PrivSGP-VR: Differentially Private Variance-Reduced Stochastic Gradient Push with Tight Utility Bounds

Zehan Zhu\textsuperscript{1}, Yan Huang\textsuperscript{1}, Xin Wang\textsuperscript{2} and Jinming Xu\textsuperscript{1}\ast

\textsuperscript{1}Zhejiang University, Hangzhou, China
\textsuperscript{2}Qilu University of Technology, Jinan, China
\{12032045, huangyan5616\}@zju.edu.cn, xinwang@qlu.edu.cn, jimmyxu@zju.edu.cn

Abstract

In this paper, we propose a differentially private decentralized learning method (termed PrivSGP-VR) which employs stochastic gradient push with variance reduction and guarantees \((\epsilon, \delta)-\)differential privacy (DP) for each node. Our theoretical analysis shows that, under DP Gaussian noise with constant variance, PrivSGP-VR achieves a sub-linear convergence rate of \(O(1/\sqrt{nK})\), where \(n\) and \(K\) are the number of nodes and iterations, respectively, which is independent of stochastic gradient variance, and achieves a linear speedup with respect to \(n\). Leveraging the moments accountant method, we further derive an optimal \(K\) to maximize the model utility under certain privacy budget in decentralized settings. With this optimized \(K\), PrivSGP-VR achieves a tight utility bound of \(O\left(\sqrt{d \log \left(\frac{1}{\delta}\right)} / (\sqrt{n}J)\right)\), where \(J\) and \(d\) are the number of local samples and the dimension of decision variable, respectively, which matches that of the server-client distributed counterparts, and exhibits an extra factor of \(1/\sqrt{n}\) improvement compared to that of the existing decentralized counterparts, such as A(DP)\textsuperscript{2}SGD. Extensive experiments corroborate our theoretical findings, especially in terms of the maximized utility with optimized \(K\), in fully decentralized settings.

1 Introduction

Distributed learning has been widely adopted in various application domains due to its great potential in improving computing efficiency [Langer et al., 2020]. In particular, we assume that each computing node has \(J\) data samples in this paper\textsuperscript{1}, and we use \(f_i(x; j)\) to denote the loss of the \(j\)-th data sample at node \(i\) with respect to the model parameter \(x \in \mathbb{R}^d\). We are then interested in solving the following non-convex finite-sum optimization problem via a group of \(n\) nodes:

\[
\min_{x \in \mathbb{R}^d} f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x),
\]

where \(f_i(x) \triangleq \frac{1}{J} \sum_{j=1}^{J} f_i(x; j)\) is the loss function of node \(i\) and all nodes collaborate to find a common model parameter \(x\) minimizing their average loss functions. We also assume that each node \(i\) can only evaluate local stochastic gradient \(\nabla f_i(x; \xi_i), \xi_i \in \{1, 2, ..., J\}\).

For distributed parallel methods [Li et al., 2014; McMahan et al., 2017] where there is a center (e.g., parameter server), they suffer from high communication burden and single point failure of the central node [Lian et al., 2017]. These potential bottlenecks motivate researchers to study fully decentralized methods [Lian et al., 2017; Lian et al., 2018] to solve Problem (1), where the central node is not required and each node only communicates with its neighbors. The existing decentralized learning algorithms usually employ undirected graphs for communication, which can not be easily implemented due to the existence of deadlocks [Assran et al., 2019]. It is desirable to consider more practical scenarios where communication graphs may be directed and even time-varying. Stochastic gradient push (SGP) proposed in [Assran et al., 2019], which builds on push-sum protocol [Kempe et al., 2003], is proven to be very effective in solving Problem (1) over directed and time-varying communication graphs.

It has been well known that the frequent exchange of model parameters in decentralized learning may lead to severe concern on privacy leakage as the disclosure of intermediate parameters could potentially compromise the original data [Wang et al., 2019c]. For instance, previous studies [Truex et al., 2019; Carlini et al., 2019] have shown that the exposed parameters can be utilized to crack original data samples. To address the above issue, differential privacy (DP), as a theoretical tool to provide rigorous privacy guarantees and quantify privacy loss, can be incorporated into each node in decentralized learning systems to enhance the privacy protection. DP techniques usually inject certain noises to disturb parameters for privacy preservation, which inevitably degrades the model accuracy. Besides, the variance of the added DP noise needs to be increased with the total number of iterations \(K\) to ensure certain privacy guarantee due to the accumulated privacy leakage over communication rounds [Dwork et al., 2014; Abadi et al., 2016; Wu et al., 2020; Wei et al., 2021]. In this regard, an excessive total number of iterations \(K\) may severely deteriorate the model accuracy, and hence prohibits the implementation of DP in real decentralized learning sys-

\astCorresponding author.

\textsuperscript{1}Full version of this paper can be found in [Zhu et al., 2024].
tems. Therefore, given certain privacy budget, it is necessary to optimize $K$ to achieve a useful model with high accuracy. However, the few existing decentralized learning algorithms with DP guarantee for non-convex problems either do not consider optimizing $K$ under certain privacy guarantee [Yu et al., 2021], or their derived theoretical utility bound under the optimized $K$ cannot match that of the server-client distributed counterparts [Xu et al., 2022].

In this paper, we aim to design a differentially private decentralized learning algorithm for non-convex problems, find the optimal $K$ that attains maximized model accuracy given certain privacy budget, and achieve a tight utility bound matching that of the server-client distributed counterparts. We summarize our main contributions as follows:

- **New efficient algorithm with personalized DP guarantee for each node.** Different from the existing works, we propose a differentially private learning method (termed PrivSGP-VR) for non-convex problems, which can work over general time-varying directed communication topologies in fully decentralized settings. More importantly, a personalized $(\epsilon_i, \delta_i)$-differential privacy (DP) is guaranteed for each node $i$, and variance-reduced technique is adopted to eliminate the effect of stochastic gradient noise, improving the convergence performance.

- **Linear speedup and tight utility bound.** Under DP Gaussian noise with constant variance for each node, we derive a sub-linear convergence rate of $O\left(\frac{1}{\sqrt{nK}}\right)$ for PrivSGP-VR, which is independent of stochastic gradient variance and scales linearly w.r.t. the number of nodes $n$. More importantly, given certain privacy budget $(\epsilon_i, \delta_i)$ for each node $i$, leveraging the moments accountant method, we derive the optimized number of iterations $K$ to obtain a tight convergence error bound maximizing the model utility. With this optimized $K$, we achieve the utility bound of $O\left(\sqrt{d \log \left(\frac{1}{\sqrt{nJ}\epsilon}\right)}\right)^2$ for PrivSGP-VR, which matches that of the server-client distributed counterparts, and enjoys an extra factor of $1/\sqrt{n}$ improvement compared to that of the existing decentralized evaluations (c.f., Table 1).

- **Extensive experimental evaluations.** Extensive experiments on two training tasks are conducted to validate our theoretical findings. In particular, our experimental results show that properly setting the total number of iterations $K$ will significantly improve the model accuracy for the proposed PrivSGP-VR algorithm under certain privacy budget. To the best of our knowledge, this is the first empirical validation of the existence of an optimal choice of $K$ in the realm of differentially private decentralized learning. Besides, we validate the property of linear speedup for PrivSGP-VR employing DP noise with a constant variance. Moreover, we demonstrate the trade-off between maximizing model utility and ensuring privacy protection by executing PrivSGP-VR with various optimized numbers of iterations that correspond to different privacy budgets.

2 Related Works

Differential privacy (DP) was first proposed in [Dwork et al., 2006] to protect data privacy for database queries. A DP mechanism adds randomly generated zero-mean noises to the output of a query function before it is exposed, making it difficult for curious attackers to extract users’ private information from the distorted query results. The basic composition theorem [Dwork et al., 2006; Dwork and Lei, 2009] and advanced composition theorem [Dwork et al., 2010; Bun and Steinke, 2016] are commonly used for computing the overall accumulated privacy loss in iterative training processes. However, these theorems can result in loose estimates of privacy loss. To address this issue, the moments accountant method proposed in [Abadi et al., 2016] obtains a much tighter estimate on the overall privacy loss by tracking higher moments and thus provides a more accurate way for calculating the privacy spending.

There has been a recent surge in research efforts towards achieving differential privacy guarantees in large-scale machine learning systems. Abadi et al. [2016]; Wang et al. [2017]; Chen et al. [2020]; Wang et al. [2020] design differentially private stochastic learning algorithms in a centralized setting. For distributed settings, Laplace and Gaussian mechanisms have been incorporated into federated learning, with corresponding convergence rates analyzed, respectively [Wu et al., 2020; Wei et al., 2020; Wei et al., 2021]. Truex et al. [2020] explore differential privacy guarantee for each client in federated personalized model learning. In [Zhou et al., 2023], the authors consider optimizing the numbers of queries and replies in federated learning to maximize the model utility given certain privacy budget for strongly convex problems. Zhang et al. [2020]; Li et al. [2022] achieve both differential privacy and communication compression in federated learning for non-convex problems, and provide a characterization of trade-offs in terms of privacy, utility, and communication complexity. There are also other works dedicated to designing differentially private distributed learning algorithms, such as [Li et al., 2019; Zeng et al., 2021; Liu et al., 2022; Lowy et al., 2022], but all the above-mentioned distributed methods are only applicable to the server-client architecture.

Recently, there have been few works aiming to achieve differential privacy for fully decentralized learning algorithms. For example, the works in [Cheng et al., 2018; Cheng et al., 2019] achieve differential privacy in fully decentralized learning systems for strongly convex problems. Wang et al. [2024] achieve differential privacy in fully decentralized architectures by tailoring gradient methods for deterministic optimization problems. Yu et al. [2021] present a decentralized stochastic learning method for non-convex problems with differential privacy guarantee (DP$^2$-SGD) based on D-PSGD [Lian et al., 2017], which relies on a fixed communication topology and uses the basic composition theorem to bound the overall privacy loss. To have a tight privacy guarantee, Xu et al. [2022] propose a differentially private asynchronous decentralized learning algorithm (ADP$^2$-SGD) for non-convex problems based on AD-PSGD [Lian et al., 2018],

---

Here, we set $\epsilon_i = \epsilon$ and $\delta_i = \delta$ for utility bound comparison.

---

Here, by being distributed, we mean sever-client architecture.
which provides privacy guarantee in the sense of Rényi differential privacy (RDP) [Mironov, 2017]. However, it should be noted that the above-mentioned two fully decentralized differentially private algorithms [Yu et al., 2021; Xu et al., 2022] work only for undirected communication graphs, which is often not satisfied in practical scenarios, and their convergence performance suffer from the effect of stochastic gradient variance. Moreover, none of them provide experimental evaluation to verify that selecting an appropriate value of $K$ can, indeed, improve the model utility (accuracy) under given certain privacy budget.

On the theoretical level, for general non-convex problems, a utility bound of $O\left(\frac{d \log \left(\frac{1}{\delta_k}\right)}{J}\right)$ is established for centralized learning with DP [Abadi et al., 2016], and a utility bound of $O\left(\frac{d \log \left(\frac{1}{\delta_k}\right)}{\sqrt{J}e}\right)$ is provided for server-client distributed algorithms with DP [Lowy et al., 2022; Li et al., 2022], which scales as $1/\sqrt{n}$ w.r.t. the number of nodes $n$. For DP-based fully decentralized algorithms as mentioned above, DP$^2$-SGD [Yu et al., 2021] lacks a theoretical utility guarantee under a given privacy budget, while the utility bound of A(DP)$^2$-SGD [Xu et al., 2022] can not match that of the server-client distributed counterparts, losing a scaling factor of $1/\sqrt{n}$ (c.f., Table 1).

### 3 Algorithm Development

We consider solving Problem (1) over the following general network model.

**Network Model.** The communication topology considered in this work is modeled as a sequence of time-varying directed graph $G^k = (\mathcal{V}, \mathcal{E}^k)$, where $\mathcal{V} = \{1, 2, ..., n\}$ denotes the set of nodes and $\mathcal{E}^k \subset \mathcal{V} \times \mathcal{V}$ denotes the set of directed edges/links at iteration $k$. We associate each graph $G^k$ with a non-negative mixing matrix $P^k \in \mathbb{R}^{n \times n}$ such that $(i, j) \in \mathcal{E}^k$ if $P^k_{i,j} > 0$, i.e., node $i$ receiving a message from node $j$ at iteration $k$. Without loss of generality, we assume that each node is an in-neighbor of itself.

The following assumptions are made on the mixing matrix and graph for the above network model to facilitate the subsequent convergence analysis for the proposed algorithm.

**Assumption 1 (Column Stochastic Mixing Matrix).** For any iteration $k$, the non-negative mixing matrix $P^k$ is column-stochastic, i.e., $1^T P^k = 1^T$, where $1$ is a column vector with all of its elements equal to 1.

**Assumption 2 (B-strongly Connected Graph).** We assume that there exists finite, positive integers $B$ and $\triangle$, such that the graph with edge set $\bigcup_{k=1}^{(l+1)B-1} \mathcal{E}^k$ is strongly connected and has diameter at most $\triangle$ for all $\forall l \geq 0$.

Before developing our proposed algorithm, we briefly introduce the following definition of $(\epsilon, \delta)$-DP [Dwork et al., 2014], which is crucial to subsequent analysis.

**Definition 1 ($\epsilon, \delta$-DP).** A randomized mechanism $\mathcal{M}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies $(\epsilon, \delta)$-differential privacy, or $(\epsilon, \delta)$-DP for short, if for any two adjacent inputs $x, x' \in \mathcal{D}$ differing on a single entry and for any subset of outputs $S \subseteq \mathcal{R}$, it holds that

$$Pr[\mathcal{M}(x) \in S] \leq e^\epsilon Pr[\mathcal{M}(x') \in S] + \delta,$$

where the privacy budget $\epsilon$ denotes the privacy lower bound to measure a randomized query and $\delta$ is the probability of breaking this bound.

---

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Privacy</th>
<th>Utility</th>
<th>Communication rounds</th>
<th>Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP-SGD [Abadi et al., 2016]</td>
<td>$(\epsilon, \delta)$-DP</td>
<td>$\frac{d \log \left(\frac{1}{\delta_k}\right)}{J}$</td>
<td>—</td>
<td>single node centralized</td>
</tr>
<tr>
<td>Distributed DP-SRM$^1$ [Wang et al., 2019a]</td>
<td>$(\epsilon, \delta)$-DP global</td>
<td>$\frac{d \log \left(\frac{1}{\delta_k}\right)}{nJ}$</td>
<td>$\frac{n^2 J \epsilon \sqrt{d}}{\sqrt{\log \left(\frac{1}{\delta_k}\right)}}$</td>
<td>$n$ nodes server-client</td>
</tr>
<tr>
<td>LDP SVRG/SPIDER [Lowy et al., 2022] for each node</td>
<td>$(\epsilon, \delta)$-DP</td>
<td>$\frac{d \log \left(\frac{1}{\delta_k}\right)}{\sqrt{nJ}}$</td>
<td>$\frac{n^2 J \epsilon \sqrt{d}}{\sqrt{\log \left(\frac{1}{\delta_k}\right)}}$</td>
<td>$n$ nodes server-client</td>
</tr>
<tr>
<td>SoteriaFL-SAGA/SVRG [Li et al., 2022] for each node</td>
<td>$(\epsilon, \delta)$-DP</td>
<td>$\frac{(1+\omega)d \log \left(\frac{1}{\delta_k}\right)}{\sqrt{\log \left(\frac{1}{\delta_k}\right)}}$</td>
<td>$\frac{n^2 J \epsilon \sqrt{d}}{\sqrt{\log \left(\frac{1}{\delta_k}\right)}}$</td>
<td>$n$ nodes server-client</td>
</tr>
<tr>
<td>A(DP)$^2$-SGD$^2$ [Xu et al., 2022] for each node</td>
<td>$(\epsilon, \delta)$-DP</td>
<td>$\frac{d \log \left(\frac{1}{\delta_k}\right)}{J}$</td>
<td>$\frac{n^2 J \epsilon \sqrt{d}}{\sqrt{\log \left(\frac{1}{\delta_k}\right)}}$</td>
<td>$n$ nodes decentralized</td>
</tr>
<tr>
<td>PrivSGP-VR (Our Algorithm 1) for each node</td>
<td>$(\epsilon, \delta)$-DP</td>
<td>$\frac{d \log \left(\frac{1}{\delta_k}\right)}{\sqrt{nJ}}$</td>
<td>$\frac{n^2 J \epsilon \sqrt{d}}{\sqrt{\log \left(\frac{1}{\delta_k}\right)}}$</td>
<td>$n$ nodes decentralized</td>
</tr>
</tbody>
</table>

$^1$ Wang et al. [2019a] consider global $(\epsilon, \delta)$-DP that merely protects the privacy for the entire dataset while we consider $(\epsilon, \delta)$-DP for each node, which can protect the local dataset at the node’s level.

$^2$ For A(DP)$^2$-SGD, the authors only provide the utility bound under global $(\epsilon, \delta)$-DP for the entire dataset. We thus derive their utility bound in the sense of ensuring $(\epsilon, \delta)$-DP for each node for fair comparison.
It can be observed that the smaller the values of $\epsilon$ and $\delta$ are, the higher the level of privacy guarantee will be. In this paper, we allow each node $i$ to tolerate different level of privacy loss, yielding personalized privacy budget $(\epsilon_i, \delta_i)$ for each node $i$. Now, we are ready to present our differentially private decentralized learning algorithm as follows.

**Stochastic gradient push over time-varying directed graphs.** We first introduce decentralized stochastic gradient push method based on Push-Sum protocol [Kempe et al., 2003], which can tackle the unblancheness of directed topologies by asymptotically estimating the Perron–Frobenius eigenvector of transition matrices. In particular, each node $i$ maintains three variables during the learning process: i) the model parameter $x^k_i$; ii) the scalar Push-Sum weight $w^k_i$ and iii) the de-biased parameter $\tilde{x}^k_i = x^k_i / w^k_i$, with the same initialization of $x^0_i = \tilde{x}^0_i = x^0 = 0$, $\forall i \in \mathbb{R}^d$ and $w^0_i = 1$ for all nodes $i \in \{1, 2, ..., n\}$. At each iteration $k$, each node $i$ updates as follows:

**Local SGD:**

\[
x_i^{k+\frac{1}{2}} = x_i^k - \gamma \nabla f_i(z_i^k; \xi_i^k),
\]

**Averaging:**

\[
x_i^{k+1} = \frac{1}{n} \sum_{j=1}^{n} P_{i,j} x_j^{k+\frac{1}{2}},
\]

**De-bias:**

\[
x_i^{k+1} = x_i^{k+1}/w_i^{k+1},
\]

where $\gamma > 0$ is the step size and $\nabla f_i(z_i^k; \xi_i^k)$ is the stochastic gradient evaluated on the de-biased parameter $z_i^k$. Note that, during the training process, each node exchanges model parameter with its neighbors frequently for averaging, resulting in potential privacy leakage as the original data could be recovered based on the disclosed model parameters.

**Ensuring differential privacy guarantee for each node.** We apply the differential privacy mechanism to protect the exchanged sensitive model parameters of each node. In particular, for each node $i$, the exchanged model parameter $x_i^{k+\frac{1}{2}}$ is obtained by performing a Local SGD step using the gradient $\nabla f_i(z_i^k; \xi_i^k)$. Since perturbing the gradient is equivalent to perturbing the model parameter, we thus inject randomly generated noise to the gradient $\nabla f_i(z_i^k; \xi_i^k)$ instead of directly adding noise to the exchanged model parameter in the proposed approach as follows:

\[
\nabla f_i(z_i^k; \xi_i^k) = \nabla f_i(z_i^k; \xi_i^k) + N_i^k
\]

where the noise $N_i^k$ is drawn from the Gaussian distribution $\mathcal{N}(0, \sigma_i^2 I_d)$ and $I_d$ represents the identity matrix with $d$ dimension. Then, the Local SGD step becomes:

\[
x_i^{k+\frac{1}{2}} = x_i^k - \gamma \nabla f_i(z_i^k; \xi_i^k) = x_i^k - \gamma \left( \nabla f_i(z_i^k; \xi_i^k) + N_i^k \right).
\]

We will refer to the above generated differentially private algorithm as PrivSGP (its pseudo-code can be found in Appendix C in our full version [Zhu et al., 2024]).

**Eliminating the stochastic gradient noise.** We now introduce the variance reduction technique [Defazio et al., 2014] to eliminate the effect of stochastic gradient noise of each node on convergence performance. Specifically, each node $i$ maintains a stochastic gradient table for all of its own local data samples. At each iteration $k$, after computing the stochastic gradient $\nabla f_i(z_i^k; \xi_i^k)$, node $i$ does not perform a local differentially private SGD step using $\nabla f_i(z_i^k; \xi_i^k)$ directly (c.f., (4)). Instead, $\nabla f_i(z_i^k; \xi_i^k)$ is corrected by subtracting the previously stored stochastic gradient corresponding to the $\xi_i^k$-th data sample, and then adding the average of all stored stochastic gradients. With such a corrected stochastic gradient, node $i$ performs a local differentially private SGD step and replaces the stochastic gradient of the $\xi_i^k$-th data sample in the table with $\nabla f_i(z_i^k; \xi_i^k)$. To better understand this process, let

\[
\phi_{i,j}^{k+1} = \left\{ \begin{array}{ll}
\phi_{i,j}^k, & j \neq \xi_i^k \\
0, & j = \xi_i^k.
\end{array} \right.
\]

where $\phi_{i,j}^k$ is the most recent model parameter used for computing $\nabla f_i(:, j)$ prior to iteration $k$. Thus, $\nabla f_i(\phi_{i,j}^k; j)$ represents the previously stored stochastic gradient for the $j$-th data sample of node $i$ prior to iteration $k$, and

\[
g_i^k \triangleq \nabla f_i(z_i^k; \xi_i^k) - \nabla f_i(\phi_{i,\xi_i^k}^k; \xi_i^k) + \frac{1}{J} \sum_{j=1}^{J} \nabla f_i(\phi_{i,j}^k; j)
\]
is the corrected stochastic gradient of node $i$ at iteration $k$. As a result, we replace the original stochastic gradient $\nabla f_i(\zeta_i; \xi_i)$ in (4) with $g_i^k$, leading to the following new local differentially private SGD step, i.e.,
\begin{equation}
x_i^{k+\frac{1}{2}} = x_i^k - \gamma (g_i^k + N_i^k),
\end{equation}
which yields the proposed differentially private decentralized learning method PrivSGP-VR, whose complete pseudocode is summarized in Algorithm 1.

4 Theoretical Analysis

In this section, we provide utility and privacy guarantees for the proposed PrivSGP-VR method.

4.1 Convergence Guarantee

To facilitate our convergence analysis, we make the following commonly used assumptions.

Assumption 3 (Smoothness). For each node $i$, $\forall x \in \mathbb{R}^d$ and sample $\forall \xi_i \in \{1, 2, ..., J\}$, the local sample loss function $f_i(x; \xi_i)$ has $L$-Lipschitz continuous gradients.

Assumption 4 (Unbiased Gradient). For any model $x \in \mathbb{R}^d$, the stochastic gradient $\nabla f_i(x; \xi_i), \xi_i \sim \{1, 2, ..., J\}$ generated by each node $i$ is unbiased, i.e.,
\begin{equation}
E[\nabla f_i(x; \xi_i)] = \nabla f_i(x).
\end{equation}

Assumption 5 (Bounded Data Heterogeneity). There exists a positive constant $b^2$ such that for any node $i$ and $\forall x \in \mathbb{R}^d$,
\begin{equation}
\|\nabla f_i(x) - \nabla f(x)\|^2 \leq b^2.
\end{equation}

With the above assumptions, we have the following convergence result for PrivSGP-VR (Algorithm 1).

Theorem 1 (Convergence Rate). Suppose Assumptions 1-5 hold. Let $K$ be the total number of iterations and $f^* = \min_{x \in \mathbb{R}^d} f(x)$. If the step-size is set as $\gamma = \sqrt{\frac{L}{K}}$ then there exist constants $C$ and $q \in [0, 1)$, which depend on the diameter of the network $\triangle$ and the sequence of mixing matrices $P^k$, such that, for any $K$ satisfying $K \geq \bar{K}(C,q)$, we have
\begin{equation}
\frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{n} \sum_{i=1}^{n} E\left[\|\nabla f_i(z_i^k)\|^2\right] \leq 13F_0^2 + 6L\|x_0\|^2 + 18Lb^2 + 24L \cdot \frac{d}{n} \sum_{i=1}^{n} \sigma_i^2,
\end{equation}
where $F_0 = f(x_0) - f^*$, $C$ and $q$ can be found in Lemma 7, and the definition of $\bar{K}(C,q)$ can be found at (92) in the appendix in our full version [Zhu et al., 2024], respectively.

Proof. See Appendix A in our full version [Zhu et al., 2024].

Remark 1. Under DP Gaussian noise with a constant variance, the above result suggests that the convergence rate for PrivSGP-VR is $O\left(\frac{1}{\sqrt{K}}\right)$, which is independent of stochastic gradient variance $\zeta_i^2$ with $E\left[\|\nabla f_i(x; \xi_i) - \nabla f_i(x)\|^2\right] \leq \zeta_i^2$ that appears in the convergence error bound in [Xu et al., 2022; Yu et al., 2021], and achieves linear speedup with respect to the number of nodes. Although it converges to an exact stationary point as the total number of iterations $K$ goes to infinity, the privacy loss will also become infinite according to the composition theorem [Dwork et al., 2006]. As a result, it is necessary to consider the trade-off between the model utility and privacy guarantee for the proposed PrivSGP-VR algorithm under certain given privacy budget.

4.2 Privacy and Utility Guarantee

Leveraging the moments accountant method [Abadi et al., 2016], we can calculate the variance $\sigma_i^2$ of the DP Gaussian noise needed to be added according to the total number of iterations $K$ and the given privacy budget $(\epsilon_1, \delta_1)$, which is provided in the following theorem.

Theorem 2 (Privacy Guarantee). Suppose that the stochastic gradient of each $f_i$ is uniformly bounded, i.e., $G = \max_{k,i} \|\nabla f_i(z_i^k; \xi_i)\| < \infty$. There exist constants $c_1$ and $c_2$ such that, given the total number of iterations $K$ for Algorithm 1, $(\epsilon_1, \delta_1)$-differential privacy can be guaranteed for each node $i$, for any $\epsilon_i < \frac{c_1 K}{Lb}$ and $\delta_i \in (0, 1)$, if $N_i^k$ is drawn from the Gaussian distribution $N\left(0, \sigma_i^2 I_d\right)$ with
\begin{equation}
\sigma_i = 3c_2 G \sqrt{K \log \left(\frac{1}{\delta_i}\right) / J_{\epsilon_i}}.
\end{equation}

Proof. See Appendix B in our full version [Zhu et al., 2024].

As highlighted in Theorem 2, it is evident that when a certain privacy budget $(\epsilon_i, \delta_i)$ is given, a larger value of $K$ requires the added DP Gaussian noise with a larger variance $\sigma_i^2$. This can potentially impact the model utility negatively. Therefore, our objective is to optimize the value of $K$ in order to maximize the final model accuracy under certain privacy budget $(\epsilon_i, \delta_i)$ for each node $i$.

Plugging (11) into (10) in Theorem 1, we obtain the following utility guarantee.

Corollary 1 (Maximized Utility Guarantee). Given certain privacy budget $(\epsilon_i, \delta_i)$ for each node $i \in \{1, 2, ..., n\}$, under the same conditions of Theorem 1 and 2, if the total number of iterations $K$ further satisfies
\begin{equation}
K = \frac{(13F_0^2 + 6L\|x_0\|^2 + 18Lb^2)J_{F\epsilon n}}{216Ldc_2^2 G^2 \sum_{i=1}^{n} \frac{1}{\epsilon_i} \log \left(\frac{1}{\delta_i}\right)},
\end{equation}
then we have
\begin{equation}
\frac{1}{K} \sum_{k=0}^{K-1} \frac{1}{n} \sum_{i=1}^{n} E\left[\|\nabla f_i(z_i^k)\|^2\right] \leq 12c_2 G \sqrt{6Ld \left(13F_0^2 + 6L\|x_0\|^2 + 18Lb^2\right) \sum_{i=1}^{n} \frac{1}{\epsilon_i} \log \left(\frac{1}{\delta_i}\right)}.
\end{equation}

5747
A(DP) is tighter than that of the existing decentralized counterpart can be observed that the utility bound of our PrivSGP-VR.

Remark 3. Note that if one set \( \epsilon_i = \epsilon \) and \( \delta_i = \delta \) for each node \( i \), the utility bound of the proposed PrivSGP-VR (13) will be reduced to \( O\left( \sqrt{\log(\frac{1}{\epsilon})}/(\sqrt{n}J) \right) \), achieving the same utility guarantee as differentially private learning algorithms with server-client structure, such as LDP SVRG/SPIDER [Lowy et al., 2022], and SoteriaFL-SAGA/SVRG [Li et al., 2022] without communication compression (\( \omega = 0 \)), see Table 1. Due to the presence of network dynamics in a fully decentralized time-varying setting, it is not surprising that the proposed PrivSGP-VR requires more communication rounds than that of other server-client distributed counterparts. In addition, PrivSGP-VR recovers the same utility \( O\left( \sqrt{\log(\frac{1}{\epsilon})}/(\sqrt{n}J) \right) \) as the baseline DP-SGD [Abadi et al., 2016] when \( n = 1 \). Furthermore, it can be observed that the utility bound of our PrivSGP-VR is tighter than that of the existing decentralized counterpart A(DP):SGD [Xu et al., 2022], exhibiting an extra factor of \( 1/\sqrt{n} \) improvement. To the best of our knowledge, we are the first to derive such a utility bound scaling as \( 1/\sqrt{n} \) with respect to the number of nodes in the realm of decentralized learning with DP guarantee for each node, for general non-convex problems.

5 Experiments

We conduct extensive experiments to validate the theoretical findings for the proposed PrivSGP-VR under various settings. All experiments are deployed in a high performance computer with Intel Xeon E5-2680 v4 CPU @ 2.40GHz and 8 Nvidia RTX 3090 GPUs, and are implemented with distributed communication package torch.distributed in PyTorch, where a process serves as a node, and inter-process communication is used to mimic communication among nodes. We consider two non-convex learning tasks (i.e., deep CNN ResNet-18 [He et al., 2016] training on Cifar-10 dataset [Krizhevsky, 2009] and shallow 2-layer neural network training on Mnist dataset [Deng, 2012]), in fully decentralized setting. For all experiments, we split shuffled datasets evenly to \( n \) nodes. For communication topology, unless otherwise stated, we use time-varying directed exponential graph (refer to Appendix D in our full version [Zhu et al., 2024] for its specific definition) for our PrivSGP-VR algorithm.

5.1 Deep CNN ResNet-18 Training

We first report the experiments of training CNN model ResNet-18 on Cifar-10 dataset. Once the dataset and learning model are given, the problem-related parameters such as \( L \) and \( b^c \) can be estimated by leveraging the method introduced in [Wang et al., 2019b; Luo et al., 2021]. The values of these parameters are \( L = 25, G^2 = 100, f^0(x) - f^* = 2.8, b^* = 500000 \) and \( \| x^0 \|^2 = 780000 \), for ResNet-18 training task.

![Comparison of convergence performance for PrivSGP-VR over 8, 16, 32 and 64 nodes under the same DP Gaussian noise variance, when training ResNet-18 on Cifar-10.](image)

Figure 1: Comparison of convergence performance for PrivSGP-VR over 8, 16, 32 and 64 nodes under the same DP Gaussian noise variance, when training ResNet-18 on Cifar-10.

Linear speedup under constant DP Gaussian noise variance. We first illustrate the convergence and scalability in terms of number of nodes \( n \) of PrivSGP-VR. In our experimental setup, we implement PrivSGP-VR on 4 distinct network configurations, comprising 8, 16, 32 and 64 nodes, respectively. All configurations utilize the same DP Gaussian noise variance \( \sigma^2 = 0.03 \) for each node \( i \). It can be observed from Figure 1 that, by increasing the number of nodes by a factor of 2, we can achieve comparable final training loss and model testing accuracy by running only half the total number of iterations. This observation illustrates the linear speedup property exhibited by our PrivSGP-VR algorithm.

Optimizing number of iterations under certain privacy budget. We investigate the significance of selecting an appropriate total number of iterations \( K \) for our proposed PrivSGP-VR given a specific privacy budget. To demonstrate this, we conduct experiments using PrivSGP-VR on a network consisting of 16 nodes. For each node \( i \), we set the privacy budget to \( \epsilon_i = 3 \) and \( \delta_i = 10^{-5} \). By varying the
value of $K$, we aim to highlight the impact of this parameter on the overall performance of PrivSGP-VR. Using (12), we determine the optimal value of $K$ to be approximately 3120. In addition to this optimal choice, we also consider other values of $K$ for comparison: 500, 1125, 6250, and 12500. For each chosen value of $K$, to guarantee the given privacy budget, we add DP Gaussian noise with variance $\sigma_i^2$ calculated according to (11). The results illustrated in Figure 2 demonstrate that the total number of iterations $K$ has a substantial impact on both training loss and testing accuracy. It is evident that selecting the proper value of $K = 3120$, as determined by our proposed approach, leads to the minimized loss and maximized accuracy. On the other hand, if a larger value of $K$ (e.g., 12500) or a smaller value (e.g., 500) is chosen, the training loss becomes larger and the model testing accuracy is lower. These findings validate the importance of selecting an appropriate value for $K$ to ensure optimal performance of PrivSGP-VR under a certain privacy budget.

Trade off between the maximized model utility and privacy guarantee. We conduct experiments by deploying the PrivSGP-VR algorithm on a network consisting of 16 nodes with a fixed value of $\delta = 10^{-5}$ for each node. The $\epsilon$ value for each node is varied from the set $\{1, 1.5, 2, 2.5, 3, 3.5, 4\}$. For each value of $\epsilon$, we determine the optimal total number of iterations $K^*$ using equation (12). Then, we execute PrivSGP-VR for $K^*$ iterations, along with two other $K$ values for comparative analysis. We incorporate the corresponding DP Gaussian noise with variance calculated according to equation (11). Figure 3 illustrates the trade-off between model utility (testing accuracy) and privacy under the optimized number of iterations. As the privacy budget $\epsilon$ diminishes (indicating a higher level of privacy protection), the maximized model utility deteriorates. This trade-off between privacy and maximized utility aligns with the theoretical insights outlined in Remark 2.

Verifying the effectiveness of variance reduction technique. To validate the effectiveness of the variance reduction technique employed by PrivSGP-VR, we conducted experiments to compare PrivSGP-VR with PrivSGP. For fair comparisons, we applied DP Gaussian noise with an identical variance of $\sigma_i^2 = 0.03$ for both PrivSGP-VR and PrivSGP. Moreover, both algorithms were executed for a fixed number of 3000 iterations. The results, as depicted in Figure 4, clearly illustrate that PrivSGP-VR outperforms PrivSGP in terms of both training loss and model testing accuracy. This validates the effectiveness of the variance reduction technique integrated into PrivSGP-VR.

Comparison with existing decentralized counterparts. We present experiments to compare the performance of PrivSGP-VR with other fully decentralized private stochastic algorithms DP$^2$-SGD and A(DP)$^2$SGD over 16 nodes with $(3, 10^{-5})$-DP guarantee for each node, when training ResNet-18 on Cifar-10.
nodes. The results shown in Figure 5 demonstrate that, under (3, 10^{-5})-DP guarantee for each node, our PrivSGP-VR outperforms DP²-SGD and A(DP)²-SGD in that PrivSGP-VR converges faster than the other two algorithms in both training loss and testing accuracy.

5.2 Shallow 2-layer Neural Network Training

We also provide additional experimental results for training 2-layer neural network on Mnist dataset which can be found in Appendix E in our full version [Zhu et al., 2024], and the experimental results under various settings are aligned with that of training ResNet-18 on Cifar-10 dataset.

6 Conclusion

We have proposed a differentially private decentralized learning method over time-varying directed communication topologies, termed PrivSGP-VR. Our analysis shows that under DP Gaussian noise with constant variance, PrivSGP-VR converges at a sub-linear rate \( O(1/\sqrt{nK}) \) which is independent of stochastic gradient variance. When given a certain privacy budget for each node, leveraging the moments account method, we derive an optimal number of iterations \( K \) to maximize the model utility. With this optimized \( K \), we achieve a tight utility bound which matches that of the server-client distributed counterparts, and exhibits an extra factor of \( 1/\sqrt{n} \) improvement compared to that of the existing decentralized counterparts. Extensive experiments are conducted to validate our theoretical findings.

Acknowledgments

This work is supported in parts by the National Key R&D Program of China under Grant No. 2022YFB3102100, and in parts by National Natural Science Foundation of China under Grants 62373323, 62088101, 62003302.

References


