Differentially Private Gradient-Tracking-Based Distributed Stochastic Optimization Over Directed Graphs

Jialong Chen, Jimin Wang, *Member, IEEE*, and Ji-Feng Zhang, *Fellow, IEEE*

***Abstract*—This article proposes a differentially private gradient-tracking-based distributed stochastic optimization algorithm over directed graphs. In particular, privacy noises are incorporated into each agent's state and tracking variable to mitigate information leakage, after which the perturbed states and tracking variables are transmitted to neighbors. We design two novel schemes for the step-sizes and the sampling number within the algorithm. The sampling parameter-controlled subsampling method employed by both schemes enhances the differential privacy level and ensures a finite cumulative privacy budget even over infinite iterations. The algorithm achieves both almost sure and mean square convergence for nonconvex objectives. Furthermore, when nonconvex objectives satisfy the Polyak–Łojasiewicz condition, *Scheme (S1)* achieves a polynomial mean square convergence rate, and *Scheme (S2)* achieves an exponential mean square convergence rate. The tradeoff between privacy and convergence is presented. The effectiveness of the algorithm and its superior performance compared to existing works are illustrated through numerical examples of distributed training on the benchmark datasets "MNIST" and "CIFAR-10."**

***Index Terms*—Differential privacy, directed graphs, distributed stochastic optimization, exponential mean square convergence rate, gradient tracking.**

I. Introduction

DISTRIBUTED optimization allows cooperative agents to compute and update their state variables through interagent communication to obtain an optimal solution of a common optimization problem [1]. Distributed stochastic optimization, a branch of distributed optimization, addresses scenarios where objectives are stochastic [2]. This approach has found extensive applications across multiple domains, including distributed machine learning [3], cloud-based control systems [4], and the Internet of Things [5]. While it is frequently utilized in distributed stochastic optimization because of its adaptability in communication-efficient methods [6] and simplicity in algorithm structure [7], stochastic gradient descent (SGD) does not guarantee the convergence over directed graphs [8, eq. (6)], and cannot achieve the exponential convergence rate [9, Thm. 2], [10, eq. (2)]. To address these issues, the gradient-tracking method has been proposed over undirected graphs (see [11] and [12]). By developing tracking variables to track global stochastic gradients, the authors in , [11], [12], and [13] initially achieve the exponential convergence rate. The convergence analysis is further extended to directed graphs in [14], [15], [16], and [17]. However, the authors in [14], [15], and [16] prove convergence under the assumption that weight matrices are row-stochastic and column-stochastic, which is often difficult to be satisfied in various practical scenarios (see, e.g., [4] and [5]). Wang and Başar [17] achieved the convergence by employing the two-time-scale step-sizes method, which removes the assumption that weight matrices are row-stochastic and column-stochastic, while requiring that the level sets of objectives are bounded.

When cooperative agents exchange information to address a distributed stochastic optimization problem, adversaries can infer stochastic gradients from the exchanged information, and further obtain agents' sensitive information through model inversion attacks [18], [19]. To address this issue, various privacy-preserving techniques have been developed [20], such as homomorphic encryption [21], [22], state decomposition [23], random coupling weights [24], uncoordinated step-sizes [25], network augmentation [26], and adding noises [27], [28], [29], [30]. Because of its simplicity of use and immunity to postprocessing, differential privacy [28], [29] has attracted considerable interest and has been extensively applied in distributed optimization for both deterministic and stochastic objectives. When objectives are deterministic, based on the gradient-tracking method, differentially private distributed optimization has been well developed in [31], [32], [33], [34], [35], and [36]. Among others, the authors [31], [32], [33], [34], [36] successfully achieved the finite cumulative differential privacy budget over infinite iterations. However, the difficulty caused by stochastic objectives makes the methods in [31], [32], [33], [34], [35], and [36] unsuitable to differentially private distributed stochastic optimization. In

Received 25 December 2025; revised 19 March 2026; accepted 25 March 2026. Date of publication 30 March 2026; date of current version 31 August 2026. This work was supported by the National Natural Science Foundation of China under Grant 62433020 and Grant T2293770. Recommended by Associate Editor Y. Wang. *(Corresponding author: Jimin Wang.)*

Jialong Chen is with the State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: chenjialong23@mails.ucas.ac.cn).

Jimin Wang is with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China, and also with the Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, Beijing 100083, China (e-mail: jimwang@ustb.edu.cn).

Ji-Feng Zhang is with the School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zheng Zhou 450007, China, and also with the State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China (e-mail: jif@iss.ac.cn).

Digital Object Identifier 10.1109/TAC.2026.3678834

1558-2523 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

TABLE I
COMPARISON WITH EXISTING WORKS IN DISTRIBUTED OFFLINE STOCHASTIC OPTIMIZATION

	Privacy budget	Convergence	Convergence rate	Graph topology	Key assumptions on objectives to achieve convergence
[10]	N/A	Mean square	$O(\rho^K)$	Row-stochastic and column-stochastic connected directed graphs	Strongly convex
[13]	N/A	✗	$O(\rho^K)$	Strongly connected directed graphs	Strongly convex
[17]	N/A	Almost sure	-	Strongly connected directed graphs	Convex, bounded level sets
[37]	Per-iteration	Mean square	$O(\frac{1}{K^{\frac{1}{3}}})$	Row-stochastic and column-stochastic undirected graphs	Strongly convex and nonconvex, bounded gradients
[40]	Per-iteration	Almost sure	-	Row-stochastic and column-stochastic undirected graphs	Nonconvex, bounded gradients
[41]	Per-iteration	Mean square	$O(1)$	Connected undirected graphs	Convex and nonconvex, bounded gradients
[47]	Finite cumulative	Mean square	$O(\frac{1}{K^{\frac{1}{3}}})$	Row-stochastic and column-stochastic undirected graphs	Nonconvex, Polyak–Łojasiewicz condition
Scheme *(S1)* (This work)	Finite cumulative	Mean square and almost sure	$O(\frac{1}{K^{\frac{1}{3}}})$	Directed graphs with spanning trees	Nonconvex, Polyak–Łojasiewicz condition
Scheme *(S2)* (This work)	Finite cumulative	Mean square and almost sure	$O(\rho^K)$	Directed graphs with spanning trees	Nonconvex

"N/A" means that the privacy protection is not considered, "✗" means that the convergence is not achieved, and "-" means that the convergence rate is not given.

addition, to achieve convergence, (strongly) convex objectives [31], [32], [33], [34], [36] and nonconvex objectives with the Polyak–Łojasiewicz condition [35] are required. However, these requirements may be hard to be satisfied or verified in practice.

When objectives are stochastic, a method based on SGD has been proposed for differentially private distributed stochastic optimization. Some interesting works can be found in [37], [38], [39], [40], [41], [42], [43], while these works only give the per-iteration differential privacy budget, and thus cannot protect the sensitive information over infinite iterations. Fortunately, by using sequentially acquired data samples inherent in online learning [44], [45], the time-varying sampling number method [46], and the sampling parameter-controlled subsampling method [47], the finite cumulative differential privacy budget over infinite iterations is given. However, the differential privacy is tailored for distributed SGD in [44], [46], and [47], respectively. Since the gradient-tracking method has shown advantages over the distributed-SGD method regarding the convergence over directed graphs, the differentially private gradient tracking-based distributed online stochastic optimization with streaming data over directed graphs is first studied in [45]. To the best of the authors' knowledge, differentially private gradient-tracking-based distributed *offline* stochastic optimization with predownloaded/local datasets over directed graphs has not been studied yet. As a result, the differentially private distributed *offline* stochastic optimization based on the gradient-tracking method is a challenging issue, especially on how to achieve the finite cumulative differential privacy budget even over infinite iterations, the almost sure and mean square convergence for nonconvex objectives without the Polyak–Łojasiewicz condition, and the exponential mean square convergence rate.

Summarizing the discussion above, we propose a new differentially private gradient-tracking-based distributed stochastic optimization algorithm with two schemes of step-sizes and the sampling number over directed graphs. *Scheme (S1)* employs the polynomially decreasing step-sizes and the increasing sampling number with the maximum iteration number. *Scheme (S2)* employs constant step-sizes and the exponentially increasing sampling number with the maximum iteration number. Comparison with existing works in distributed offline stochastic optimization is presented in Table I, and the main contribution of this article is as follows.

1) The sampling parameter-controlled subsampling method is employed to enhance the differential privacy level of the algorithm. The algorithm with both schemes achieves the finite cumulative differential privacy budget even over infinite iterations. To the best of the authors' knowledge, a finite cumulative differential privacy budget over infinite iterations is achieved in differentially private gradient-tracking-based distributed *offline* stochastic optimization for the first time.
2) The almost sure and mean square convergence of the algorithm is given for nonconvex objectives without the Polyak–Łojasiewicz condition. Furthermore, when nonconvex objectives satisfy the Polyak–Łojasiewicz condition, the polynomial mean square convergence rate is achieved for *Scheme (S1)*, and the exponential mean square convergence rate is achieved for *Scheme (S2)*.
3) Two schemes are shown to achieve the finite cumulative differential privacy budget over infinite iterations and mean square convergence simultaneously. For *Scheme (S1)*, the polynomial mean square convergence rate and the cumulative differential privacy budget are achieved simultaneously even over infinite iterations for general privacy noises, including decreasing, constant, and increasing privacy noises. For *Scheme (S2)*, the exponential mean square convergence rate and the cumulative differential privacy budget are achieved simultaneously even over infinite iterations.

The rest of this article is organized as follows. Section II presents preliminaries and the problem formulation. Section III provides the algorithm with its convergence and privacy analysis. Section IV verifies the effectiveness of the algorithm through numerical examples of distributed training on the benchmark datasets "MNIST" and "CIFAR-10." Finally, Section V concludes this article.

Notations. $\mathbb{R}$, $\mathbb{C}$, and $\mathbb{R}^n$ denote the set of real numbers, the set of complex numbers, and n-dimensional Euclidean space,

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

respectively. $\mathbf{1}_n$ denotes an n-dimensional vector whose elements are all 1, and $\|x\|$ denotes the standard Euclidean norm of a vector x. $X \sim \text{Lap}(b)$ refers to a random variable that has a Laplacian distribution with the variance parameter $b > 0$, and the probability density function of the random variable X is given by $p(x;b) = \frac{1}{2b}\exp(-\frac{|x|}{b})$. For a matrix $A \in \mathbb{R}^{n\times n}$, $A^\top$ and $\rho(A)$ stand for its transpose and spectral radius, respectively. $\langle\cdot,\cdot\rangle$ denotes the inner product. $(\Omega, \mathcal{F}, \mathbb{P})$ denotes a probability space, $\mathbb{P}(B)$ and $\mathbb{E}X$ stand for the probability of an event $B \in \mathcal{F}$ and the expectation of the random variable X, respectively. $\otimes$ denotes the Kronecker product of matrices. $\lfloor a \rfloor$ denotes the largest integer, which is not larger than a. For a differentiable function $f(x)$, $\nabla f(x)$ denotes its gradient at the point x. For a vector $x = [x_1, x_2, \ldots, x_n]^\top \in \mathbb{R}^n$, the notation $\text{diag}(x)$ denotes the diagonal matrix with diagonal elements being $x_1, x_2, \ldots, x_n$. For a complex number $\varpi \in \mathbb{C}$, $\text{Re}(\varpi)$ stands for its real part. $\mathbb{I}_{\{\cdot\}}$ denotes the indicator function, whose value is 1 if its argument is true, and 0, otherwise.

II. Preliminaries and Problem Formulation

A. Graph Theory

In this article, we consider a network of n agents that exchange the information over two different directed graphs $\mathcal{G}_\mathcal{R} = (\mathcal{V}, \mathcal{E}_\mathcal{R})$ and $\mathcal{G}_\mathcal{C} = (\mathcal{V}, \mathcal{E}_\mathcal{C})$. $\mathcal{V} = \{1, 2, \ldots, n\}$ is the set of all agents, and $\mathcal{E}_\mathcal{R}$ and $\mathcal{E}_\mathcal{C}$ are sets of directed edges in $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_\mathcal{C}$, respectively. In our gradient-tracking algorithm, agents exchange state variables over $\mathcal{G}_\mathcal{R}$ and tracking variables over $\mathcal{G}_\mathcal{C}$. Directed graphs $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_\mathcal{C}$ are induced by the weight matrix $\mathcal{R} = (\mathcal{R}_{ij})_{i,j=1,\ldots,n}$ and $\mathcal{C} = (\mathcal{C}_{ij})_{i,j=1,\ldots,n}$, respectively. Any element $\mathcal{R}_{ij}$ of $\mathcal{R}$ is either strictly positive if Agent i can receive Agent j's state variable, or 0, otherwise. The same property holds for any element $\mathcal{C}_{ij}$ of C. For any $i \in \mathcal{V}$, its in-neighbor and out-neighbor sets of over $\mathcal{G}_\mathcal{R}$ are defined as $\mathcal{N}^-_{\mathcal{R},i} = \{j \in \mathcal{V} : \mathcal{R}_{ij} > 0\}$ and $\mathcal{N}^+_{\mathcal{R},i} = \{j \in \mathcal{V} : \mathcal{R}_{ji} > 0\}$, respectively. Similarly, Agent i's in-neighbor and out-neighbor sets over $\mathcal{G}_\mathcal{C}$ are defined as $\mathcal{N}^-_{\mathcal{C},i}$ and $\mathcal{N}^+_{\mathcal{C},i}$, respectively. Denote the in-Laplacian matrix of $\mathcal{R}$ and the out-Laplacian matrix of $\mathcal{C}$ as $\mathcal{L}_1 = \text{diag}(\mathcal{R}\cdot\mathbf{1}_n) - \mathcal{R}$ and $\mathcal{L}_2 = \text{diag}(\mathbf{1}_n^\top \cdot \mathcal{C}) - \mathcal{C}$, respectively. Then, the assumption about directed graphs $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_\mathcal{C}$ is given as follows.

Assumption 1: Let $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_{\mathcal{C}^\top}$ be directed graphs induced by nonnegative matrices $\mathcal{R}$ and $\mathcal{C}^\top$, respectively. Then, both $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_{\mathcal{C}^\top}$ contain at least one spanning tree. Moreover, there exists at least one agent being a root of spanning trees in both $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_{\mathcal{C}^\top}$.

Remark 1: Directed graphs in Assumption 1 are allowed to have self-loops, which are commonly used in distributed stochastic optimization (see, e.g., [9], [10], [11], and [15]). More importantly, directed graphs in Assumption 1 are more general than connected undirected graphs in [12], [31], [35], [37], [38], [39], [40], [41], [44], [46], and [47], row-stochastic and column-stochastic connected directed graphs in [6], [10], [11], [15], and [24], and strongly connected directed graphs in [13], [17], and [45]. In addition, by [48, Thm. 3.8], Assumption 1 is a necessary condition for the consensus of agents' state and tracking variables.

Based on Assumption 1, we have the following useful lemma for weight matrices $\mathcal{R}$ and $\mathcal{C}$.

Lemma 1: If Assumption 1 holds, then following statements hold.

(i) Let $\{\varpi_1^{(1)}, \ldots, \varpi_n^{(1)}\}$ be eigenvalues of the matrix $\mathcal{L}_1$ such that $|\varpi_1^{(1)}| \le \cdots \le |\varpi_n^{(1)}|$, and $\{\varpi_1^{(2)}, \cdots, \varpi_n^{(2)}\}$ be eigenvalues of the matrix $\mathcal{L}_2$ such that $|\varpi_1^{(2)}| \le \cdots \le |\varpi_n^{(2)}|$. Then, $\varpi_1^{(1)} = \varpi_1^{(2)} = 0$, and $\text{Re}(\varpi_l^{(1)}) > 0$ and $\text{Re}(\varpi_l^{(2)}) > 0$ for any $l = 2, \ldots, n$.

(ii) Let matrices $W_1 = I_n - \frac{1}{n}\mathbf{1}_n v_1^\top$, $W_2 = I_n - \frac{1}{n}v_2\mathbf{1}_n^\top$, and step-sizes α_K, β_K satisfy

$$0 < \alpha_K < \min\{\min_{i\in\mathcal{V}}\{\frac{1}{\sum_{j\in\mathcal{N}^-_{\mathcal{R},i}}\mathcal{R}_{ij}}\}, \min_{l=2,\ldots,n}\{\frac{\text{Re}(\varpi_l^{(1)})}{1+|\varpi_l^{(1)}|^2}\}\},$$

$$0 < \beta_K < \min\{\min_{i\in\mathcal{V}}\{\frac{1}{\sum_{j\in\mathcal{N}^+_{\mathcal{C},i}}\mathcal{C}_{ji}}\}, \min_{l=2,\ldots,n}\{\frac{\text{Re}(\varpi_l^{(2)})}{1+|\varpi_l^{(2)}|^2}\}\}$$

Then, there exist unique nonnegative vectors $v_1, v_2 \in \mathbb{R}^n$ such that $v_1^\top(I_n - \alpha_K\mathcal{L}_1) = v_1^\top$, $(I_n - \beta_K\mathcal{L}_2)v_2 = v_2$, $v_1^\top\mathbf{1}_n = n$, $v_2^\top\mathbf{1}_n = n$, and $v_1^\top v_2 > 0$, and there exist $r_1, r_2 > 0$ such that $\rho(W_1 - \alpha_K\mathcal{L}_1) \le 1 - r_1\alpha_K$ and $\rho(W_2 - \beta_K\mathcal{L}_2) \le 1 - r_2\beta_K$.

Proof: Due to the space limitation, we leave the proof to [52]. ■

B. Problem Formulation

In this article, the following distributed stochastic optimization problem is considered:

$$\min_{x\in\mathbb{R}^d} F(x) = \min_{x\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^n f_i(x), f_i(x) = \mathbb{E}_{\xi_i\sim\mathscr{D}_i}[\ell_i(x,\xi_i)] \quad (1)$$

where x is available to all agents, $\ell_i(x,\xi_i)$ is a local objective, which is private to Agent i, ξ_i is a random variable drawn from an unknown probability distribution $\mathscr{D}_i$, and $\mathscr{D}_i$ is not required to be independent and identically distributed for any $i \in \mathcal{V}$. In practice, since the probability distribution $\mathscr{D}_i$ is difficult to obtain, it is usually replaced by the dataset $\mathcal{D}_i = \{\xi_{i,l} \in \mathbb{R}^r, l = 1, \ldots, D\}$. Then, (1) can be rewritten as the following empirical risk minimization problem:

$$\min_{x\in\mathbb{R}^d} F(x) = \min_{x\in\mathbb{R}^d}\frac{1}{n}\sum_{i=1}^n f_i(x), f_i(x) = \frac{1}{D}\sum_{l=1}^D \ell_i(x,\xi_{i,l}). \quad (2)$$

When solving the problem (2), a stochastic first-order oracle is often required [49], which returns a sampled gradient $g_i(x,\lambda_i)$ of the local objective $\ell(x,\lambda_i)$ for any $i\in\mathcal{V}$, $x\in\mathbb{R}^d$, and λ_i generated by uniformly sampling ξ_i from $\mathcal{D}_i$, i.e., $g_i(x,\lambda_i) = \nabla\ell_i(x,\lambda_i)$. Then, the following standard assumption is given.

Assumption 2: (i) There exist $L_1, L_2 > 0, \tau \ge 0$ satisfying $\|g_i(x,\lambda_i) - g_i(y,\lambda_i)\| \le L_1\|x - y\|$ and $\|g_i(x,\lambda_i) - g_i(x,\lambda_i')\| \le L_2\|\lambda_i - \lambda_i'\|^\tau$ $\forall i\in\mathcal{V}$ $\forall x, y\in\mathbb{R}^d$ $\forall\lambda_i, \lambda_i'\in\mathbb{R}^r$.

(ii) There exists $\sigma_g > 0$ satisfying $\mathbb{E}[g_i(x,\lambda_i)] = \nabla f_i(x)$ and $\mathbb{E}[\|g_i(x,\lambda_i) - \nabla f_i(x)\|^2] \le \sigma_g^2$.

Remark 2: Assumption 2(i) requires that the sampled gradient $g_i(x,\lambda_i)$ is L_1-Lipschitz continuous with respect to x and (τ, L_2)-Hölder continuous with respect to λ_i, which is more general than [40], [47] with $\tau = 1$ and [44] with $\tau = 0$. Assumption 2(ii) requires that each sampled gradient $g_i(x,\lambda_i)$ is unbiased with a bounded variance σ_g^2, which is standard for distributed stochastic optimization (see, e.g., [10], [11], [12], [13], [14], [16], [17], [37], [39], [41], [42], [44] [46], and [47]).

Next, assumptions for the nonconvex and strongly convex global objective are, respectively, given as follows.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

Assumption 3: There exists $x^*\in\mathbb{R}^d$ such that $F(x^*)$ is the global minimum of the nonconvex global objective $F(x)$. Moreover, the Polyak–Łojasiewicz condition holds, i.e., there exists $\mu>0$ such that $2\,\mu(F(x)-F(x^*))\leq\|\nabla F(x)\|^2\ \forall x\in\mathbb{R}^d$.

Remark 3: Assumption 3 requires the gradient $\nabla F(x)$ to grow faster than a quadratic function as we move away from the global minimum, which is commonly used (see, e.g., [7], [16], [35], [38], and [47]).

Remark 4: There exists functions that satisfy Assumptions 2 and 3 simultaneously. We give two examples. One example is $l_i(x,\xi_i)=\frac{1}{2n}\|\mathbf{A}x-\mathbf{d}\|^2+\frac{\|x\|\xi_i}{1+\|x\|}$, where $x\in\mathbb{R}^d$, the matrix $\mathbf{A}\in\mathbb{R}^{m\times d}$ has full column rank, $\mathbf{d}\in\mathbb{R}^d$ is a constant vector, and $\xi_i\sim N(0,4)$ is a Gaussian noise. Denote $\rho(\mathbf{A}),\Theta_{\mathbf{A}^\top\mathbf{A}}>0$ as the spectral radius of $\mathbf{A}$ and the minimum eigenvalue of $\mathbf{A}^\top\mathbf{A}$, respectively. Then, by [50, Thm. 2], $l_i(x,\xi_i)$ satisfies Assumption 2 with $L_1=\frac{\rho(\mathbf{A})^2}{2n}$, $L_2=1$, $\tau=1$, and $\sigma_g=2$, and $F(x)$ satisfies Assumption 3 with $\mu=2\Theta^2_{\mathbf{A}^\top\mathbf{A}}$. Another example is $l_i(x,\xi_i)=x^2+(3+\xi_i)(\sin x)^2+2\xi_i\cos x$, where $x\in\mathbb{R}$, and $\xi_i\sim\text{Lap}(\frac{1}{2})$ is a Laplacian noise. Then, by [50, Sect. 2.2], $l_i(x,\xi_i)$ satisfies Assumption 2 with $L_1=8$, $L_2=2$, $\tau=1$, and $\sigma_g=\frac{5}{2}$, and $F(x)$ satisfies Assumption 3 with $\mu=\frac{n}{32}$.

In practice, since finding the exact optimal solution is computationally expensive and time-consuming, suboptimal solutions within a given error $\varphi>0$ are often preferred. Inspired by the works of Bhavsar and Prashanth [2], the φ-suboptimal solution and the oracle complexity are defined as follows.

Definition 1: Let $\varphi>0$, $K=0,1,\ldots,$ and $x_K=[x_{1,K}^\top,\ldots,x_{n,K}^\top]^\top$ be the output of an algorithm. Then, x_K is a φ-suboptimal solution if $\mathbb{E}\|\nabla F(x_{i,K+1})\|^2<\varphi\ \forall i\in\mathcal{V}$.

Definition 2: Let $\varphi>0$, $N(\varphi)=\min\{K: x_K \text{ is a } \varphi-\text{suboptimal solution}\}$, and m_k be the sampling number at the kth iteration. Then, the oracle complexity of the algorithm is $\sum_{k=0}^{N(\varphi)}m_k$.

C. Local Differential Privacy

As shown in [40], [41], and [44], there are the following two kinds of adversary models widely used in the privacy-preserving issue for distributed stochastic optimization.

1) A *semihonest* adversary. This kind of adversary is defined as an agent within the network that has access to certain internal information (such as state variable $x_{i,k}$ and tracking variable $y_{i,k}$ of Agent i), follows the prescribed protocols, and accurately computes iterative state and tracking correctly. However, it aims to infer the sensitive information of other agents.
2) An *eavesdropper*. This kind of adversary refers to an external adversary who has capability to wiretap and monitor all communication channels, allowing them to capture distributed messages from any agent. This enables the eavesdropper to infer the sensitive information of agents.

When cooperative agents exchange information to solve the empirical risk minimization problem (2), these two kinds of adversaries can use the model inversion attack [18] to infer sampled gradients, and further obtain the sensitive information in agents' data samples from sampled gradients [19]. Inspired by the works of the authors in [44] and [45], a symmetric binary relation called *adjacency relation* is defined as follows.

Definition 3: For any $i\in\mathcal{V}$, let $\mathcal{D}_i=\{\xi_{i,l},l=1,\ldots,D\}$ and $\mathcal{D}_i'=\{\xi_{i,l}',l=1,\ldots,D\}$ be two sets of Agent i's data samples. If there exists $C>0$ and exactly one pair of data samples ξ_{i,l_0},ξ_{i,l_0}' in $\mathcal{D}_i,\mathcal{D}_i'$ such that for any $x\in\mathbb{R}^d$, $l=1,\ldots,D$

$$\begin{cases}0<\|g_i(x,\xi_{i,l})-g_i(x,\xi_{i,l}')\|_1\leq C, & \text{if } l=l_0\\ \|g_i(x,\xi_{i,l})-g_i(x,\xi_{i,l}')\|_1=0, & \text{if } l\neq l_0\end{cases}\tag{3}$$

then $\mathcal{D}_i$ and $\mathcal{D}_i'$ are said to be adjacent, denoted by $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$.

Remark 5: The constant C is an upper bound of the magnitude of sampled gradients when changing one data sample in $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$. The larger the constant C is, the larger the allowed magnitude of sampled gradients between adjacent datasets is. As long as there exist $\mathcal{D}_i,\mathcal{D}_i'$ satisfying the adjacency relation defined by a constant C, then the privacy analysis in Section III-C holds for $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$. For more details, refer to [47, Sect. II-D].

Remark 6: Definition 3 allows us to avoid the assumption of bounded gradients required in [37], [38], [40], [41], [43], and [45]. Since $\mathcal{D},\mathcal{D}'$ have finite data samples, it follows that $\max_{\xi\in\mathcal{D}\cup\mathcal{D}'}\|\xi\|^\tau<\infty$. Then, for any $C\geq(2^\tau+1)\sqrt{d}L_2\max_{\xi\in\mathcal{D}\cup\mathcal{D}'}\|\xi\|^\tau$ and $x\in\mathbb{R}^d$, by [51, Thm. 2.8 and 2.13] and Assumption 2(i), we have

$$\begin{cases}\|g_i(x,\xi_{i,l})-g_i(x,\xi_{i,l}')\|_1\leq\sqrt{d}L_2\|\xi_{i,l}-\xi_{i,l}'\|^\tau\\ \quad\leq(2^\tau+1)\sqrt{d}L_2\max_{\xi\in\mathcal{D}\cup\mathcal{D}'}\|\xi\|^\tau\leq C, & \text{if } l=l_0\\ \|g_i(x,\xi_{i,l})-g_i(x,\xi_{i,l}')\|_1=0, & \text{if } l\neq l_0.\end{cases}$$

Thus, there exists a constant C such that (3) holds for any $x\in\mathbb{R}^d$, no matter whether gradients are bounded or not.

Remark 7: Different from the adjacency relation defined in differentially private distributed optimization [31], [32], [33], [34], [35], [36], Definition 3 is given with respect to data samples. Moreover, by allowing one data sample of each agent to be different, Definition 3 corresponds to local differential privacy in the offline setting, inspired by the online setting in [44, Def. 2], [45, Def. 2], and then is more stringent than the one of [37], [38], [39], [40], [41], [42], [43], [46], [47], which only allow one data sample of one agent to be different.

Next, the definition of differential privacy is given to show the privacy-preserving level of the algorithm.

Definition 4 (See [28]): Let $\varepsilon_i\geq0$ be the differential privacy budget of Agent i. Then, the mechanism $\mathcal{M}$ achieves ε_i-local differential privacy for $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$ if $\mathbb{P}(\mathcal{M}(\mathcal{D}_i)\in\mathcal{O})\leq e^{\varepsilon_i}\mathbb{P}(\mathcal{M}(\mathcal{D}_i')\in\mathcal{O})$ holds for any Borel-measurable observation set $\mathcal{O}\subseteq\text{Range}(\mathcal{M})$.

Remark 8: As shown in [37], [38], [39], [40], [41], [42], [43], [44], [46], and [47], the differential privacy budget ε_i measures the similarity of the mechanism $\mathcal{M}$'s output distributions under $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$. The smaller the differential privacy budget ε_i is, the higher the differential privacy level is.

Remark 9: Both ε-local differential privacy and (ε,δ)-local differential privacy have been used in differentially private distributed stochastic optimization. ε-local differential privacy is achieved by Laplacian noises, while (ε,δ)-local differential privacy is achieved by Gaussian noises. To simplify the analysis, ε-local differential privacy is used in this article. If (ε,δ)-local differential privacy is used, then the framework of the convergence and privacy analysis still holds.

Problem of interest: In this article, we first aim to propose a new differentially private gradient-tracking-based algorithm for the problem (2) over directed graphs; then, design schemes of step-sizes and the sampling number to enhance the differential privacy level, achieve the almost sure and mean square convergence for nonconvex objectives without the Polyak–Łojasiewicz condition, and further accelerate the convergence rate.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

III. Main Results

A. Proposed Algorithm

In this section, we propose a differentially private gradient-tracking-based distributed stochastic optimization algorithm over directed graphs. Detailed steps are given in Algorithm 1.

Algorithm 1: Differentially Private Gradient-Tracking-Based Distributed Stochastic Optimization Algorithm Over Directed Graphs.

Initialization: $x_{i,0} \in \mathbb{R}^d$ for any $i \in \mathcal{V}$, m_K different data samples $\lambda_{i,0,1}, \ldots, \lambda_{i,0,m_K}$ in $\mathcal{D}_i$, $y_{i,0} = g_{i,0} = \frac{1}{m_K}\sum_{l=1}^{m_K} g_i(x_{i,0}, \lambda_{i,0,l})$ for any $i \in \mathcal{V}$, weight matrices $\mathcal{R} = (\mathcal{R}_{ij})_{i,j=1,\ldots,n}$, $\mathcal{C} = (\mathcal{C}_{ij})_{i,j=1,\ldots,n}$, the maximum iteration number K, step-sizes $\alpha_K, \beta_K, \gamma_K$ and the sampling number m_K.

for $k = 0, 1, \ldots, K$, **do**

1: Agent i adds independent d-dimensional Laplacian noises $\zeta_{i,k}$, $\eta_{i,k}$ to its state variable $x_{i,k}$ and tracking variable $y_{i,k}$, respectively: $\breve{x}_{i,k} = x_{i,k} + \zeta_{i,k}$, $\breve{y}_{i,k} = y_{i,k} + \eta_{i,k}$, where each coordinate of $\zeta_{i,k}$, $\eta_{i,k}$ has the distribution $\mathrm{Lap}(\sigma_k^{(\zeta_i)})$ and $\mathrm{Lap}(\sigma_k^{(\eta_i)})$, respectively.

2: Agent i broadcasts its perturbed state variable $\breve{x}_{i,k}$ to all its out-neighbors in $\mathcal{N}_{\mathcal{R},i}^+$, and broadcasts its perturbed tracking variable $\breve{y}_{i,k}$ to all its out-neighbors in $\mathcal{N}_{\mathcal{C},i}^+$.

3: Agent i receives $\breve{x}_{j,k}$ from all its in-neighbors in $\mathcal{N}_{\mathcal{R},i}^-$ and $\breve{y}_{j,k}$ from all its in-neighbors in $\mathcal{N}_{\mathcal{C},i}^-$.

4: Agent i updates its state variable by

$$x_{i,k+1} = \left(1 - \alpha_K \sum_{j\in\mathcal{N}_{\mathcal{R},i}^-} \mathcal{R}_{ij}\right) x_{i,k} + \alpha_K \sum_{j\in\mathcal{N}_{\mathcal{R},i}^-} \mathcal{R}_{ij}\breve{x}_{j,k} - \gamma_K y_{i,k}. \tag{4}$$

5: Agent i takes m_K different samples $\lambda_{i,k+1,1}, \ldots, \lambda_{i,k+1,m_K}$ uniformly from $\mathcal{D}_i$ to generate sampled gradients $g_i(x_{i,k+1}, \lambda_{i,k+1,1}), \ldots, g_i(x_{i,k+1}, \lambda_{i,k+1,m_K})$. Then, Agent i puts these data samples back into $\mathcal{D}_i$.

6: Agent i computes the averaged sampled gradient by

$$g_{i,k+1} = \frac{1}{m_K}\sum_{l=1}^{m_K} g_i(x_{i,k+1}, \lambda_{i,k+1,l}). \tag{5}$$

7: Agent i updates its tracking variable by

$$y_{i,k+1} = \left(1 - \beta_K \sum_{j\in\mathcal{N}_{\mathcal{C},i}^+} \mathcal{C}_{ji}\right) y_{i,k} + \beta_K \sum_{j\in\mathcal{N}_{\mathcal{C},i}^-} \mathcal{C}_{ij}\breve{y}_{j,k} + g_{i,k+1} - g_{i,k}. \tag{6}$$

end for

Return $x_{1,K+1}, \ldots, x_{n,K+1}$

For the convenience of the analysis, let $x_k = [x_{1,k}^\top, \ldots, x_{n,k}^\top]^\top$ $y_k = [y_{1,k}^\top, \ldots, y_{n,k}^\top]^\top$, $\zeta_k = [\zeta_{1,k}^\top, \ldots, \zeta_{n,k}^\top]^\top$, $\eta_k = [\eta_{1,k}^\top, \ldots, \eta_{n,k}^\top]^\top$, and $g_k = [g_{1,k}^\top, \ldots, g_{n,k}^\top]^\top$. Then, (4) and (6) can be written in the following compact form:

$$x_{k+1} = ((I_n - \alpha_K\mathcal{L}_1)\otimes I_d)x_k + \alpha_K(\mathcal{R}\otimes I_d)\zeta_k - \gamma_K y_k \tag{7}$$

$$y_{k+1} = ((I_n - \beta_K\mathcal{L}_2)\otimes I_d)y_k + \beta_K(\mathcal{C}\otimes I_d)\eta_k + g_{k+1} - g_k. \tag{8}$$

B. Convergence Analysis

In this section, we will give the convergence analysis of Algorithm 1. First, we give the following key lemma:

Lemma 2: For any $K = 0, 1, \ldots$, $k = 0, \ldots, K$. Let $V_k = [\mathbb{E}\|(W_1\otimes I_d)x_k\|^2, \mathbb{E}\|(W_2\otimes I_d)y_k\|^2, \mathbb{E}(F(x_k) - F(x^*))]^\top$. Under Assumptions 1–3, if step-sizes $\alpha_K, \beta_K, \gamma_K$ satisfy the following conditions:

$$0 < \alpha_K < \min\left\{\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}}\right\}, \min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(1)})}{1+|\varpi_l^{(1)}|^2}\right\}\right\}$$

$$0 < \beta_K < \min\left\{\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}}\right\}, \min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(2)})}{1+|\varpi_l^{(2)}|^2}\right\}\right\}$$

$$0 < \gamma_K < \frac{n}{4(v_1^\top v_2)L_1}$$

then the following inequality holds:

$$\mathbb{E}V_{k+1} \le A_K\mathbb{E}V_k + u_k \tag{9}$$

where

$$u_k = \begin{bmatrix} u_k^{(1)} \\ u_k^{(2)} \\ u_k^{(3)} \end{bmatrix}, A_K = \begin{bmatrix} A_K^{(11)} & A_K^{(12)} & A_K^{(13)} \\ A_K^{(21)} & A_K^{(22)} & A_K^{(23)} \\ A_K^{(31)} & A_K^{(32)} & A_K^{(33)} \end{bmatrix},$$

$u_k^{(1)} = 2nd\rho(\mathcal{R})^2\alpha_K^2\max_{i\in\mathcal{V}}\{(\sigma_k^{(\zeta_i)})^2\} + \frac{2(1+r_1\alpha_K)\|v_2\|^2\gamma_K^2\sigma_g^2}{n^2 r_1\alpha_K m_K} + \frac{4d(1+r_1\alpha_K)\|v_2\|^2\rho(\mathcal{C})^2\beta_K^2\gamma_K^2}{n^3 r_1\alpha_K}\sum_{l=0}^{k-1}\max_{i\in\mathcal{V}}\{(\sigma_k^{(\eta_i)})^2\}$, $u_k^{(2)} = \frac{12d(1+r_2\beta_K)\|v_2\|^2\rho(\mathcal{C})^2\beta_K\gamma_K^2L_1^2}{nr_2}\sum_{l=0}^{k-1}\max_{i\in\mathcal{V}}\{(\sigma_k^{(\eta_i)})^2\} + \frac{(2n+3nr_2\beta_K+(6+6r_2\beta_K)\|v_2\|^2\gamma_K^2L_1^2)\sigma_g^2}{r_2 m_K\beta_K} + 2nd\rho(\mathcal{C})^2\beta_K^2\max_{i\in\mathcal{V}}\{(\sigma_k^{(\eta_i)})^2\} + \frac{4(1+r_2\beta_K)nd\rho(\mathcal{R})^2\alpha_K^2L_1^2\max_{i\in\mathcal{V}}\{(\sigma_k^{(\zeta_i)})^2\}}{r_2\beta_K}$, $u_k^{(3)} = \frac{(v_1^\top v_2)(3n+2(v_1^\top v_2)\gamma_K L_1)\gamma_K\sigma_g^2}{2n^2 m_K} + \frac{2d\|v_1\|^2\rho(\mathcal{R})^2\alpha_K\max_{i\in\mathcal{V}}\{(\sigma_k^{(\zeta_i)})^2\}}{n} + \frac{(v_1^\top v_2)d\rho(\mathcal{C})^2(3n+2(v_1^\top v_2)\gamma_K L_1)\beta_K^2\gamma_K}{n^3}\sum_{l=0}^{k-1}\max_{i\in\mathcal{V}}\{(\sigma_k^{(\eta_i)})^2\}$,

$A_K^{(11)} = 1 - r_1\alpha_K + \frac{4(1+r_1\alpha_K)\|v_2\|^2\gamma_K^2L_1^2}{n^3 r_1\alpha_K}$, $A_K^{(12)} = \frac{2(1+r_1\alpha_K)\gamma_K^2}{r_1\alpha_K}$,
$A_K^{(13)} = \frac{8(1+r_1\alpha_K)\|v_2\|^2\gamma_K^2L_1}{n^2 r_1\alpha_K}$, $A_K^{(31)} = \frac{(v_1^\top v_2)(3n+4(v_1^\top v_2)\gamma_K L_1)\gamma_K L_1^2}{2n^3}$,
$A_K^{(22)} = 1 - r_2\beta_K + \frac{6(1+r_2\beta_K)\gamma_K^2L_1^2}{r_2\beta_K}$, $A_K^{(23)} = \frac{24(1+r_2\beta_K)\|v_2\|^2\gamma_K^2L_1^3}{r_2\beta_K}$,
$A_K^{(21)} = \frac{6(1+r_2\beta_K)(n\rho(\mathcal{L}_1)^2\alpha_K^2+2\|v_2\|^2\gamma_K^2L_1^2)L_1^2}{nr_2\beta_K}$, $A_K^{(32)} = \frac{3\|v_1\|^2\gamma_K}{2n(v_1^\top v_2)}$,
and $A_K^{(33)} = 1 - \frac{(v_1^\top v_2)\mu\gamma_K}{n} + \frac{4(v_1^\top v_2)^2\gamma_K^2L_1}{n^2}$.

Proof: Due to the space limitation, we leave the proof to [52]. ■

Next, we give two different schemes of step-sizes and the sampling number for Algorithm 1.

Scheme (S1): For any $K = 0, 1, \ldots$, the following holds.

(I) *Step-sizes:* $\alpha_K = \frac{a_1}{(K+1)^{p_\alpha}}$, $\beta_K = \frac{a_2}{(K+1)^{p_\beta}}$, and $\gamma_K = \frac{a_3}{(K+1)^{p_\gamma}}$.

(II) *Sampling number:* $m_K = \lfloor a_4 K^{p_m}\rfloor + 1$.

Here, $a_1, a_2, a_3, a_4 > 0, p_\alpha, p_\beta, p_\gamma > 0, p_m \ge 0$.

Scheme (S2): For any $K = 0, 1, \ldots$, the following holds.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

(I) *Step-sizes:* $\alpha_K=\alpha, \beta_K=\beta$, and $\gamma_K=\gamma$ are constants.
(II) *Sampling number:* $m_K=\lfloor p_m^K\rfloor+1$.
Here, $\alpha,\beta,\gamma>0$, $p_m\geq 0$.

To get the almost sure and mean square convergence of Algorithm 1, we need the following assumptions.

Assumption 4: Under *Scheme (S1)*, step-sizes α_K, β_K, and γ_K, the sampling number m_K, and privacy noise parameters $\sigma_k^{(\zeta_i)}=(k+1)^{p_{\zeta_i}}$ and $\sigma_k^{(\eta_i)}=(k+1)^{p_{\eta_i}}$ satisfy the following conditions:

$$a_1<\min\left\{\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}}\right\},\min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(1)})}{1+|\varpi_l^{(1)}|^2}\right\}\right\}$$

$$a_2<\min\left\{\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}}\right\},\min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(2)})}{1+|\varpi_l^{(2)}|^2}\right\}\right\}$$

$$a_3<\frac{n}{4(v_1^\top v_2)L},\frac{1}{2}<p_\beta<p_\alpha<p_\gamma<1,p_m-p_\beta\geq 1$$

$$2p_\gamma-p_\alpha\geq 1,2p_\alpha-p_\beta-2\max\left\{\max_{i\in\mathcal{V}}\{p_{\zeta_i}\},0\right\}\geq 1$$

$$p_\gamma+2p_\beta-2\max\left\{\max_{i\in\mathcal{V}}\{p_{\eta_i}\},0\right\}\geq 2.$$

Assumption 5: Under *Scheme (S2)*, step-sizes α,β, and γ, the sampling number m_K, and privacy noise parameters $\sigma_k^{(\zeta_i)}=p_{\zeta_i}^K$ and $\sigma_k^{(\eta_i)}=p_{\eta_i}^K$ satisfy the following conditions:

$$\beta<\min\left\{\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}}\right\},\min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(2)})}{1+|\varpi_l^{(2)}|^2}\right\}\right\}$$

$$\alpha<\min\left\{\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}}\right\},\min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(1)})}{1+|\varpi_l^{(1)}|^2}\right\},\frac{\sqrt{2}(v_1^\top v_2)r_2\beta}{12\rho(\mathcal{L}_1)\|v_1\|L_1}\right\}$$

$$\gamma<\min\left\{1,\frac{n}{20(v_1^\top v_2)L},Q_1\alpha,Q_2\beta\right\},0<p_{\zeta_i},p_{\eta_i}<1,p_m>1$$

where

$$Q_1=\min\left\{\frac{n\sqrt{3n}r_1}{24\|v_2\|L_1},\frac{r_1}{2\|v_2\|L_1}\sqrt{\frac{\mu}{12L_1+2\mu}+\frac{\mathbb{I}_{\{\mu=0\}}}{2}}\right\}$$

$$Q_2=\min\left\{\frac{\sqrt{3}r_2}{6nL_1},\frac{\sqrt{3}(v_1^\top v_2)r_2}{36\|v_1\|\|v_2\|L_1},\frac{\sqrt{6}(v_1^\top v_2)r_1r_2}{144\rho(\mathcal{L}_1)\|v_1\|\|v_2\|L_1},\frac{\sqrt{6}(v_1^\top v_2)r_2}{12\|v_1\|\|v_2\|L_1}\sqrt{\frac{\mu}{36L_1+7\mu}+\frac{\mathbb{I}_{\{\mu=0\}}}{7}}\right\}.$$

Theorem 1: If Assumptions 1, 2, and 4 hold under *Scheme (S1)*, then $\liminf_{K\to\infty}\|\nabla F(x_{i,K+1})\|^2=0$ a.s., $\liminf_{K\to\infty}\mathbb{E}\|\nabla F(x_{i,K+1})\|^2=0$ $\forall i\in\mathcal{V}$. If Assumptions 1, 2, and 5 hold under *Scheme (S2)*, then $\lim_{K\to\infty}\|\nabla F(x_{i,K+1})\|^2=0$ a.s., $\lim_{K\to\infty}\mathbb{E}\|\nabla F(x_{i,K+1})\|^2=0$ $\forall i\in\mathcal{V}$.

Proof: Due to the space limitation, we leave the proof to [52]. ■

Remark 10: Algorithm 1 achieves the almost sure and mean square convergence for nonconvex objectives without the Polyak–Łojasiewicz condition. The condition imposed on objectives is weaker than (strongly) convex objectives [10], [11], [12], [13], [14], [15], [17], [31], [32], [33], [34], [36] or the Polyak–Łojasiewicz condition [16], [35]. Thus, Algorithm 1 has wider applicability than [10], [11], [12], [13], [14], [15], [16], [17], [31], [32], [33], [34], [35], and [36].

The polynomial mean square convergence rate and the oracle complexity of Algorithm 1 with *Scheme (S1)* are given as follows.

Theorem 2: Under Assumptions 1–3 and 4, Algorithm 1 with *Scheme (S1)* achieves the following polynomial mean square convergence rate for any $i\in\mathcal{V}$:

$$\mathbb{E}\|\nabla F(x_{i,K+1})\|^2=O\left(\frac{1}{(K+1)^{\theta-p_\gamma}}\right)\tag{10}$$

where $\theta=\min\{p_m-p_\beta,2p_\alpha-p_\beta-2\max\{\max_{i\in\mathcal{V}}\{p_{\zeta_i}\},0\}$ and $2p_\beta-2\max\{\max_{i\in\mathcal{V}}\{p_{\eta_i}\},0\}\}$. Furthermore, for any $\varphi>0$, if $p_\alpha=\max\left\{1-\frac{\varphi}{5},\frac{9}{10}\right\}$, $p_\beta=\max\left\{\frac{2}{3}\left(1-\frac{\varphi}{5}\right),\frac{3}{5}\right\}$, $p_\gamma=\max\left\{1-\frac{\varphi}{10},\frac{9}{10}\right\}$, $p_m=\max\left\{2-\frac{\varphi}{10},\frac{39}{20}\right\}$, and $p_{\zeta_i}=p_{\eta_i}=\max\left\{\frac{\varphi}{10},\frac{1}{20}\right\}$, then the oracle complexity of Algorithm 1 with *Scheme (S1)* is $O\left(\varphi^{-\frac{177+3\max\{1-2\varphi,0\}}{9-11\max\{1-2\varphi,0\}}}\right)$.

Proof: See Appendix B. ■

Remark 11: In Theorem 2, the polynomial mean square convergence rate is given for privacy noises with decreasing, constant (see, e.g., [37], [38], [40], and [43]), and increasing variances (see, e.g., [44], [46], and [47]). This is nontrivial even without considering the privacy protection. For example, let step-sizes $\alpha_K=\frac{1}{(K+1)^{0.96}}$, $\beta_K=\frac{1}{(K+1)^{0.7}}$, and $\gamma_K=\frac{1}{(K+1)^{0.98}}$. Then, Theorem 2 holds as long as privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ have the increasing rate no more than $O(k^{0.19})$.

Remark 12: The key to achieving the polynomial mean square convergence rate without the assumption of bounded gradients is to use polynomially decreasing step-sizes and the increasing sampling number, which reduces the effect of stochastic gradient noises and privacy noises. This is different from [6], [7], [37], [38], [39], [40], [41], [43], and [44], where the assumption of bounded gradients is required.

Next, the exponential mean square convergence rate and the oracle complexity of Algorithm 1 with *Scheme (S2)* are given.

Theorem 3: Under Assumptions 1–3 and 5, Algorithm 1 with *Scheme (S2)* achieves the following exponential mean square convergence rate for any $i\in\mathcal{V}$

$$\mathbb{E}\|\nabla F(x_{i,K+1})\|^2=O\left(\max\left\{\rho(A_K),\frac{1}{p_m},\max_{i\in\mathcal{V}}\{p_{\zeta_i}^2,p_{\eta_i}^2\}\right\}^K\right).$$

Furthermore, for any $\varphi>0$, if $\beta=\min\left\{\frac{1}{2},\frac{n}{40(v_1^\top v_2)L},\min_{i\in\mathcal{V}}\left\{\frac{1}{2\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}}\right\},\min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(2)})}{2+2|\varpi_l^{(2)}|^2}\right\}\right\}$, $\alpha=\min\left\{\beta,\min_{i\in\mathcal{V}}\left\{\frac{1}{2\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}}\right\},\min_{l=2,\ldots,n}\left\{\frac{\mathrm{Re}(\varpi_l^{(1)})}{2+2|\varpi_l^{(1)}|^2}\right\},\frac{\sqrt{2}(v_1^\top v_2)r_2\beta}{12\rho(\mathcal{L}_1)\|v_1\|L_1}\right\}$, $\gamma=\min\left\{\frac{1}{2},\frac{n}{40(v_1^\top v_2)L},\frac{Q_1\alpha}{2},\frac{Q_2\beta}{2}\right\}$, $p_m=\max\left\{\frac{1}{\varphi},\frac{1}{\rho(A_K)}\right\}$, and $p_{\zeta_i}=p_{\eta_i}=\min\{\varphi,\rho(A_K)\}$, then the oracle complexity of Algorithm 1 with *Scheme (S2)* is $O\left(\frac{|\ln\varphi|}{\varphi}\right)$.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

Proof: Due to the space limitation, we leave the proof to [52]. ■

Remark 13: By Theorems 2 and 3, *Scheme (S2)* achieves the exponential mean square convergence rate, while *Scheme (S1)* and methods in [6], [7], [15], [17], [37], [38], [39], [40], [41], [42], [43], [44], [46], and [47] achieve the polynomial mean square convergence rate. For example, when the index of convergence rate is $\frac{1}{K+1}\sum_{k=0}^{K}\mathbb{E}(F(\bar{x}_k)-F(x^*))$, methods in [40] and [41] achieve convergence rates of $O\left(\frac{1}{\sqrt{K}}\right)$ and $O(1)$, respectively. Since the method in [40] is the same as the one in [53], by [53, Thm. 2], the method in [40] achieves the convergence rate of $O\left(\frac{1}{\sqrt{K}}\right)$. By [41, Thm. 2], the method in [41] achieves the convergence rate of $O(1)$. Thus, *Scheme (S2)* is suitable for the scenario where the convergence rate is prioritized. However, by Theorem 1, *Scheme (S1)* achieves the almost sure and mean square convergence under decreasing, constant, and increasing privacy noises, while *Scheme (S2)* achieves the almost sure and mean square convergence only under decreasing privacy noises. This shows the tradeoff of Algorithm 1 between the convergence rate and the added privacy noises.

Remark 14: When the global objective $F(x)$ is strongly convex (i.e., there exists $s>0$ such that $F(y)\geq F(x)+\langle\nabla F(x),y-x\rangle+\frac{s}{2}\|y-x\|^2\ \ \forall x,y\in\mathbb{R}^d$), by [49, Lemma 6.9], we have $2s(F(x)-F(x^*))\leq\|\nabla F(x)\|^2$. Then, Assumption 3 is satisfied with $\mu=s$, and thus, Theorems 2 and 3 also hold for strongly convex objectives. Hence, we provide a general frame for Algorithm 1's convergence rate analysis under both nonconvex objectives with the Polyak–Łojasiewicz condition and strongly convex objectives.

Remark 15: The oracle complexity of *Scheme (S2)* shows that the sampling number required to achieve the desired accuracy is lower than existing works (see, e.g., [14]). By Theorem 3, the oracle complexity of *Scheme (S2)* is $O(\frac{|\ln\varphi|}{\varphi})$, which is smaller than the oracle complexity $O(\frac{1}{\varphi^2})$ of the gradient-tracking-based algorithm in [14]. For example, when the error $\varphi=0.02$, $O(10^2)$ data samples are required in *Scheme (S2)*, while $O(10^3)$ data samples are required in the gradient-tracking-based algorithm in [14]. Moreover, the increasing sampling number in both *Schemes (S1)* and *(S2)* is feasible in machine learning scenarios, such as the speech recognition problem [55], the simulated annealing problem [56], and the noun-phrase chunking problem [57].

C. Privacy Analysis

In the following, the definition of the sensitivity is provided to compute the cumulative differential privacy budget ε_i for any $i\in\mathcal{V}$.

Definition 5: Given $\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')$ for any $i\in\mathcal{V}$ and query q. For any $k=0,\ldots,K$, let $\mathcal{D}_{i,k}=\{\lambda_{i,k,l},l=1,\ldots,m_K\}$ and $\mathcal{D}_{i,k}'=\{\lambda_{i,k,l}',l=1,\ldots,m_K\}$ be the data samples taken from $\mathcal{D}_i$ and $\mathcal{D}_i'$ at the kth iteration, respectively. Then, the sensitivity of Algorithm 1 for Agent i at the kth iteration is defined follows:

$$\Delta_{i,k}^{q}\triangleq\begin{cases}\sup\limits_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\mathrm{nd}}\\(\breve{x}_0,\breve{y}_0)\in\mathcal{O}\\\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}}\|q(\mathcal{D}_{i,0})-q(\mathcal{D}_{i,0}')\|_1, & \text{if } k=0\\ \sup\limits_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\mathrm{nd}}\\(\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O}\\\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}}\|q(\mathcal{D}_{i,k})-q(\mathcal{D}_{i,k}')\|_1, & \text{if } k=1,\ldots,K.\end{cases}\tag{11}$$

Remark 16: Definition 5 captures the magnitude by which Agent i's data sample can change the query q in the worst case. It is the key quantity showing how many noises should be added to achieve $\varepsilon_{i,k}$-local differential privacy for Agent i at the kth iteration. In Algorithm 1, the query $q(\mathcal{D}_{i,k})=[x_{i,k}^\top,y_{i,k}^\top]^\top$, and the mechanism $\mathcal{M}(\mathcal{D}_{i,k})=[\breve{x}_{i,k}^\top,\breve{y}_{i,k}^\top]^\top$.

The following lemma gives the sensitivity $\Delta_{i,k}^q$ of Algorithm 1 for any $k=0,\ldots,K$.

Lemma 3: If Assumption 2(i) holds, then the sensitivity of Algorithm 1 for Agent i at the kth iteration satisfies $\Delta_{i,k}^q=\|\Delta x_{i,k}\|_1+\|\Delta y_{i,k}\|_1$, where $\|\Delta x_{i,k}\|_1$ and $\|\Delta y_{i,k}\|_1$ are given as follows:

$$\|\Delta x_{i,k}\|_1\leq\begin{cases}0, & \text{if } k=0\\ \gamma_K\sum_{l=0}^{k-1}|1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}|^{k-l-1}\|\Delta y_{i,l}\|_1, & \text{if } k=1,\ldots,K\end{cases}$$

$$\|\Delta y_{i,k}\|_1\leq\begin{cases}\frac{C}{m_K}, & \text{if } k=0\\ \sum_{l=0}^{k-1}|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}|^l\frac{2C}{m_K}\\ \quad+|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}|^k\frac{C}{m_K}, & \text{if } k=1,\ldots,K.\end{cases}$$

Proof: See Appendix C. ■

Lemma 4: If Assumption 2(i) holds, then for any $K=0,1,\ldots,$ Algorithm 1 achieves ε_i-local differential privacy for Agent i over K iterations, where $\varepsilon_i=\sum_{k=0}^{K}\left(\frac{\|\Delta x_{i,k}\|_1}{\sigma_k^{(\zeta_i)}}+\frac{\|\Delta y_{i,k}\|_1}{\sigma_k^{(\eta_i)}}\right)$.

Proof: Due to the space limitation, we leave the proof to [52]. ■

Remark 17: By Lemma 4, the larger privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the smaller the cumulative differential privacy budget ε_i is. While by Theorem 2, the larger privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the higher the oracle complexity is. For example, if the error $\varphi=0.02$, then the oracle complexity $O(10^{15})$ of Algorithm 1 with *Scheme (S1)* is higher than the oracle complexity $O(10^3)$ of the centralized SGD in [54]. As a result, Algorithm 1 with *Scheme (S1)* achieves privacy at the cost of increasing the oracle complexity.

Theorem 4: For step-sizes α_K,β_K, and γ_K, the sampling number m_K satisfying *Scheme (S1)*, and privacy noise parameters $\sigma_k^{(\zeta_i)}=(k+1)^{p_{\zeta_i}}$ and $\sigma_k^{(\eta_i)}=(k+1)^{p_{\zeta_i}}$, if Assumption 2(i) and the following conditions hold:

$$p_m-p_\beta+\min\{\min_{i\in\mathcal{V}}\{p_{\eta_i}\}-1,0\}>0$$

$$p_m+\min\{0,p_\gamma-p_\alpha-p_\beta\}+\min\{\min_{i\in\mathcal{V}}\{p_{\zeta_i}\}-1,0\}>0$$

$$0<a_1<\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}}\right\},0<a_2<\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}}\right\}$$

then the cumulative privacy budget ε_i is finite for any $i\in\mathcal{V}$ even over infinite iterations.

Due to the space limitation, we leave the proof to [52].

Theorem 5: For step-sizes α_K,β_K, and γ_K, the sampling number m_K satisfying *Scheme (S2)*, and privacy noise parameters $\sigma_k^{(\zeta_i)}=p_{\zeta_i}^K$, $\sigma_k^{(\eta_i)}=p_{\eta_i}^K$, if Assumption 2(i) and the

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

following conditions hold:

$$0<p_{\zeta_i},p_{\eta_i}<1, p_m>\max_{i\in\mathcal{V}}\left\{\frac{1}{p_{\zeta_i}},\frac{1}{p_{\eta_i}}\right\}$$

$$0<\alpha<\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}^-_{\mathcal{R},i}}\mathcal{R}_{ij}}\right\}, 0<\beta<\min_{i\in\mathcal{V}}\left\{\frac{1}{\sum_{j\in\mathcal{N}^+_{\mathcal{C},i}}\mathcal{C}_{ji}}\right\}$$

then the cumulative privacy budget ε_i is finite for any $i\in\mathcal{V}$ even over infinite iterations.

Due to the space limitation, we leave the proof to [52].

Remark 18: Theorems 4 and 5 establish the sufficient condition for Algorithm 1 with *Schemes (S1)* and *(S2)* to achieve the finite cumulative differential privacy budget ε_i even over infinite iterations, respectively. This is different from [6], [7], [10], [11], [12], [13], [14], [15], [16], and [17] that do not consider the privacy protection, and [37], [38], [39], [40], [41], [42], and [43] that achieve the infinite cumulative differential privacy budget ε over infinite iterations. Thus, compared to [37], [38], [39], [40], [41], [42], and [43], Algorithm 1 with both *Schemes (S1)* and *(S2)* provides a higher differential privacy level.

D. Tradeoff Between Privacy and Convergence

Based on Theorems 2–5, the tradeoff between the privacy and the convergence is given in the following corollary.

Corollary 1: (i) Under Assumptions 1–4, if $p_m-p_\beta+\min\{\min_{i\in\mathcal{V}}\{p_{\eta_i}\}-1,0\}>0, p_m+\min\{0,p_\gamma-p_\alpha-p_\beta\}+\min\{\min_{i\in\mathcal{V}}\{p_{\zeta_i}\}-1,0\}>0$ hold, then Algorithm 1 with *Scheme (S1)* achieves the polynomial mean square convergence rate and the finite cumulative differential privacy budget ε_i for any $i\in\mathcal{V}$ even over infinite iterations simultaneously.

(ii) Under Assumptions 1–3 and 5, if $0<p_{\zeta_i},p_{\eta_i}<1, p_m>\max_{i\in\mathcal{V}}\left\{\frac{1}{p_{\zeta_i}},\frac{1}{p_{\eta_i}}\right\}$ hold, then Algorithm 1 with *Scheme (S2)* achieves the exponential mean square convergence rate and the finite cumulative differential privacy budget ε_i for any $i\in\mathcal{V}$ even over infinite iterations simultaneously.

Proof: By Theorems 2 and 4, Corollary 1(i) is proved. Then, by Theorems 3 and 5, Corollary 1(ii) is proved. ■

Remark 19: *Scheme (S1)* achieves the polynomial mean square convergence rate and the finite cumulative differential privacy budget ε_i over infinite iterations simultaneously under decreasing, constant, and increasing privacy noises. For example, let $p_\alpha=0.987$, $p_\beta=0.69$, $p_\gamma=0.997$, and $p_m=2$. Then, conditions in Corollary 1(i) are satisfied as long as $-0.3<p_{\zeta_i}<0.15$, and $-0.3<p_{\eta_i}<0.15$. *Scheme (S2)* achieves the exponential mean square convergence rate and the finite cumulative differential privacy budget ε_i over infinite iterations simultaneously under decreasing privacy noises. For example, let $\alpha=0.1,\beta=0.1,\gamma=0.01$, and $p_m=1.1$. Then, conditions in Corollary 1(ii) are satisfied as long as $0.91<p_{\zeta_i}<0.95$ and $0.91<p_{\eta_i}<0.95$.

Remark 20: Corollary 1 shows the tradeoff between privacy and the convergence rate in Algorithm 1. The smaller the privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the faster the Algorithm 1 converges, while the larger the cumulative differential privacy budget ε_i is. Moreover, *Scheme (S1)* achieves the polynomial mean square convergence rate and finite cumulative differential privacy budget ε_i over infinite iterations under decreasing, constant, and increasing privacy noises, while *Scheme (S2)* achieves the exponential mean square convergence rate and finite cumulative differential privacy budget ε_i only for decreasing privacy noises. Then, the differential privacy level of *Scheme (S1)* is higher than the one of *Scheme (S2)*, while the convergence rate of *Scheme (S2)* is faster than the one of *Scheme (S1)*.

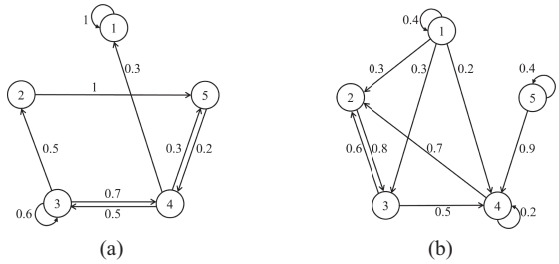


Fig. 1. Topology structures of directed graphs $\mathcal{G}_\mathcal{R}$ and $\mathcal{G}_\mathcal{C}$ induced by weight matrices $\mathcal{R}$ and $\mathcal{C}$. (a) Directed graph $\mathcal{G}_\mathcal{R}$. (b) Directed graph $\mathcal{G}_\mathcal{C}$.

Remark 21: The parameter a_4 in the sampling number $m_K=\lfloor a_4K^{p_m}\rfloor+1$ affects both convergence rate and the cumulative privacy budget. Since by (24), $\mathbb{E}\|\nabla F(x_{i,K+1})\|^2=O\left(\frac{a_4+1}{a_4(K+1)^{\theta-p_\gamma}}\right)$ is decreasing with respect to a_4. Then, the larger the parameter a_4 is, the faster the convergence rate is. By Lemma 3, the larger the parameter a_4 is, the smaller the sensitivity $\Delta_{i,k}^q$ is, and then by Lemma 4, the smaller the cumulative privacy budget ε_i is.

Based on Corollary 1, we have the following corollary as the sampling number goes to infinity.

Corollary 2: Under the conditions of Corollary 1, Algorithm 1 with both *Schemes (S1)* and *(S2)* achieves the almost sure and mean square convergence and the finite cumulative differential privacy budget ε_i for any $i\in\mathcal{V}$ over infinite iterations simultaneously as the sampling number goes to infinity.

Remark 22: The result of Corollary 2 does not contradict the tradeoff between privacy and utility. In fact, to achieve differential privacy, Algorithm 1 incurs a compromise on the utility. However, different from [38] and [42] that compromise convergence accuracy to enable differential privacy, Algorithm 1 compromises the convergence rate and the sampling number (which are also utility metrics) instead. By Corollary 1, the larger privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the slower the convergence rate is. By Corollary 2, the sampling number m_K is required to go to infinity when the convergence of Algorithm 1 and the finite cumulative privacy budget ε_i over infinite iterations are considered simultaneously. The ability to retain convergence accuracy makes our approach suitable for accuracy-critical scenarios.

IV. Numerical Examples

In this section, we train the machine learning model ResNet18 (see [58]) in a distributed manner with the benchmark datasets "MNIST" (see [59]) and "CIFAR-10" (see [60] and [61]), respectively. Specifically, five agents cooperatively train ResNet18 over the directed graphs, as shown in Fig. 1(a) and (b), which satisfy Assumption 1. Then, each benchmark dataset is divided into two subsets for training and testing, respectively. The training dataset of each benchmark dataset is uniformly divided into five subsets, each of which can only be accessed by one agent to update its model parameters. To ensure a fair comparison, we set the cumulative differential privacy budget $\varepsilon=\max_{i\in\mathcal{V}}\{\varepsilon_i\}$ for

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

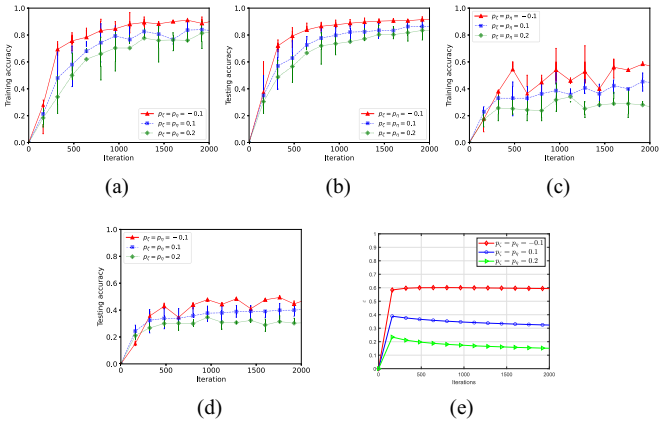


Fig. 2. Accuracy and cumulative differential privacy budget ε of Algorithm 1 with *Scheme (S1)* and $p_{\zeta_i}, p_{\eta_i} = -0.1, 0.1, 0.2$. (a) Training accuracy on the "MNIST" dataset. (b) Testing accuracy on the "MNIST" dataset. (c) Training accuracy on the "CIFAR-10" dataset. (d) Testing accuracy on the "CIFAR-10" dataset. (e) Cumulative differential privacy budget ε.

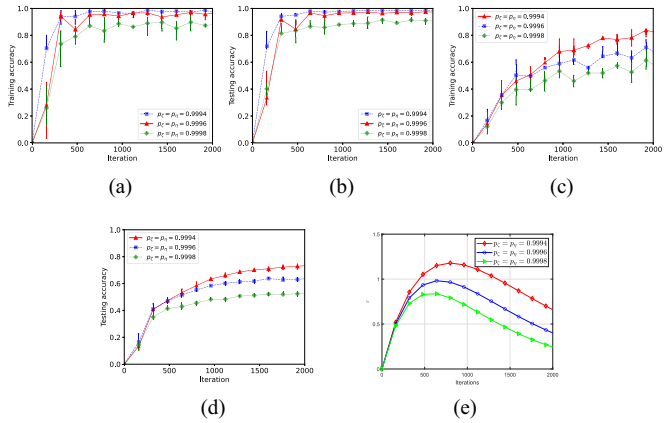


Fig. 3. Accuracy and cumulative differential privacy budget ε of Algorithm 1 with *Scheme (S2)* and $p_{\zeta_i}, p_{\eta_i} = 0.9994, 0.9996, 0.9998$. (a) Training accuracy on the "MNIST" dataset. (b) Testing accuracy on the "MNIST" dataset. (c) Training accuracy on the "CIFAR-10" dataset. (d) Testing accuracy on the "CIFAR-10" dataset. (e) Cumulative differential privacy budget ε.

Algorithm 1 with *Schemes (S1)* and *(S2)*. Then, the following three numerical experiments are given.

(a) The effect of privacy noises on Algorithm 1's convergence rate and differential privacy level.
(b) The comparison of Algorithm 1 with *Schemes (S1)* and *(S2)* between the convergence rate and the differential privacy level.
(c) The comparison between Algorithm 1 with *Schemes (S1)* and *(S2)* and methods in [38], [41], [42], [46], and [47] for the convergence rate and the differential privacy level.

A. Effect of Privacy Noises

First, let step-sizes $\alpha_K = \frac{72}{2000^{0.987}} = 0.04$, $\beta_K = \frac{0.95}{2000^{0.69}} = 0.005$, and $\gamma_K = \frac{98}{2000^{0.997}} = 0.05$, the sampling number $m_K = \lfloor 0.00007 \cdot 2000^{1.78} \rfloor + 1 = 53$, and privacy noise parameters $\sigma_k^{(\zeta_i)} = (k+1)^{p_{\zeta_i}}$, $\sigma_k^{(\eta_i)} = (k+1)^{p_{\eta_i}}$, and $p_{\zeta_i} = p_{\eta_i} = -0.11 + 0.01i, 0.09 + 0.01i, 0.19 + 0.01i$ in *Scheme (S1)* for $i = 1, \ldots, 5$, respectively. Then, the training and testing accuracy on the benchmark datasets "MNIST" and "CIFAR-10" are given in Fig. 2(a)–(d), from which one can see that the smaller the privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the faster the Algorithm 1 converges. This is consistent with the convergence rate analysis in Theorem 2. Meanwhile, the cumulative differential privacy budget ε of Algorithm 1 is given in Fig. 2(e), from which one can see that the smaller the privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the smaller the cumulative differential privacy budget ε is. This is consistent with the privacy analysis in Theorem 4, and thus consistent with the tradeoff between the privacy and the convergence rate in Corollary 1.

Next, let step-sizes $\alpha_K = 0.1$, $\beta_K = 0.01$, and $\gamma_K = 0.1$, the sampling number $m_K = \lfloor 1.002^{2000} \rfloor + 1 = 55$, and privacy noise parameters $\sigma_k^{(\zeta_i)} = {p_{\zeta_i}}^{2000}$, $\sigma_k^{(\eta_i)} = {p_{\eta_i}}^{2000}$, and $p_{\zeta_i} = p_{\eta_i} = 0.99939 + i \cdot 10^{-5}, 0.99959 + i \cdot 10^{-5}, 0.99979 + i \cdot 10^{-5}$ in *Scheme (S2)* for $i = 1, \ldots, 5$, respectively. Then, the training and testing accuracy on the benchmark datasets "MNIST" and "CIFAR-10" are given in Fig. 3(a)–(d), from which one can see that the smaller the privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the faster the Algorithm 1 converges. This is consistent with the convergence rate analysis in Theorem 3. Meanwhile, the cumulative differential privacy budget ε of Algorithm 1 is given in Fig. 3(e), from which one can see that the smaller the privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ are, the smaller the cumulative differential privacy budget ε is. This is consistent with the privacy analysis in Theorem 5, and thus consistent with the tradeoff between the privacy and the convergence rate in Corollary 1.

Remark 23: Due to the increasing sample size m_K, the cumulative differential privacy budget ε decreases in the later stages of the iterations in the numerical experiment. In *Scheme (S1)*, the sampling number $m_K = \lfloor 0.00007 \cdot K^{1.78} \rfloor + 1 = O(K^{1.78})$. By Theorem 4, the cumulative differential privacy budget $\varepsilon = O\left(\frac{\ln(K+2)}{(K+1)^{0.22}}\right)$. Denote the function $\psi_1(t) = \frac{\ln(t+2)}{(t+1)^{0.22}}$. Then, it can be seen that the function $\psi_1(t)$ decreases when t satisfies $t + 1 \leq 0.22(t+2)\ln(t+2)$, i.e., $t \geq 87.54$. Thus, the cumulative differential privacy budget ε decreases when the maximum iteration number $K \geq 88$. This result is consistent with Fig. 2(e). Similarly, in *Scheme (S2)*, the sampling number $m_K = \lfloor 1.002^K \rfloor + 1 = O(1.002^K)$. By Theorem 5, the cumulative differential privacy budget $\varepsilon = O\left(\frac{K}{1.0016^K}\right)$. Denote the function $\psi_2(t) = \frac{t}{1.0016^t}$. Then, it can be seen that the function $\psi_2(t)$ decreases when $t \geq 625.49$. Thus, the cumulative differential privacy budget ε decreases when the maximum iteration number $K \geq 626$. This result is consistent with Fig. 3(e).

B. Comparison Between Schemes (S1) and (S2)

In this section, the comparison of Algorithm 1 with *Schemes (S1)* and *(S2)* between the convergence rate and the differential privacy level is given. Let $p_{\zeta_i} = p_{\eta_i} = 0.09 + 0.01i$ in *Scheme (S1)*, and $p_{\zeta_i} = p_{\eta_i} = 0.99959 + i \cdot 10^{-5}$ in *Scheme (S2)* for $i = 1, \ldots, 5$, respectively. Then, from Fig. 4(a)–(d), one can see that Algorithm 1 with *Scheme (S2)* converges faster than Algorithm 1 with *Scheme (S1)*, while from Fig. 4(e), one can see that the cumulative differential privacy budget ε of Algorithm 1 with

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

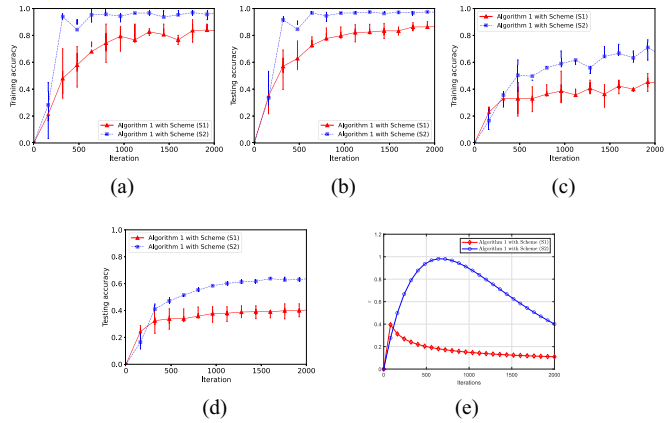


Fig. 4. Comparison of Algorithm 1 with *Schemes (S1)* and *(S2)* on accuracy and cumulative differential privacy budget ε. (a) Training accuracy on the "MNIST" dataset. (b) Testing accuracy on the "MNIST" dataset. (c) Training accuracy on the "CIFAR-10" dataset. (d) Testing accuracy on the "CIFAR-10" dataset. (e) Cumulative differential privacy budget ε.

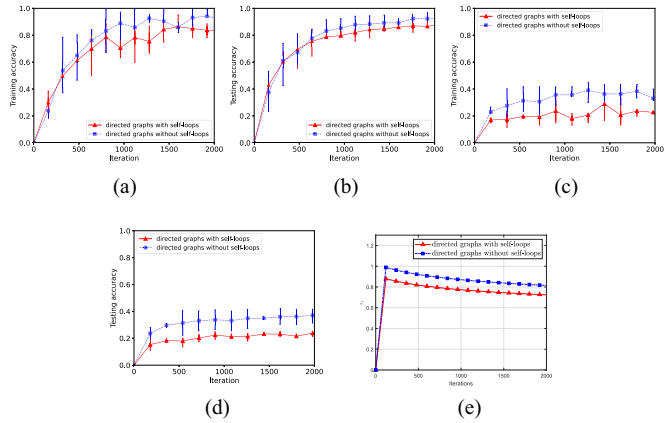


Fig. 5. Accuracy and cumulative differential privacy budget ε of Algorithm 1 with *Scheme (S1)* over directed graphs with and without self-loops. (a) Training accuracy on the "MNIST" dataset. (b) Testing accuracy on the "MNIST" dataset. (c) Training accuracy on the "CIFAR-10" dataset. (d) Testing accuracy on the "CIFAR-10" dataset. (e) Cumulative differential privacy budget ε.

Scheme (S1) is smaller than the cumulative differential privacy budget ε of Algorithm 1 with *Scheme (S1)*.

Remark 24: By [62, Corollary 8.1.19], the spectral radii $\rho(\mathcal{R})$ and $\rho(\mathcal{C})$ of directed graphs with self-loops are larger than those of directed graphs without self-loops. Then, by (24) , both *Schemes (S1)* and *(S2)* converge slower over directed graphs with self-loops than over directed graphs without self-loops. This result is consistent with Figs. 5(a) –(d) and 6(a)–(d). Note that $\sum_{j\in\mathcal{N}_{\mathcal{R},i}^-}\mathcal{R}_{ij}$ and $\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}$ of directed graphs with self-loops are larger than those of directed graphs without self-loops. Then, by Lemma 4, the cumulative differential privacy budget ε of both *Schemes (S1)* and *(S2)* over the directed graph with self-loops is smaller than the one over the directed graph without self-loops. This result is consistent with Figs. 5(e) and 6(e). As a result, self-loops in directed graphs slow down the convergence rate while enhancing the differential privacy level of Algorithm 1.

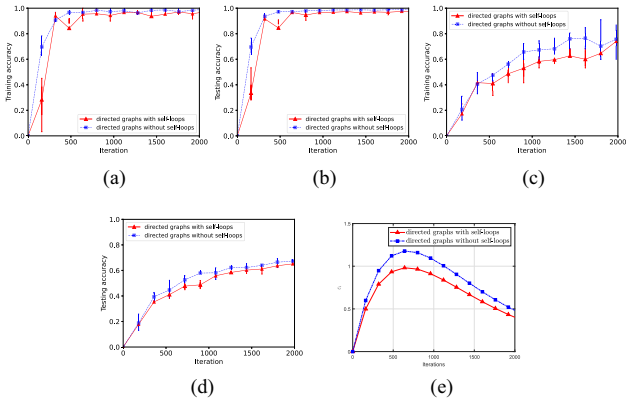


Fig. 6. Accuracy and cumulative differential privacy budget ε of Algorithm 1 with *Scheme (S2)* over directed graphs with and without self-loops. (a) Training accuracy on the "MNIST" dataset. (b) Testing accuracy on the "MNIST" dataset. (c) Training accuracy on the "CIFAR-10" dataset. (d) Testing accuracy on the "CIFAR-10" dataset. (e) Cumulative differential privacy budget ε.

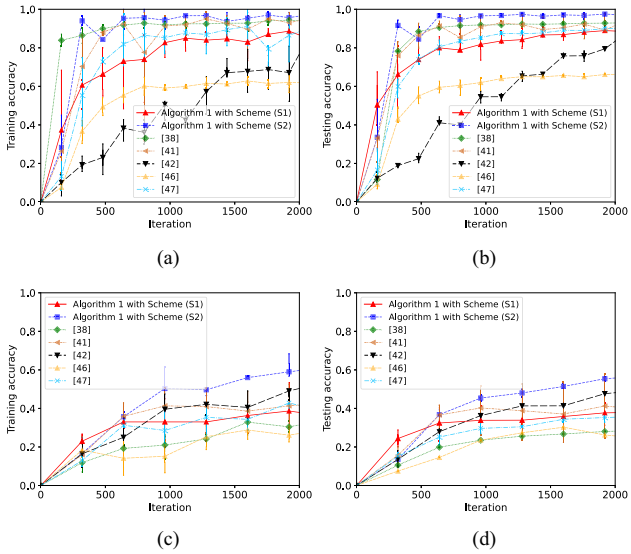


Fig. 7. Comparison of accuracy on the benchmark datasets "MNIST" and "CIFAR-10." (a) Training accuracy on the "MNIST" dataset. (b) Testing accuracy on the "MNIST" dataset. (c) Training accuracy on the "CIFAR-10" dataset. (d) Testing accuracy on the "CIFAR-10" dataset.

C. Comparison With Methods in [38], [41], [42], [46], and [47]

Let $p_{\zeta_i} = p_{\eta_i} = 0.99959 + i \cdot 10^{-5}$ in *Scheme (S2)* for $i = 1, \ldots, 5$, step-sizes $\alpha_K, \beta_K,$ and γ_K, the sampling number m_K, and privacy noise parameters $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$ in *Scheme (S1)* and [38], [41], [42], [46], [47] be the same as *Scheme (S2)* to ensure a fair comparison. Then, the comparison of the convergence rate and the differential privacy level between Algorithm 1 and the methods in [38], [41], [42], [46], and [47] is given in Fig. 7. From Fig. 7, one can see that Algorithm 1 with *Scheme (S2)* converges faster than methods in [38], [41], [42], [46], and [47].

A comparison of the differential privacy level between Algorithm 1 and the methods in [38], [41], [42], [46], and [47] is given

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

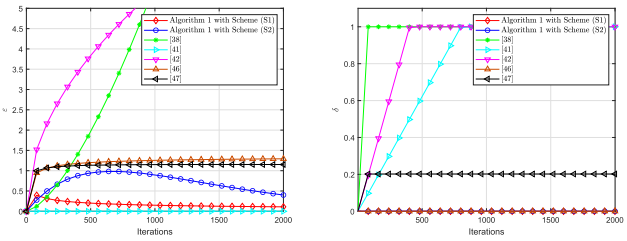


Fig. 8. Comparison of cumulative differential privacy budgets ε and δ. (a) ε under same $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$. (b) δ under same $\sigma_k^{(\zeta_i)}$ and $\sigma_k^{(\eta_i)}$.

in Fig. 8. By Fig. 8(a), the cumulative differential privacy budget ε of Algorithm 1 with both *Schemes (S1)* and *(S2)* is smaller than the ones in [38], [42], [46], and [47]. By Fig. 8(b), Wang and Başar [41] achieved the cumulative differential privacy budget $\delta = 1$ after 800 iterations, and thus, the one therein cannot protect sampled gradients after 800 iterations. Thus, Algorithm 1 with both *Schemes (S1)* and *(S2)* provides a higher differential privacy level than methods in [38], [41], [42], [46], and [47].

V. Conclusion

In this article, we have proposed a new differentially private gradient-tracking-based distributed stochastic optimization algorithm over directed graphs. Two novel schemes of step-sizes and the sampling number are given: *Scheme (S1)* uses polynomially decreasing step-sizes and the increasing sampling number with the maximum iteration number. *Scheme (S2)* uses constant step-sizes and the exponentially increasing sampling number with the maximum iteration number. By using the sampling parameter-controlled subsampling method, both schemes achieve the finite cumulative privacy budget even over infinite iterations, and thus enhance the differential privacy level compared to the existing ones. By using the gradient-tracking method, the almost sure and mean square convergence of the algorithm is shown for nonconvex objectives over directed graphs with spanning trees. Further, when nonconvex objectives satisfy the Polyak–Łojasiewicz condition, the polynomial mean square convergence rate [*Scheme (S1)*] and the exponential mean square convergence rate [*Scheme (S2)*] are given, respectively. Furthermore, the oracle complexity of the algorithm, the tradeoff between the privacy and the convergence are shown. Finally, numerical examples of distributed training on the benchmark datasets "MNIST" and "CIFAR-10" are given to show the effectiveness of the algorithm.

Appendix A
Useful Lemmas

Lemma A.1 (See [47, Lemma A.1]): If Assumption 2(i) holds for a function $h : \mathbb{R}^d \to \mathbb{R}$ with a global minimum $h(x^*)$, then following statements holds.

(i) $h(y) \le h(x) + \langle \nabla h(x), y - x\rangle + \frac{L}{2}\|y - x\|^2 \quad \forall x, y \in \mathbb{R}^d$.

(ii) $\|\nabla h(x)\|^2 \le 2L(h(x) - h(x^*)) \; \forall x \in \mathbb{R}^d$.

Lemma A.2 (See [62, Corollary 8.1.29, Thm. 8.4.4]): For any $n = 1, 2, \ldots$, let $A \in \mathbb{R}^{n\times n}$ be a nonnegative matrix and $x \in \mathbb{R}^n$ be a positive vector. Then, following statements hold.

(i) If there exists $\rho > 0$ such that $Ax \le \rho x$, then $\rho(A) \le \rho$.

(ii) If A is irreducible, then $\rho(A) > 0$ and there exists a positive vector $y = [y_1, \ldots, y_n]^\top \in \mathbb{R}^n$ such that $y^\top A = \rho(A) y^\top$.

Appendix B
Proof of Theorem 2

Let $0<\Gamma<1$ and $\omega_K = \Gamma \min\{r_1\alpha_K, r_2\beta_K, \frac{(v_1^\top v_2)\mu\gamma_K}{n}\}$. Then, the following four steps are given to prove Theorem 2.

Step 1: First, we prove that there exists a positive integer K_0 such that for any $K = K_0, K_0 + 1, \ldots$

$$\rho(A_K) \le 1 - \omega_K. \tag{12}$$

Since $2p_\alpha - p_\beta - 2\max\{\max_{i\in\mathcal{V}}\{p_{\zeta_i}\}, 0\} \ge 1$, $\frac{1}{2} < p_\beta < p_\alpha < p_\gamma < 1$, and $2p_\gamma - p_\alpha \ge 1$ in Assumption 4, there exists a positive vector $\tilde{u} \in \mathbb{R}^3$ and a positive integer K_0 such that for any $K = K_0, K_0 + 1, \ldots$, the following inequality holds:

$$A_K\tilde{u} \le (1 - \omega_K)\tilde{u}. \tag{13}$$

Then, by (13) and Lemma A.2(i), (12) holds for any $K = K_0, K_0 + 1, \ldots$.

Step 2: In this step, we prove that there exists a positive vector $\tilde{t} = [\tilde{t}_1, \tilde{t}_2, \tilde{t}_3]^\top$ such that for any $K = 0, 1, \ldots$, $\mathbb{E}(\tilde{t}^\top V_{K+1}) = O\left(\frac{1}{(K+1)^{\theta-p_\gamma}}\right)$. Note that for any $K = K_0, K_0 + 1, \ldots$, (12) holds. Then, by Lemma A.2(ii), there exists a positive vector $\tilde{t} = [\tilde{t}_1, \tilde{t}_2, \tilde{t}_3]^\top$ such that $\tilde{t}^\top A_K = \rho(A_K)\tilde{t}^\top \le (1 - \omega_K)\tilde{t}^\top$. Moreover, by Assumptions 1–4, (9) in Lemma 2 holds. Then, multiplying $\tilde{t}^\top$ on both sides of (9) implies that for any $k = 0, \ldots, K$

$$\begin{aligned}\mathbb{E}(\tilde{t}^\top V_{k+1}) &\le \tilde{t}^\top A_K \mathbb{E}V_k + \tilde{t}^\top u_k \\ &\le (1 - \omega_K)\mathbb{E}(\tilde{t}^\top V_k) + \tilde{t}^\top u_k.\end{aligned} \tag{14}$$

By Assumption 4, $\tilde{t}^\top u_k = O\left(\frac{(a_4+1)(\rho(\mathcal{R})^2+\rho(\mathcal{C})^2+1)}{a_4(K+1)^\theta}\right)$ holds for any $k = 0, \ldots, K$. Thus, iteratively computing (14) results in

$$\begin{aligned}&\mathbb{E}(\tilde{t}^\top V_{K+1}) \\ &= (1 - \omega_K)^{K+1}\mathbb{E}(\tilde{t}^\top V_0) \\ &\quad + O\left(\sum_{k=0}^{K}(1 - \omega_K)^k \frac{(a_4 + 1)(\rho(\mathcal{R})^2 + \rho(\mathcal{C})^2 + 1)}{a_4(K + 1)^\theta}\right) \\ &= (1 - \omega_K)^{K+1}\mathbb{E}(\tilde{t}^\top V_0) + O\left(\frac{(a_4 + 1)(\rho(\mathcal{R})^2 + \rho(\mathcal{C})^2 + 1)}{a_4\omega_K(K + 1)^\theta}\right).\end{aligned} \tag{15}$$

By the definition of ω_K, it can be seen that

$$\begin{aligned}&O\left(\frac{1}{\omega_K(K + 1)^\theta}\right) = O\left(\frac{1}{(K + 1)^{\theta-p_\gamma}}\right) \\ &(1 - \omega_K)^{K+1} = \exp\left((K + 1)\ln(1 - \omega_K)\right) \\ &\quad \le \exp(-(K + 1)\omega_K) = \exp(-O\left((K + 1)^{1-p_\gamma}\right)) \\ &\quad = o\left(\frac{1}{(K + 1)^{\theta-p_\gamma}}\right).\end{aligned} \tag{16}$$

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

By (16), we have $\mathbb{E}(\tilde{t}^\top V_{K+1}) = O\left(\frac{1}{(K+1)^{\theta-p_\gamma}}\right)$ for any $K = K_0, K_0+1, \ldots$. Thus, there exists $S_0 > 0$ such that $\mathbb{E}(\tilde{t}^\top V_{K+1}) \le \frac{S_0}{(K+1)^{\theta-p_\gamma}}$. Let $S = \max\{\mathbb{E}(\tilde{t}^\top V_1), 2^{\theta-p_\gamma}\mathbb{E}(\tilde{t}^\top V_2), \ldots, (K_0-1)^{\theta-p_\gamma}\mathbb{E}(\tilde{t}^\top V_{K_0-1}), S_0\}$. Then, for any $K = 0, 1, \ldots$, we have $\mathbb{E}(\tilde{t}^\top V_{K+1}) \le \frac{S}{(K+1)^{\theta-p_\gamma}}$, which leads to

$$\mathbb{E}(\tilde{t}^\top V_{K+1}) = O\left(\frac{(a_4+1)(\rho(\mathcal{R})^2+\rho(\mathcal{C})^2+1)}{a_4(K+1)^{\theta-p_\gamma}}\right) = O\left(\frac{1}{(K+1)^{\theta-p_\gamma}}\right). \tag{17}$$

Step 3: In this step, we prove that for any $i \in \mathcal{V}$ and $K = 0, 1, \ldots$, $\mathbb{E}\|\nabla F(x_{i,K+1})\|^2 = O(\frac{1}{(K+1)^{\theta-p_\gamma}})$. By Lemma A.1(i), we have

$$F(x_{i,K+1}) - F(\bar{x}_{K+1}) \le \langle \nabla F(\bar{x}_{K+1}), x_{i,K+1} - \bar{x}_{K+1}\rangle + \frac{L_1}{2}\|\bar{x}_{K+1} - x_{i,K+1}\|^2. \tag{18}$$

Note that for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, $\langle \mathbf{a}, \mathbf{b}\rangle \le \frac{\|\mathbf{a}\|^2+\|\mathbf{b}\|^2}{2}$ holds. Then, (18) can be rewritten as

$$\begin{aligned} &F(x_{i,K+1}) - F(\bar{x}_{K+1}) \\ &\le \frac{\|\nabla F(\bar{x}_{K+1})\|^2 + \|\bar{x}_{K+1} - x_{i,K+1}\|^2}{2} + \frac{L_1}{2}\|\bar{x}_{K+1} - x_{i,K+1}\|^2 \\ &= \frac{L_1+1}{2}\|\bar{x}_{K+1} - x_{i,K+1}\|^2 + \frac{\|\nabla F(\bar{x}_{K+1})\|^2}{2}. \end{aligned} \tag{19}$$

By Lemma A.1(ii), $\|\nabla F(\bar{x}_{K+1})\|^2 \le 2L_1(F(\bar{x}_{K+1}) - F(x^*))$. Substituting it into (19) gives $F(x_{i,K+1}) - F(\bar{x}_{K+1}) \le \frac{L_1+1}{2}\|\bar{x}_{K+1} - x_{i,K+1}\|^2 + L_1(F(\bar{x}_{K+1}) - F(x^*))$. Thus, we have

$$\begin{aligned} &F(x_{i,K+1}) - F(\bar{x}_{K+1}) \\ &\le \frac{L_1+1}{2}\sum_{i=1}^{n}\|\bar{x}_{K+1} - x_{i,K+1}\|^2 + L_1(F(\bar{x}_{K+1}) - F(x^*)) \\ &= \frac{L_1+1}{2}\|(W_1 \otimes I_d)x_{K+1}\|^2 + L_1(F(\bar{x}_{K+1}) - F(x^*)). \end{aligned} \tag{20}$$

Then, by (20), it can be seen that

$$\begin{aligned} &F(x_{i,K+1}) - F(x^*) \\ &= (F(x_{i,K+1}) - F(\bar{x}_{K+1})) + (F(\bar{x}_{K+1}) - F(x^*)) \\ &\le (L_1+1)\left(\mathbf{1}_3^\top \mathbb{E}V_{K+1}\right) = O\left(\mathbb{E}(\tilde{t}^\top V_{K+1})\right). \end{aligned} \tag{21}$$

Thus, combining (17) and (23) gives $\mathbb{E}(F(x_{i,K+1}) - F(x^*)) = O\left(\frac{(a_4+1)(\rho(\mathcal{R})^2+\rho(\mathcal{C})^2+1)}{a_4(K+1)^{\theta-p_\gamma}}\right)$. By Lemma A.1(ii), we have

$$\begin{aligned} \mathbb{E}\|\nabla F(x_{i,K+1})\|^2 &\le 2L\mathbb{E}(F(x_{i,K+1}) - F(x^*)) \\ &= O\left(\frac{(a_4+1)(\rho(\mathcal{R})^2+\rho(\mathcal{C})^2+1)}{a_4(K+1)^{\theta-p_\gamma}}\right) \\ &= O\left(\frac{1}{(K+1)^{\theta-p_\gamma}}\right). \end{aligned} \tag{22}$$

Hence, the polynomial mean square convergence rate is achieved.

Step 4: In this step, we prove that the oracle complexity of Algorithm 1 with *Scheme (S1)* is $O\left(\varphi^{-\frac{165(1+\varphi)}{\max\{55-27\varphi,46\}}}\right)$ for any $\varphi > 0$. Let $p_\alpha = \max\left\{1-\frac{\varphi}{5}, \frac{9}{10}\right\}$, $p_\beta = \max\left\{\frac{2}{3}\left(1-\frac{5\varphi}{5}\right), \frac{3}{5}\right\}$, $p_\gamma = \max\left\{1-\frac{\varphi}{10}, \frac{19}{20}\right\}$, $p_m = \max\left\{2-\frac{\varphi}{10}, \frac{39}{20}\right\}$, and $p_{\zeta_i} = p_{\eta_i} = \max\left\{\frac{\varphi}{10}, \frac{1}{20}\right\}$. Then, by *Step 3*, $\mathbb{E}\|\nabla F(x_{i,K+1})\|^2 = O\left(\frac{1}{(K+1)^{\frac{\max\{20-22\varphi,9\}}{60}}}\right)$ for any $i \in \mathcal{V}$ and $K = 0, 1, \ldots$. Thus, there exists $\Phi > 0$ such that the following inequality holds:

$$\mathbb{E}\|\nabla F(x_{i,K+1})\|^2 \le \frac{\Phi}{(K+1)^{\frac{\max\{20-22\varphi,9\}}{60}}}. \tag{23}$$

Let $K = \left\lfloor \left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}} \right\rfloor$. Then, by (23), we have

$$\mathbb{E}\|\nabla F(x_{i,K+1})\|^2 \le \frac{\Phi}{\left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}\frac{\max\{20-22\varphi,9\}}{60}}} = \varphi. \tag{24}$$

Thus, by (24) and Definition 1, x_{K+1} is a φ-suboptimal solution. Since $N(\varphi)$ is the smallest integer such that $x_{N(\varphi)}$ is a φ-suboptimal solution, we have

$$N(\varphi) \le \left\lfloor \left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}} \right\rfloor + 1. \tag{25}$$

Since $m_K = \lfloor a_4 K^{p_m} \rfloor + 1 = \left\lfloor a_4 \left\lfloor \left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}} \right\rfloor^{p_m} \right\rfloor + 1$, by Definition 2 and (25), the oracle complexity of Algorithm 1 with *Scheme (S1)* is given as follows:

$$\begin{aligned} &\sum_{k=0}^{N(\varphi)} m_K \\ &= (N(\varphi)+1)\left(\left\lfloor a_4 \left\lfloor \left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}} \right\rfloor^{p_m} \right\rfloor + 1\right) \\ &\le \left(\left\lfloor \left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}} \right\rfloor + 2\right) \\ &\quad \times \left(a_4 \left\lfloor \left(\frac{\Phi}{\varphi}\right)^{\frac{60}{\max\{20-22\varphi,9\}}} \right\rfloor^{p_m} + 1\right) \\ &= O(\varphi^{-\frac{177+3\max\{1-2\varphi,0\}}{9-11\max\{1-2\varphi,0\}}}). \end{aligned}$$

Therefore, this theorem is proved. ■

Appendix C
Proof of Lemma 3

For any $i \in \mathcal{V}$, let

$$\|\Delta x_{i,k}\|_1 = \begin{cases} \sup\limits_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\check{x}_0,\check{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|x_{i,0} - x_{i,0}'\|_1, & \text{if } k = 0 \\ \sup\limits_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\check{x}_{k-1},\check{y}_{k-1})\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|x_{i,k} - x_{i,k}'\|_1, & \text{if } k = 1, \ldots, K \end{cases}$$

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

$$\|\Delta y_{i,k}\|_1 = \begin{cases} \sup\limits_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_0,\breve{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|y_{i,0}-y_{i,0}'\|_1, & \text{if } k=0 \\ \sup\limits_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|y_{i,k}-y_{i,k}'\|_1, & \text{if } k=1,\dots,K. \end{cases}$$

Then, by Definition 5, we have

$$\Delta_{i,k}^q = \|\Delta x_{i,k}\|_1 + \|\Delta y_{i,k}\|_1.$$

Thus, the following two steps are given to prove Lemma 3.

Step 1: We compute $\|\Delta y_{i,k}\|_1$ for any $k=0,\dots,K$ and $i\in\mathcal{V}$. When $k=0$, we have

$$\begin{aligned}\|\Delta y_{i,0}\|_1 &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_0,\breve{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|y_{i,0}-y_{i,0}'\|_1 = \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_0,\breve{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|g_{i,0}-g_{i,0}'\|_1 \\ &= \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \|g_{i,0}-g_{i,0}'\|_1. \end{aligned} \tag{26}$$

Note that by Step 5 of Algorithm 1, m_K different data samples are taken uniformly from $\mathcal{D}_i$ and $\mathcal{D}_i'$, respectively. Then, there exists at most one pair of data samples $\lambda_{i,0,l_1}$ and $\lambda_{i,0,l_1}'$ such that $\lambda_{i,0,l_1}=\xi_{i,l_0}$ and $\lambda_{i,0,l_1}'=\xi_{i,l_0}'$. Thus, by (5), (26) can be rewritten as

$$\begin{aligned}&\|\Delta y_{i,0}\|_1 \\ &= \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \left\| \frac{1}{m_K}\sum_{l=1}^{m_K}(g_i(x_{i,0},\lambda_{i,0,l})-g_i(x_{i,0},\lambda_{i,0,l}')) \right\|_1 \\ &= \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \left\| \frac{1}{m_K}(g_i(x_{i,0},\lambda_{i,0,l_1})-g_i(x_{i,0},\lambda_{i,0,l_1}')) \right\|_1 \\ &\le \frac{1}{m_K}\left\|g_i(x_{i,0},\xi_{i,l_0})-g_i(x_{i,0},\xi_{i,l_0}')\right\|_1. \end{aligned} \tag{27}$$

By Assumption 2(i), there exists a constant C such that $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$ holds. Then, there exists exactly one pair of data samples ξ_{i,l_0} and ξ_{i,l_0}' such that (3) holds. Thus, (27) can be rewritten as

$$\|\Delta y_{i,0}\|_1 \le \frac{C}{m_K}. \tag{28}$$

When $k=1$, we have

$$\begin{aligned}&\|\Delta y_{i,1}\|_1 \\ &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_0,\breve{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|y_{i,1}-y_{i,1}'\|_1 \\ &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_0,\breve{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \left\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right)(y_{i,0}-y_{i,0}') \right. \\ &\quad \left. -\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^-}\mathcal{C}_{ji}(\breve{y}_{j,0}-\breve{y}_{j,0}')+(g_{i,1}-g_{i,1}')+(g_{i,0}-g_{i,0}') \right\|_1. \end{aligned} \tag{29}$$

By Definition 5, $\breve{y}_{j,0}=z_2=\breve{y}_{j,0}'$ holds for any $j\in\mathcal{N}_{\mathcal{C},i}^+$. Then, (29) can be rewritten as

$$\begin{aligned}\|\Delta y_{i,1}\|_1 = \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_0,\breve{y}_0)\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} &\left\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right)(y_{i,0}-y_{i,0}') \right. \\ &\left. +(g_{i,1}-g_{i,1}')+(g_{i,0}-g_{i,0}') \right\|_1. \end{aligned} \tag{30}$$

Since $y_{i,0}=g_{i,0}$ and $y_{i,0}'=g_{i,0}'$ hold for any $i\in\mathcal{V}$, by (28), (30) can be rewritten as

$$\begin{aligned}&\|\Delta y_{i,1}\|_1 \\ &= \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \left\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right)(y_{i,0}-y_{i,0}')+(g_{i,1}-g_{i,1}') \right. \\ &\quad \left. +(g_{i,0}-g_{i,0}') \right\|_1 \\ &\le \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \left\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right)(y_{i,0}-y_{i,0}') \right\|_1 \\ &\quad + \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|g_{i,1}-g_{i,1}'\|_1 + \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|g_{i,0}-g_{i,0}'\|_1 \\ &\le \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \left\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right)(y_{i,0}-y_{i,0}') \right\|_1 \\ &\quad + \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|g_{i,1}-g_{i,1}'\|_1 + \frac{C}{m_K}. \end{aligned} \tag{31}$$

By $\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')$ in Definition 3, there exists at most one pair of data samples $\lambda_{i,1,l_2}$ and $\lambda_{i,1,l_2}'$ such that $\lambda_{i,1,l_2}=\xi_{i,l_0}$ and $\lambda_{i,1,l_2}'=\xi_{i,l_0}'$. Then by (28), (31) can be rewritten as

$$\begin{aligned}\|\Delta y_{i,1}\|_1 &\le \sup_{\text{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \left\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right)(y_{i,0}-y_{i,0}') \right\|_1 + \frac{2C}{m_K} \\ &= \left|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right|\|\Delta y_{i,0}\|_1 + \frac{2C}{m_K} \\ &\le \left|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^+}\mathcal{C}_{ji}\right|\frac{C}{m_K} + \frac{2C}{m_K}. \end{aligned} \tag{32}$$

When $k=2,\dots,K$, we have

$$\|\Delta y_{i,k}\|_1 = \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2\text{nd}} \\ (\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O} \\ \text{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|y_{i,k}-y_{i,k}'\|_1$$

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

$$= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd}\\(\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \Bigg\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{+}}\mathcal{C}_{ji}\right)(y_{i,k-1}-y'_{i,k-1}) - \beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{-}}\mathcal{C}_{ji}(\breve{y}_{j,k-1}-\breve{y}'_{j,k-1}) + (g_{i,k}-g'_{i,k}) + (g_{i,k-1}-g'_{i,k-1})\Bigg\|_1. \tag{33}$$

By Definition 5, $\breve{y}_{j,k} = z_2 = \breve{y}'_{j,k}$ holds for any $j \in \mathcal{N}_{\mathcal{C},i}^{+}$. Then, (33) can be rewritten as

$$\begin{aligned}\|\Delta y_{i,k}\|_1 &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd}\\(\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \Bigg\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{+}}\mathcal{C}_{ji}\right)(y_{i,k-1}-y'_{i,k-1}) + (g_{i,k}-g'_{i,k}) + (g_{i,k-1}-g'_{i,k-1})\Bigg\|_1 \\ &\le \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \Bigg\| \left(1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{+}}\mathcal{C}_{ji}\right)(y_{i,k-1}-y'_{i,k-1})\Bigg\|_1 + \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|g_{i,k}-g'_{i,k}\|_1 + \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|g_{i,k-1}-g'_{i,k-1}\|_1.\end{aligned} \tag{34}$$

By $\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')$ in Definition 3, there exists at most one pair of data samples $\lambda_{i,k-1,l_k}$ and $\lambda'_{i,k-1,l_k}$ in $\mathcal{D}_{i,k-1}$ and $\mathcal{D}'_{i,k-1}$ such that $\lambda_{i,k-1,l_k} = \xi_{i,l_0}$ and $\lambda'_{i,k-1,l_k} = \xi'_{i,l_0}$, and there exists at most one pair of data samples $\lambda_{i,k,l_{k+1}}$ and $\lambda'_{i,k,l_{k+1}}$ such that $\lambda_{i,k,l_{k+1}} = \xi_{i,l_0}$ and $\lambda'_{i,k,l_{k+1}} = \xi'_{i,l_0}$. Thus, (34) can be rewritten as

$$\|\Delta y_{i,k}\|_1 \le \left|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{+}}\mathcal{C}_{ji}\right|\|\Delta y_{k-1}\|_1 + \frac{2C}{m_K}. \tag{35}$$

Iteratively computing (35) implies

$$\|\Delta y_{i,k}\|_1 \le \sum_{l=0}^{k-1}\left|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{+}}\mathcal{C}_{ji}\right|^{l}\frac{2C}{m_K} + \left|1-\beta_K\sum_{j\in\mathcal{N}_{\mathcal{C},i}^{+}}\mathcal{C}_{ji}\right|^{k}\frac{C}{m_K}. \tag{36}$$

Step 2: Next, we compute $\|\Delta x_{i,k}\|_1$ for any $k = 0,\dots,K$ and $i \in \mathcal{V}$. When $k = 0$, since the initial value $x_{i,0} = x'_{i,0}$ for any $i \in \mathcal{V}$, we have $\|\Delta x_{i,0}\|_1 = 0$. When $k = 1$, we have

$$\begin{aligned}\|\Delta x_{i,1}\|_1 &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd}\\(\breve{x}_0,\breve{y}_0)\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|x_1-x'_1\|_1 \\ &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd}\\(\breve{x}_0,\breve{y}_0)\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \Bigg\| \left(1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}\right)(x_{i,0}-x'_{i,0}) - \alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}(\breve{x}_{j,0}-\breve{x}'_{j,0}) - \gamma_K(y_{i,0}-y'_{i,0})\Bigg\|_1.\end{aligned} \tag{37}$$

By Definition 5, $\breve{x}_{j,0} = z_1 = \breve{x}'_{j,0}$ holds for any $i \in \mathcal{V}, j \in \mathcal{N}_{\mathcal{R},i}^{-}$. Moreover, since the initial value $x_{i,0} = x'_{i,0}$, (37) can be rewritten as

$$\begin{aligned}\|\Delta x_{i,1}\|_1 &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd}\\(\breve{x}_0,\breve{y}_0)\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|\gamma_K(y_{i,0}-y'_{i,0})\|_1 \\ &= \gamma_K \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|\Delta y_{i,0}\|_1.\end{aligned} \tag{38}$$

Then, substituting (28) into (39) implies

$$\|\Delta x_{i,1}\|_1 \le \frac{\gamma_K C}{m_K}. \tag{39}$$

When $k = 2,\dots,K$, we have

$$\begin{aligned}\|\Delta x_{i,k}\|_1 &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd}\\(\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \|x_k-x'_k\|_1 \\ &= \sup_{\substack{\mathcal{O}\subseteq\mathbb{R}^{2nd},\\(\breve{x}_{k-1},\breve{y}_{k-1})\in\mathcal{O}\\ \mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}} \Bigg\| \left(1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}\right)(x_{i,k-1}-x'_{i,k-1}) - \alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}(\breve{x}_{j,k-1}-\breve{x}'_{j,k-1}) - \gamma_K(y_{i,k-1}-y'_{i,k-1})\Bigg\|_1.\end{aligned} \tag{40}$$

By Definition 5, $\breve{x}_{j,k-1} = z_1 = \breve{x}'_{j,k-1}$ holds for any $j \in \mathcal{N}_{\mathcal{R},i}^{-}$, $i \in \mathcal{V}$. Then, (40) can be rewritten as

$$\begin{aligned}\|\Delta x_{i,k}\|_1 &= \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \Bigg\| \left(1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}\right)(x_{i,k-1}-x'_{i,k-1}) - \gamma_K(y_{i,k-1}-y'_{i,k-1})\Bigg\|_1 \\ &\le \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')} \Bigg\| \left(1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}\right)(x_{i,k-1}-x'_{i,k-1})\Bigg\|_1 + \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|\gamma_K(y_{i,k-1}-y'_{i,k-1})\|_1 \\ &= \left|\left(1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}\right)\right| \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|x_{i,k-1}-x'_{i,k-1}\|_1 + \gamma_K \sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|y_{i,k-1}-y'_{i,k-1}\|_1.\end{aligned} \tag{41}$$

Note that $\sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|x_{i,k-1}-x'_{i,k-1}\|_1 = \|\Delta x_{i,k-1}\|_1$ and $\sup_{\mathrm{Adj}(\mathcal{D}_i,\mathcal{D}_i')}\|y_{i,k-1}-y'_{i,k-1}\|_1 = \|\Delta y_{i,k-1}\|_1$. Then, (41) can be rewritten as

$$\|\Delta x_{i,k}\|_1 \le |1-\alpha_K\sum_{j\in\mathcal{N}_{\mathcal{R},i}^{-}}\mathcal{R}_{ij}|\|\Delta x_{i,k-1}\|_1 + \gamma_K\|\Delta y_{i,k-1}\|_1. \tag{42}$$

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

Iteratively computing (42) implies

$$\|\Delta x_{i,k}\|_1 \leq \gamma_K \sum_{l=0}^{k-1} |1 - \alpha_K \sum_{j \in \mathcal{N}_{\mathcal{R},i}^-} \mathcal{R}_{ij}|^{k-l-1} \|\Delta y_{i,l}\|_1. \quad (43)$$

Therefore, by (28), (32), (36), (39), and (43), this lemma is proved. ■

ACKNOWLEDGMENT

The material in this paper was not presented at any conference.

REFERENCES

[1] S. Pu, W. Shi, J. M. Xu, and A. Nedić, "Push-pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, Jan. 2021.

[2] N. Bhavsar and L. A. Prashanth, "Nonasymptotic bounds for stochastic optimization with biased noisy gradient oracles," *IEEE Trans. Autom. Control*, vol. 68, no. 3, pp. 1628–1641, Mar. 2023.

[3] O. Shamir and N. Srebro, "Distributed stochastic optimization and learning," in *Proc. 52nd Annu. Allerton Conf. Commun. Control Comput.*, Monticello, IL, USA, 2014, pp. 850–857.

[4] T. Y. Chen, A. G. Marques, and G. B. Giannakis, "DGLB: Distributed stochastic geographical load balancing over cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 7, pp. 1866–1880, Jul. 2017.

[5] M. Wang, C. Q. Xu, X. Y. Chen, H. Hao, L. J. Zhong, and D. O. Wu, "Design of multipath transmission control for information-centric Internet of Things: A distributed stochastic optimization framework," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9475–9488, Dec. 2019.

[6] T. T. Doan, S. T. Maguluri, and J. Romberg, "Convergence rates of distributed gradient methods under random quantization: A stochastic approximation approach," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4469–4484, Oct. 2021.

[7] K. H. Lu, H. X. Wang, H. S. Zhang, and L. Wang, "Convergence in high probability of distributed stochastic gradient descent algorithms," *IEEE Trans. Autom. Control*, vol. 69, no. 4, pp. 2189–2204, Apr. 2024.

[8] P. Xie, K. Y. You, R. Tempo, S. J. Song, and C. Wu, "Distributed convex optimization with inequality constraints over time-varying unbalanced digraphs," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4331–4337, Dec. 2018.

[9] G. N. Qu and N. Li, "Harnessing smoothness to accelerate distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 1245–1260, Sep. 2018.

[10] R. Xin, U. A. Khan, and S. Kar, "Variance-reduced decentralized stochastic optimization with accelerated convergence," *IEEE Trans. Signal Process.*, vol. 68, pp. 6255–6271, 2020.

[11] S. Pu and A. Nedić, "Distributed stochastic gradient tracking methods," *Math. Prog.*, vol. 187, no. 1, pp. 409–457, 2021.

[12] A. Koloskova, T. Lin, and S. U. Stich, "An improved analysis of gradient tracking for decentralized machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 11422–11435.

[13] R. Xin, A. K. Sahu, U. A. Khan, and S. Kar, "Distributed stochastic optimization with gradient tracking over strongly-connected networks," in *Proc. IEEE 58th Conf. Decis. Control*, Nice, France, 2019, pp. 8353–8358.

[14] J. L. Lei, P. Yi, J. Chen, and Y. G. Hong, "Distributed variable sample-size stochastic optimization with fixed step-sizes," *IEEE Trans Autom. Control*, vol. 67, no. 10, pp. 5630–5637, Oct. 2022.

[15] S. C. Zhao and Y. C. Liu, "Asymptotic properties of $\mathcal{S}$-$\mathcal{AB}$ method with diminishing step-size," *IEEE Trans. Autom. Control*, vol. 69, no. 5, pp. 3222–3229, May 2024.

[16] Y. Y. Chen, A. Hashemi, and H. Vikalo, "Accelerated distributed stochastic non-convex optimization over time-varying directed networks," *IEEE Trans. Autom. Control*, vol. 70, no. 4, pp. 2196–2211, Apr. 2025.

[17] Y. Q. Wang and T. Başar, "Gradient-tracking-based distributed optimization with guaranteed optimality under noisy information sharing," *IEEE Trans. Autom. Control*, vol. 68, no. 8, pp. 4796–4811, Aug. 2023.

[18] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. Conf. Comput. Commun. Secur.*, Denver, CO, USA, 2015, pp. 1322–1333.

[19] L. G. Zhu, Z. J. Liu, and S. Han, "Deep leakage from gradients," in *Proc. 33rd Int. Conf Neural Inf. Process Syst.*, 2019, pp. 14774–14784.

[20] J. F. Zhang, J. W. Tan, and J. M. Wang, "Privacy security in control systems," *Sci. China Inf. Sci.*, vol. 64, no. 7, 2021, Art. no. 176201.

[21] Y. Lu and M. H. Zhu, "Privacy preserving distributed optimization using homomorphic encryption," *Automatica*, vol. 96, pp. 314–325, 2018.

[22] J. W. Tan, J. M. Wang, and J. F. Zhang, "Cooperative secure parameter identification of multi-participant ARX systems—A threshold Paillier cryptosystem-based least-squares identification algorithm," *Scientia Sinica Informationis*, vol. 53, no. 12, 2472–2492, 2023.

[23] Y. Q. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711–4716, Nov. 2019.

[24] H. Gao, Y. Q. Wang, and A. Nedić, "Dynamics based privacy preservation in decentralized optimization," *Automatica*, vol. 151, 2023, Art. no. 110878.

[25] Y. Q. Wang and A. Nedić, "Decentralized gradient methods with time-varying uncoordinated stepsizes: Convergence analysis and privacy design," *IEEE Trans. Autom. Control*, vol. 69, no. 9, pp. 5352–5367, Aug. 2024.

[26] G. Ramos, A. P. Aguiarz, S. Kar, and S. Pequito, "Privacy preserving average consensus through network augmentation," *IEEE Trans. Autom. Control*, vol. 69, no. 10, pp. 6907–6919, Oct. 2024.

[27] Y. L. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[28] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.

[29] J. L. Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.

[30] L. P. Guo, J. M. Wang, Y. L. Zhao, and J. F. Zhang, "State estimation with protecting exogenous inputs via Cramér–Rao lower bound approach," 2025, *arXiv:2410.08756v2*.

[31] T. Ding, S. Y. Zhu, J. P. He, C. L. Chen, and X. P. Guan, "Differentially private distributed optimization via state and direction perturbation in multiagent systems," *IEEE Trans. Autom. Control*, vol. 67, no. 2, pp. 722–737, Feb. 2022.

[32] Y. Xuan and Y. Q. Wang, "Gradient-tracking based differentially private distributed optimization with enhanced optimization accuracy," *Automatica*, vol. 155, 2023, Art. no. 111150.

[33] Y. Q. Wang and A. Nedić, "Tailoring gradient methods for differentially private distributed optimization," *IEEE Trans. Autom. Control*, vol. 69, no. 2, pp. 872–887, Feb. 2024.

[34] L. Y. Huang, J. F. Wu, D. W. Shi, S. Dey, and L. Shi, "Differential privacy in distributed optimization with gradient tracking," *IEEE Trans. Autom. Control*, vol. 69, no. 9, pp. 5727–5742, Sep. 2024.

[35] A. T. Xie, X. L. Yi, X. F. Wang, M. Cao, and X. Q. Ren, "Differentially private and communication-efficient distributed nonconvex optimization algorithms," *Automatica*, vol. 177, 2025, Art. no. 112338.

[36] W. Huo, X. M. Chen, L. Y. Huang, K. H. Johansson, and L. Shi, "Differentially private dual gradient tracking for distributed resource allocation," *Automatica*, vol. 182, 2025, Art. no. 112521.

[37] J. H. Ding, G. N. Liang, J. B. Bi, and M. Pan, "Differentially private and communication efficient collaborative learning," in *Proc. AAAI Conf. Artif. Intell.*, Palo Alto, CA, USA, 2021, pp. 7219–7227.

[38] Y. L. Kang, Y. Liu, B. Niu, and W. P. Wang, "Weighted distributed differential privacy ERM: Convex and non-convex," *Comput. Secur.*, vol. 106, 2021, Art. no. 102275.

[39] J. Xu, W. Zhang, and F. Wang, "A(DP)^{2}SGD: Asynchronous decentralized parallel stochastic gradient descent with differential privacy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8036–8047, Nov. 2022.

[40] Y. Q. Wang and T. Başar, "Decentralized nonconvex optimization with guaranteed privacy and accuracy," *Automatica*, vol. 150, 2023, Art. no. 110858.

[41] Y. Q. Wang and T. Başar, "Quantization enabled privacy protection in decentralized stochastic optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 7, pp. 4038–4052, Jul. 2023.

[42] G. F. Yan, T. Li, K. Wu, and L. Q. Song, "Killing two birds with one stone: Quantization achieves privacy in distributed learning," *Digit. Signal Process.*, vol. 146, 2024, Art. no. 104353.

[43] C. X. Liu, K. H. Johansson, and Y. Shi, "Distributed empirical risk minimization with differential privacy," *Automatica*, vol. 162, 2024, Art. no. 111514.

[44] Z. Q. Chen and Y. Q. Wang, "Locally differentially private distributed online learning with guaranteed optimality," *IEEE Trans. Autom. Control*, vol. 70, no. 4, pp. 2521–2536, Apr. 2025.

[45] Z. Q. Chen and Y. Q. Wang, "Locally differentially private gradient tracking for distributed online learning over directed graphs," *IEEE Trans. Autom. Control*, vol. 70, no. 5, pp. 3040–3055, May 2025.

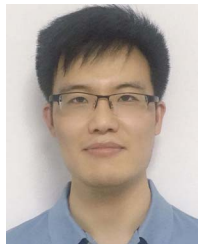
[46] J. M. Wang and J. F. Zhang, "Differentially private distributed stochastic optimization with time-varying sample sizes," *IEEE Trans. Autom. Control*, vol. 69, no. 9, pp. 6341–6348, Sep. 2024.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.

[47] J. L. Chen, J. M. Wang, and J. F. Zhang, "Differentially private distributed nonconvex stochastic optimization with quantized communication," *IEEE Trans. Autom. Control*, vol. 71, no. 1, pp. 368–383, Jan. 2026.
[48] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. Autom. control*, vol. 50, no. 5, pp. 655–661, May 2005.
[49] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Theor. Comput. Sci.*, vol. 8, no. 3/4, pp. 231–358, 2015.
[50] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak–Łojasiewicz condition," in *Proc. Mach. Learn. Knowl. Discov. Databases Euro. Conf.*, Riva del Garda, Italy, 2016, pp. 795–811.
[51] R. A. Adams and J. J. F. Fournier, "The Lebesgue spaces $L^p(\Omega)$," in *Sobolev Spaces*, Oxford, U.K.: Academic Press, 2003, ch. 2, sec. 1, pp. 25–28.
[52] J. L. Chen, J. M. Wang, and J. F. Zhang, "Differentially private gradient-tracking-based distributed stochastic optimization over directed graphs," 2025, *arXiv:2501.06793v6*.
[53] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich, "A unified theory of decentralized SGD with changing topology and local updates," in *Proc. Int. Conf., Mach. Learn.*, Vienna, Austria, 2020, pp. 5381–5393.
[54] X. R. Lian, C. Zhang, H. Zhang, C. J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, vol. 30, Long Beach, CA, USA, 2017, pp. 5336–5346.
[55] R. H. Byrd, G. M. Chin, J. Nocedal, and Y. C. Wu, "Sample size selection in optimization methods for machine learning," *Math. Prog.*, vol. 134, no. 1, pp. 127–155, 2012.
[56] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," in *Proc. Int. Conf. Learn. Representations*, Vancouver, BC, Canada, 2018, pp. 1–11.
[57] M. P. Friedlander and M. Schmidt, "Hybrid deterministic-stochastic methods for data fitting," *SIAM J. Sci. Comput.*, vol. 34, no. 3, pp. 1380–1405, 2012.
[58] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 770–778.
[59] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," National Institute of Standards and Technology, 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[60] A. Krizhevsky, V. Nair, and G. Hinton, "Canadian institute for advanced research, 10 classes," Department of Computer Science of University of Toronto, 2009. [Online]. Avalable: http://www.cs.toronto.edu/kriz/cifar.html
[61] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, *Dept. Comput. Sci., Univ. Toronto*, Toronto, ON, Canada, 2009. [Online]. Avalable: http://www.cs.utoronto.ca/kriz/learning-features-2009-TR.pdf
[62] R. A. Horn and C. R. Johnson, "Positive and nonnegative matrices," in *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012, ch. 8, sec. 4, pp. 529–545.
[63] V. A. Zorich, "Differential calculus," in *Mathematical Analysis I*. Berlin, German: Springer, 2015, ch. 5, sec. 4, pp. 238–239.
[64] Y. S. Chow and H. Teicher, "Integration in a Probability Space," in *Probability Theory: Independence, Interchangeability, Martingales*. New York, NY, USA: Springer, 2012, ch. 4, sec. 1, pp. 84–92.

Jialong Chen received the B.S. degree in mathematics and applied mathematics from Beihang University, Beijing, China, in 2023. He is currently working toward the Ph.D. degree majoring in system theory with the Academy of Mathematics and Systems Science, Chinese Academy of Science, Beijing.

His research focuses on privacy and security in stochastic systems.



Jimin Wang (Member, IEEE) received the B.S. degree in mathematics from Shandong Normal University, Jinan, China, in 2012, and the Ph.D. degree in operational research and cybernetics from the School of Mathematics, Shandong University, Jinan, in 2018.

From 2017 to 2018, he was a Joint Ph.D. Student with the School of Electrical Engineering and Computing, The University of Newcastle, Callaghan, NSW, Australia. From 2018 to 2020, he was a Postdoctoral Researcher with the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing. His current research interests include privacy and security in cyber-physical systems, stochastic systems, and networked control systems.

Dr. Wang was the recipient of the IEEE CSS Beijing Chapter Young Author Prize. He is a Member of the IEEE CSS Technical Committee on Security and Privacy, the IEEE CSS Technical Committee on Networks and Communication Systems, and the IFAC Technical Committee 1.5 on Networked Systems. Heis also an Associate Editor for *Systems and Control Letters* and *Journal of Automation and Intelligence*.

Ji-Feng Zhang (Fellow, IEEE) received the B.S. degree in mathematics from Shandong University, Jinan, China, in 1985, and the Ph.D. degree in operational research and cybernetics from the Institute of Systems Science, Chinese Academy of Sciences (CAS), Beijing, China, in 1991.

He is currently with the School of Automation and Electrical Engineering, Zhongyuan University of Technology, Zhengzhou, China, and the State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, CAS. His current research interests include system modeling, adaptive control, stochastic systems, and multiagent systems.

Dr. Zhang is an IFAC Fellow, CAA Fellow, SIAM Fellow, Member of the European Academy of Sciences and Arts, and Academician of the International Academy for Systems and Cybernetic Sciences. He was the recipient of the Second Prize of the State Natural Science Award of China in 2010 and 2015, respectively. He was the Vice-President of the Chinese Association of Automation, the Chinese Mathematical Society, and the Systems Engineering Society of China. He was the Vice-Chair of the IFAC Technical Board, Member of the Board of Governors, IEEE Control Systems Society; Convenor of Systems Science Discipline, and Academic Degree Committee of the State Council of China. He was the Editor-in-Chief, Deputy Editor-in-Chief, or Associate Editor for more than ten journals, including *Science China Information Sciences*, IEEE TRANSACTIONS ON AUTOMATIC CONTROL, and *SIAM Journal on Control and Optimization*.

Authorized licensed use limited to: Jifeng Zhang. Downloaded on August 29,2026 at 00:39:29 UTC from IEEE Xplore. Restrictions apply.