Locally Differentially Private Distributed Online Learning with Guaranteed Optimality

Ziqin Chen and Yongqiang Wang, *Senior Member, IEEE*

***Abstract*—Distributed online learning is gaining increased traction due to its unique ability to process large-scale datasets and streaming data. To address the growing public awareness and concern on privacy protection, plenty of algorithms have been proposed to enable differential privacy in distributed online optimization and learning. However, these algorithms often face the dilemma of trading learning accuracy for privacy. By exploiting the unique characteristics of online learning, this paper proposes an approach that tackles the dilemma and ensures both differential privacy and learning accuracy in distributed online learning. More specifically, while ensuring a diminishing expected instantaneous regret, the approach can simultaneously ensure a finite cumulative privacy budget, even in the infinite time horizon. To cater for the fully distributed setting, we adopt the local differential-privacy framework, which avoids the reliance on a trusted data curator that is required in the classic "centralized" (global) differential-privacy framework. To the best of our knowledge, this is the first algorithm that successfully ensures both rigorous local differential privacy and learning accuracy. The effectiveness of the proposed algorithm is evaluated using machine learning tasks, including logistic regression on the the "mushrooms" datasets and CNN-based image classification on the "MNIST" and "CIFAR-10" datasets.**

***Index Terms*—Distributed online optimization and learning, local differential privacy, instantaneous regret.**

I. Introduction

The modern data landscape, fueled by advances in web technologies, social media, and sensory devices, calls for evolved machine learning methods to handle the "big data" challenge [2]. Due to its unique ability to handle streaming data, online learning has emerged as an attractive paradigm to address this challenge [3]. In online learning, data are accessed and processed in a sequential manner, thereby obviating the requirement to process the entire dataset at once. This feature makes online learning algorithms particularly appealing for large-scale datasets and dynamic scenarios, where data are continually generated, ranging from financial markets, social media streams, to real-time sensor interpretation.

Traditional online learning algorithms (e.g., [4]–[6]) require transmitting all data streams to a central location for processing, leading to potential security risks like information leakage or model compromises in the event of a server attack [7]–[9]. Distributed online learning algorithms mitigate these risks by dispersing data among multiple networked learners, each updating its model with local streaming data, and then sharing updates across the network for parameter synchronization (see, e.g., [10]–[15]). While these algorithms eliminate the need for centralized data storage and associated security risks, information leakage during parameter transmission remains a concern, particularly via unencrypted communication channels. In fact, using these shared parameters, not only can an adversary infer sensitive attributes of the original data [8], but it can also precisely reversely infer raw training data (pixel-wise accurate for images [9]). To mitigate privacy breaches in distributed online learning, one natural approach is to patch an online learning algorithm with existing privacy mechanisms. For example, partially homomorphic encryption has been employed in both our prior results as well as others' to ensure privacy in distributed optimization [16]–[18]. However, such approaches suffer from heavy communication and computation overheads. Alternatively, time or spatially correlated noise-based approaches preserve privacy while maintaining accuracy by canceling out injected noises [19]–[23]. However, such approaches require each learner to have at least one neighbor not sharing information with potential adversaries, a condition that is difficult to guarantee in many multi-agent networks.

A. Related Literature

As differential privacy (DP) is gaining increased traction due to its mathematical rigor, implementation simplicity, and post-processing immunity [24], [25], plenty of results have been proposed to enable differential privacy in distributed optimization/learning [26]–[37]. However, most existing differential-privacy results for distributed optimization/learning explicitly rely on a trusted curator to aggregate and publish data in a centralized manner [26]–[29]. Recently, some differential-privacy solutions have been proposed for fully distributed optimization algorithms, including [30]–[35] as well as our own prior work [36]–[38]. However, since these results still use the classical centralized differential-privacy framework[1], they do not explicitly address protection against information inference by participating learners [38]. To ensure privacy in the scenario where a learner does not trust anyone else (including other participating learners) and aims to protect

The work was supported in part by the National Science Foundation under Grants ECCS-1912702, CCF-2106293, CCF-2215088, CNS-2219487, CCF-2334449, and CNS-2422312.

Ziqin Chen and Yongqiang Wang are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: yongqiw@clemson.edu).

[1] By centralized differential privacy, we mean the traditional differential-privacy framework, where a data aggregator/curator is needed to collect data from all learners and inject differential-privacy noises. Note that although results such as [31], [33]–[38] do not explicitly assume the existence of a data aggregator/curator, they still require participating learners to trust each other to cooperatively determine the amount of noises needed to achieve a certain level of privacy protection (privacy budget). Hence, they are also somewhat "centralized," and hence, different from the local model of differential privacy in this paper.

against an adversary that can observe every message shared in the network, we have to use local differential privacy (LDP), which obviates the need for a data curator/aggregator that is required in the traditional centralized differential-privacy framework to collect data and inject noises [39]–[41].

Unfortunately, the benefit of LDP comes at a great cost in optimization/learning accuracy. To the best of our knowledge, all existing differential-privacy solutions for distributed online learning have to either sacrifice learning accuracy [42]–[44] or allow the cumulative privacy budget to grow to infinity with time, implying diminishing privacy protection as time tends to infinity [45]–[53]. It is worth noting that our own prior work [36], [37] as well as others' [35], [54] have managed to retain provable convergence accuracy and differential privacy in distributed offline optimization. However, they still use the classical centralized DP framework, and it is unclear if the offline learning approaches can be extended to the online learning scenario, where data arrive sequentially.

B. Contributions

In this paper, we propose a locally differentially private distributed online learning algorithm that efficiently circumvents the tradeoff between privacy and learning accuracy. Our key idea is to exploit both the unique data patterns in online learning and a decaying interaction strength which enables the injection of DP noises with increasing variances (in contrast to decreasing DP-noise variances commonly used in the literature). The main contributions are summarized as follows:

- We demonstrate that for both strongly convex and general convex objective functions, our proposed locally differentially private distributed online learning algorithm ensures that the expected instantaneous regret decreases to zero, even in the presence of increasing DP-noise variances. Moreover, in the strongly convex scenario, we further prove that the expected tracking error (the deviation between the online algorithm's output and the optimal solution) also converges to zero. To the best of our knowledge, no such results have been reported before.
- In addition to ensuring provable convergence, we prove that our algorithm can simultaneously ensure rigorous LDP, even in the infinite time horizon. To our knowledge, this is the first time that both goals of LDP and provable convergence are achieved simultaneously in distributed online learning. This is in sharp contrast to existing results on differentially private distributed online learning in [45]–[53], where the cumulative privacy budget grows to infinity when time tends to infinity.
- Moreover, our LDP framework allows individual learners to choose heterogeneous privacy budgets in a fully distributed manner, making individual learners free to choose desired privacy strengths depending on practical needs.
- Besides providing a theoretic approach to selecting stepsizes based on global parameters such as graph Laplacian and the global Lipschitz constant (which is common in most existing distributed online optimization algorithms [10]–[15]), we also provide an approach for individual learners to select stepsizes independently of any global parameters, which is more amenable to distributed implementations.
- We evaluated the performance of our approach using several benchmark machine learning datasets, including the "mushrooms" dataset for logistic regression and the "MNIST" and "CIFAR-10" datasets for CNN-based image classification. The results corroborate the effectiveness of our approach. Notably, compared with existing differentially private distributed learning/optimization methods in [30], [45], [55], our algorithm demonstrates higher training and test accuracies.

The organization of the paper is as follows. Sec. II introduces the problem formulation and definitions for LDP. Sec. III presents a locally differentially private distributed online learning algorithm and discusses its computational complexity. Sec. IV analyzes the learning accuracy of the proposed algorithm. Sec. V provides an approach to selecting stepsizes independently of any global parameters. Sec. VI establishes the LDP guarantees. Sec. VII presents experimental results on benchmark datasets. Sec. VIII concludes the paper.

Notations: We use $\mathbb{R}^n$ to denote the n-dimensional Euclidean space. We also use $\mathbb{N}$ and $\mathbb{N}^+$ to denote the natural number and the positive natural number, respectively. I_n represents the identity matrix of dimension n and $\mathbf{1}_n$ represents the n-dimensional column vector with all entries equal to 1. We use $\|\cdot\|$ and $\|\cdot\|_1$ to represent the Euclidean norm and l^1-norm of a vector, respectively. The Kronecker product is denoted by $\otimes$. The stacked column vector of vectors or scalars $\theta_1, \cdots, \theta_m$ is denoted by $\text{col}\{\theta_1, \cdots, \theta_m\}$. The transpose of a matrix A is written as A^T. The notation $\lceil a \rceil$ refers to the smallest integer no less than a, and $\lfloor a \rfloor$ represents the largest integer no greater than a. We use $[m]$ to denote the set $\{1, 2, \cdots, m\}$. For any $\theta \in \mathbb{R}^n$, we use $\text{Pro}_\Theta(\theta) = \text{argmin}_{\theta' \in \Theta}\|\theta - \theta'\|_2$ to represent the Euclidean projection onto a set $\Theta \subseteq \mathbb{R}^n$. We also use $\text{Lap}(\varrho)$ to denote Laplace distribution with parameter $\varrho > 0$, featuring a probability density function $p_\varrho(x) \triangleq \frac{1}{2\varrho}e^{\frac{-|x|}{\varrho}}$. $\text{Lap}(\varrho)$ has a mean of zero and a variance of $2\varrho^2$.

II. Problem Statement

A. Distributed online learning

In distributed online learning, each Learner $i,\ i \in [m]$ must perform learning on streaming data that arrive sequentially. More specifically, at time t, Learner i acquires a data point a_t^i, which is independently and identically sampled from an unknown distribution over a sample space Ω_i. Using model parameter θ_t^i learned from data prior to time t, which is usually constrained in a convex subset Θ of $\mathbb{R}^n$, Learner i predicts a label $\hat{b}_t^i$ for the data x_t^i acquired at time t. When the true label $b_t^i \in \mathbb{R}$ is revealed, Learner i experiences a loss $l(\theta_t^i, \xi_t^i)$, where $\xi_t^i = (a_t^i, b_t^i)$ resides in $\mathcal{P}_i = \Omega_i \times \mathbb{R}$. The loss prompts Learner i to adjust its model parameter θ_t^i. The goal of distributed online learning is to let the m learners cooperatively find a common optimal parameter, based on sequentially acquired streaming data, for the following stochastic optimization problem:

$$\min_{\theta\in\Theta} \quad F(\theta) := \frac{1}{m}\sum_{i=1}^{m} f_i(\theta), \tag{1}$$

where $f_i(\theta) = \mathbb{E}_{\xi^i\sim\mathcal{P}_i}\left[l(\theta,\xi^i)\right]$ satisfies the following assumption:

Assumption 1. *(i) Θ is a convex and compact subset of $\mathbb{R}^n$ with nonempty interior; (ii) for all $i \in [m]$ and $x, y \in \Theta$, there exists some $\mu \geq 0$ such that $f_i(y) \geq f_i(x) + \nabla f_i(x)^T(y-x) + \frac{\mu}{2}\|x-y\|^2$ holds; and (iii) there exists some positive constant D such that $\|\nabla f_i(\theta)\| \leq D$ holds for all $\theta \in \Theta$.*

We describe the communication pattern among learners using an $m\times m$ matrix W. If Learners i and j can communicate with each other, then w_{ij} is positive, and $w_{ij} = 0$ otherwise. The set of learners that can directly interact with Learner i is called the neighboring set of Learner i and is represented as $\mathcal{N}_i$. We let $w_{ii} = -\sum_{j\in\mathcal{N}_i} w_{ij}$. The matrix W satisfies the following assumption:

Assumption 2. *The matrix W^2 satisfies $\mathbf{1}^T W = \mathbf{0}^T$ and $W\mathbf{1} = \mathbf{0}$. The eigenvalues of W satisfy (after arranged in an increasing order) $-2 < \delta_m \leq \cdots \leq \delta_2 < \delta_1 = 0$.*

To solve for (1) with f_i equal to the expected value of the loss function $l(\theta,\xi^i)$, we have to know the distribution $\mathcal{P}_i$ of ξ^i. However, in practice, the distribution $\mathcal{P}_i$ is usually unknown, which makes it impossible to directly compute $\mathbb{E}_{\xi^i\sim\mathcal{P}_i}\left[l(\theta,\xi^i)\right]$. To circumvent this problem, a common approach is reformulating (1) as the following Empirical Risk Minimization (ERM) problem:

$$\min_{\theta\in\Theta} F_t(\theta) \triangleq \frac{1}{m}\sum_{i=1}^{m} f_t^i(\theta),\ f_t^i(\theta) = \frac{1}{t+1}\sum_{k=0}^{t} l(\theta,\xi_k^i). \tag{2}$$

According to the law of large numbers, we have $\lim_{t\to\infty} F_t(\theta) = F(\theta)$, implying that the solution θ_t^* to the ERM problem (2) will gradually approach the solution θ^* to the problem (1) as time t tends to infinity (detailed proofs can be found in Lemma 2 in [57] and Section 5.1.2 in [58]). This is an intrinsic property of our ERM problem setting. It is worth noting that different from the conventional ERM problem, where all data are accumulated prior to performing training, here we have to perform online training from experience as more data are observed. In addition, since $\xi^i \sim \mathcal{P}_i$ are randomly streaming data, the gradients $\nabla l(\theta,\xi^i)$ are stochastic, which we assume to satisfy the following standard assumption [54]:

Assumption 3. *The random data points $\{\xi^i\}$ are independent of each other. In addition, (i) $\mathbb{E}[\nabla l(\theta,\xi^i)] = \nabla f_i(\theta)$; (ii) $\mathbb{E}[\|\nabla l(\theta,\xi^i) - \nabla f_i(\theta)\|^2] \leq \kappa^2$; and (iii) $\|\nabla l(x,\xi^i) - \nabla l(y,\xi^i)\| \leq L\|x-y\|$ for any $x, y \in \Theta$.*

Furthermore, given the streaming nature of data, the objective function $F_t(\theta)$ in (2) varies with time, which further leads to time-varying optimal solutions θ_t^*. Hence, to evaluate the quality of the parameters learned by learners through a distributed online learning algorithm at each time instant, we employ metrics of the *expected tracking error* $\mathbb{E}[\|\theta_t^i - \theta_t^*\|^2]$ and the *expected instantaneous regret* $\mathbb{E}[F_t(\theta_t^i) - F_t(\theta_t^*)]$.

Remark 1. The expected tracking error and the expected instantaneous regret are commonly used metrics in existing literature on online optimization and learning [59]–[62]. They capture the real-time performance of an online algorithm, and, hence, are well-suited in the online learning setting where data arrive sequentially [63].

B. Local differential privacy

Local differential privacy is a local (distributed) model of differential privacy for scenarios where no trusted data aggregator (curator) exists to aggregate data and execute a privacy mechanism. It is contrasted with the classic centralized differential privacy, where a trusted aggregator gathers all raw data and then executes a differentially private data publishing mechanism. In distributed learning and optimization, each learner maintains a local dataset and shares learned parameters with neighbors to collaboratively optimize these parameters. This information exchange has the risk of information leakage as malicious external attackers or curious neighbors might try to recover raw training data from shared parameters [7], [8]. To protect the privacy of all learners, we adopt LDP to address the most severe scenario: all communication channels can be compromised by malicious attackers and no learners are trustworthy. Consequently, not only does our LDP-based approach deters external adversaries from extracting raw data through shared information, but it also shields against neighboring curious learners within the network.

To facilitate privacy analysis, we need the definition of adjacency of local datasets [45], [50], [54]:

Definition 1. *(Adjacency) For any $t \in \mathbb{N}^+$ and any learner $i \in [m]$, given two local datasets $\mathcal{D}_t^i = \{\xi_1^i, \cdots, \xi_k^i, \cdots, \xi_t^i\}$ and $\mathcal{D}_t^{i'} = \{\xi_1^i, \cdots, \xi_k^{i'}, \cdots, \xi_t^i\}$, $\mathcal{D}_t^i$ is said to be adjacent to $\mathcal{D}_t^{i'}$ if there exists a time instant $k \in [1,t]$ such that $\xi_k^i \neq \xi_k^{i'}$ while $\xi_p^i = \xi_p^{i'}$ for all $p \in [1,t]$ and $p \neq k$.*

It can be seen that for any given time t, $\mathcal{D}_t^i$ is adjacent to $\mathcal{D}_t^{i'}$ if and only if $\mathcal{D}_t^i$ and $\mathcal{D}_t^{i'}$ differ in a single entry while all other entries are the same. Definition 1 also implies that for any given time t, two adjacent datasets $\mathcal{D}_t = \mathcal{D}_t^1 \cup \cdots \mathcal{D}_t^i \cup \cdots \cup \mathcal{D}_t^m$ and $\mathcal{D}_t' = \mathcal{D}_t^{1'} \cup \cdots \mathcal{D}_t^{i'} \cup \cdots \cup \mathcal{D}_t^{m'}$ differ in m entries. We use $\mathrm{Adj}(\mathcal{D}_t^i, \mathcal{D}_t^{i'})$ to denote the adjacent relationship between two local datasets $\mathcal{D}_t^i$ and $\mathcal{D}_t^{i'}$.

Remark 2. It is worth noting that our definition of adjacency corresponds to event-level LDP in the literature [25]. It allows m entries in the global datasets of all learners to be different, and is more stringent than most existing online results using the traditional centralized version of DP (e.g., [46]–[49], [51]–[53]), where at each time instant t, only one agent's one entry is allowed to be different. It is also worth noting that allowing one learner to have all data entries to be different (called user-level DP [25]) has been proven infeasible in distributed optimization/learning under the local model of DP [64]–[66].

[2] Our matrix $I + \epsilon W$ corresponds to the Perron matrix $P_\epsilon = I - \epsilon L$ used in [56], where L is the Laplacian matrix.

Algorithm 1 Locally differentially private distributed online learning for $i \in [m]$

```
**Input:** Random initialization $\theta_0^i \in \Theta$; $\lambda_t = \frac{\lambda_0}{(t+1)^v}$ with $\lambda_0 > 0$ and $v \in (\frac{1}{2}, 1)$; decaying sequence $\gamma_t = \frac{\gamma_0}{(t+1)^u}$ with $\gamma_0 > 0$ and $u \in (\frac{1}{2}, 1)$.
**for** $t = 0, 1, \cdots, T-1$ **do**
    Use all available data up to time $t$, i.e., $\xi_k^i \in \mathcal{D}_t^i,\ k \in [0, t]$ and the current parameter $\theta_t^i$ to compute:
    $d_t^i(\theta_t^i) = \frac{1}{t+1}\sum_{k=0}^{t} \nabla l(\theta_t^i, \xi_k^i)$.
    Add DP noises $\zeta_t^i$ to $\theta_t^i$, and then send the obscured value $y_t^i \triangleq \theta_t^i + \zeta_t^i$ to neighbors $j \in \mathcal{N}_i$.
    Receive $y_t^j$ from neighbors $j \in \mathcal{N}_i$.
    $\hat{\theta}_{t+1}^i = \theta_t^i + \sum_{j \in \mathcal{N}_i} \gamma_t w_{ij}(y_t^j - \theta_t^i) - \lambda_t d_t^i(\theta_t^i)$;
    $\theta_{t+1}^i = \text{Pro}_\Theta(\hat{\theta}_{t+1}^i)$.
**end for**
```

Given a distributed online learning problem (2), we denote the implementation of an online algorithm by Learner $i \in [m]$ as $\mathcal{A}_i$. Now we are in a position to present the definition of LDP [25]:

Definition 2. *(Local differential privacy). Let $\mathcal{A}_i(\mathcal{D}^i, \theta^{-i})$ denote the output of Learner i under a distributed learning algorithm with its local dataset $\mathcal{D}^i$ and all received information from neighbors θ^{-i}. Then, Learner i's implementation $\mathcal{A}_i$ is ϵ_i locally differentially private if the following inequality always holds for any two adjacent datasets $\mathcal{D}^i$, $\mathcal{D}^{i'}$:*

$$\mathbb{P}[\mathcal{A}_i(\mathcal{D}^i, \theta^{-i}) \in \mathcal{O}^i] \le e^{\epsilon_i}\mathbb{P}[\mathcal{A}_i(\mathcal{D}^{i'}, \theta^{-i}) \in \mathcal{O}^i], \quad (3)$$

where $\mathcal{O}^i$ represents the set of all possible observations.

The parameter ϵ_i measures the similarity (indistinguishability) of Learner i's output distributions under two adjacent datasets. A smaller value of ϵ_i indicates greater indistinguishability between the outputs for two adjacent datasets, implying a higher level of privacy protection.

In our definition of LDP, for Learner i, all received information from neighbors, i.e., θ^{-i}, is regarded as external information and beyond its control. This is different from the classic centralized DP definition used in existing differentially private distributed optimization/learning approaches [31], [33]–[38], which, in the absence of a data aggregator/curator, requires participating learners to trust each other and cooperatively determine the amount of noises needed to achieve a certain level of privacy protection (privacy budget). In fact, when no data aggregator/curator exists, such a centralized DP framework even allows agents to cooperatively decide (like a centralized data curator) how to mask shared information [38].

III. Locally Differentially Private Distributed online learning algorithm

A. Algorithm design

Our locally differentially private distributed online learning algorithm to solve problem (2) is summarized in Algorithm 1, in which DP noises $\zeta_t^i \in \mathbb{R}^n$ satisfy the following assumption:

Assumption 4. *For every learner $i \in [m]$ and $t \in \mathbb{N}$, each element of the DP-noise vector ζ_t^i follows Laplace distribution $Lap(\varrho_t^i)$ with $\varrho_t^i = \frac{\sigma^i}{\sqrt{2}}(t+1)^{\varsigma^i}$, where σ^i is a positive constant and the increasing rate of noise variances $\varsigma^i \in (0, \frac{1}{2})$ satisfies*

$$\max_{i \in [m]}\{\varsigma^i\} + \frac{1}{2} < u < v < 1, \quad (4)$$

with u and v the decaying rates of the decaying sequence γ_t and the stepsize λ_t in Algorithm 1, respectively.

Instead of using DP noises with decaying variances, we employ DP noises with increasing variances in Algorithm 1. This is fundamentally different from existing results on differentially private distributed optimization, such as [30]–[32], [34], [45]–[52], and is key for us to ensure both accurate convergence and strong differential privacy with a finite cumulative privacy budget even in the infinite time horizon. In fact, most existing results on differentially private distributed optimization have to either sacrifice accurate convergence [42]–[44] or allow the cumulative privacy budget to grow to infinity (meaning diminishing privacy protection as iteration tends to infinity) [45]–[53], and, to our knowledge, our approach is the first to achieve both accurate convergence and differential privacy in the infinite time horizon for online learning.

One key reason for our algorithm to ensure robustness to DP noises is using a decaying sequence γ_t, which can effectively suppress the influence of DP noises with increasing variances, and, hence, ensuring accurate convergence. This approach is inspired by our recent result on distributed offline optimization [36]. Nevertheless, it is worth noting that compared with the result in [36], where the objective function is predetermined and the same for all iterations, the objective function here changes over iterations due to sequentially arriving data. Furthermore, unlike [36] where the optimal solutions can be any point in $\mathbb{R}^n$, here we consider optimization problems where the optimal solutions have to be restricted in a convex set Θ. This constraint makes convergence analysis much more challenging because the nonlinearity induced by projection (necessary to address set constraints) poses challenges to both optimality analysis and consensus characterization.

Moreover, we propose a novel gradient computation strategy that exploits historical data. This strategy improves learning accuracy and reduces the sensitivity of our algorithm, which is key to ensuring a finite cumulative privacy budget even in the infinite time horizon. This is in sharp contrast to existing DP solutions for distributed online optimization/learning [45]–[53], whose cumulative privacy budgets explode to infinity as the number of iterations tends to infinity, implying diminishing privacy protection in the infinite time horizon. In addition, as the data point at any single iteration t might be lost or corrupted, our strategy of using all available data up to time t also enhances the robustness of the learning algorithm. The advantage of this strategy is clearly demonstrated later in experimental results (see Fig. 1-Fig. 3) and privacy analysis (see Eq. (25)).

Remark 3. Note that all existing results on differentially private distributed online optimization follow the approach of patching DP noises with a given existing distributed optimization/learning algorithm (e.g., [45]–[53]), which does not fully exploit the flexibilities in DP design and optimization algorithm design. In fact, almost all existing distributed optimization algorithms (which are designed without considering privacy) are not robust to DP noises (since directly incorporating DP noises into these optimization algorithms renders them unable to guarantee convergence to the exact optimal solution). Hence, a direct combination of these existing algorithms with DP designs has to sacrifice either DP strength or convergence accuracy. In contrast, by incorporating a judiciously designed decaying factor γ_t to gradually attenuate the influence of DP noises, we co-design the optimization algorithm and DP-noise injection mechanism, which enables us to achieve both differential privacy and accurate convergence.

Remark 4. A commonly used approach to enabling privacy protection in distributed optimization/learning is to broadcast $\theta^i_t + \gamma_t\zeta^i_t$ and make the consensus of optimization variables θ^i_t unaffected by the decaying sequence γ_t [30]. Although this approach reduces the amount of noises injected into the algorithm, and, hence, will make convergence easier to happen, its diminishing noise variance also jeopardizes the strength of privacy protection, leading to an exploding cumulative privacy budget (implying diminishing privacy protection as iteration proceeds) under the stepsize strategy used in our paper.

B. Algorithm complexity discussion

In this subsection, we discuss the computational complexity of our strategy that uses historical data in our Algorithm 1. It is intuitive that using all data available at time t can increase execution time of the algorithm compared with traditional online optimization/learning algorithms [10]–[15] that use only one current data sample. However, here we show that the increased computational complexity can be mitigated by exploiting the characteristics of learning problems. More specifically, if the loss function $l(\theta, \xi^i)$ is a polynomial function of θ, we can make sure that our strategy of using all historical data has the same order of computational complexity as those only using one data point at each time instant.

We illustrate the idea by using the Ridge regression problem [67]. In the Ridge regression problem, the loss function is a quadratic function of θ, i.e., $l(\theta, \xi^i_k) = (b^i_k - a^i_k\theta)^T(b^i_k - a^i_k\theta) + \alpha_t\|\theta\|^2$. The gradient $d^i_t(\theta^i_t)$ at each time t is given as $d^i_t(\theta^i_t) = \frac{1}{t+1}\sum_{k=0}^{t}\nabla l(\theta^i_t, \xi^i_k)$ with $\nabla l(\theta, \xi^i_k) = -2(a^i_k)^T(b^i_k - a^i_k\theta) + 2\alpha_t\theta$. Hence, we have

$$d^i_t(\theta^i_t) = \frac{d^i_{t-1}(\theta^i_t) \times t + \nabla l(\theta^i_t, \xi^i_t)}{t+1}. \quad (5)$$

Using the linear interpolation (two-point interpolation) method, we can obtain $d^i_{t-1}(\theta^i_t)$ as follows:

$$d^i_{t-1}(\theta^i_t) = d^i_{t-1}(\theta^i_{t-1})\frac{\theta^i_t - \theta^i_{t-2}}{\theta^i_{t-1} - \theta^i_{t-2}} + d^i_{t-1}(\theta^i_{t-2})\frac{\theta^i_t - \theta^i_{t-1}}{\theta^i_{t-2} - \theta^i_{t-1}}. \quad (6)$$

In the preceding equality, $d^i_{t-1}(\theta^i_{t-2})$ can be expressed as

$$d^i_{t-1}(\theta^i_{t-2}) = \frac{d^i_{t-2}(\theta^i_{t-2}) \times (t-1) + \nabla l(\theta^i_{t-2}, \xi^i_{t-1})}{t}, \quad (7)$$

where the term $d^i_{t-2}(\theta^i_{t-2})$ has been calculated at time $t-2$ and $\nabla l(\theta^i_{t-2}, \xi^i_{t-1})$ can be computed at time t.

Therefore, by combining (5), (6), and (7), we can see that the gradient $d^i_t(\theta^i_t) = \frac{1}{t+1}\sum_{k=0}^{t}\nabla l(\theta^i_t, \xi^i_k)$ needed at time t can be computed in a recursive manner. By simply storing two gradients computed in the prior two time instants, we can keep the computational complexity invariant with time.

Using a similar argument, we can show that when the loss function is a polynomial function (like in Lasso and polynomial regression) of θ of order n, we can exploit the iterative formulation in (5) and the Lagrange interpolation method to control the computational complexity of the gradient to be $\mathcal{O}(n+1)$.

It is worth noting that since every continuous function can be approximated as closely as desired by a polynomial function according to the Weierstrass approximation theorem [68], the interpolation-based approach can be used in other non-polynomial loss functions to mitigate the computational complexity of our gradient computation strategy. In fact, the sigmoid and logarithmic loss functions in logistic regression have been shown to be easily approximated by polynomials [69]. Even the cross-entropy and focal losses in neural networks have also been shown to be approximatable efficiently by a series of weighted polynomial bases [70].

IV. Tracking accuracy analysis

In this section, we systematically analyze the learning accuracy of Algorithm 1 under both strongly convex and general convex objective functions.

A. Tracking analysis with strongly convex objective functions (μ in Assumption 1 is positive)

We first analyze the time variation of the optimal parameter:

Lemma 1. *Denote θ^*_t as the optimal solution to the online optimization problem* (2) *at time t. Under Assumption 1 with $\mu > 0$ and Assumption 3, we have*

$$\mathbb{E}[\|\theta^*_{t+1} - \theta^*_t\|^2] \leq \mathcal{O}\left((t+1)^{-2}\right), \quad (8)$$

which implies $\lim_{t\to\infty}\mathbb{E}[\|\theta^*_{t+1} - \theta^*_t\|^2] = 0$.

Proof. Due to space limitations, we leave the proof to the extended version available at [1]. □

Remark 5. Lemma 1 reveals a key property of our learning problem (2): as learning progresses, the variation in optimal parameters decreases with time at a rate of $\mathcal{O}((t+1)^{-2})$. This decreasing rate is an intrinsic property of the problem setting in (2). Specifically, the objective function is the average of loss functions over a growing number of samples. As more data points are acquired, any single data point's impact on the overall loss becomes progressively smaller. The cumulative moving average acts as a form of memory, which makes the learning process smoother and more stable.

Notably, only when the data distribution $\mathcal{P}_i$ is time-invariant, the optimal parameter to the problem (2) could converge to a fixed constant. However, our result in Lemma 1 is applicable even when the data distribution $\mathcal{P}_i$ is not time-invariant, or in other words, the optimal parameter does not have to converge to a constant value. For example, if the optimal parameter follows the sequence $\theta_t^* = 1, 1+\frac{1}{2}, 1+\frac{1}{2}+\frac{1}{3}, \cdots$, it can be verified that the result in Lemma 1 still applies, whereas the sequence never converges.

We now characterize the expected tracking error of Algorithm 1 for strongly convex objective functions.

Theorem 1. *Under Assumptions 1-4 with $\mu > 0$, if $0 < \gamma_0 \leq \frac{1}{-3\delta_m}$ and $0 < \lambda_0 \leq \frac{-\gamma_0\delta_2\mu}{\mu^2+8L^2}$ hold, the expected tracking error of Algorithm 1 satisfies*

$$\mathbb{E}[\|\theta_{t+1}^i - \theta_{t+1}^*\|^2] \leq \mathcal{O}(t^{-\beta}), \tag{9}$$

for all $t > 0$, where the rate β satisfies $\beta = \min\{1-v, 2u-2\varsigma-1\}$ with $\varsigma \triangleq \min_{i\in[m]}\{\varsigma^i\}$.

Proof. See Appendix B. □

Theorem 1 shows that even in the presence of time-increasing DP-noise variances ϱ_t^i ($\varsigma^i > 0$), Algorithm 1 can still track time-varying optimal parameters with time, with the expected tracking error diminishing at a rate of $\mathcal{O}(t^{-\beta})$. This proves that Algorithm 1 is capable of preserving learning accuracy even in the presence of large DP noises.

In the following corollary, we quantify the dynamic regret of Algorithm 1, which measures accumulated losses [10] of our algorithm in all T iterations:

Corollary 1. *Under the conditions in the statement of Theorem 1, the dynamic regret of Algorithm 1 satisfies*

$$\sum_{t=1}^{T}\mathbb{E}\left[F_t(\theta_t^i)\right] - \sum_{t=1}^{T}\mathbb{E}\left[F_t(\theta_t^*)\right] \leq \mathcal{O}\left(T^{1-\frac{\beta}{2}}\right), \tag{10}$$

for any $i \in [m]$.

Proof. According to the definition $f_j(\theta)=\mathbb{E}[l(\theta,\xi^j)]$, we have

$$\begin{aligned}
&\mathbb{E}\left[F_t(\theta_t^i)\right] - \mathbb{E}\left[F_t(\theta_t^*)\right]\\
&= \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}\left[\frac{1}{t+1}\sum_{k=0}^{t} l(\theta_t^i,\xi_k^j) - \frac{1}{t+1}\sum_{k=0}^{t} l(\theta_t^*,\xi_k^j)\right]\\
&= \frac{1}{m}\sum_{j=1}^{m}(f_j(\theta_t^i) - f_j(\theta_t^*)) = \frac{1}{m}\sum_{j=1}^{m}\mathbb{E}\left[\nabla f_j(\varphi_t^{ij})^T(\theta_t^i - \theta_t^*)\right]\\
&\leq D\mathbb{E}\left[\|\theta_t^i - \theta_t^*\|\right] \leq \mathcal{O}(t^{-\frac{\beta}{2}}),
\end{aligned} \tag{11}$$

with $\varphi_t^{ij} \triangleq q_j\theta_t^i + (1-q_j)\theta_t^*$ for any $q_j \in (0,1)$. Here, we have used the mean value theorem in the third equality, Assumption 1-(iii) in the first inequality, and relationship (9) in the last inequality.

By using (11) and the relation $\sum_{t=2}^{T} t^{-\alpha} \leq \int_{t=1}^{T}\frac{1}{x^{\alpha}}dx \leq \frac{1}{1-\alpha}T^{1-\alpha}$ valid for any $\alpha \in (0,1)$, we arrive at

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbb{E}\left[F_t(\theta_t^i)\right] - \sum_{t=1}^{T}\mathbb{E}\left[F_t(\theta_t^*)\right]\\
&\leq \mathcal{O}\left(T^{1-\frac{\beta}{2}}\right) + D\mathbb{E}\left[\|\theta_1^i - \theta_1^*\|\right] \leq \mathcal{O}\left(T^{1-\frac{\beta}{2}}\right),
\end{aligned} \tag{12}$$

where we have omitted the constant $\frac{2D}{2-\beta}$ in the first inequality and $\mathcal{O}(1)$ in the last inequality. □

Corollary 1 proves that Algorithm 1 can achieve a sublinear dynamic regret even under LDP constraints. This result is consistent with the dynamic regret result in [10], which shows that the dynamic regret is bounded by the path length of an online optimization problem. In fact, under our ERM formulation in (2), the path length can be quantitatively bounded by $\mathbb{E}[\|\theta_{t+1}^* - \theta_t^*\|] \leq \mathcal{O}((t+1)^{-1})$, as established in Lemma 1 (see Eq. (8)). Moreover, this upper bound has been incorporated into our convergence result in Theorem 1 (see Eq. (38) for details). Therefore, we can derive a sublinear dynamic regret in Corollary 1 based on Theorem 1.

B. Tracking analysis with convex objective functions (μ in Assumption 1 is nonnegative)

In this section, we examine the tracking performance of Algorithm 1 for general convex objective functions.

Theorem 2. *Under Assumptions 1-4 with $\mu \geq 0$, if $\frac{2}{3} < \frac{1+2u}{3} < v < 1$, $0 < \gamma_0 \leq \frac{1}{-3\delta_m}$, and $0 < \lambda_0 \leq \frac{-\delta_2\gamma_0}{2(L^2+\kappa^2+D^2)}$ hold, the expected instantaneous regret of Algorithm 1 satisfies*

$$\mathbb{E}\left[F_t(\theta_t^i) - F_t(\theta_t^*)\right] \leq \mathcal{O}(t^{-\beta}), \tag{13}$$

for all $t > 0$, where the rate β satisfies $\beta = \frac{1-v}{2}$.

Proof. See Appendix C. □

Theorem 2 presents the expected instantaneous regret of Algorithm 1 when the objective functions are convex. However, analyzing parameter tracking errors is challenging for convex objective functions due to the possible existence of multiple optimal solutions with identical gradients. In such cases, the gradient's change does not provide sufficient information to establish an upper bound on the parameter tracking error.

V. Extension: stepsize selection without global parameters

In Theorem 1 and Theorem 2, the design of the stepsize sequence λ_t and the decaying sequence γ_t for Algorithm 1 requires knowledge of global parameters, such as the eigenvalues of the matrix W, the Lipschitz constant L, and the strongly convex coefficient μ of the objective function. Obtaining these global parameters might be challenging for individual learners in practical distributed implementations. Therefore, in this section, we discuss the tracking performance of Algorithm 1 when the stepsize and decaying sequences are designed without any knowledge of global parameters.

More specifically, we establish the following theorems for strongly convex and convex objective functions, respectively.

Theorem 3. *Under Assumptions 1-4 with $\mu > 0$, if $\frac{1}{2} < u < v < 1$ holds, then for any positive constants λ_0 and γ_0, the expected tracking error of Algorithm 1 satisfies*

$$\mathbb{E}[\|\theta_{t+1}^i - \theta_{t+1}^*\|^2] \leq \mathcal{O}\left((t-t_0)^{-\beta}\right), \tag{14}$$

for all $t > t_0$, where the rate β satisfies $\beta = \min\{1-v, 2u-2\varsigma-1\}$ with $\varsigma \triangleq \min_{i\in[m]}\{\varsigma^i\}$ and the positive constant t_0 is given by

$$t_0 = \left\lceil \max\left\{(-3\delta_m\gamma_0)^{\frac{1}{u}} - 1, \left(\frac{(\mu^2+8L^2)\lambda_0}{-\delta_2\mu\gamma_0}\right)^{\frac{1}{v-u}} - 1\right\}\right\rceil.$$

Proof. Due to space limitations, we leave the proof to the extended version available at [1]. □

Theorem 4. *Under Assumptions 1-4 with $\mu \geq 0$, if $\frac{2}{3} < \frac{2u+1}{3} < v < 1$ holds, then for any positive constants λ_0 and γ_0, the expected instantaneous regret of Algorithm 1 satisfies*

$$\mathbb{E}\left[F_t(\theta_t^i) - F_t(\theta_t^*)\right] \leq \mathcal{O}\left(t^{-\beta}\right), \tag{15}$$

for all $t > t_0'$, where the rate β satisfies $\beta = \frac{1-v}{2}$ and the positive constant t_0' is given by

$$t_0' = \Bigg\lceil \max\Bigg\{(-3\delta_m\gamma_0)^{\frac{1}{u}} - 1, \\ \left(\frac{2(L^2+\kappa^2+D^2)\lambda_0}{-\delta_2\gamma_0}\right)^{\frac{2}{3v-2u-1}} - 1\Bigg\}\Bigg\rceil.$$

Proof. Due to space limitations, we leave the proof to the extended version available at [1]. □

The compactness of the parameter set Θ in Algorithm 1 ensures that both the expected tracking error and the expected instantaneous regret are bounded before time instant t_0 in Theorem 3 (or t_0' in Theorem 4).

Remark 6. The convergence results in Theorems 1 and 2 need global information, such as the eigenvalues δ_2 and δ_m of the matrix W, the Lipschitz constant L, and the strongly convex coefficient μ, to determine the values of λ_0 and γ_0. To the contrary, the results in Theorems 3 and 4 hold for any positive constants λ_0 and γ_0, and, hence, are applicable even when global information, such as the eigenvalues of the matrix W, the Lipschitz constant L, and the strongly convex coefficient μ, are inaccessible.

Remark 7. The decaying sequence γ_t leads to a decaying coupling strength. However, we prove in Theorems 1 through 4 that this decaying coupling strength is still sufficient to ensure that all learners converge to the global optimal solution. Of course, the decaying coupling strength will reduce the convergence rate. We use the convergence result in Theorem 2 as an example to illustrate this tradeoff. It is clear that the convergence rate $\mathcal{O}(t^{-\frac{1-v}{2}})$ in Theorem 2 decreases with an increase in the decaying rate v of the stepsize λ_t. Given the condition $\frac{2}{3} < \frac{1+2u}{3} < v < 1$ presented in the statement of Theorem 2, we can see that an increase in the parameter u (corresponding to a faster decaying sequence γ_t) corresponds to an increase in the parameter v, resulting in a decreased convergence rate $\mathcal{O}(t^{-\frac{1-v}{2}})$ from Theorem 2.

VI. Local-differential-privacy analysis

In this section, we prove that besides accurate convergence, Algorithm 1 can simultaneously ensure rigorous ϵ_i-LDP for each learner, with the cumulative privacy budget guaranteed to be finite even when the number of iterations T tends to infinity. To this end, we first provide a definition for the sensitivity of Learner i's implementation $\mathcal{A}_i$ of Algorithm 1:

Definition 3. *(Sensitivity) The sensitivity of Learner i's implementation $\mathcal{A}_i$ at each time instant t is defined as*

$$\Delta_t^i = \max_{Adj(\mathcal{D}_t^i, \mathcal{D}_t^{i'})} \|\mathcal{A}_i(\mathcal{D}_t^i, \theta_t^{-i}) - \mathcal{A}_i(\mathcal{D}_t^{i'}, \theta_t^{-i})\|_1, \tag{16}$$

where $\mathcal{D}_t^i$ represents Learner i's dataset and θ_t^{-i} represents all messages received by Learner i at time instant t.

With the defined sensitivity, we have the following lemma:

Lemma 2. *For any given $T \in \mathbb{N}^+$ or $T = \infty$, if Learner i injects to each of its transmitted messages at each time $t \in \{1, \cdots, T\}$ a noise vector ζ_t^i consisting of n independent Laplace noises with parameter ϱ_t^i, then Learner i's implementation $\mathcal{A}_i$ is ϵ_i locally differentiable private with the cumulative privacy budget from time $t = 1$ to $t = T$ upper bounded by $\sum_{t=1}^{T} \frac{\Delta_t^i}{\varrho_t^i}$.*

Proof. Due to space limitations, we leave the proof to the extended version available at [1]. □

For our privacy analysis, we also utilize the ensuing result:

Lemma 3. *([71]) Let $\{v_t\}$ denote a nonnegative sequence, and $\{\alpha_t\}$ and $\{\beta_t\}$ be positive non-increasing sequences satisfying $\sum_{t=0}^{\infty} \alpha_t = \infty$, $\lim_{t\to\infty} \alpha_t = 0$, and $\lim_{t\to\infty} \frac{\beta_t}{\alpha_t} = 0$. If there exists a $T \geq 0$ such that $v_{t+1} \leq (1-\alpha_t)v_t + \beta_t$ holds for all $t \geq T$, and then we always have $v_t \leq c\frac{\beta_t}{\alpha_t}$ for all t, where c is some positive constant.*

For the convenience of privacy analysis, we represent the different data points between two adjacent datasets $\mathcal{D}_t^i$ and $\mathcal{D}_t^{i'}$ as k-th one, i.e., ξ_k^i in $\mathcal{D}_t^i$ and $\xi_k^{i'}$ in $\mathcal{D}_t^{i'}$, without loss of generality. We further denote θ_t^i and $\theta_t^{i'}$ as the parameters generated by Algorithm 1 based on $\mathcal{D}_t^i$ and $\mathcal{D}_t^{i'}$, respectively. We also use the following assumption, which is standard in existing DP analysis for distributed optimization/learning (see e.g., [35]):

Assumption 5. *For any data ξ and ξ', there exists some constant C such that $sup_{\theta\in\Theta}\|\nabla l(\theta,\xi) - \nabla l(\theta,\xi')\|_2 \leq C$ holds.*

Remark 8. Assumption 5 is standard for privacy analysis [35]. It relaxes the bounded-gradient assumption in [27]–[31], [33] because if one has $\|\nabla l(\theta,\xi)\|_2 \leq C$, then one always has $\|\nabla l(\theta,\xi) - \nabla l(\theta,\xi')\|_2 \leq 2C$. In general, Assumption 5 can be satisfied under our problem setting since the optimization variable is restricted in a compact set Θ. For example, under the commonly used loss function $l(\theta,\xi) = \theta^T Q\theta + \xi^T\theta$ for given data ξ and $Q > 0$, we can easily obtain $\|\nabla l(\theta,\xi) - \nabla l(\theta,\xi')\|_2 \leq \|\xi - \xi'\|_2$ and, hence, the boundedness of gradient differences in Assumption 5. In addition, in many machine learning applications, gradient clipping is used to make the norm of the gradient vector be at most C [72], [73]. In this case, we can easily obtain the upper bound in Assumption 5 by using the inequality $\|\nabla l(\theta,\xi) - \nabla l(\theta,\xi')\|_2 \leq 2\|\nabla l(\theta,\xi)\|_2 \leq 2C$.

Theorem 5. *Under Assumptions 1-5, if nonnegative sequences λ_t and γ_t satisfy the conditions in the statement of Theorem 1, and each element of ζ_t^i independently follows a Laplace distribution $Lap(\varrho_t^i)$ satisfying Assumption 4, then the tracking error of Algorithm 1 will converge in mean square to zero. Furthermore,*

(i) For any finite number of iterations T, under Algorithm 1, Learner i is ensured to be ϵ_i locally differentially private with the cumulative privacy budget bounded by $\sum_{t=1}^{T} \frac{\sqrt{2n}C\tau_t}{\sigma^i(t+1)^{\varsigma^i}}$. Here, C is given in the statement of Assumption 5 and τ_t is given by $\tau_t \triangleq \sum_{p=1}^{t-1}\left(\prod_{q=p}^{t-1}(1 - \bar{w}\gamma_q + \lambda_q L)\right)\lambda_{p-1} + \lambda_{t-1}$.

(ii) The cumulative privacy budget is finite for $T \to \infty$.

Proof. Since the Laplace DP noise satisfies Assumption 4, the tracking result follows naturally from Theorem 1.

(i) To prove the statements on privacy, we first analyze the sensitivity of Learner i under Algorithm 1.

According to the definition of sensitivity in (16), we have $\theta_t^j + \zeta_t^j = \theta_t^{j'} + \zeta_t^{j'}$ for all $t \geq 0$ and $j \in \mathcal{N}_i$. Since we assume that only the k-th data point is different between $\mathcal{D}_t^i$ and $\mathcal{D}_t^{i'}$, when $t < k$, we have $\theta_t^i = \theta_t^{i'}$. However, when $t \geq k$, since the difference in loss functions kicks in at time k, i.e., $l(\theta, \xi_k^i) \neq l(\theta, \xi_k^{i'})$, we have $\theta_t^i \neq \theta_t^{i'}$. Hence, for Learner i's implementation of Algorithm 1, we use the projected inequality to obtain

$$\begin{aligned} &\|\theta_{t+1}^i - (\theta_{t+1}^i)'\|_2 = \|\text{Pro}_\Theta(\hat{\theta}_{t+1}^i) - \text{Pro}_\Theta((\hat{\theta}_{t+1}^i)')\|_2 \\ &\leq \|\hat{\theta}_{t+1}^i - (\hat{\theta}_{t+1}^i)'\|_2 \leq \big\|(1+w_{ii}\gamma_t)(\theta_t^i - \theta_t^{i'}) \\ &\quad - \frac{\lambda_t}{t+1}\sum_{p=k}^{t}(\nabla l(\theta_t^i, \xi_p^i) - \nabla l(\theta_t^{i'}, \xi_p^{i'}))\big\|_2, \end{aligned} \tag{17}$$

for all $t \geq k$ and any $k \geq 0$, where we have used the definition $w_{ii} = -\sum_{j \in \mathcal{N}_i} w_{ij}$. Letting $\bar{w} \triangleq \min\{|w_{ii}|\}$, $i \in [m]$ and $\Phi_t^i \triangleq \|\theta_t^i - \theta_t^{i'}\|_2$, we obtain

$$\begin{aligned} \Phi_{t+1}^i &\leq (1 - \bar{w}\gamma_t)\Phi_t^i + \frac{\lambda_t}{t+1}\sum_{p=k}^{t}\|\nabla l(\theta_t^i, \xi_p^i) - \nabla l(\theta_t^{i'}, \xi_p^{i'})\|_2 \\ &\leq (1 - \bar{w}\gamma_t)\Phi_t^i + \frac{\lambda_t}{t+1}\sum_{p=0}^{t}\|\nabla l(\theta_t^i, \xi_p^i) - \nabla l(\theta_t^{i'}, \xi_p^{i'})\|_2, \end{aligned} \tag{18}$$

where we have used $\sum_{p=0}^{k-1}\nabla l(\theta_t^i, \xi_p^i) = \sum_{p=0}^{k-1}\nabla l(\theta_t^{i'}, \xi_p^{i'})$ in the second inequality. Since $\|\nabla l(\theta_t^i, \xi_p^i) - \nabla l(\theta_t^{i'}, \xi_p^{i'})\|_2 = \|\nabla l(\theta_t^i, \xi_p^i) - \nabla l(\theta_t^i, \xi_p^{i'}) + \nabla l(\theta_t^i, \xi_p^{i'}) - \nabla l(\theta_t^{i'}, \xi_p^{i'})\|_2 \leq C + L\Phi_t^i$ holds, the inequality (18) can be rewritten as

$$\Phi_{t+1}^i \leq (1 - \bar{w}\gamma_t + \lambda_t L)\Phi_t^i + \lambda_t C. \tag{19}$$

By iterating (19) from 0 to t and using the relationship $\|\theta\|_1 \leq \sqrt{n}\|\theta\|_2$ valid for any $\theta \in \mathbb{R}^n$, we obtain

$$\Delta_t^i \leq \sqrt{n}C\left(\sum_{p=1}^{t-1}\left(\prod_{q=p}^{t-1}(1 - \bar{\omega}\gamma_q + \lambda_q L)\right)\lambda_{p-1} + \lambda_{t-1}\right).$$

Therefore, for Learner i, the cumulative privacy budget for any finite T iterations is bounded by

$$\sum_{t=1}^{T}\frac{\Delta_t^i}{\varrho_t^i} \leq \sum_{t=1}^{T}\frac{\sqrt{2n}C\tau_t}{\sigma^i(t+1)^{\varsigma^i}}, \tag{20}$$

where τ_t is defined in the theorem statement.

(ii) Based on (18) and $\xi_p^i = \xi_p^{i'}$ for $p \neq k$, we have

$$\begin{aligned} \Phi_{t+1}^i &\leq (1 - \bar{w}\gamma_t)\Phi_t^i + \frac{\lambda_t}{t+1}\|\nabla l(\theta_t^i, \xi_k^i) - \nabla l(\theta_t^i, \xi_k^{i'})\|_2 \\ &\quad + \frac{\lambda_t}{t+1}\|\nabla l(\theta_t^i, \xi_k^{i'}) - \nabla l(\theta_t^{i'}, \xi_k^{i'})\|_2 \\ &\quad + \frac{\lambda_t}{t+1}\sum_{p=0,\ p\neq k}^{t}\|\nabla l(\theta_t^i, \xi_p^i) - \nabla l(\theta_t^{i'}, \xi_p^i)\|_2, \end{aligned} \tag{21}$$

for all $t \geq k$ and any $k \geq 0$. By using Assumption 3-(iii) and Assumption 5, we can rewrite (21) as follows:

$$\Phi_{t+1}^i \leq \left(1 - \bar{w}\gamma_t + \frac{L\lambda_t(t+1)}{t+1}\right)\Phi_t^i + \frac{\lambda_t C}{t+1}. \tag{22}$$

Recalling the definitions $\gamma_t = \frac{\gamma_0}{(t+1)^u}$ and $\lambda_t = \frac{\lambda_0}{(t+1)^v}$ with $v > u$ from the statement of Theorem 1, there must exist a $T_0 > 0$ and some constant $C_1 > 0$ such that

$$\bar{w}\gamma_t - L\lambda_t = \frac{\bar{w}\gamma_0}{(t+1)^u} - \frac{L\lambda_0}{(t+1)^v} \geq \frac{C_1}{(t+1)^u}, \tag{23}$$

holds for all $t \geq T_0$. Combining (22) and (23) yields $\Phi_{t+1}^i \leq \left(1 - \frac{C_1}{(t+1)^u}\right)\Phi_t^i + \frac{\lambda_0 C}{(t+1)^{1+v}}$ for all $t \geq T_0$. Using Lemma 3 yields for some constant C_2, we have $\Phi_t^i = \|\theta_t^i - \theta_t^{i'}\|_2 \leq C_2\frac{\lambda_0 C}{C_1(t+1)^{1+v-u}}$ for all $t > 0$.

Based on the relationship $\|x\|_1 \leq \sqrt{n}\|x\|_2$ valid for any $x \in \mathbb{R}^n$, we can prove that the sensitivity Δ_t^i satisfies

$$\Delta_t^i \leq \sqrt{n}\Phi_t^i \leq \frac{\sqrt{n}\lambda_0 C C_2}{C_1(t+1)^{1+v-u}}, \tag{24}$$

for all $t > 0$. Recalling the Laplace-noise parameter $\varrho_t^i = \frac{\sigma^i(t+1)^{\varsigma^i}}{\sqrt{2}}$ from the statement of Assumption 4, we have the cumulative privacy budget bounded by

$$\sum_{t=1}^{\infty}\frac{\Delta_t^i}{\varrho_t^i} \leq \sum_{t=1}^{\infty}\frac{\sqrt{2n}\lambda_0 C C_2}{C_1\sigma^i(t+1)^{1+v-u+\varsigma^i}}, \tag{25}$$

according to Lemma 2 when $T \to \infty$. Since $v - u + \varsigma^i > 0$, the cumulative privacy budget is finite when $T \to \infty$. □

Theorem 5-(i) implies that for any given cumulative privacy budget ϵ_i, Learner i's implementation $\mathcal{A}_i$ of Algorithm 1 is ϵ_i locally differentially private when the noise parameter satisfies $\sigma^i = \sum_{t=1}^{T}\frac{\sqrt{2n}C\tau_t}{\epsilon_i(t+1)^{\varsigma^i}}$ with τ_t defined in Theorem 5. Therefore, each learner can choose its desired privacy budget based on its own practical and **personalized** need. This differs from existing centralized DP frameworks used in differentially private distributed optimization/learning approaches [31], [33]–[38], which, in the absence of a data aggregator/curator, require participating learners to trust each other and cooperatively determine the amount of noises needed to achieve a **universal** global privacy budget ϵ.

Theorem 5-(ii) proves that in addition to accurate convergence, Algorithm 1 can ensure a finite cumulative privacy budget even when the number of iterations tends to infinity. The key reason for our approach to achieve rigorous LDP is the judicious design of the decaying factor γ_t, gradient computation strategy, and the stepsize λ_t. These designs can ensure a fast diminishing sensitivity (see Eq. (24)), which, combined with increasing DP-noise variances, ensures a finite cumulative privacy budget even in the infinite time horizon (see Eq. (25)).

Remark 9. Theorem 5 proves that our algorithm can circumvent the tradeoff between privacy and learning accuracy. However, this does not mean that our algorithm achieves privacy protection for free. In fact, resolving the tradeoff between privacy and learning accuracy comes at the expense of sacrificing the convergence rate. Specifically, the rate $\mathcal{O}(t^{-\beta})$

in Theorem 1 is determined by the decaying parameter u of the sequence γ_t, the decaying parameter v of the stepsize sequence λ_t, and the noise parameter $\max_{i\in[m]}\{\varsigma^i\}$. The condition $\max_{i\in[m]}\{\varsigma^i\}<u<v<1$ indicates that an increase in noise parameter ς^i (corresponding to stronger privacy protection) necessitates an increase in decaying parameter v, resulting in a slower convergence rate $\mathcal{O}(t^{-\beta})$ from Theorem 1.

Remark 10. The parameters u, v, and ς^i are crucial for our algorithm's performance. More specifically, according to the convergence results in Theorems 1 through 4, a smaller v, a larger u, and a smaller ς^i lead to a faster convergence rate. Therefore, for applications requiring fast convergence, a small v, a large u, and a small ς^i are preferable. In addition, according to (25) in our privacy analysis, a smaller v, a larger u, and a smaller ς^i result in weaker privacy protection. Hence, for privacy-sensitive applications, a large v, a small u, and a large ς^i are preferable. Therefore, there is a tradeoff between convergence rate and privacy. In applications, we can select these parameters based on practical needs.

Remark 11. Our approach can ensure both DP and **mean square convergence** of the optimization variable to the optimal solution (the variance of the distance between the optimization variable and the optimal solution converges to zero). It is much stronger than [74] that only characterizes the convergence of the **expected** value of the optimization variable to the optimal solution in the presence of DP noises (which cannot exclude the possibility that the optimization error can have an arbitrarily large variance). In addition, [74] only ensures DP of the data (sample) label but does not consider the privacy of the content of data. In contrast, we enable DP for both the label and the content of data.

VII. Numerical Experiments

In this section, we use three numerical experiments to validate our theoretical results. In the first experiment, we consider distributed online training of a logistic regression classifier using the "mushrooms" dataset [75]. In the second experiment, we consider distributed online training of a convolutional neural network (CNN) using the "MNIST" dataset [76]. In the third experiment, we train a CNN distributively using the "CIFAR-10" dataset [77], which is a more diverse and challenging dataset than "MNIST". For each test, we considered heterogeneous data distributions, which are particularly likely in distributed learning where data are collected by multiple learners from multiple sources. In all three experiments, we compared Algorithm 1 with the distributed stochastic gradient descent algorithm (DSGD) in [55], the DP approach for distributed online learning (DOLA) in [45], and the DP approach for distributed optimization (PDOP) in [30]. The convex set was set as $\Theta=\{\theta\in\mathbb{R}^n|\|\theta\|\leq 10^5\}$. We considered five learners connected in a circle, where each learner can only communicate with its two immediate neighbors. For the matrix W, we set $w_{ij}=0.3$ if Learners i and j are neighbors, and $w_{ij}=0$ otherwise.

A. Logistic regression using the "mushrooms" dataset

We first evaluated the effectiveness of Algorithm 1 by using an l_2-logistic regression classification problem on the "mushrooms" dataset [75]. We spread data samples among the learners according to their target values. Specifically, Learners 1, 2, and 3 have samples with the target value of 0, while Learners 4 and 5 have samples with the target value of 1. All learners cooperatively track the optimal parameter θ_t^* to the online optimization problem (2), in which the loss function is given by $l(\theta,\xi^i)=\frac{1}{N_i}\sum_{j=1}^{N_i}(1-b_j^i)(a_j^i)^T\theta-\log(s((a_j^i)^T\theta))+\frac{r_i}{2}\|\theta\|^2$. Here, N_i represents the number of data points per iteration, $r_i>0$ is a regularization parameter proportional to $\frac{1}{N_i}$, $\xi^i=(a_j^i,b_j^i)$ represents the j-th data sample on Learner i, and $s(q)=1/(1+e^{-q})$ is the sigmoid function.

In each iteration, we incorporated Laplace DP noises with parameter $\varrho_t^i=(t+1)^{\varsigma^i}$ to all shared messages, where $\varsigma^i=0.1+0.01i$. Note that the multiplier i in ς^i leads to different noise amplitudes and further different privacy budgets ϵ_i for different learners. We configured the stepsize sequence and diminishing sequence as $\lambda_t=\frac{1}{(t+1)^{0.77}}$ and $\gamma_t=\frac{1}{(t+1)^{0.65}}$, respectively. All configurations satisfy the conditions in Theorems 1-5. The algorithm was implemented for $2,000$ iterations, during which time-varying optimal parameters θ_t^* were calculated using a noise-free, centralized gradient descent algorithm.

In the comparison, we selected the near-optimal stepsize sequences for DSGD, DOLA, and PDOP such that doubling the stepsize results in nonconverging behavior. The resulting average tracking error and average instantaneous regret are shown in Fig. 1-(a) and Fig. 1-(b), respectively. It is clear that the proposed approach has a much better learning accuracy under the constraint of local differential privacy. We also plotted the cumulative privacy budgets of all algorithms in Fig. 1-(c), which shows that our algorithm always has a finite cumulative privacy budget whereas the cumulative privacy budgets for DSGD, PDOP, and DOLA all grow with time to infinity as iteration proceeds, implying diminishing privacy protection as iteration proceeds.

To show that Algorithm 1's achievement of both rigorous LDP and optimization accuracy comes at the expense of sacrificing convergence rate, we compared the number of iterations needed to achieve a certain optimization accuracy under different cumulative privacy budgets. The results, summarized in Table I, clearly show that a smaller cumulative privacy budget (i.e., stronger privacy protection) corresponds to a greater number of iterations (i.e., a slower convergence rate).

B. Neural network training using the "MNIST" dataset

In the second experiment, we assessed Algorithm 1's performance through distributed online training of a convolutional neural network (CNN) using the "MNIST" dataset [76]. We assigned 40% of the data from the i-th class to Learner i, while splitting the remaining 60% evenly among the other learners. The training process spanned 600 iterations.

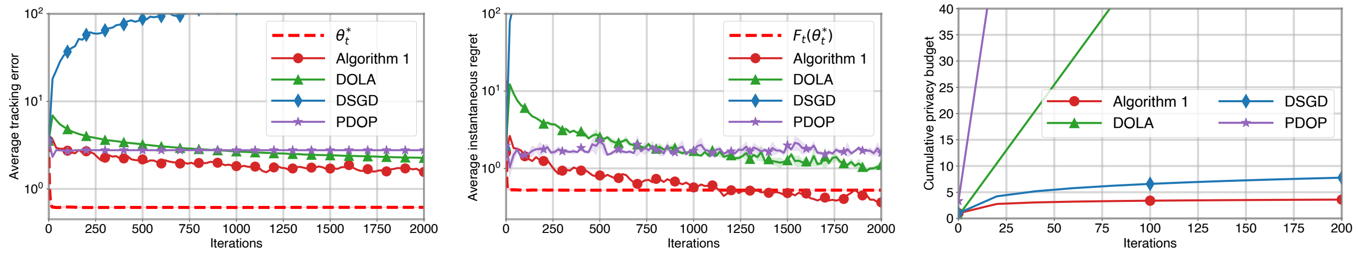
In this experiment, we utilize Laplace DP noises with parameter $\varrho_t^i=\sqrt{2}(t+1)^{\varsigma^i}$ for all shared messages, where $\varsigma^i=0.1+0.02i$. We set the stepsize sequence and decaying sequence to $\lambda_t=\frac{1}{(t+1)^{0.71}}$ and $\gamma_t=\frac{0.01}{(t+1)^{0.7}}$, respectively.

We compared our algorithm with the algorithm DSGD in [55] by training the same CNN, utilizing the same stepsize

TABLE I
THE NUMBER OF ITERATIONS TO ACHIEVE $\|\frac{1}{m}\sum_{i=1}^{m}\theta_t^i-\theta_t^*\|_2\leq 1$ UNDER DIFFERENT CUMULATIVE PRIVACY BUDGETS

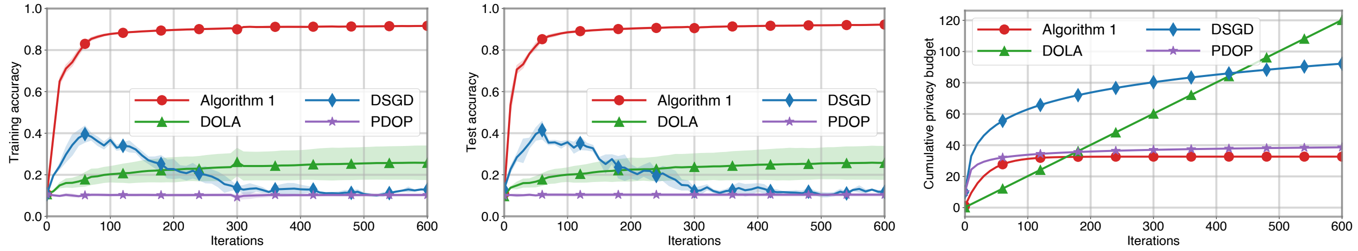
Noise level[a]	×1	×1.5	×2	×2.5	×3	×3.5	×4	×4.5	×5	×5.5	×6
Cumulative privacy budget	23.34	16.59	12.65	11.97	11.54	10.50	9.64	8.79	7.98	7.47	7.03
Iteration number	8	11	12	34	127	269	575	934	1119	2292	4999

[a] Considering the Laplace noise Lap$(0.1(t+1)^{0.1+0.01^i})$ as the base level.



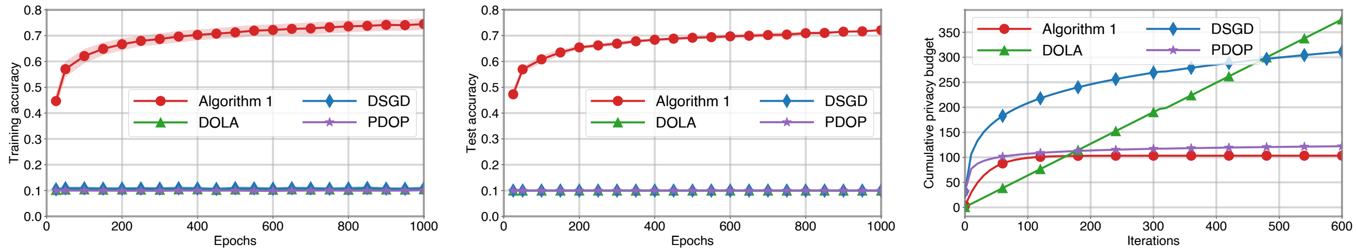
(a) Tracking error (b) Loss (c) Cumulative privacy budget

Fig. 1. Comparison of online logistic regression results by using the "mushrooms" dataset.



(a) Training accuracy (b) Test accuracy (c) Cumulative privacy budget

Fig. 2. Comparison of neural network training results by using the "MNIST" dataset.



(a) Training accuracy (b) Test accuracy (c) Cumulative privacy budget

Fig. 3. Comparison of neural network training results by using the "CIFAR-10" dataset.

sequence and the same Laplace DP noise. Additionally, we implemented existing DP methods, DOLA in [45] and PDOP in [30], using DP noises with decaying and homogeneous parameters $\varrho_t=\frac{0.07}{0.2(t+1)}$ and $\varrho_t=0.5(0.98)^t$ for DOLA and PDOP, respectively). The stepsize sequences for DOLA and PDOP followed their default parameters suggested in [45] and [30], respectively. Fig. 2-(a) and Fig. 2-(b) illustrate the training and test accuracies, respectively.

The results reveal that under the given DP noise, the DSGD algorithm falls short in training the CNN model. Besides, both the DOLA and PDOP algorithms are incapable of effectively training the CNN model (Note that when the test data led to an exploding loss function — all happened when the training accuracy stalled around 0.1 in existing algorithms —, we used the initial parameter for validation, which always gave a test accuracy of 0.1). These results confirm the advantages of our proposed algorithm.

To compare the strength of enabled privacy protection, we ran the DLG attack model proposed in [78], which is a powerful inference algorithm capable of reconstructing raw data from shared gradient/model updates. The training/test accuracies and the DLG attacker's inference errors under different levels of DP noise are summarized in Table II. It can be seen that stronger privacy protection (i.e., a larger DLG inference error) leads to lower training/test accuracies under a fixed number of 3,000 iterations (implying a slower convergence rate).

C. Neural network training using the "CIFAR-10" dataset

In our third experiment, we appraised Algorithm 1's performance via distributed online training of a CNN on the

"CIFAR-10" dataset, which is one of the most widely used datasets for machine learning research (it is also more difficult to train than the "MNIST" dataset). In this experiment, the CNN architecture and all parameter designs are identical to those employed in the previous "MNIST" dataset experiment.

The results in Fig. 3 once again confirms the effectiveness of our distributed online learning algorithm for training the complex CNN model under the constraint of LDP.

TABLE II
TRAINING/TEST ACCURACIES AND DLG ATTACKER'S INFERENCE ERRORS UNDER DIFFERENT LEVELS OF DP NOISE

Noise level[a]	×0.5	×1	×1.5	×2
Training accuracy	0.9402	0.9350	0.8862	0.8180
Test accuracy	0.9449	0.9380	0.8964	0.8259
DLG inference error	0.2696	0.2786	0.2898	0.3110

[a] Considering Laplace noise $\text{Lap}(0.05(t+1)^{0.1+0.01^i})$ as the base level.

VIII. CONCLUSION

In this study, we have introduced a differentially private distributed online learning algorithm that successfully circumvents the tradeoff between privacy and learning accuracy. More specifically, our proposed approach ensures a finite cumulative privacy budget in the infinite time horizon. This is in sharp contrast to existing DP methods for distributed online learning/optimization, which allow the privacy budget to grow to infinity, implying losing privacy protection when time tends to infinity. In addition, our approach also guarantees the convergence of expected instantaneous regret to zero. To the best of our knowledge, our approach is the first to achieve both rigorous local differential privacy and provable convergence in distributed online learning. Our numerical experiments on benchmark datasets confirm the advantages of the proposed approach over existing counterparts.

APPENDIX

A. Technical Lemmas

In this subsection, we introduce two auxiliary lemmas. For the sake of notational simplicity, we add an overbar to a letter to denote the average of all learners, e.g., $\bar{\theta}_t = \frac{1}{m}\sum_{i=1}^{m}\theta_t^i$. We also use bold font to represent the stacked vectors of all learners, e.g., $\boldsymbol{\theta}_t = \text{col}(\theta_t^1, \cdots, \theta_t^m)$. We also denote $\tilde{\boldsymbol{\theta}}_t \triangleq \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^*$, $\check{\boldsymbol{\theta}}_t \triangleq \boldsymbol{\theta}_t - \bar{\boldsymbol{\theta}}_t$, $\boldsymbol{d}_t(\boldsymbol{\theta}_t) \triangleq \text{col}(d_t^1(\theta_t^1), \cdots, d_t^m(\theta_t^m))$, $\zeta_t^{wi} \triangleq \sum_{j\in\mathcal{N}_i} w_{ij}\zeta_t^j$, $\sigma_t^i \triangleq \sigma^i(t+1)^{\varsigma^i}$, and $\sigma^+ \triangleq \max_{i\in[m]}\{\sigma^i\}$.

Lemma 4. *Under the conditions in the statement of Theorem 1, the following inequality always holds:*

$$\begin{aligned}\mathbb{E}\left[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2\right] \leq \left(1+\frac{\mu\lambda_t}{8}\right)\Bigg[\left(1-\frac{\lambda_t\mu}{4}\right)\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_t\|^2] \\ +2\gamma_t\mathbb{E}[\tilde{\boldsymbol{\theta}}_t^T(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t] + 3\gamma_t^2\mathbb{E}[\|(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t\|^2] \\ +\lambda_t\left(\mu+\frac{8L^2}{\mu}\right)\mathbb{E}[\|\check{\boldsymbol{\theta}}_t\|^2] + 3\gamma_t^2\|\boldsymbol{\sigma}_t\|^2 + 3\lambda_t^2\mathbb{E}[\|\boldsymbol{d}_t(\boldsymbol{\theta}_t)\|^2] \\ +\frac{12m\kappa^2\lambda_t}{\mu(t+1)}\Bigg] + \left(1+\frac{8}{\lambda_t\mu}\right)\mathbb{E}\left[\|\boldsymbol{\theta}_{t+1}^* - \boldsymbol{\theta}_t^*\|^2\right]. \qquad (26)\end{aligned}$$

Proof. Due to space limitations, we leave the proof to the extended version available at [1]. □

For the convenience of analysis, we introduce $s \in [0, t]$ and denote $\tilde{\boldsymbol{\theta}}_{t-s} \triangleq \boldsymbol{\theta}_{t-s} - \boldsymbol{\theta}_{t+1}^*$ and $\tilde{\boldsymbol{\theta}}_{t+1-s} \triangleq \boldsymbol{\theta}_{t+1-s} - \boldsymbol{\theta}_{t+1}^*$.

Lemma 5. *Under the conditions in the statement of Theorem 2, the following inequality always holds:*

$$\begin{aligned}&\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1-s}\|^2] \leq (1+\lambda_{t-s}(\eta_s+\eta_{t-s}))\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t-s}\|^2] \\ &-2m\lambda_{t-s}\mathbb{E}[F_{t+1}(\theta_{t-s}^i) - F_{t+1}(\theta_{t+1}^*)] + 3\gamma_{t-s}^2\|\boldsymbol{\sigma}_{t-s}\|^2 \\ &+3\lambda_{t-s}^2\mathbb{E}[\|\boldsymbol{d}_{t-s}(\boldsymbol{\theta}_{t-s})\|^2] + 2\gamma_{t-s}\mathbb{E}[\tilde{\boldsymbol{\theta}}_{t-s}^T(W\otimes I_n)\tilde{\boldsymbol{\theta}}_{t-s}] \\ &+3\gamma_{t-s}^2\mathbb{E}[\|(W\otimes I_n)\tilde{\boldsymbol{\theta}}_{t-s}\|^2] + m\lambda_{t-s}\eta_{t-s} \\ &+\frac{8\lambda_{t-s}(\kappa^2+D^2)(s+1)}{\eta_s(t+2)(t-s+1)} + \frac{4m\kappa^2\lambda_{t-s}}{\eta_{t-s}(t+1)} + \frac{2m\kappa R\lambda_{t-s}}{\sqrt{t+1}} \\ &+\frac{2(L^2+\kappa^2+D^2)\lambda_{t-s}\mathbb{E}[\|\check{\boldsymbol{\theta}}_{t-s}\|^2]}{\eta_{t-s}}, \qquad (27)\end{aligned}$$

where the sequence η_t is given by $\eta_t = \frac{1}{(t+1)^r}$ with $r = \frac{1-v}{2}$.

Proof. Due to space limitations, we leave the proof to the extended version available at [1]. □

B. Proof of Theorem 1

The proof is divided into three steps: in Step 1), we simplify the result in Lemma 4 to obtain (32); in Step 2), we iterate (32) from 0 to t to derive (35); in Step 3), we estimate an upper bound on each term on the right hand side of (35) and obtain (54).

1) To simplify the result in Lemma 4, we first prove that the sum of the following three terms in (26) is negative:

$$\begin{aligned}2\gamma_t\mathbb{E}[\tilde{\boldsymbol{\theta}}_t^T(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t] + 3\gamma_t^2\mathbb{E}[\|(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t\|^2] \\ +\lambda_t\left(\mu+\frac{8L^2}{\mu}\right)\mathbb{E}[\|\check{\boldsymbol{\theta}}_t\|^2] \leq 0.\end{aligned} \qquad (28)$$

Given $\gamma_t \leq \gamma_0 \leq \frac{1}{-3\delta_m}$ in the statement of Theorem 1, we have $\gamma_t\delta_i + 3\gamma_t^2\delta_i^2 \leq 0,\ \forall i \in [m]$, which implies

$$\gamma_t\mathbb{E}[\tilde{\boldsymbol{\theta}}_t^T(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t] + 3\gamma_t^2\mathbb{E}[\|(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t\|^2] \leq 0. \qquad (29)$$

By using the relation $\check{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_t - \boldsymbol{\theta}_t^* - (\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t^*)$, we obtain

$$\check{\boldsymbol{\theta}}_t = \tilde{\boldsymbol{\theta}}_t - \left(\left(\frac{\mathbf{1}_m\mathbf{1}_m^T}{m}\otimes I_n\right)\boldsymbol{\theta}_t - \mathbf{1}_m\left(\frac{\mathbf{1}_m^T\mathbf{1}_m}{m}\right)\otimes\theta_t^*\right).$$

Given $\mathbf{1}_m\left(\frac{\mathbf{1}_m^T\mathbf{1}_m}{m}\right)\otimes\theta_t^* = \left(\frac{\mathbf{1}_m\mathbf{1}_m^T}{m}\otimes I_n\right)(\mathbf{1}_m\otimes\theta_t^*)$, we have

$$\check{\boldsymbol{\theta}}_t = \left(I_{mn} - \left(\frac{\mathbf{1}_m\mathbf{1}_m^T}{m}\otimes I_n\right)\right)\tilde{\boldsymbol{\theta}}_t,$$

which further leads to

$$\tilde{\boldsymbol{\theta}}_t = \check{\boldsymbol{\theta}}_t + \left(\frac{\mathbf{1}_m\mathbf{1}_m^T}{m}\otimes I_n\right)\tilde{\boldsymbol{\theta}}_t.$$

By using $\check{\boldsymbol{\theta}}_t^T(W\otimes I_n)\check{\boldsymbol{\theta}}_t \leq \delta_2\|\check{\boldsymbol{\theta}}_t\|^2$, $\mathbf{1}^TW = \mathbf{0}^T$, and $W\mathbf{1} = \mathbf{0}$, we obtain

$$\gamma_t\tilde{\boldsymbol{\theta}}_t^T(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t = \gamma_t\check{\boldsymbol{\theta}}_t^T(W\otimes I_n)\check{\boldsymbol{\theta}}_t \leq \gamma_t\delta_2\|\check{\boldsymbol{\theta}}_t\|^2. \qquad (30)$$

Noting the relationships $\lambda_t \leq \lambda_0 \leq \frac{-\gamma_0\delta_2\mu}{\mu^2+8L^2}$ and $\gamma_t \leq \gamma_0 \leq \frac{1}{-3\delta_m}$ with $v > u$ from the statement of Theorem 1, we have

$$\gamma_t\mathbb{E}[\tilde{\boldsymbol{\theta}}_t^T(W\otimes I_n)\tilde{\boldsymbol{\theta}}_t] + \lambda_t\left(\mu+\frac{8L^2}{\mu}\right)\mathbb{E}[\|\check{\boldsymbol{\theta}}_t\|^2] \leq 0. \qquad (31)$$

Combining (29) with (31) yields (28).

By using the inequality (31), we can rewrite (26) as follows:

$$\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] \leq \left(1-\frac{\mu\lambda_t}{8}\right)\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_t\|^2]+\Delta_t, \tag{32}$$

where in the derivation we have used the definitions $a_t = 1+\frac{\mu\lambda_t}{8}$ and $b_t = 1+\frac{8}{\lambda_t\mu}$ from the statement of Lemma 4. Moreover, the term Δ_t in (32) is given by

$$\begin{aligned}\Delta_t \triangleq& \left(1+\frac{\mu\lambda_t}{8}\right)\frac{12m\kappa^2\lambda_t}{\mu(t+1)}+\left(1+\frac{8}{\lambda_t\mu}\right)\mathbb{E}[\|\boldsymbol{\theta}_{t+1}^*-\boldsymbol{\theta}_t^*\|^2]\\ &+\left(3+\frac{3\mu\lambda_t}{8}\right)\left(\lambda_t^2\mathbb{E}[\|\boldsymbol{d}_t(\boldsymbol{\theta}_t)\|^2]+\gamma_t^2\|\boldsymbol{\sigma}_t\|^2\right).\end{aligned} \tag{33}$$

2) Iterating (32) from 0 to t, we arrive at

$$\begin{aligned}\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] \leq& \prod_{p=0}^{t}\left(1-\frac{\mu\lambda_p}{8}\right)\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_0\|^2]\\ &+\sum_{p=1}^{t}\prod_{q=p}^{t}\left(1-\frac{\mu\lambda_q}{8}\right)\Delta_{p-1}+\Delta_t.\end{aligned} \tag{34}$$

Since $\ln(1-u) \leq -u$ holds for all $u>0$, and, hence, we have $\prod_{p=0}^{t}\left(1-\frac{\mu\lambda_p}{8}\right) \leq e^{-\frac{1}{8}\mu\sum_{p=0}^{t}\lambda_p}$. Then, the inequality (34) can be rewritten as follows:

$$\begin{aligned}\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] \leq& e^{-\frac{1}{8}\mu\sum_{p=0}^{t}\lambda_p}\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_0\|^2]\\ &+\sum_{p=1}^{t}\Delta_{p-1}e^{-\frac{1}{8}\mu\sum_{q=p}^{t}\lambda_q}+\Delta_t,\end{aligned} \tag{35}$$

where the term Δ_t is given in (33).

3) We proceed to estimate an upper bound on the right hand side of (35).

By using the relationships $e^{-\frac{\mu}{8}\sum_{q=p}^{t}\lambda_q} \leq e^{-\frac{\mu}{8}\sum_{q=\lceil\frac{t}{2}\rceil}^{t}\lambda_q}$ valid for all $p\in[1,\lceil\frac{t}{2}\rceil]$ and $e^{-\frac{\mu}{8}\sum_{q=\lceil\frac{t}{2}\rceil+1}^{t}\lambda_q} < 1$, the last two terms on the right hand side of (35) satisfies

$$\begin{aligned}&\sum_{p=1}^{t}\Delta_{p-1}e^{-\frac{\mu}{8}\sum_{q=p}^{t}\lambda_q}+\Delta_t\\ &\leq\sum_{p=1}^{\lceil\frac{t}{2}\rceil}\Delta_{p-1}e^{-\frac{\mu}{8}\sum_{q=\lceil\frac{t}{2}\rceil}^{t}\lambda_q}+\sum_{p=\lceil\frac{t}{2}\rceil+1}^{t}\Delta_{p-1}+\Delta_t\\ &\leq\sum_{p=0}^{\lfloor\frac{t}{2}\rfloor}\Delta_p e^{-\frac{\mu}{8}\sum_{q=\lceil\frac{t}{2}\rceil}^{t}\lambda_q}+\sum_{p=\lceil\frac{t}{2}\rceil}^{t}\Delta_p.\end{aligned} \tag{36}$$

Next, we estimate an upper bound on Δ_t in (33):

(a) Since $\lambda_t \leq \lambda_0$ always holds, we have

$$\left(1+\frac{\mu\lambda_t}{8}\right)\frac{12m\kappa^2\lambda_t}{\mu(t+1)} \leq c_1\frac{\lambda_t}{t+1}, \tag{37}$$

where c_1 is given by $c_1 = \frac{12m\kappa^2}{\mu}(1+\frac{\mu\lambda_0}{8})$.

(b) By using (8) in Lemma 1, we have

$$\left(1+\frac{8}{\lambda_t\mu}\right)\mathbb{E}\left[\|\boldsymbol{\theta}_{t+1}^*-\boldsymbol{\theta}_t^*\|^2\right] \leq \frac{c_2}{\lambda_t(t+1)^2}, \tag{38}$$

where c_2 is given by $c_2 = \frac{16m(\kappa^2+D^2)(\lambda_0+8)}{\mu}(\frac{2}{\mu^2}+\frac{1}{L^2})$.

(c) Assumption 1-(iii) and Assumption 3-(ii) imply $\mathbb{E}[\|\nabla l(\theta_t^i,\xi^i)\|^2] \leq 2(\kappa^2+D^2)$, which further leads to

$$\left(3+\frac{3\mu\lambda_t}{8}\right)\lambda_t^2\mathbb{E}\left[\|\boldsymbol{d}_t(\boldsymbol{\theta}_t)\|^2\right] \leq c_3\lambda_t^2, \tag{39}$$

where c_3 is given by $c_3 = 6m(1+\frac{\lambda_0\mu}{8})(\kappa^2+D^2)$.

(d) We denote $\varsigma \triangleq \max_{i\in[m]}\{\varsigma^i\}$ and $\sigma^+ \triangleq \max_{i\in[m]}\{\sigma^i\}$. Then, we have

$$\left(3+\frac{3\mu\lambda_t}{8}\right)\gamma_t^2\|\boldsymbol{\sigma}_t\|^2 \leq c_4\gamma_t^2(t+1)^{2\varsigma}, \tag{40}$$

where c_4 is given by $c_4 = m(\sigma^+)^2(3+\frac{3\mu\lambda_0}{8})$.

By substituting (37)-(40) into (33), we obtain

$$\Delta_t \leq \frac{c_1\lambda_t}{t+1}+\frac{c_2}{\lambda_t(t+1)^2}+c_3\lambda_t^2+c_4\gamma_t^2(t+1)^{2\varsigma}. \tag{41}$$

Substituting (41) into the second term on the right hand side of (36), one yields

$$\begin{aligned}\sum_{p=\lceil\frac{t}{2}\rceil}^{t}\Delta_p \leq& \sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}\frac{c_1\lambda_p}{p+1}+\sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}\frac{c_2}{\lambda_p(p+1)^2}\\ &+\sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}c_3\lambda_p^2+\sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}c_4\gamma_p^2(p+1)^{2\varsigma}.\end{aligned} \tag{42}$$

Recalling the definition $\lambda_p = \frac{\lambda_0}{(p+1)^v}$, we have

$$\sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}\frac{c_1\lambda_p}{p+1} \leq c_1\lambda_0\int_{\lceil\frac{t}{2}\rceil}^{\infty}\frac{1}{x^{1+v}}dx \leq \frac{c_1\lambda_0 2^v}{vt^v}. \tag{43}$$

Following an argument similar to that of (43), we can derive that the following inequalities always hold:

$$\begin{cases}\sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}\frac{c_2}{\lambda_p(p+1)^2} \leq \frac{c_2}{\lambda_0}\int_{\lceil\frac{t}{2}\rceil}^{\infty}\frac{1}{x^{2-v}}dx \leq \frac{c_2 2^{1-v}}{(1-v)\lambda_0 t^{1-v}},\\ \sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}c_3\lambda_p^2 \leq c_3\lambda_0^2\int_{\lceil\frac{t}{2}\rceil}^{\infty}\frac{1}{x^{2v}}dx \leq \frac{c_3\lambda_0^2 2^{2v-1}}{(2v-1)t^{2v-1}},\\ \sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}c_4\gamma_p^2(p+1)^{2\varsigma} \leq \frac{c_4\gamma_0^2 2^{2u-2\varsigma-1}}{(2u-2\varsigma-1)t^{2u-2\varsigma-1}},\end{cases} \tag{44}$$

where we have used $\sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}\gamma_p^2(p+1)^{2\varsigma} \leq \gamma_0^2\int_{\lceil\frac{t}{2}\rceil}^{\infty}\frac{1}{x^{2u-2\varsigma}}dx$ and $\max_{i\in[m]}\{\varsigma^i\}+\frac{1}{2} < u$ in the last inequality.

Substituting (43)-(44) into (42), we have

$$\begin{aligned}\sum_{p=\lceil\frac{t}{2}\rceil}^{t}\Delta_p \leq \sum_{p=\lceil\frac{t}{2}\rceil}^{\infty}\Delta_p \leq& \frac{c_1\lambda_0 2^v}{vt^v}+\frac{c_2 2^{1-v}}{(1-v)\lambda_0 t^{1-v}}\\ &+\frac{c_3\lambda_0^2 2^{2v-1}}{(2v-1)t^{2v-1}}+\frac{c_4\gamma_0^2 2^{2u-2\varsigma-1}}{(2u-2\varsigma-1)t^{2u-2\varsigma-1}},\end{aligned} \tag{45}$$

where the positive constants c_1, c_2, c_3, and c_4 are given in (37), (38), (39), and (40), respectively.

Based on the result in (45), we proceed to characterize the first term on the right hand side of (36). To this end, we first characterize the term $e^{-\frac{\mu}{8}\sum_{q=\lceil\frac{t}{2}\rceil}^{t}\lambda_q}$. By using the inequality $(t+1)^v \leq 3(t-1)^v$ valid for all $t\geq 2$ and the fact $t-\lceil\frac{t}{2}\rceil = \lfloor\frac{t}{2}\rfloor$, we can obtain

$$\sum_{q=\lceil \frac{t}{2} \rceil}^{t} \lambda_q \geq \int_{\lceil \frac{t}{2} \rceil}^{t} \frac{\lambda_0}{(x+1)^v} dx = \frac{\lambda_0}{1-v}(x+1)^{1-v}\Big|_{\lceil \frac{t}{2} \rceil}^{t}$$
$$\geq \frac{\lambda_0}{(1-v)^2} \times \xi^{-v} \times \left\lfloor \frac{t}{2} \right\rfloor \geq \frac{\lambda_0 (t-1)^{1-v}}{6(1-v)^2}, \quad (46)$$

with $\xi \in \left(\left\lceil \frac{t}{2} \right\rceil + 1, t+1\right)$. By combining the relations $e^{-x} < \frac{1}{x}$ valid for all $x > 0$, $(t-1)^{v-1} \leq 2^{1-v} t^{v-1}$ valid for all $t \geq 2$, and the inequality (46), we have

$$e^{-\frac{\mu}{8} \sum_{q=\lceil \frac{t}{2} \rceil}^{t} \lambda_q} < \frac{48(1-v)^2 2^{1-v} t^{v-1}}{\mu \lambda_0}, \quad (47)$$

for all $t \geq 2$. Moreover, when we consider the case $t = 1$ (i.e., $q = 1$), the relationship $e^{-\frac{\mu}{8} \sum_{q=1}^{1} \lambda_q} < 1$ always holds. Combining $e^{-\frac{\mu}{8} \sum_{q=1}^{1} \lambda_q} < 1$ and (47), we obtain the following inequality for all $t \geq 1$:

$$e^{-\frac{\mu}{8} \sum_{q=\lceil \frac{t}{2} \rceil}^{t} \lambda_q} < \frac{c_5}{t^{1-v}}, \quad (48)$$

where c_5 is given by $c_5 = \max\{1, \frac{48(1-v)^2 2^{1-v}}{\mu \lambda_0}\}$.

Substituting (48) into the first term on the right hand side of (36) yields

$$\sum_{p=0}^{\lfloor \frac{t}{2} \rfloor} \Delta_p e^{-\frac{\mu}{8} \sum_{q=\lceil \frac{t}{2} \rceil}^{t} \lambda_q} < \left(\Delta_0 + \sum_{p=1}^{\infty} \Delta_p\right) \frac{c_5}{t^{1-v}}. \quad (49)$$

By using the relationship $\Delta_0 \leq c_1 \lambda_0 + \frac{c_2}{\lambda_0} + c_3 \lambda_0^2 + c_4 \gamma_0^2$ derived from (41) and the upper bound obtained in (45), we can rewrite (49) as

$$\sum_{p=0}^{\lfloor \frac{t}{2} \rfloor} \Delta_p e^{-\frac{\mu}{8} \sum_{q=\lceil \frac{t}{2} \rceil}^{t} \lambda_q} < \frac{c_6}{t^{1-v}}, \quad (50)$$

where the positive constant c_6 is given by $c_6 = c_5[c_1 \lambda_0 + \frac{c_2}{\lambda_0} + c_3 \lambda_0^2 + c_4 \gamma_0^2 + \frac{c_1 \lambda_0 2^v}{v} + \frac{c_2 2^{1-v}}{(1-v)\lambda_0} + \frac{c_3 \lambda_0^2 2^{2v-1}}{(2v-1)} + \frac{c_4 \gamma_0^2 2^{2u-2\varsigma-1}}{(2u-2\varsigma-1)}]$.

By substituting (45) and (50) into (36), we have

$$\sum_{p=1}^{t} \Delta_{p-1} e^{-\frac{\mu}{8} \sum_{q=p}^{t} \lambda_q} + \Delta_t$$
$$\leq \frac{c_1 \lambda_0 2^v}{v t^v} + \frac{c_2 2^{1-v}}{(1-v)\lambda_0 t^{1-v}} + \frac{c_3 \lambda_0^2 2^{2v-1}}{(2v-1) t^{2v-1}} \quad (51)$$
$$+ \frac{c_4 \gamma_0^2 2^{2u-2\varsigma-1}}{(2u-2\varsigma-1) t^{2u-2\varsigma-1}} + \frac{c_6}{t^{1-v}}.$$

We further incorporate (51) into (35) to arrive at

$$\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] \leq e^{-\frac{1}{8}\mu \sum_{p=0}^{t} \lambda_p} \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_0\|^2] + \frac{c_1 \lambda_0 2^v}{v t^v} + \frac{c_2 2^{1-v}}{(1-v)\lambda_0 t^{1-v}}$$
$$+ \frac{c_3 \lambda_0^2 2^{2v-1}}{(2v-1) t^{2v-1}} + \frac{c_4 \gamma_0^2 2^{2u-2\varsigma-1}}{(2u-2\varsigma-1) t^{2u-2\varsigma-1}} + \frac{c_6}{t^{1-v}}. \quad (52)$$

Using the relation $(t+1)^v \leq 2t^v$ valid for all $t > 0$, we have

$$\sum_{p=0}^{t} \lambda_p \geq \int_0^t \frac{\lambda_0}{(x+1)^v} dx > \frac{\lambda_0 t}{(1-v)^2 (t+1)^v} \geq \frac{\lambda_0 t^{1-v}}{2(1-v)^2},$$

which further leads to

$$e^{-\frac{\mu}{8} \sum_{p=0}^{t} \lambda_p} < \frac{1}{\frac{\mu}{8} \sum_{p=0}^{t} \lambda_p} < \frac{16(1-v)^2}{\mu \lambda_0 t^{1-v}}. \quad (53)$$

Incorporating (53) into (52), we arrive at

$$\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] \leq \frac{16(1-v)^2 \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_0\|^2]}{\mu \lambda_0 t^{1-v}} + \frac{c_1 \lambda_0 2^v}{v t^v} + \frac{c_2 2^{1-v}}{(1-v)\lambda_0 t^{1-v}}$$
$$+ \frac{c_3 \lambda_0^2 2^{2v-1}}{(2v-1) t^{2v-1}} + \frac{c_4 \gamma_0^2 2^{2u-2\varsigma-1}}{(2u-2\varsigma-1) t^{2u-2\varsigma-1}} + \frac{c_6}{t^{1-v}}, \quad (54)$$

which implies (9) in Theorem 1 since $\min\{1-v, v, 2v-1, 2u-2\varsigma-1\} = \min\{1-v, 2u-2\varsigma-1\}$ always holds.

C. *Proof of Theorem 2*

The proof is divided into three steps: in Step 1), we simplify the result in Lemma 5 to obtain (59); in Step 2), we estimate an upper bound on the item on the right hand side of (59) and obtain (70); in Step 3), we characterize (70) to arrive at (71).

1) Given $\gamma_{t-s} \leq \frac{1}{-3\delta_m}$, $\lambda_{t-s} \leq \frac{-\delta_2 \gamma_0}{2(L^2+\kappa^2+D^2)}$, and $v > \frac{1+2u}{3}$ in the statement of Theorem 2, the sum of the following three terms in (27) is negative, whose proof is similar to that of (28) and thus is omitted here.

$$2\gamma_{t-s} \mathbb{E}[\tilde{\boldsymbol{\theta}}_{t-s}^T (W \otimes I_n) \tilde{\boldsymbol{\theta}}_{t-s}] + 3\gamma_{t-s}^2 \mathbb{E}[\|(W \otimes I_n) \tilde{\boldsymbol{\theta}}_{t-s}\|^2]$$
$$+ \frac{\lambda_{t-s}}{\eta_{t-s}} 2(L^2 + \kappa^2 + D^2) \mathbb{E}[\|\check{\boldsymbol{\theta}}_{t-s}\|^2] \leq 0. \quad (55)$$

Substituting the inequality (55) into (27) and further summing both sides of (27) from $s = 0$ to $s = t$, we obtain

$$\sum_{s=0}^{t} \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1-s}\|] \leq -2m \sum_{s=0}^{t} \lambda_{t-s} \mathbb{E}[F_{t+1}(\theta_{t-s}^i) - F_{t+1}(\theta_{t+1}^*)]$$
$$+ \sum_{s=0}^{t} \left(1 + \frac{\lambda_{t-s}}{(s+1)^r} + \frac{\lambda_{t-s}}{(t-s+1)^r}\right) \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t-s}\|^2]$$
$$+ 3m(\sigma^+)^2 \sum_{s=0}^{t} \gamma_{t-s}^2 (t-s+1)^{2\varsigma} + m \sum_{s=0}^{t} \frac{\lambda_{t-s}}{(t-s+1)^r}$$
$$+ 8m(\kappa^2 + D^2) \sum_{s=0}^{t} \frac{\lambda_{t-s}(s+1)^{r+1}}{(t+2)(t-s+1)}$$
$$+ 4m\kappa^2 \sum_{s=0}^{t} \frac{\lambda_{t-s}(t-s+1)^r}{t+1} + 2m\kappa R \sum_{s=0}^{t} \frac{\lambda_{t-s}}{\sqrt{t+1}}, \quad (56)$$

where in the derivation we have used the definition $\eta_s = \frac{1}{(s+1)^r}$ with $r = \frac{1-v}{2}$ from the statement of Lemma 5.

We characterize the second term on the right hand side of (56). To this end, we make the following decomposition:

$$\sum_{s=0}^{t} \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1-s}\|^2] = \sum_{s=0}^{t} \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t-s}\|^2] + \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] - \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_0\|^2]. \quad (57)$$

By using (57), we have

$$\sum_{s=0}^{t} \left(1 + \frac{\lambda_{t-s}}{(s+1)^r} + \frac{\lambda_{t-s}}{(t+1-s)^r}\right) \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t-s}\|^2]$$
$$- \sum_{s=0}^{t} \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1-s}\|^2] \quad (58)$$
$$\leq \sum_{s=1}^{t} \left(\frac{\lambda_{t-s}}{(s+1)^r} + \frac{\lambda_{t-s}}{(t+1-s)^r}\right) \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1-s}\|^2]$$
$$- \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2] + \left(1 + \frac{\lambda_0}{(t+1)^r} + \lambda_0\right) \mathbb{E}[\|\tilde{\boldsymbol{\theta}}_0\|^2],$$

where we have used the relation $\sum_{s=0}^{t} \eta_{t-s} = \sum_{s=1}^{t} \eta_{t+1-s} + \eta_0$ valid for any sequence $\{\eta_p\}, p \in \mathbb{N}$ in the last inequality.

Furthermore, since θ_t^i is restricted in a compact set Θ, we have $\mathbb{E}[\|\boldsymbol{\theta}_p - \boldsymbol{\theta}_{t+1}^*\|^2] \leq mR^2$ valid for all $p \geq 0$. By using the relationship $\mathbb{E}[F_{t+1}(\theta_{t-s}^i) - F_{t+1}(\theta_{t+1}^*)] \geq \mathbb{E}[F_{t+1}(\theta_{t+1}^i) - F_{t+1}(\theta_{t+1}^*)]$, we can substitute (58) into (56) and omit the negative term $-\mathbb{E}[\|\tilde{\boldsymbol{\theta}}_{t+1}\|^2]$ to obtain

$$2\sum_{s=0}^{t} \lambda_{t-s} \mathbb{E}[F_{t+1}(\theta_{t+1}^i) - F_{t+1}(\theta_{t+1}^*)] \leq \Delta_t, \tag{59}$$

where

$$\begin{aligned}
\Delta_t &= \sum_{s=1}^{t} \left(\frac{\lambda_{t-s}}{(s+1)^r} + \frac{\lambda_{t-s}}{(t+1-s)^r} \right) R^2 \\
&+ \left(1 + \frac{\lambda_0}{(t+1)^r} + \lambda_0 \right) R^2 + 3(\sigma^+)^2 \sum_{s=0}^{t} \gamma_{t-s}^2 (t-s+1)^{2\varsigma} \\
&+ 8(\kappa^2 + D^2) \sum_{s=0}^{t} \frac{\lambda_{t-s}(s+1)^{r+1}}{(t+2)(t-s+1)} + \sum_{s=0}^{t} \frac{\lambda_{t-s}}{(t+1-s)^r} \\
&+ 4\kappa^2 \sum_{s=0}^{t} \frac{\lambda_{t-s}(t-s+1)^r}{t+1} + 2\kappa R \sum_{s=0}^{t} \frac{\lambda_{t-s}}{\sqrt{t+1}}.
\end{aligned} \tag{60}$$

2) Using the relation $\sum_{s=0}^{t} \lambda_{t-s} = \sum_{s=0}^{t} \lambda_s$, the first term on the left hand side of (59) satisfies

$$2\sum_{s=0}^{t} \lambda_{t-s} \geq 2\int_0^{t+1} \frac{\lambda_0}{(x+1)^v} dx = \frac{2\lambda_0((t+2)^{1-v} - 1)}{1-v}. \tag{61}$$

We proceed with the calculation of the upper bound on Δ_t.

(a) The rearrangement inequality states

$$\begin{aligned}
x_1 y_n + \cdots + x_n y_1 &\leq x_1 y_{\sigma(1)} + \cdots + x_n y_{\sigma(n)} \\
&\leq x_1 y_1 + \cdots + x_n y_n,
\end{aligned}$$

for all real numbers satisfying $x_1 \leq \cdots \leq x_n$ and $y_1 \leq \cdots \leq y_n$ and for all permutations $y_{\sigma(1)}, \cdots, y_{\sigma(1)}$ of $y_1, \cdots, y_n$.

Therefore, for two decreasing sequences λ_{s-1} and $\frac{1}{(s+1)^r}$ with $s = 1, \cdots, t$, we have

$$\begin{aligned}
&\sum_{s=1}^{t} \frac{\lambda_{t-s}}{(s+1)^r} = \sum_{s=1}^{t} \frac{\lambda_{s-1}}{(s+1)^r} \leq \sum_{s=1}^{t} \frac{2\lambda_s}{(s+1)^r} \\
&\leq \int_1^{t+1} \frac{2\lambda_0}{x^{r+v}} dx \leq \frac{2\lambda_0}{1-r-v}((t+1)^{1-r-v} - 1).
\end{aligned} \tag{62}$$

(b) Given the fact $\sum_{s=1}^{t} \frac{\lambda_{t-s}}{(t-s+1)^r} = \sum_{s=0}^{t-1} \frac{\lambda_s}{(s+1)^r}$, we obtain

$$\begin{aligned}
&\sum_{s=1}^{t} \frac{\lambda_{t-s}}{(t-s+1)^r} = \sum_{s=0}^{t-1} \frac{\lambda_s}{(s+1)^r} \leq \sum_{s=0}^{t} \frac{\lambda_s}{(s+1)^r} \\
&\leq \lambda_0 + \int_1^{t+1} \frac{\lambda_0}{x^{r+v}} dx \leq \frac{\lambda_0}{1-r-v}(t+1)^{1-r-v}.
\end{aligned} \tag{63}$$

(c) By using Assumption 4 with $u > \varsigma + \frac{1}{2}$, we have

$$\begin{aligned}
&\sum_{s=0}^{t} \gamma_{t-s}^2 (t-s+1)^{2\varsigma} = \sum_{s=0}^{t} \gamma_s^2 (s+1)^{2\varsigma} \\
&\leq \gamma_0^2 + \int_1^{\infty} \frac{\gamma_0^2}{x^{2u-2\varsigma}} dx = \frac{\gamma_0^2(2u-2\varsigma)}{2u-2\varsigma-1}.
\end{aligned} \tag{64}$$

(d) Given $\lambda_t = \frac{\lambda_0}{(t+1)^v}$ with $v \in (\frac{1}{2})$, we have

$$\begin{aligned}
&\sum_{s=0}^{t} \frac{\lambda_{t-s}(s+1)^{r+1}}{(t+2)(t-s+1)} \\
&= \frac{\lambda_0}{t+2} \sum_{s=0}^{t} \frac{(s+1)^{r+1}}{(t-s+1)^{1+v}} \\
&= \frac{\lambda_0}{t+2} \left(\sum_{s=0}^{t-1} \frac{(s+1)^{r+1}}{(t-s+1)^{1+v}} + (t+1)^{r+1} \right),
\end{aligned} \tag{65}$$

which can be further simplified by using the following inequality:

$$\begin{aligned}
&\sum_{s=0}^{t-1} \frac{(s+1)^{r+1}}{(t-s+1)^{1+v}} \leq \int_0^t \frac{(x+1)^{r+1}}{(t-x+1)^{1+v}} dx \\
&= \frac{1}{v} \int_0^t (x+1)^{r+1} d(t-x+1)^{-v} \\
&= \frac{1}{v}((t+1)^{r+1} - (t+1)^{-v}) \\
&\qquad - \frac{1}{v}(1+r) \int_0^t (t-x+1)^{-v}(x+1)^r dx \\
&\leq \frac{1}{v}((t+1)^{r+1} - (t+1)^{-v}).
\end{aligned} \tag{66}$$

We substitute (66) into (65) to obtain

$$\sum_{s=0}^{t} \frac{\lambda_{t-s}(s+1)^{r+1}}{(t+2)(t-s+1)} \leq \lambda_0 \left(\frac{1}{v} + 1 \right) (t+1)^r. \tag{67}$$

(e) Using the relation $\sum_{s=0}^{t} \lambda_{t-s}(t-s+1)^r = \sum_{s=1}^{t+1} \frac{1}{s^{v-r}}$, we have

$$\begin{aligned}
&\sum_{s=0}^{t} \frac{\lambda_{t-s}(t-s+1)^r}{t+1} = \frac{1}{t+1} \sum_{s=1}^{t+1} \frac{1}{s^{v-r}} \\
&\leq \frac{1}{t+1} \int_0^{t+1} \frac{1}{x^{v-r}} dx \leq \frac{1}{(1+r-v)(t+1)^{v-r}}.
\end{aligned} \tag{68}$$

(f) Following an argument similar to that of (68), we have

$$\sum_{s=0}^{t} \frac{\lambda_{t-s}}{\sqrt{t+1}} = \frac{1}{\sqrt{t+1}} \sum_{s=1}^{t+1} \frac{1}{s^v} \leq \frac{1}{(1-v)(t+1)^{v-\frac{1}{2}}}. \tag{69}$$

Incorporating (61)-(69) into (60) and further multiplying both sides of (59) by $\frac{1-v}{4\lambda_0((t+2)^{1-v}-1)}$ yield

$$\begin{aligned}
&\mathbb{E}\left[F_{t+1}(\theta_{t+1}^i) - F_{t+1}(\theta_{t+1}^*)\right] \\
&\leq \frac{c_1(1-v)}{4\lambda_0((t+2)^{1-v} - 1)} + \frac{R^2(1-v)}{4(t+1)^r((t+2)^{1-v} - 1)} \\
&\quad + \frac{(3R^2+1)(1-v)}{4(1-r-v)(t+1)^{r+v-1}((t+2)^{1-v} - 1)} \\
&\quad + \frac{2(\kappa^2 + D^2)(1+v)(1-v)}{v(t+1)^{-r}((t+2)^{1-v} - 1)} \\
&\quad + \frac{\kappa R}{2\lambda_0(t+1)^{v-\frac{1}{2}}((t+2)^{1-v} - 1)} \\
&\quad + \frac{\kappa^2(1-v)}{\lambda_0(1+r-v)(t+1)^{v-r}((t+2)^{1-v} - 1)},
\end{aligned} \tag{70}$$

with $c_1 = (1+\lambda_0)R^2 + \frac{3(\sigma^+)^2\gamma_0^2(2u-2\varsigma)}{2u-2\varsigma-1}$.

3) Using that the relation $(t+2)^{1-v}(t+1)^r \geq 2^{1-v}(t+1)^r$ implies $(t+1)^r \leq \frac{1}{2^{1-v}}(t+2)^{1-v}(t+1)^r$, we obtain

$$(t+1)^r((t+2)^{1-v}-1) \geq \left(1-\frac{1}{2^{1-v}}\right)(t+1)^{1-v+r}.$$

By using a similar argument for each item on the right hand side of (70) and substituting $r=\frac{1-v}{2}$ given in the statement of Lemma 5 into (70), we can arrive at

$$\begin{aligned}&\mathbb{E}\left[F_{t+1}(\theta_{t+1}^i)-F_{t+1}(\theta_{t+1}^*)\right]\\ &\leq \frac{c_2}{(t+1)^{1-v}}+\frac{c_3}{(t+1)^{\frac{3(1-v)}{2}}}+\frac{c_4+c_5}{(t+1)^{\frac{1-v}{2}}}\\ &\quad+\frac{c_6}{(t+1)^{\frac{1}{2}}}+\frac{c_7}{(t+1)^{\frac{1+v}{2}}}=\mathcal{O}((t+1)^{-\beta}),\end{aligned} \tag{71}$$

for any $t \geq 0$, where β is given by $\beta=\frac{1-v}{2}$ and the constants c_2 to c_6 are given by $c_2=\frac{c_1(1-v)2^{1-v}}{4\lambda_0(2^{1-v}-1)}$, $c_3=\frac{R^2(1-v)2^{1-v}}{4(2^{1-v}-1)}$, $c_4=\frac{(3R^2+1)(1-v)2^{1-v}}{4(1-r-v)(2^{1-v}-1)}$, $c_5=\frac{2(\kappa^2+D^2)(1-v^2)2^{1-v}}{v(2^{1-v}-1)}$, $c_6=\frac{\kappa R2^{1-v}}{2\lambda_0(2^{1-v}-1)}$, and $c_7=\frac{\kappa^2(1-v)2^{1-v}}{\lambda_0(1+r-v)(2^{1-v}-1)}$, respectively. The inequality (71) implies (13) in Theorem 2.

References

[1] Z. Chen and Y. Wang, "Locally differentially private distributed online learning with guaranteed optimality," *arXiv preprint arXiv:2306.14094*, 2023.

[2] H. V. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Commun. ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[3] V. Singh and A. Thurman, "How many ways can we define online learning? a systematic literature review of definitions of online learning (1988-2018)," *Am. J. Distance Educ.*, vol. 33, no. 4, pp. 289–306, 2019.

[4] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[5] E. Hazan, "Introduction to online convex optimization," *Found. Trends Mach. Learn.*, vol. 2, no. 3-4, pp. 157–325, 2016.

[6] H. Yu, M. Neely, and X. Wei, "Online convex optimization with stochastic constraints," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1428–1438.

[7] C. Song, T. Ristenpart, and V. Shmatikov, "Machine learning models that remember too much," in *2017 ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 587–601.

[8] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symp. Secur. Privacy*. IEEE, 2017, pp. 3–18.

[9] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 17–31.

[10] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Trans. Autom. Control*, vol. 63, no. 3, pp. 714–725, 2017.

[11] S. Lee, A. Nedić, and M. Raginsky, "Stochastic dual averaging for decentralized online optimization on time-varying communication graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6407–6414, 2017.

[12] K. Lu, G. Jing, and L. Wang, "Online distributed optimization with strongly pseudoconvex-sum cost functions," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 426–433, 2019.

[13] Z. Chen, P. Yi, L. Li, and Y. Hong, "Distributed time-varying convex optimization with dynamic quantization," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 1078–1092, 2021.

[14] X. Yi, X. Li, T. Yang, L. Xie, T. Chai, and H. Karl, "Regret and cumulative constraint violation analysis for distributed online constrained convex optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 5, pp. 2875–2890, 2022.

[15] Y. Ding, W. Ren, and Z. Meng, "Distributed optimal time-varying resource allocation for networked high-order systems," *IEEE Trans. Autom. Control*, 2024.

[16] C. Zhang, M. Ahmad, and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 3, pp. 565–580, 2018.

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

[18] C. N. Hadjicostis and A. D. Domínguez-García, "Privacy-preserving distributed averaging via homomorphically encrypted ratio consensus," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3887–3894, 2020.

[19] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, 2016.

[20] Y. Wang and T. Başar, "Quantization enabled privacy protection in decentralized stochastic optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 7, pp. 4038–4052, 2022.

[21] H. Gao, Y. Wang, and A. Nedić, "Dynamics based privacy preservation in decentralized optimization," *Automatica*, vol. 151, p. 110878, 2023.

[22] C. Altafini, "A system-theoretic framework for privacy preservation in continuous-time multiagent dynamics," *Automatica*, vol. 122, p. 109253, 2020.

[23] G. Ramos, A. P. Aguiarz, S. Karx, and S. Pequito, "Privacy preserving average consensus through network augmentation," *IEEE Trans. Autom. Control (Early Access)*, 2024.

[24] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proc. 42nd ACM Symp. Theory Comput.*, 2010, pp. 715–724.

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

[26] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 50–64, 2016.

[27] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "DP-ADMM: ADMM-based distributed learning with differential privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 15, pp. 1002–1012, 2020.

[28] N. Wu, F. Farokhi, D. Smith, and M. A. Kaafar, "The value of collaboration in convex machine learning with differential privacy," in *2020 IEEE Symp. Secur. Priv.* IEEE, 2020, pp. 304–317.

[29] K. Wei, J. Li, C. Ma, M. Ding, W. Chen, J. Wu, M. Tao, and H. V. Poor, "Personalized federated learning with differential privacy and convergence guarantee," *IEEE Trans. Inf. Forensics Secur.*, vol. 18, pp. 4488–4503, 2023.

[30] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. 16th Int. Conf. Distrib. Comput. Netw.*, no. 4, 2015, pp. 1–10.

[31] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Int. Conf. Mach. Learn.* PMLR, 2018, pp. 5796–5805.

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

[33] X. Chen, L. Huang, L. He, S. Dey, and L. Shi, "A differentially private method for distributed optimization in directed networks via state decomposition," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 4, pp. 2165–2177, 2023.

[34] C. Liu, K. H. Johansson, and Y. Shi, "Distributed empirical risk minimization with differential privacy," *Automatica*, vol. 162, p. 111514, 2024.

[35] L. Huang, J. Wu, D. Shi, S. Dey, and L. Shi, "Differential privacy in distributed optimization with gradient tracking," *IEEE Trans. Autom. Control (Early Access)*, 2024.

[36] Y. Wang and A. Nedić, "Tailoring gradient methods for differentially-private distributed optimization," *IEEE Trans. Autom. Control*, vol. 69, no. 2, pp. 872–887, 2023.

[37] Y. Wang and T. Başar, "Decentralized nonconvex optimization with guaranteed privacy and accuracy," *Automatica*, vol. 150, p. 110858, 2023.

[38] Y. Wang and A. Nedić, "Differentially-private distributed algorithms for aggregative games with guaranteed convergence," *IEEE Trans. Autom. Control (Early Access)*, 2024.

[39] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," in *Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2879–2887.

[40] Z. Chen and Y. Wang, "Locally differentially private decentralized stochastic bilevel optimization with guaranteed convergence accuracy," in *Int. Conf. Mach. Learn.* PMLR, 2024.

[41] J. Wang, J. Ke, and J.-F. Zhang, "Differentially private bipartite consensus over signed networks with time-varying noises," *IEEE Trans. Autom. Control (Early Access)*, 2024.

[42] H. Zheng, H. Hu, and Z. Han, "Preserving user privacy for machine learning: Local differential privacy or federated machine learning?" *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 5–14, 2020.

[43] 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, 2021.

[44] J. Ding, G. Liang, J. Bi, and M. Pan, "Differentially private and communication efficient collaborative learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 8, 2021, pp. 7219–7227.

[45] C. Li, P. Zhou, L. Xiong, Q. Wang, and T. Wang, "Differentially private distributed online learning," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1440–1453, 2018.

[46] J. Zhu, C. Xu, J. Guan, and D. O. Wu, "Differentially private distributed online algorithms over time-varying directed networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 1, pp. 4–17, 2018.

[47] Y. Xiong, J. Xu, K. You, J. Liu, and L. Wu, "Privacy-preserving distributed online optimization over unbalanced digraphs via subgradient rescaling," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 3, pp. 1366–1378, 2020.

[48] Q. Lü, X. Liao, T. Xiang, H. Li, and T. Huang, "Privacy masking stochastic subgradient-push algorithm for distributed online optimization," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3224–3237, 2020.

[49] D. Han, K. Liu, Y. Lin, and Y. Xia, "Differentially private distributed online learning over time-varying digraphs via dual averaging," *Int. J. Robust Nonlinear Control*, vol. 32, no. 5, pp. 2485–2499, 2022.

[50] H. Cheng, X. Liao, and H. Li, "Distributed online private learning of convex nondecomposable objectives," *IEEE Trans. Netw. Sci. Eng.*, no. 2, pp. 1716–1728, 2023.

[51] M. Yuan, J. Lei, and Y. Hong, "Differentially private distributed online mirror descent algorithm," *Neurocomputing*, vol. 551, p. 126531, 2023.

[52] Q. Lü, K. Zhang, S. Deng, Y. Li, H. Li, S. Gao, and Y. Chen, "Privacy-preserving decentralized dual averaging for online optimization over directed networks," *IEEE Trans. Ind. Cyber-Phys. Syst.*, vol. 1, pp. 79–91, 2023.

[53] Z. Zhao, J. Yang, W. Gao, Y. Wang, and M. Wei, "Differentially private distributed online optimization via push-sum one-point bandit dual averaging," *Neurocomputing*, vol. 572, p. 127184, 2024.

[54] J. Wang and J.-F. Zhang, "Differentially private distributed stochastic optimization with time-varying sample sizes," *IEEE Trans. Autom. Control (Early Access)*, 2024.

[55] S. Pu, A. Olshevsky, and I. C. Paschalidis, "A sharp estimate on the transient time of distributed stochastic gradient descent," *IEEE Trans. Autom. Control*, vol. 67, no. 11, pp. 5900–5915, 2021.

[56] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[57] Z. Chen and Y. Wang, "Locally differentially private gradient tracking for distributed online learning over directed graphs," *arXiv preprint arXiv:2310.16105*, 2023.

[58] A. Shapiro, D. Dentcheva, and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*. Soc. Ind. Appl. Math., Philadelphia, PA, USA: SIAM, 2014.

[59] A. Simonetto, "Dual prediction–correction methods for linearly constrained time-varying convex programs," *IEEE Trans. Autom. Control*, vol. 64, no. 8, pp. 3355–3361, 2019.

[60] A. I. Maass, C. Manzie, D. Nesic, J. H. Manton, and I. Shames, "Tracking and regret bounds for online zeroth-order euclidean and riemannian optimization," *SIAM J. Optim.*, vol. 32, no. 2, pp. 445–469, 2022.

[61] S. M. Fosson, "Centralized and distributed online learning for sparse time-varying optimization," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2542–2557, 2021.

[62] R. H. L. Sim, Y. Zhang, B. K. H. Low, and P. Jaillet, "Collaborative Bayesian optimization with fair regret," in *Int. Conf. Mach. Learn.* PMLR, 2021, pp. 9691–9701.

[63] B. Haydon, K. D. Mishra, P. Keyantuo, D. Panagou, F. Chow, S. Moura, and C. Vermillion, "Dynamic coverage meets regret: Unifying two control performance measures for mobile agents in spatiotemporally varying environments," in *2021 60th IEEE Conf. Decis. Control*, 2021, pp. 521–526.

[64] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[65] H. B. McMahan, D. Ramage, K. Talwar, and L. Zhang, "Learning differentially private recurrent language models," in *Int. Conf. Learn. Represent.*, 2018.

[66] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symp. Secur. Priv.*, 2019, pp. 691–706.

[67] F. Gao, X. Song, L. Jian, and X. Liang, "Toward budgeted online kernel ridge regression on streaming data," *IEEE Access*, vol. 7, pp. 2169–3536, 2019.

[68] J. Llavona, *Approximation of Continuously Differentiable Functions*, Math. Stud., North Holland, Amsterdam, Netherlands, 1986.

[69] Z. Pan, Z. Gu, X. Jiang, G. Zhu, and D. Ma, "A modular approximation methodology for efficient fixed-point hardware implementation of the sigmoid function," *IEEE Trans. Ind. Electron.*, vol. 69, no. 10, pp. 10 694–10 703, 2022.

[70] Z. Leng, M. Tan, C. Liu, E. D. Cubuk, J. Shi, S. Cheng, and D. Anguelov, "Polyloss: A polynomial expansion perspective of classification loss functions," in *Int. Conf. Learn. Represent.*, 2022.

[71] K. L. Chung, "On a stochastic approximation method," *Ann. Math. Stat.*, vol. 25, no. 3, pp. 463–483, 1954.

[72] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar, "Can gradient clipping mitigate label noise?" in *Int. Conf. Learn. Represent.*, 2019.

[73] X. Chen, S. Z. Wu, and M. Hong, "Understanding gradient clipping in private sgd: A geometric perspective," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 13 773–13 782.

[74] Y. Liu, J. Liu, and T. Basar, "Differentially private gossip gradient descent," in *2018 IEEE Conf. Decis. Control*. IEEE, 2018, pp. 2777–2782.

[75] D. Dua, C. Graff *et al.*, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2007. [Online]. Available: http://archive.ics.uci.edu/ml

[76] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11. IEEE, 1998, pp. 2278–2324.

[77] A. Krizhevsky, G. Hinton *et al.*, *Learning Multiple Layers of Features from Tiny Images*, Master's thesis, Univ. Toronto, Canada, 2009.

[78] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 14 774–14 784.

Ziqin Chen received the Ph.D. degree in automation from the University of Science and Technology of China, Hefei, China, in 2020. She is currently a postdoctoral associate at the Department of Electrical Computer and Engineering, Clemson University, USA. She was a postdoctoral fellow at Tongji University, China, from 2020 to 2022. Her current research interests include differential privacy, distributed optimization/learning, and game theory.

Yongqiang Wang (Senior Member, IEEE) was born in Shandong, China. He received the dual B.S. degrees in electrical engineering and automation and computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2004, and the M.Sc. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2009. From 2007 to 2008, he was with the University of Duisburg-Essen, Duisburg, Germany, as a Visiting Student. He was a Project Scientist with the University of California, Santa Barbara, CA, USA before joining Clemson University, SC, USA, where he is currently an Associate Professor. His current research interests include decentralized control, optimization, and learning, with an emphasis on privacy and security.

Prof. Wang currently serves as an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL and IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS.