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

Zehan Zhu[1] , **Yan Huang**[1] , **Xin Wang**[2] and **Jinming Xu**[1*]
[1]Zhejiang University, Hangzhou, China
[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 (ϵ, δ)-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 $\mathcal{O}(1/\sqrt{nK})$, 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 $\mathcal{O}\left(\sqrt{d\log\left(\frac{1}{\delta}\right)}/(\sqrt{n}J\epsilon)\right)$, 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)^{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[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:

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

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 disclose 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-

[*]Corresponding author.
[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 conterparts [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 conterparts. 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 (ϵ_i, δ_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 $\mathcal{O}(\frac{1}{\sqrt{nK}})$ 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 (ϵ_i, δ_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 $\mathcal{O}\left(\sqrt{d\log\left(\frac{1}{\delta}\right)}/(\sqrt{n}J\epsilon)\right)^2$ 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 counterparts (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]Here, we set $\epsilon_i = \epsilon$ and $\delta_i = \delta$ for utility bound comparison.

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[3] 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 provided 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 (A(DP)^2SGD) for non-convex problems based on AD-PSGD [Lian *et al.*, 2018],

[3]Here, by being distributed, we mean sever-client architecture.

Algorithm	Privacy	Utility	Communication rounds	Architecture
DP-SGD [Abadi *et al.*, 2016]	(ϵ,δ)-DP	$\frac{\sqrt{d\log\left(\frac{1}{\delta}\right)}}{J\epsilon}$	–	single node centralized
Distributed DP-SRM[1] [Wang *et al.*, 2019a]	(ϵ,δ)-DP global	$\frac{\sqrt{d\log\left(\frac{1}{\delta}\right)}}{nJ\epsilon}$	$\frac{n^2J\epsilon\sqrt{d}}{\sqrt{\log\left(\frac{1}{\delta}\right)}}$	n nodes server-client
LDP SVRG/SPIDER [Lowy *et al.*, 2022]	(ϵ,δ)-DP for each node	$\frac{\sqrt{d\log\left(\frac{1}{\delta}\right)}}{\sqrt{n}J\epsilon}$	$\frac{n^{\frac{3}{2}}J\epsilon\sqrt{d}}{\sqrt{\log\left(\frac{1}{\delta}\right)}}$	n nodes server-client
SoteriaFL-SAGA/SVRG [Li *et al.*, 2022]	(ϵ,δ)-DP for each node	$\frac{\sqrt{(1+\omega)d\log\left(\frac{1}{\delta}\right)}}{\sqrt{n}J\epsilon}$	$\frac{\sqrt{n}J\epsilon}{\sqrt{(1+\omega)d\log\left(\frac{1}{\delta}\right)}}$	n nodes server-client
A(DP)^{2}SGD[2] [Xu *et al.*, 2022]	(ϵ,δ)-DP for each node	$\frac{\sqrt{d\log\left(\frac{1}{\delta}\right)}}{J\epsilon}$	$\frac{J^2\epsilon^2}{d\log\left(\frac{1}{\delta}\right)}$	n nodes decentralized
PrivSGP-VR (Our Algorithm 1)	(ϵ,δ)-DP for each node	$\frac{\sqrt{d\log\left(\frac{1}{\delta}\right)}}{\sqrt{n}J\epsilon}$	$\frac{J^2\epsilon^2}{d\log\left(\frac{1}{\delta}\right)}$	n nodes decentralized

[1] Wang *et al.* [2019a] consider global (ϵ,δ)-DP that merely protects the privacy for the entire dataset while we consider (ϵ,δ)-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 (ϵ,δ)-DP for the entire dataset. We thus derive their utility bound in the sense of ensuring (ϵ,δ)-DP for each node for fair comparison.

Table 1: Comparison of existing differentially private stochastic algorithms for the non-convex problem in terms of privacy, utility and communication complexity. DP-SGD is the centralized (single-node) stochastic learning algorithm serving as a baseline. Distributed DP-SRM and LDP SVRG/SPIDER are server-client distributed learning algorithms without communication compression. SoteriaFL-SAGA/SVRG are server-client distributed learning algorithms with communication compression. ω is the parameter for unbiased compression in SoteriaFL-SAGA/SVRG ($\omega=0$ corresponds to no compression). A(DP)^{2}SGD is decentralized learning algorithm. For comparison, we set $\epsilon_i=\epsilon$ and $\delta_i=\delta$ for each node i in our PrivSGP-VR. The Big $\mathcal{O}$ notation is omitted for simplicity.

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 $\mathcal{O}\left(\sqrt{d\log\left(\frac{1}{\delta}\right)}/(J\epsilon)\right)$ is established for centralized learning with DP [Abadi *et al.*, 2016], and a utility bound of $\mathcal{O}\left(\sqrt{d\log\left(\frac{1}{\delta}\right)}/(\sqrt{n}J\epsilon)\right)$ 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 $\mathcal{G}^k=\left(\mathcal{V},\mathcal{E}^k\right)$, 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 $\mathcal{G}^k$ 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., $\mathbf{1}^\top P^k=\mathbf{1}^\top$, where $\mathbf{1}$ 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 $\bigcup_{k=lB}^{(l+1)B-1}\mathcal{E}^k$ is strongly connected and has diameter at most $\triangle$ for $\forall l\geqslant 0$.*

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

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

$$Pr\left[\mathcal{M}\left(\mathrm{x}\right)\in S\right]\leqslant e^{\epsilon}Pr\left[\mathcal{M}\left(\mathrm{x}'\right)\in S\right]+\delta,\qquad(2)$$

where the privacy budget ϵ denotes the privacy lower bound to measure a randomized query and δ is the probability of breaking this bound.

It can be observed that the smaller the values of ϵ and δ 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 (ϵ_i, δ_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 unblanceness 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_i^k; ii) the scalar Push-Sum weight w_i^k and iii) the de-biased parameter $z_i^k = x_i^k / w_i^k$, with the same initialization of $x_i^0 = z_i^0 = x^0 \in \mathbb{R}^d$ and $w_i^0 = 1$ for all nodes $i \in \{1, 2, ..., n\}$. At each iteration k, each node i updates as follows:

$$\texttt{Local SGD:} \quad x_i^{k+\frac{1}{2}} = x_i^k - \gamma \nabla f_i\left(z_i^k; \xi_i^k\right),$$

$$\texttt{Averaging:} \quad x_i^{k+1} = \sum_{j=1}^{n} P_{i,j}^k x_j^{k+\frac{1}{2}}, w_i^{k+1} = \sum_{j=1}^{n} P_{i,j}^k w_j^k,$$

$$\texttt{De-bias:} \quad z_i^{k+1} = x_i^{k+1} / w_i^{k+1},$$

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:

$$\nabla \tilde{f}_i(z_i^k; \xi_i^k) = \nabla f_i(z_i^k; \xi_i^k) + N_i^k \tag{3}$$

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

$$x_i^{k+\frac{1}{2}} = x_i^k - \gamma \nabla \tilde{f}_i(z_i^k; \xi_i^k) = x_i^k - \gamma\left(\nabla f_i(z_i^k; \xi_i^k) + N_i^k\right). \tag{4}$$

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

Algorithm 1 PrivSGP-VR

Initialization: $x_i^0 = z_i^0 = x^0 \in \mathbb{R}^d$, $w_i^0 = 1$ and privacy budget (ϵ_i, δ_i) for all $i \in \mathcal{V}$, step size $\gamma > 0$, and total number of iterations K.

for $j \in \{1, 2, ..., J\}$, at node i, **do**
  Initializes $\nabla f_i(\phi_{i,j}; j) = \nabla f_i(z_i^0; j)$
end for
for $k = 0, 1, 2, ..., K-1$, at node i, **do**
  Randomly samples a local training data ξ_i^k with the sampling probability $\frac{1}{J}$;
  Computes the corrected gradient by

$$g_i^k = \nabla f_i(z_i^k; \xi_i^k) - \nabla f_i(\phi_{i,\xi_i^k}; \xi_i^k) + \frac{1}{J} \sum_{j=1}^{J} \nabla f_i(\phi_{i,j}; j);$$

  Stores gradient: $\nabla f_i(\phi_{i,\xi_i^k}; \xi_i^k) = \nabla f_i(z_i^k; \xi_i^k)$;
  Adds noise $\tilde{g}_i^k = g_i^k + N_i^k$, where $N_i^k \in \mathbb{R}^d \sim \mathcal{N}\left(0, \sigma_i^2 \mathbb{I}_d\right)$ and σ_i is defined in Theorem 2;
  Generates intermediate model parameter: $x_i^{k+\frac{1}{2}} = x_i^k - \gamma \tilde{g}_i^k$;
  Sends $\left(x_i^{k+\frac{1}{2}}, w_i^k\right)$ to out-neighbors;
  Receives $\left(x_j^{k+\frac{1}{2}}, w_j^k\right)$ from in-neighbors;
  Updates x_i^{k+1} by: $x_i^{k+1} = \sum_{j=1}^{n} P_{i,j}^k x_j^{k+\frac{1}{2}}$;
  Updates w_i^{k+1} by: $w_i^{k+1} = \sum_{j=1}^{n} P_{i,j}^k w_j^k$;
  Updates z_i^{k+1} by: $z_i^{k+1} = x_i^{k+1} / w_i^{k+1}$;
end for

i maintains a stochastic gradient table for all of its own local data samples. At each iteration k, after computing the stochastic gradient $\nabla f_i\left(z_i^k; \xi_i^k\right)$, node i does not perform a local differentially private SGD step using $\nabla f_i\left(z_i^k; \xi_i^k\right)$ directly (c.f., (4)). Instead, $\nabla f_i\left(z_i^k; \xi_i^k\right)$ is corrected by subtracting the previously stored stochastic gradient corresponding to the ξ_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 ξ_i^k-th data sample in the table with $\nabla f_i\left(z_i^k; \xi_i^k\right)$. To better understand this process, let

$$\phi_{i,j}^{k+1} = \begin{cases} \phi_{i,j}^k & , j \neq \xi_i^k \\ z_i^k & , j = \xi_i^k \end{cases}, \tag{5}$$

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

$$g_i^k \triangleq \nabla f_i\left(z_i^k; \xi_i^k\right) - \nabla f_i\left(\phi_{i,\xi_i^k}^k; \xi_i^k\right) + \frac{1}{J} \sum_{j=1}^{J} \nabla f_i\left(\phi_{i,j}^k; j\right) \tag{6}$$

is the corrected stochastic gradient of node i at iteration k. As a result, we replace the original stochastic gradient $\nabla f_i\left(z_i^k;\xi_i^k\right)$ in (4) with g_i^k, leading to the following new local differentially private SGD step, i.e.,

$$x_i^{k+\frac{1}{2}} = x_i^k - \gamma\left(g_i^k + N_i^k\right), \tag{7}$$

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.,*

$$\mathbb{E}\left[\nabla f_i(x;\xi_i)\right] = \nabla f_i(x). \tag{8}$$

Assumption 5 (Bounded Data Heterogeneity). *There exists a positive constant b^2 such that for any node i and $\forall x \in \mathbb{R}^d$,*

$$\left\|\nabla f_i(x) - \nabla f(x)\right\|^2 \leqslant b^2. \tag{9}$$

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 $\gamma = \sqrt{\frac{n}{K}}$, 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 $K \geqslant \hat{K}(C,q)$, we have*

$$\begin{aligned}&\frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f\left(z_i^k\right)\right\|^2\right]\\ &\leqslant \frac{13F^0 + 6L\left\|x^0\right\|^2 + 18Lb^2 + 24L\cdot\frac{d}{n}\sum_{i=1}^{n}\sigma_i^2}{\sqrt{nK}},\end{aligned} \tag{10}$$

where $F^0 = f\left(x^0\right) - f^*$, C and q can be found in Lemma 7, and the definition of $\hat{K}(C,q)$ 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]. $\square$

Remark 1. *Under DP Gaussian noise with a constant variance, the above result suggests that the convergence rate for PrivSGP-VR is $\mathcal{O}(\frac{1}{\sqrt{nK}})$, which is independent of stochastic gradient variance ζ^2 with $\mathbb{E}\left[\left\|\nabla f_i(x;\xi_i) - \nabla f_i(x)\right\|^2\right] \leqslant \zeta^2$ 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 σ_i^2 of the DP Gaussian noise needed to be added according to the total number of iterations K and the given privacy budget (ϵ_i, δ_i), 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., $G = \max_{k,i}\left\|\nabla f_i\left(z_i^k;\xi_i^k\right)\right\| < \infty$. There exist constants c_1 and c_2 such that, given the total number of iterations K for Algorithm 1, (ϵ_i,δ_i)-differential privacy can be guaranteed for each node i, for any $\epsilon_i < \frac{c_1 K}{J^2}$ and $\delta_i \in (0,1)$, if N_i^k is drawn from the Gaussian distribution $\mathcal{N}\left(0,\sigma_i^2\mathbb{I}_d\right)$ with*

$$\sigma_i = 3c_2G\frac{\sqrt{K\log\left(\frac{1}{\delta_i}\right)}}{J\epsilon_i}. \tag{11}$$

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

As highlighted in Theorem 2, it is evident that when a certain privacy budget (ϵ_i,δ_i) is given, a larger value of K requires the added DP Gaussian noise with a larger variance σ_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 (ϵ_i,δ_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 (ϵ_i,δ_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*

$$K = \frac{\left(13F^0 + 6L\left\|x^0\right\|^2 + 18Lb^2\right)J^2 n}{216Ldc_2^2G^2\sum_{i=1}^{n}\frac{1}{\epsilon_i^2}\log\left(\frac{1}{\delta_i}\right)}, \tag{12}$$

then we have

$$\begin{aligned}&\frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f\left(z_i^k\right)\right\|^2\right]\\ &\leqslant 12c_2G\cdot\\ &\frac{\sqrt{6Ld\left(13F^0 + 6L\left\|x^0\right\|^2 + 18Lb^2\right)\sum_{i=1}^{n}\frac{1}{\epsilon_i^2}\log\left(\frac{1}{\delta_i}\right)}}{nJ}.\end{aligned} \tag{13}$$

Proof. Substituting the Gaussian noise level in (11) into (10), we have

$$\begin{aligned}&\frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\nabla f\left(z_i^k\right)\right\|^2\right]\\ \leqslant&\frac{13F^0+6L\left\|x^0\right\|^2+18Lb^2}{\sqrt{nK}}\\ &+\sqrt{K}\cdot\frac{216Ldc_2^2G^2}{J^2\sqrt{n}}\cdot\frac{1}{n}\sum_{i=1}^{n}\frac{1}{\epsilon_i^2}\log\left(\frac{1}{\delta_i}\right).\end{aligned}\tag{14}$$

Regarding the right hand side of (14) as a function of K, we can obtain the optimal value of K (c.f., (12)) and error bound (c.f., (13)) by minimizing this function. □

Remark 2. *Corollary 1 provides a valuable insight regarding the optimization of the total number of iterations K under certain privacy budget. It shows that there exists an optimal value of K that minimizes the error bound and thus maximizes the model accuracy. This suggests that simply increasing the value of K will not necessarily lead to improved results, and it is important to carefully select the appropriate number of iterations. Furthermore,* (13) *highlights the trade-off between privacy and maximized model utility. It shows that stronger privacy protection, represented by smaller privacy budget (ϵ_i,δ_i), leads to a larger minimum error bound and thus worse maximized model utility. This finding underscores the inherent tension between privacy and model performance in decentralized learning scenarios.*

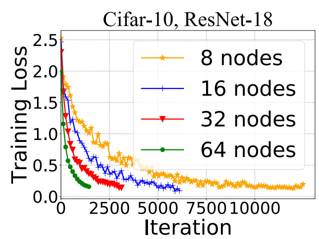
Remark 3. *Note that if one set $\epsilon_i=\epsilon$ and $\delta_i=\delta$ for each node i, the utility bound of the proposed PrivSGP-VR* (13) *will be reduced to $\mathcal{O}\left(\sqrt{d\log\left(\frac{1}{\delta}\right)}/(\sqrt{n}J\epsilon)\right)$, achieving the same utility guarantee as differentially 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 sever-client distributed counterparts. In addition, PrivSGP-VR recovers the same utility $\mathcal{O}\left(\sqrt{d\log\left(\frac{1}{\delta}\right)}/(J\epsilon)\right)$ 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)^{2}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

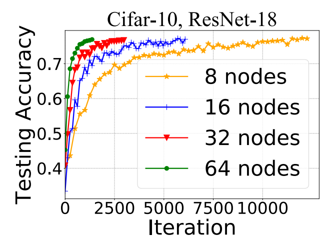
We conduct extensive experiments to validate the theoretical findings for the proposed PrivSGP-VR under various settings. All experiments are deployed in a high performance computer with Intel Xeon E5-2680 v4 CPU @ 2.40GHz and 8 Nvidia RTX 3090 GPUs, and are implemented with distributed communication 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^2 can be estimated by leveraging the method introduced in [Wang *et al.*, 2019b; Luo *et al.*, 2021]. The values of these parameters are $L=25, G^2=100, f(\bar{x}^0)-f^*=2.8, b^2=500000$ and $\left\|x^0\right\|^2=780000$, for ResNet-18 training task.



(a) Training loss

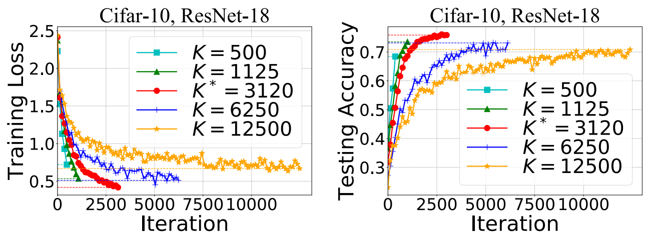


(b) Testing accuracy

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 $\sigma_i^2=0.03$ 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



(a) Training loss (b) Testing accuracy

Figure 2: Comparison of convergence performance for PrivSGP-VR over 16 nodes by setting different total number of iterations K under a certain privacy budget, when training ResNet-18 on Cifar-10.

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 σ_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.

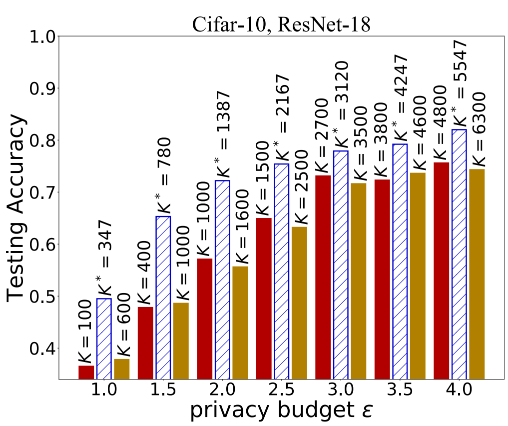
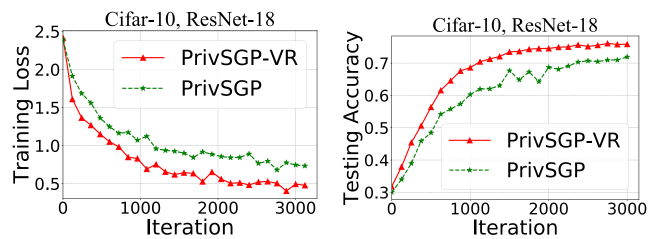


Figure 3: Performance of running PrivSGP-VR for K^* (K) iterations under different certain privacy budgets ϵ, when training ResNet-18 on Cifar-10.

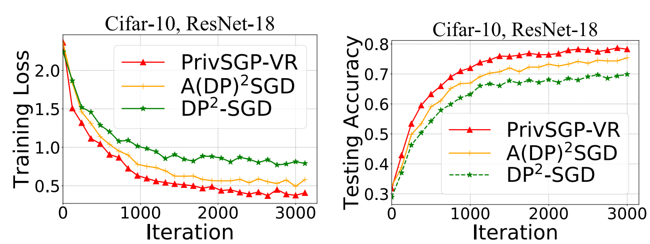
Trade off between the maximized model utility and privacy guarantee. We conduct experiments by deploying the PrivSGP-VR algorithm on a network consisting of 16 nodes with a fixed value of $\delta = 10^{-5}$ for each node. The ϵ value for each node is varied from the set $\{1, 1.5, 2, 2.5, 3, 3.5, 4\}$. For each value of ϵ, 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 ϵ 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.



(a) Training loss (b) Testing accuracy

Figure 4: Comparison of convergence performance for PrivSGP-VR with PrivSGP over 16 nodes under the same DP Gaussian noise variance, when training ResNet-18 on Cifar-10.

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.



(a) Training loss (b) Testing accuracy

Figure 5: Comparison of convergence performance for PrivSGP-VR with 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.

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. We implement all three algorithms on an undirected ring graph consisting of 16 nodes. The results shown in Figure 5 demonstrate that, under $(3, 10^{-5})$-DP guarantee for each node, our PrivSGP-VR outperforms DP^2-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 $\mathcal{O}(1/\sqrt{nK})$ which is independent of stochastic gradient variance. When given a certain privacy budget for each node, leveraging the moments accountant 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

[Abadi *et al.*, 2016] 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, 2016.

[Assran *et al.*, 2019] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019.

[Bun and Steinke, 2016] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th International Conference on Theory of Cryptography*, pages 635–658. Springer, 2016.

[Carlini *et al.*, 2019] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, volume 267, 2019.

[Chen *et al.*, 2020] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33:13773–13782, 2020.

[Cheng *et al.*, 2018] 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.

[Cheng *et al.*, 2019] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Syed Zawad, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. Towards decentralized deep learning with differential privacy. In *International Conference on Cloud Computing*, pages 130–145. Springer, 2019.

[Defazio *et al.*, 2014] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems*, 27, 2014.

[Deng, 2012] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[Dwork and Lei, 2009] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *the 41st ACM Annual Symposium on Theory of Computing*, pages 371–380, 2009.

[Dwork *et al.*, 2006] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[Dwork *et al.*, 2010] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *the 51st IEEE Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[Dwork *et al.*, 2014] 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.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[Kempe *et al.*, 2003] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *the 44th IEEE Annual Symposium on Foundations of Computer Science*, pages 482–491. IEEE, 2003.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[Langer *et al.*, 2020] Matthias Langer, Zhen He, Wenny Rahayu, and Yanbo Xue. Distributed training of deep learning models: A taxonomic perspective. *IEEE Transactions on Parallel and Distributed Systems*, 31(12):2802–2818, 2020.

[Li *et al.*, 2014] Mu Li, David G Andersen, Jun Woo Park, Alexander J Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 583–598, 2014.

[Li *et al.*, 2019] Yanan Li, Shusen Yang, Xuebin Ren, and Cong Zhao. Asynchronous federated learning with differential privacy for edge intelligence. *arXiv preprint arXiv:1912.07902*, 2019.

[Li *et al.*, 2022] Zhize Li, Haoyu Zhao, Boyue Li, and Yuejie Chi. SoteriaFL: A unified framework for private federated learning with communication compression. *Advances in Neural Information Processing Systems*, 35:4285–4300, 2022.

[Lian *et al.*, 2017] 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. *Advances in Neural Information Processing Systems*, 30, 2017.

[Lian *et al.*, 2018] Xiangru Lian, Wei Zhang, Ce Zhang, and Ji Liu. Asynchronous decentralized parallel stochastic gradient descent. In *International Conference on Machine Learning*, pages 3043–3052. PMLR, 2018.

[Liu *et al.*, 2022] Tianyu Liu, Boya Di, Bin Wang, and Lingyang Song. Loss-privacy tradeoff in federated edge learning. *IEEE Journal of Selected Topics in Signal Processing*, 16(3):546–558, 2022.

[Lowy *et al.*, 2022] Andrew Lowy, Ali Ghafelebashi, and Meisam Razaviyayn. Private non-convex federated learning without a trusted server. *arXiv preprint arXiv:2203.06735*, 2022.

[Luo *et al.*, 2021] Bing Luo, Xiang Li, Shiqiang Wang, Jianwei Huang, and Leandros Tassiulas. Cost-effective federated learning design. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 1–10. IEEE, 2021.

[McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *International Conference on Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, 2017.

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

[Truex *et al.*, 2019] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[Truex *et al.*, 2020] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. LDP-Fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 61–66, 2020.

[Wang and Nedic, 2024] Yongqiang Wang and Angelia Nedic. Tailoring gradient methods for differentially private distributed optimization. *IEEE Transactions on Automatic Control*, 69(2):872–887, 2024.

[Wang *et al.*, 2017] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. *Advances in Neural Information Processing Systems*, 30, 2017.

[Wang *et al.*, 2019a] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving stochastic nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

[Wang *et al.*, 2019b] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[Wang *et al.*, 2019c] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 2512–2520. IEEE, 2019.

[Wang *et al.*, 2020] Di Wang, Hanshen Xiao, Srinivas Devadas, and Jinhui Xu. On differentially private stochastic convex optimization with heavy-tailed data. In *International Conference on Machine Learning*, pages 10081–10091. PMLR, 2020.

[Wei *et al.*, 2020] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[Wei *et al.*, 2021] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Hang Su, Bo Zhang, and H Vincent Poor. User-level privacy-preserving federated learning: Analysis and performance optimization. *IEEE Transactions on Mobile Computing*, 21(9):3388–3401, 2021.

[Wu *et al.*, 2020] Nan Wu, Farhad Farokhi, David Smith, and Mohamed Ali Kaafar. The value of collaboration in convex machine learning with differential privacy. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 304–317. IEEE, 2020.

[Xu *et al.*, 2022] Jie Xu, Wei Zhang, and Fei Wang. A(DP)^2SGD: Asynchronous decentralized parallel stochastic gradient descent with differential privacy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8036–8047, 2022.

[Yu *et al.*, 2021] Dongxiao Yu, Zongrui Zou, Shuzhen Chen, Youming Tao, Bing Tian, Weifeng Lv, and Xiuzhen Cheng. Decentralized parallel sgd with privacy preservation in vehicular networks. *IEEE Transactions on Vehicular Technology*, 70(6):5211–5220, 2021.

[Zeng *et al.*, 2021] Yiming Zeng, Yixuan Lin, Yuanyuan Yang, and Ji Liu. Differentially private federated temporal difference learning. *IEEE Transactions on Parallel and Distributed Systems*, 33(11):2714–2726, 2021.

[Zhang *et al.*, 2020] Xin Zhang, Minghong Fang, Jia Liu, and Zhengyuan Zhu. Private and communication-efficient edge learning: a sparse differential gaussian-masking distributed sgd approach. In *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, pages 261–270, 2020.

[Zhou *et al.*, 2023] Yipeng Zhou, Xuezheng Liu, Yao Fu, Di Wu, Jessie Hui Wang, and Shui Yu. Optimizing the numbers of queries and replies in convex federated learning with differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 20(6):4823–4837, 2023.

[Zhu *et al.*, 2024] Zehan Zhu, Yan Huang, Xin Wang, and Jinming Xu. PrivSGP-VR: Differentially private variance-reduced stochastic gradient push with tight utility bounds. *arXiv preprint arXiv:2405.02638*, 2024.