamma L)\theta^2\big)} + \frac{(1-\lambda_n+\gamma L)\theta\gamma^2 TC_2}{2p - (1-\lambda_n+\gamma L)\theta}\Big) \qquad (41)
\end{aligned}
$$

where (a) is by $\mathbb{E}[f(\bar{x}_T); \mathcal{D}] \geq f(x^*_{\mathcal{D}}; \mathcal{D})$ and Fubini's theorem, (b) is by $\beta_\theta \in (0, 1)$ and Lemma 6 and (c) is from step 1. Hence,

$$
\begin{aligned}
&\sum_{t=0}^{T-1} \|\nabla f(\bar{x}_t; \mathcal{D})\|_2^2 \leq \frac{2C_1}{\theta\gamma} + \frac{2TLC_3}{n}(\frac{\gamma}{1-\beta})^2 + \frac{2T\theta\gamma^2 LC_2}{n(1-\beta)}(\frac{1}{p} - 1) \\
&+ \frac{LT\theta\gamma}{n^2 p}C_2 + (\frac{2\theta\gamma L}{n(1-\beta)} + \frac{L\theta}{n^2})(\frac{1}{p} - 1)\Big(\frac{2pnC_1}{\big(2p\theta - (1-\lambda_n+\gamma L)\theta^2\big)} \\
&+ \frac{(1-\lambda_n+\gamma L)\theta\gamma TC_2}{2p - (1-\lambda_n+\gamma L)\theta}\Big) \qquad (42)
\end{aligned}
$$

where $C_1 = f(0; \mathcal{D}) - f(x^*_{\mathcal{D}}; \mathcal{D})$, $C_2 = n\tilde{\sigma}^2/m\tau + nd\sigma^2$ and $C_3 = (nG)^2 + (nd\sigma)^2$ are constants. □

A.4 Proof of Corollary 3

PROOF. By setting $\theta = \min\{p/(1-\lambda_n+\gamma L), p/2\} \leq 1$, we have

$$(1-\lambda_n+\gamma L)\theta = \min\{p, p(1-\lambda_n+\gamma L)/2\} \leq p \qquad (43)$$

$$1/\theta = \max\{(1-\lambda_n+\gamma L)/p, 2/p\} \overset{(a)}{\leq} 3/p \qquad (44)$$

where (a) is by $1 - \lambda_n + \gamma L \leq 3$ with a very small γ (i.e. large enough T). Thus for (I) in (7), $2C_1/\theta\gamma T \leq 6C_1/\gamma Tp = O(1/\gamma Tp)$; (II) is $O\big(n\gamma^2/(1-\beta)^2\big)$; for (III) is

$$\frac{2\theta\gamma^2 LC_2}{n(1-\beta)}(\frac{1}{p} - 1) + \frac{L\theta\gamma C_2}{n^2 p} \leq \frac{\gamma^2 LC_2}{n(1-\beta)} + \frac{L\gamma C_2}{2n^2} = O\big(\frac{\gamma^2}{(1-\beta)} + \frac{\gamma}{n}\big); \qquad (45)$$

and (IV) is

$$
\begin{aligned}
&(\frac{2\gamma L}{n(1-\beta)} + \frac{L}{n^2})(\frac{1}{p} - 1)\Big(\frac{2pnC_1}{\big(2p - (1-\lambda_n+\gamma L)\theta\big)T} + \frac{(1-\lambda_n+\gamma L)\theta^2\gamma C_2}{2p-(1-\lambda_n+\gamma L)\theta}\Big) \\
&\leq (\frac{2\gamma L}{n(1-\beta)} + \frac{L}{n^2})\Big(\frac{2nC_1}{Tp} + 2\gamma C_2\Big) = O\big((\frac{\gamma}{1-\beta} + \frac{1}{n})(\frac{1}{Tp} + 2\gamma)\big) \qquad (46)
\end{aligned}
$$

To summarize, the convergence error has the order:

$$
\begin{aligned}
&O\Big(\frac{1}{\gamma Tp} + \frac{n\gamma^2}{(1-\beta)^2} + \frac{\gamma^2}{(1-\beta)} + \frac{\gamma}{n} + (\frac{\gamma}{1-\beta} + \frac{1}{n})(\frac{1}{Tp} + 2\gamma)\Big) \\
&= O\Big(\frac{1}{\gamma Tp} + \frac{n\gamma^2}{(1-\beta)^2} + \frac{\gamma^2}{(1-\beta)} + \frac{\gamma}{n} + \frac{\gamma}{(1-\beta)Tp} + \frac{1}{nTp}\Big) \\
&= O\Big(\frac{1}{\gamma Tp} + \frac{n\gamma^2}{(1-\beta)^2} + \frac{\gamma}{n} + \frac{\gamma}{(1-\beta)Tp} + \frac{1}{nTp}\Big) \qquad (47)
\end{aligned}
$$

Set $\gamma = c\sqrt{n\log(T)/T}$, then order of the convergence error is:

$$O\Big(\frac{1}{\sqrt{n\log(T)Tp}} + \frac{n^2\log(T)}{(1-\beta)^2 Tp} + \sqrt{\frac{\log(T)}{nT}} + \frac{\sqrt{n\log(T)}}{(1-\beta)\sqrt{(Tp)^3}} + \frac{1}{nTp}\Big). \qquad (48)$$

With the large iteration number, i.e. $T/\log(T)^4 > n^5/(1-\beta)^4$, then the order of the convergence error is bounded by:

$$O\Big(\frac{1}{\sqrt{nT}} + \sqrt{\frac{\log(T)}{nT}} + \frac{\log(T)}{nT}\Big) = O\Big(\sqrt{\frac{\log(T)}{nT}}\Big). \qquad (49)$$

□