Decentralized Nonconvex Optimization with Guaranteed Privacy and Accuracy ⋆

Yongqiang Wang [a], Tamer Başar [b]

[a] *Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA*

[b] *Coordinated Science Lab, University of Illinois Urbana-Champaign, Urbana, IL 61801, USA*

Abstract

Privacy protection and nonconvexity are two challenging problems in decentralized optimization and learning involving sensitive data. Despite some recent advances addressing each of the two problems separately, no results have been reported that have theoretical guarantees on both privacy protection and saddle/maximum avoidance in decentralized nonconvex optimization. We propose a new algorithm for decentralized nonconvex optimization that can enable both rigorous differential privacy and saddle/maximum avoiding performance. The new algorithm allows the incorporation of persistent additive noise to enable rigorous differential privacy for data samples, gradients, and intermediate optimization variables without losing provable convergence, and thus circumventing the dilemma of trading accuracy for privacy in differential privacy design. More interestingly, the algorithm is theoretically proven to be able to efficiently guarantee accuracy by avoiding convergence to local maxima and saddle points, which has not been reported before in the literature on decentralized nonconvex optimization. The algorithm is efficient in both communication (it only shares one variable in each iteration) and computation (it is encryption-free), and hence is promising for large-scale nonconvex optimization and learning involving high-dimensional optimization parameters. Numerical experiments for both a decentralized estimation problem and an Independent Component Analysis (ICA) problem confirm the effectiveness of the proposed approach.

Key words: Nonconvex optimization; Distributed optimization; Privacy; Saddle avoidance.

1 Introduction

Decentralized optimization is gaining increased traction across disciplines due to its fundamental role in cooperative control [52], distributed sensing [3], multi-agent systems [34], sensor networks [55], and large-scale machine learning [42]. In many of these applications, the problem can be formulated in the following general form, in which a network of m agents cooperatively solve a common optimization problem through on-node computation and local communication:

$$\min_{\theta\in\mathbb{R}^d} F(\theta) \triangleq \frac{1}{m}\sum_{i=1}^{m} f_i(\theta) \tag{1}$$

where the optimization variable θ is common to all agents but each $f_i(\theta) : \mathbb{R}^d \to \mathbb{R}$ is a local objective function private to agent i.

Plenty of results have been reported to solve the above decentralized optimization problem since the seminal work of [43], with some of the popular approaches including gradient-descent (e.g., [29,33,46]), distributed alternating direction method of multipliers (e.g., [36,54]), and distributed Newton methods (e.g., [50]). Results have also emerged incorporating various communication and computation constraints in decentralized optimization (e.g., [27,6,7]). Most of the reported results focus on convex objective functions, whereas results are relatively sparse for nonconvex objective functions. However, in many practical applications, the objective functions are essentially nonconvex. For example, in the resource allocation problem of communication net-

⋆ The work of Yongqiang Wang was supported in part by the National Science Foundation under Grants ECCS-1912702, CCF-2106293, CCF-2215088, and CNS-2219487. The work of Tamer Başar was supported in part by the ONR MURI Grant N00014-16-1- 2710 and in part by the Army Research Laboratory, under Cooperative Agreement W911NF-17-2-0196. This paper was not presented at any IFAC meeting. Corresponding author Y. Q. Wang. Tel. +1-864.656.5923.

Email addresses: yongqiw@clemson.edu (Yongqiang Wang), basar1@illinois.edu (Tamer Başar).

works, the utility functions cannot be modeled by convex/concave functions when the communication traffic is non-elastic [44]; in most machine learning applications, the objective functions are essentially nonconvex due to the presence of multi-layer neural networks [42]; in policy optimization for linear-quadratic regulator [14] as well as for robust and risk-sensitive control [56], nonconvex optimization naturally arises. Therefore, it is imperative to study decentralized optimization under nonconvex objective functions.

In recent years, results have emerged on decentralized nonconvex optimization [4,11,41,45,53], which address the convergence of participating agents' optimization variables to a first-order stationary point of the global objective function. Nevertheless, these results do not address the avoidance of saddle points (stationary points that are not local extrema), which is a major concern in many nonconvex optimization problems [16]. For example, in machine learning applications, the main bottleneck in parameter optimization is not due to the existence of multiple local minima, but the existence of many saddle points which trap gradient updates [16]. To escape from saddle points, classical approaches resort to second-order information, in particular the Hessian matrix of second derivatives (see, e.g., [31,8]). The Hessian matrix based approach, however, incurs a high cost in both computation and storage in every iteration since the Hessian matrix scales quadratically with the dimension of optimization variables, which can be hundreds of millions in modern deep learning applications [40]. Recently, first-order gradient methods have been shown to be able to escape saddle points with the help of random perturbations in centralized optimization (see, e.g., [16,23]). However, it is unclear if this is still true in decentralized nonconvex optimization since the decentralized architecture brings in fundamental differences in optimization dynamics. For example, the saddle points of individual objective functions $f_i(\cdot)$ in decentralized optimization are different from those of the aggregated objective function $F(\cdot)$, which is the only function that needs to be considered in centralized optimization. Furthermore, the inter-agent coupling also complicates the optimization dynamics. Note that random initialization has been shown to be able to asymptotically avoid saddles in centralized nonconvex optimization [25], which is further extended to the decentralized case in [9]. However, the result in [12] shows that this approach to avoiding saddles may take exponentially long time, rendering it impractical.

In this paper, we propose an approach to avoiding maxima/saddles in decentralized nonconvex optimization by leveraging differential privacy design. More specifically, we propose a new algorithm for first-order decentralized optimization that enables exploiting differential-privacy noise to achieve guaranteed saddle-avoiding performance without losing provable convergence. This is significant because differential-privacy noise is generally known to sacrifice provable convergence while enabling privacy protection [2]. Moreover, we rigorously establish that the differential-privacy noise can prevent the algorithm from converging to undesired stationary points such as saddle points within polylogarithmic time, and hence can enhance optimization accuracy. This extends recent results on saddle avoidance in centralized nonconvex optimization [23], and to our knowledge, has not been reported for decentralized optimization with guaranteed convergence. It is worth noting that although diminishing noises, known as annealing, has been shown to be able to facilitate global convergence in distributed nonconvex optimization [38,39], such an approach may not be efficient as it could result in convergence time increasing exponentially with the dimension of optimization variables [15].

The considered privacy protection aspect is becoming an increasingly pressing need in decentralized optimization involving sensitive data. For example, in sensor network based localization, the positions of sensor agents should be kept private in sensitive or hostile environments [54,20]. This requires that decentralized-optimization based localization approaches protect the privacy of individual agents' gradients, which are linear functions of sensor positions and whose disclosure will directly reveal a sensor's position [54]. Another example underscoring the importance of privacy preservation in decentralized optimization is machine learning, where the data involved may contain sensitive information such as medical records or salary information [51]. In fact, recent results in [57] (as well as our own result [47,49]) show that without a strong privacy mechanism in place, an adversary can precisely recover the raw data used for training through shared gradients (pixel-wise accurate for images and token-wise matching for texts).

The main contributions of this paper are as follows: 1) We propose a new algorithm for decentralized optimization that enables the achievement of differential privacy without losing provable convergence. This is significant since in general differential privacy has to trade algorithmic accuracy for privacy; 2) We rigorously establish that the proposed algorithm can guarantee accuracy by efficiently avoiding saddle points in decentralized nonconvex optimization under a diminishing stepsize, which, to our knowledge, has not been reported before. We would like to emphasize that allowing the stepsize to diminish with time is crucial to ensure provable convergence under persistent differential-privacy noise; 3) We prove that the proposed approach can enable rigorous differential privacy for individual agents' data samples and shared gradients under guaranteed convergence. Moreover, the proposed approach also provides differential privacy for participating agents' optimization variables. However, different from the differential-privacy protection on data samples and gradients, the differential-privacy protection on optimization variables decreases with time and reduces to zero when the algo-

rithm converges. Note that this is completely acceptable because the objective of decentralized optimization is to let individual agents learn the same optimal optimization variable.

The organization of the paper is as follows. Sec. II provides formulation of the problem. Sec. III presents the proposed algorithm and Sec. IV provides a rigorous analysis on convergence, including the guaranteed performance in avoiding maxima and saddle points in decentralized nonconvex optimization. Sec. V discusses the privacy preservation performance of the algorithm. Sec. VI presents numerical experimental results in both a decentralized estimation application and an Independent Component Analysis (ICA) application. Finally, Sec. VII concludes the paper.

Notations: $\mathbb{R}^m$ denotes the Euclidean space of dimension m. I_d denotes the identity matrix of dimension d, and $\mathbf{1}_d$ denotes a d dimensional column vector with all entries equal to 1; in both cases we suppress the dimension when clear from the context. A vector is viewed as a column vector. For a vector x, x_i denotes its ith element. A^T denotes the transpose of matrix A. $\|x\|$ denotes the standard Euclidean norm $\|x\| = \sqrt{\sum_{i=1}^d (x_i)^2}$ and $\|x\|_1$ denotes the Taxicab norm $\|x\|_1 = \sum_{i=1}^d |x_i|$. A matrix is column-stochastic when its entries are nonnegative and elements in every column add up to one. A square matrix A is said to be doubly-stochastic when both A and A^T are column-stochastic.

2 Problem Formulation

2.1 Objective functions

In decentralized optimization, agent i's objective function $f_i(\cdot)$ is determined by its loss function and locally accessible data samples. Therefore, we consider $f_i(\cdot)$ of the following form

$$f_i(\theta) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell_i(\theta, s_{i,j}) \tag{2}$$

where $\ell_i(\cdot,\cdot)$ denotes the cost function of agent i, n_i denotes the number of data samples available to agent i, and $s_{i,j}$ represents the jth data sample of agent i. We represent the set of all data samples available to agent i as $\mathbb{D}_i$.

We make the following assumption on cost functions:

Assumption 1 *Every $\ell_i(\cdot,\cdot)$ satisfies $\lim_{\|u\|\to\infty} \ell_i(u,\cdot) \to \infty$ and has Lipschitz gradient and Lipschitz Hessian over $\mathbb{R}^d$, i.e., for some $\nu > 0$ and $\rho > 0$,*

$$\begin{aligned}
\|\nabla \ell_i(u,\cdot) - \nabla \ell_i(v,\cdot)\| &\le \nu \|u - v\|, \quad \forall u, v \in \mathbb{R}^d. \\
\|\nabla \ell_i(\cdot, s_p) - \nabla \ell_i(\cdot, s_q)\|_1 &\le \nu \|s_p - s_q\|_1, \quad \forall s_p, s_q \in \mathbb{D}_i. \\
\|\nabla^2 \ell_i(u,\cdot) - \nabla^2 \ell_i(v,\cdot)\| &\le \rho \|u - v\|, \quad \forall u, v \in \mathbb{R}^d.
\end{aligned}$$

always hold for all i, where $\nabla^2 \ell_i(\cdot)$ denotes the Hessian matrix of $\ell_i(\cdot,\cdot)$ with respect to the first argument.

From the definition of $f_i(\cdot)$ in (2), it can be easily verified that $f_i(\cdot)$ always has Lipschtiz gradient and Hessian, i.e.,

$$\begin{aligned}
\|\nabla f_i(u,\cdot) - \nabla f_i(v,\cdot)\| &\le \nu \|u - v\|, \quad \forall u, v \in \mathbb{R}^d. \\
\|\nabla^2 f_i(u,\cdot) - \nabla^2 f_i(v,\cdot)\| &\le \rho \|u - v\|, \quad \forall u, v \in \mathbb{R}^d.
\end{aligned}$$

The coercivity assumption $\lim_{\|u\|\to\infty} \ell_i(u,\cdot) \to \infty$ is used here because we need the stochastic approximation theory in [32] to prove the avoidance of local maxima. It is also recently used in [41] to analyze the push-sum based distributed optimization under the assumption of no saddle points. The Lipschitz gradient and Hessian condition is a standard assumption in saddle-avoidance studies [16,23,10].

We also assume that the gradient is bounded, which is commonly used in (distributed) nonconvex optimization [10,28,22,24] and differential-privacy analysis [20,1]:

Assumption 2 *For every i, we always have $\|\nabla \ell_i(\cdot,\cdot)\| \le G$ for some positive constant $G < \infty$, which further implies $\|\nabla f_i(\cdot)\| \le G$ according to the definition of $f_i(\cdot)$ in (2).*

Remark 1 *Note that the Lipschitz function assumption in [22] implies bounded gradients.*

For a twice differentiable aggregated objective function $F(\cdot)$, we call θ a stationary point if $\nabla F(\theta) = 0$ holds. A stationary point θ can be viewed as belonging to one of three categories:

- local minimum: there exists a $\gamma > 0$ such that $F(\theta) \le F(\vartheta)$ for any $\|\vartheta - \theta\| \le \gamma$;
- local maximum: there exists a $\gamma > 0$ such that $F(\theta) \ge F(\vartheta)$ for any $\|\vartheta - \theta\| \le \gamma$;
- saddle point: neither of the above two cases is true, i.e., for any $\gamma > 0$, there exist ϑ_1 and ϑ_2 satisfying $\|\vartheta_1 - \theta\| \le \gamma$ and $\|\vartheta_2 - \theta\| \le \gamma$ such that $F(\vartheta_1) < F(\theta) < f(\vartheta_2)$.

Since distinguishing saddle points from local minima for smooth functions is NP-hard in general [30], we focus on a subclass of saddle points, i.e., strict saddle points:

Assumption 3 *All saddle points θ of the aggregated function $F(\cdot)$ are strict saddles, i.e., the minimum (resp.*

maximum) eigenvalue of the Hessian matrix $\nabla^2 F(\theta)$ at any saddle θ is negative (resp. positive).

A generic saddle point must satisfy that the minimum (resp. maximum) eigenvalue of its Hessian matrix is non-positive (resp. non-negative). Our assumption of strict saddles rules out the case where the minimum or maximum eigenvalue of the Hessian matrix is zero. A line of recent work in the machine learning literature shows that for many popular models in machine learning, all saddle points are indeed strict saddle points, with examples ranging from tensor decomposition [16], dictionary learning [37], smooth semidefinite programs [5], to robust principal component analysis [17].

Recently, [16] and [23] have shown that in centralized nonconvex optimization, saddle points could be avoided efficiently (in a polylogarithmic number of iterations) by adding perturbations in the classical single-variable gradient descent algorithm. In this paper, we extend this result to the decentralized case and prove that the added differential-privacy noise can be leveraged to avoid saddles without sacrificing provable convergence. It is worth noting that the extension from centralized optimization to the decentralized case is highly nontrivial because the saddle points of the aggregated function $F(\theta)$ are different from those of individual objective functions $f_i(\theta)$. Furthermore, in decentralized optimization, the interaction between agents brings in an additional element that affects the evolution of dynamics around saddle points, which makes state evolution analysis around saddle points more involved compared with the centralized optimization case.

2.2 Interaction topology

We consider a network of m agents. The agents interact on an undirected graph, which can be described by a weight matrix $W = \{w_{ij}\}$. More specifically, if agents i and j can interact with each other, then w_{ij} is positive. Otherwise, w_{ij} will be zero. We assume that an agent is always able to affect itself, i.e., $w_{ii} > 0$ for all $1 \leq i \leq m$. The neighbor set $\mathbb{N}_i$ of agent i is defined as the set of agents $\{j | w_{ij} > 0\}$. So the neighbor set of agent i always includes itself. To ensure that the agents can cooperatively solve the decentralized optimization problem (1), we make the following standard assumption on the interaction topology:

Assumption 4 *$W = \{w_{ij}\} \in \mathbb{R}^{m\times m}$ is symmetric and satisfies $\mathbf{1}^T W = \mathbf{1}^T$, $W\mathbf{1} = \mathbf{1}$, and $\eta = \|W - \frac{\mathbf{1}\mathbf{1}^T}{m}\| < 1$.*

The optimization problem (1) can now be reformulated as the following equivalent multi-agent optimization problem:

$$\min_{x\in\mathbb{R}^{md}} f(x) \triangleq \frac{1}{m}\sum_{i=1}^{m} f_i(x_i) \text{ s.t. } x_1 = x_2 = \cdots = x_m \quad (3)$$

where $x_i \in \mathbb{R}^d$ is the local estimate of agent i about the optimization solution and $x = [x_1^T, x_2^T, \cdots, x_m^T]^T \in \mathbb{R}^{md}$ is the collection of the estimates made by the agents.

2.3 Privacy preservation in decentralized optimization

In decentralized optimization, the sensitive information that has to be protected from disclosure could be the data samples (raw data), gradients, or optimization variables. Although data samples are not directly shared in decentralized optimization, their information are abstracted and embedded in gradients. For example, in decentralized-optimization based rendezvous and localization, disclosing the gradient of an agent amounts to disclosing its (initial) position [54,20,48], which directly correlates with sampled range/angle measurements. In machine learning, it has been shown that shared gradients can be used by an adversary to reversely recover the raw data used for training (pixel-wise accurate for images and token-wise matching for texts) [57,47,49]. The optimization variables (models in machine learning) could also carry sensitive information abstracted from raw data. However, note that the objective of decentralized optimization is for individual agents to learn the same optimal optimization variable (model), and hence the final consensual optimization variable should be disclosed to all agents, and not be a target of privacy protection. Therefore, in this paper, we restrict the privacy to individual agents' data samples and gradients, and individual agents' intermediate optimization variables (by "intermediate," we mean the evolution of optimization variables before achieving consensus among agents).

We consider two potential attacks in decentralized optimization, which are the two most commonly used models of attacks in privacy research [18]:

- *Honest-but-curious attacks* are attacks in which a participating agent or multiple participating agents (colluding or not) follow all protocol steps correctly but are curious and collect all received intermediate data to learn the sensitive information about other participating agents.
- *Eavesdropping attacks* are attacks in which an external eavesdropper eavesdrops upon all communication channels to intercept exchanged messages so as to learn sensitive information about the sending agents.

An honest-but-curious adversary (e.g., agent i) has access to the internal state x_i, which is unavailable to external eavesdroppers. However, an eavesdropper has access to all shared information in the network, whereas an honest-but-curious agent only has access to shared information that is destined to it.

In this paper, the proposed new decentralized optimization algorithm enables us to leverage differential-privacy

noises to facilitate the avoidance of maxima and saddle points in nonconvex optimization. We adopt the popular definition of (ϵ, δ)-differential privacy following standard conventions [13]:

Definition 1 *A randomized function $h(x)$ is (ϵ, δ)-differentially private if for all $S \subset \text{Range}(h)$ and for all x, y with $\|x - y\|_1 \leq 1$, we have*

$$\text{Prob}(h(x) \in S) \leq e^{\epsilon}\text{Prob}(h(y) \in S) + \delta$$

where Range(h) *denotes the image (the set of all output values) of the function h and* Prob$(\cdot)$ *denotes probability.*

Note that ϵ and δ are always non-negative, and a smaller ϵ (or δ) corresponds to a stronger privacy protection.

3 The proposed decentralized optimization algorithm

Conventional single-variable decentralized optimization algorithms usually take the following form [29]:

$$x_i^{k+1} = \sum\nolimits_{j\in\mathbb{N}_i} w_{ij}x_j^k + \lambda^k g_i^k \tag{4}$$

where x_i^k denotes the local copy of optimization variable of agent i at iteration k, λ^k is a positive scalar denoting the stepsize, and g_i^k denotes the gradient of agent i evaluated at x_i^k, i.e., $g_i^k = \nabla f_i(x_i^k)$. It is well-known that under Assumption 1 and Assumption 4, when λ^k is such that $\sum_{k=0}^{\infty}\lambda^k = \infty$ and $\sum_{k=0}^{\infty}(\lambda^k)^2 < \infty$, then all x_i^k will converge to the same optimal solution when $f(\cdot)$ is convex.

However, in the above decentralized optimization algorithm, agent i has to share x_i^k with all its neighbors $j \in \mathbb{N}_i$, which breaches the privacy of optimization variable x_i^k. Furthermore, if an adversary has access to the optimization variable x_i^k of agent i and the updates that agent i receives from all its neighbors x_j^k for $j \in \mathbb{N}_i$, then the adversary can easily infer g_i^k based on the update rule (4) and publicly known W and λ^k. To protect the privacy of individual agents' optimization variable x_i^k, existing decentralized optimization approaches usually choose to inject additive noise on shared x_i^k (see, e.g., [20]), which, however, will compromise the accuracy of the final optimization result.

We propose the following decentralized optimization algorithm:

$$x_i^{k+1} = \sum_{j\in\mathbb{N}_i} w_{ij}(x_j^k - \lambda^k g_j^k) \tag{5}$$

The detailed implementation procedure for individual agents is provided in Algorithm 1. Compared with the conventional decentralized optimization algorithm, it can be seen that instead of letting agent j share x_j^k with neighboring agents, we let agent j share $v_{ij}^k \triangleq w_{ij}(x_j^k - \lambda^k g_j^k)$ with all its neighbors $i \in \mathbb{N}_j$. This new algorithm has two advantages over the conventional one in (4): First, it includes the server based distributed optimization like federated learning as a special case. More specifically, when all x_j^k $(1 \leq j \leq m)$ are forced to be the same, then different agents use the same parameter x_j^k but different local data sets to calculate gradients, the average of which is used to update the universal state. This is exactly the architecture used in federated learning. Note that since the conventional decentralized algorithms share x_j^k among participating agents, they cannot be used to describe the server based distributed optimization like federated learning. Secondly, in the shared message v_{ij}^k, the optimization variable x_j^k and the gradient g_j^k are blended together, which makes it impossible for a receiving agent to uniquely determine x_j^k or g_j^k based on received information. In fact, the transmission of $x_i^k - \lambda^k g_i^k$ in our scheme amounts to transforming x_i^k to a different reference frame (by adding an unknown displacement $\lambda^k g_i^k$), which avoids the receiver from inferring the value of x_i^k or gradient g_i^k. In contrast, the conventional scheme directly discloses the optimization variable x_i^k, which also makes g_i^k inferrable by an adversary that has access to all messages shared in the network. Note that, because v_{ij}^k has the same dimension as x_j^k, the new algorithm does not increase the communication overhead compared with conventional decentralized algorithms. This one-variable only information-sharing scheme is important in many applications such as machine learning because in these applications the dimension of the optimization variables can scale to hundreds of millions, which causes significant communication overhead and even communication bottlenecks [40].

Algorithm 1: Decentralized nonconvex optimization algorithm

Parameters: W, λ^k

(1) **for** $k = 1, 2, \cdots$ **do**

(a) Every agent j computes and sends to agent $i \in \mathbb{N}_j$

$$v_{ij}^k \triangleq w_{ij}(x_j^k - \lambda^k g_j^k) \tag{6}$$

(b) After receiving v_{ij}^k from all $j \in \mathbb{N}_i$, agent i updates its state as follows:

$$x_i^{k+1} = \sum_{j\in\mathbb{N}_i} v_{ij}^k = \sum\nolimits_{j\in\mathbb{N}_i} w_{ij}^k(x_j^k - \lambda^k g_j^k) \tag{7}$$

(c) **end**

Although the new algorithm provides inherent privacy protection by sharing the mixture of the optimization variable and the gradient, the achieved privacy may not be strong enough. Therefore, we propose to inject additional additive noise to the gradient to ensure rigorous differential privacy. More specifically, instead of letting agent j send $v_{ij}^k = w_{ij}(x_j^k - \lambda^k g_j^k)$ to agent i, we let agent j send $v_{ij}^k = w_{ij}(x_j^k - \lambda^k(g_j^k + n_j^k))$ to agent i, where $n_j^k \in \mathbb{R}^d$ is a d dimensional Gaussian noise with mean zero and covariance matrix σI_d. The detailed implementation procedure for individual agents is provided in Algorithm 2.

Algorithm 2: Decentralized nonconvex optimization algorithm with differential privacy

Parameters: W, λ^k

(1) **for** $k = 1, 2, \cdots$ **do**

(a) Every agent j computes and sends to agent $i \in \mathbb{N}_j$

$$v_{ij}^k \triangleq w_{ij}(x_j^k - \lambda^k(g_j^k + n_j^k)) \tag{8}$$

(b) After receiving v_{ij}^k from all $j \in \mathbb{N}_i$, agent i updates its state as follows:

$$x_i^{k+1} = \sum_{j \in \mathbb{N}_i} v_{ij}^k = \sum\nolimits_{j \in \mathbb{N}_i} w_{ij}^k(x_j^k - \lambda^k(g_j^k + n_j^k)) \tag{9}$$

(c) **end**

We will prove that not only can the noise n_j^k ensure rigorous differential privacy for the data samples and gradient of agent j, but it will also bring differential privacy protection to the optimization variable x_j^k before the algorithm converges. (Note that we do not need to protect the privacy of the final optimization variable after convergence because the objective of decentralized optimization is to let individual agents learn the same optimal optimization variable.) In fact, as illustrated later in Sec. V, compared with the conventional decentralized algorithm, our algorithm architecture greatly facilitates differential privacy design. For example, to protect g_i^k with a designated privacy strength, since the transmitted message is $x_i^k - \lambda^k g_i^k$, the sender can easily calculate the amount of noise that it should add to shared messages. In contrast, in the conventional distributed optimization framework, as the shared information is x_i^k, it is not directly clear how much noise should be added to x_i^k to achieve a certain privacy strength for g_i^k.

What is more interesting is that the injected additive noise does not compromise the provable convergence of the algorithm, but instead, it ensures avoidance of undesired stationary points like maxima and saddle points, and hence enhances the accuracy of decentralized nonconvex optimization. This is significant because differential-privacy noise is known to sacrifice algorithmic accuracy for privacy. In the following two sections, we will rigorously analyze the convergence and maximum/saddle avoidance of Algorithm 2 and characterize its privacy-preservation performance in Sec. 4 and Sec. 5, respectively.

For the convenience of the convergence analysis, we augment the individual-agent dynamics in (9) and obtain the network-level dynamics:

$$x^{k+1} = (W \otimes I_d)(x^k - \lambda^k(g^k + N^k)) \tag{10}$$

where $\lambda^k \geq 0$ denotes the stepsize at iteration k, and x^k, g^k, and N^k denote the stacked optimization variables, gradients, and noise respectively, i.e.,

$$\begin{aligned} x^k &= [(x_1^k)^T, (x_2^k)^T, \cdots (x_m^k)^T]^T, \\ g^k &= [(g_1^k)^T, (g_2^k)^T, \cdots (g_m^k)^T]^T, \\ N^k &= [(n_1^k)^T, (n_2^k)^T, \cdots (n_m^k)^T]^T. \end{aligned}$$

The symbol $\otimes$ denotes the Kronecker product.

4 Convergence Analysis

In this section, we first prove that even in the presence of differential-privacy noise, all x_i^k in Algorithm 2 will reach consensus almost surely (Sec. 4.1). Then, we will prove in Sec. 4.2 and Sec. 4.3 that the differential-privacy noise in Algorithm 2 guarantees avoidance of, respectively, local maxima and saddle points.

For the convenience of analysis, we define the average vector $\bar{x}^k$ as

$$\bar{x}^k = \frac{\sum_{i=1}^m x_i^k}{m}$$

Since W is column stochastic, from (10), we have

$$\bar{x}^{k+1} = \bar{x}^k - \frac{\lambda^k}{m} \sum_{i=1}^m (g_i^k + n_i^k) \tag{11}$$

4.1 Consensus of all x_i^k

We first prove that all x_i^k $(1 \leq i \leq m)$ converge to the average $\bar{x}^k$ almost surely.

Theorem 1 *Under Assumption 1, Assumption 2, and Assumption 4, when the stepsize λ^k is square summable, i.e., $\sum_{k=0}^{\infty}(\lambda^k)^2 < \infty$, we have $\lim_{k\to\infty} \|x_i^k - \bar{x}^k\| = 0$ almost surely for all i.*

Proof: Using (9) and (11), one obtains

$$x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1} = \bar{W} x^k - \lambda^k \bar{W}(g^k + N^k) \tag{12}$$

where $\bar{W} = (W - \frac{\mathbf{1}\mathbf{1}^T}{m}) \otimes I_d$.

Since $(W - \frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{1} = 0$, we have for any element $\bar{x}^k[\ell]$ of $\bar{x}^k$ (where $1 \leq \ell \leq d$) that, $(W - \frac{\mathbf{1}\mathbf{1}^T}{m})\mathbf{1}\bar{x}^k[\ell] = 0$ holds, which further leads to

$$(W - \frac{\mathbf{1}\mathbf{1}^T}{m}) \otimes I_d \cdot \bar{x}^k \otimes \mathbf{1} = 0 \tag{13}$$

Combining (12) and (13) yields

$$x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1} = \bar{W}(x^k - \bar{x}^k \otimes \mathbf{1}) - \lambda^k \bar{W}(g^k + N^k) \tag{14}$$

Using the definition of $\eta = \|W - \frac{\mathbf{1}\mathbf{1}^T}{m}\|$ in Assumption 4, we obtain

$$\|x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1}\| \leq \eta \|x^k - \bar{x}^k \otimes \mathbf{1}\| + \eta \lambda^k \|g^k + N^k\| \tag{15}$$

By taking squares on both sides and using the inequality

$$2ab \leq \epsilon a^2 + \epsilon^{-1} b^2$$

which holds for any a, $b \in \mathbb{R}$ and $\epsilon > 0$, we obtain

$$\begin{aligned} \|x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1}\|^2 \leq & \eta^2(1+\epsilon)\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2 \\ & + \eta^2(\lambda^k)^2(1+\epsilon^{-1})\|g^k + N^k\|^2 \end{aligned} \tag{16}$$

i.e.,

$$\begin{aligned} \|x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1}\|^2 \leq & \|x^k - \bar{x}^k \otimes \mathbf{1}\|^2 \\ & - \left(1 - \eta^2(1+\epsilon)\right) \|x^k - \bar{x}^k \otimes \mathbf{1}\|^2 \\ & + \eta^2(\lambda^k)^2(1+\epsilon^{-1})\|g^k + N^k\|^2 \end{aligned} \tag{17}$$

By setting $1+\epsilon = \frac{1}{\eta}$ which further leads to $\eta^2(1+\epsilon^{-1}) = \frac{\eta^2}{1-\eta}$, one yields

$$\begin{aligned} \|x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1}\|^2 \leq & \|x^k - \bar{x}^k \otimes \mathbf{1}\|^2 \\ & - (1-\eta) \|x^k - \bar{x}^k \otimes \mathbf{1}\|^2 \\ & + (\lambda^k)^2 \frac{\eta^2}{1-\eta} \|g^k + N^k\|^2 \end{aligned} \tag{18}$$

Summing the preceding inequality over $k = 0, 1, \cdots$ yields

$$\begin{aligned} & (1-\eta) \sum_{k=0}^{\infty} \|x^k - \bar{x}^k \otimes \mathbf{1}\|^2 + \|x^\infty - \bar{x}^\infty \otimes \mathbf{1}\|^2 \\ & - \|x^0 - \bar{x}^0 \otimes \mathbf{1}\|^2 \\ & \quad \leq \sum_{k=0}^{\infty} (\lambda^k)^2 \frac{\eta^2}{1-\eta} \|g^k + N^k\|^2 \\ & \quad \leq \sum_{k=0}^{\infty} (\lambda^k)^2 \frac{\eta^2}{1-\eta} \|g^k\|^2 + \sum_{k=0}^{\infty} (\lambda^k)^2 \frac{\eta^2}{1-\eta} \|N^k\|^2 \end{aligned} \tag{19}$$

According to Assumption 2, g^k is bounded. Therefore, we have $\sum_{k=0}^{\infty} (\lambda^k)^2 \frac{\eta^2}{1-\eta} \|g^k\|^2 < \infty$ when λ^k is square summable. For the second term on the right hand side of (19), according to [19], we have $\|N^k\|^2$ being finite almost surely (note that almost surely finite is different from almost surely bounded). Therefore, under square summable λ^k, the second term on the right hand side of (19) is finite almost surely. In summary, the right hand side of (19) is finite almost surely, meaning that $(1-\eta) \sum_{k=0}^{\infty} \|x^k - \bar{x}^k \otimes \mathbf{1}\|^2$ is finite almost surely, which further implies that $\lim_{k\to\infty} \|x_i^k - \bar{x}^k\| = 0$ holds almost surely for all i. □

Remark 2 *The theorem can also be proven using our proof technique in [48].*

Remark 3 *Besides almost sure convergence, we can also prove that* $\lim_{k\to\infty} \mathbb{E}\left[\|x_i^k - \bar{x}^k\|^2\right] \to 0$, *where* $\mathbb{E}[\cdot]$ *is taken with respect to the* σ*-field generated by the Gaussian noise sequence* $\{N^k\}$. *More specifically, in the derivation of Theorem 1, (18) also implies*

$$\begin{aligned} & \mathbb{E}\left[\|x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1}\|^2\right] \\ & \leq \mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] - (1-\eta)\,\mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] \\ & \quad + (\lambda^k)^2 \frac{\eta^2}{1-\eta} \mathbb{E}\left[\|g^k + N^k\|^2\right] \\ & \leq \mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] - (1-\eta)\,\mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] \\ & \quad + (\lambda^k)^2 \frac{\eta^2}{1-\eta} \mathbb{E}\left[2\|g^k\|^2 + 2\|N^k\|^2\right] \\ & \leq \mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] - (1-\eta)\,\mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] \\ & \quad + (\lambda^k)^2 \frac{\eta^2}{1-\eta} 2mG^2 + (\lambda^k)^2 \frac{\eta^2}{1-\eta} 2m\sigma \end{aligned} \tag{20}$$

where we have made use of Assumption 2 and the fact that n_i^k *has covariance matrix* σI_d.

Summing the preceding inequality over $k = 0, 1, \cdots$ yields

$$(1-\eta)\sum_{k=0}^{\infty}\mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right] + \mathbb{E}\left[\|x^\infty - \bar{x}^\infty \otimes \mathbf{1}\|^2\right] - \mathbb{E}\left[\|x^0 - \bar{x}^0 \otimes \mathbf{1}\|^2\right] \leq \sum_{k=0}^{\infty}(\lambda^k)^2\frac{\eta^2}{1-\eta}2mG^2 + \sum_{k=0}^{\infty}(\lambda^k)^2\frac{\eta^2}{1-\eta}2m\sigma \tag{21}$$

Therefore, the right hand side of (21) is finite, meaning that $(1-\eta)\sum_{k=0}^{\infty}\mathbb{E}\left[\|x^k - \bar{x}^k \otimes \mathbf{1}\|^2\right]$ is finite, which further implies that $\lim_{k\to\infty}\mathbb{E}\left[\|x_i^k - \bar{x}^k\|^2\right] = 0$ holds for all i.

4.2 Avoidance of local maxima

Theorem 1 states that all x_i^k will converge to each other almost surely. However, it is still unclear what the convergence point is. In the centralized case, it has been shown that additive noise can enable a stochastic-approximation based optimization process to converge to a local minimum when there are no saddle points:

Lemma 1 *[32] For a stochastic approximation process*

$$x^{k+1} = x^k - a^k(\nabla f(x^k) + q(k, x^k) + w^k),$$

if the following conditions are satisfied:

(1) $f(x)$ satisfies $\lim_{\|x\|\to\infty} f(x) \to \infty$ and $\|\nabla f(x)\| < C$ for some C;
(2) $\sum_{k=0}^{\infty} a^k q(k, x^k) < \infty$ holds almost surely;
(3) w^k are independent random variables satisfying $\mathbb{E}\{w_i^k\} = 0$ and $\mathbb{E}\{(w_i^k)^2\} = \sigma I$ with $\sigma < \infty$ for each element w_i^k of w^k;
(4) a^k is not summable but square summable,

then state x^k converges almost surely to a point of the union of saddle and minima or the boundary of the union. (It will avoid the local maxima almost surely).

Recently, the above result has been extended in [41] to push-sum based distributed optimization under the assumption of no saddle points. However, the push-sum based distributed optimization approach in [41] has to share two variables in every iteration (one optimization variable and an additional gradient-tracking variable), which is undesirable when the dimension of the optimization variable is high. In fact, in modern deep learning applications, the dimensions of optimization variables can scale to hundreds of millions, and hence information sharing could create significant communication overhead and even communication bottlenecks [40]. In this paper, we show that the result in Lemma 1 can be extended to our decentralized optimization framework which only shares one variable in every iteration:

Theorem 2 *Under Assumption 1, Assumption 2, and Assumption 4, when λ^k is non-increasing, is not summable but square summable, i.e., $\sum_{k=0}^{\infty}\lambda^k = \infty$ and $\sum_{k=0}^{\infty}(\lambda^k)^2 < \infty$, then all states x_i^k converge almost surely to the same point in the union of saddles and minima or the boundary of the union. (It will avoid local maxima almost surely).*

Proof: It can be seen that under the conditions of the theorem, the conditions in Theorem 1 are satisfied and hence all x_i^k will converge to the average state $\bar{x}^k$ almost surely. So we only need to prove that $\bar{x}^k$ will converge to a point in the union of saddles and minima or the boundary of the union.

From (11), we can rewrite the dynamics of $\bar{x}^k$ as follows

$$\begin{aligned}\bar{x}^{k+1} &= \bar{x}^k - \frac{\lambda^k}{m}\sum_{i=1}^{m}(g_i^k + n_i^k)\\ &= \bar{x}^k - \lambda^k\nabla f(\bar{x}^k) - \lambda^k\frac{\sum_{i=1}^m n_i^k}{m}\\ &\quad + \lambda^k\left(\nabla f(\bar{x}^k) - \frac{\sum_{i=1}^m g_i^k}{m}\right)\end{aligned} \tag{22}$$

Since when n_i^k follows Gaussian distribution, $\frac{\sum_{i=1}^m n_i^k}{m}$ also follows Gaussian distribution, it can be seen that (22) resembles the dynamics in Lemma 1 with $q(k, x^k) = \frac{\sum_{i=1}^m g_i^k}{m} - \nabla f(\bar{x}^k)$ and $a^k = \lambda^k$. Therefore, according to Lemma 1, if we can prove that $\sum_{k=0}^{\infty}\lambda^k q(k, x^k)$ is finite almost surely, then it will follow that $\bar{x}^k$ will converge to a point in the union of saddles and minima or the boundary of the union almost surely, and hence that all x_i^k will converge to the same point in the union of saddles and minima or the boundary of the union, almost surely.

One can verify that $q(k, x^k)$ satisfies the following relationship:

$$\begin{aligned}\|q(k, x^k)\| &= \left\|\frac{\sum_{i=1}^m g_i^k}{m} - \nabla f(\bar{x}^k)\right\|\\ &= \left\|\frac{\sum_{i=1}^m (g_i^k - \nabla f_i(\bar{x}^k))}{m}\right\|\\ &\leq \frac{L}{m}\sum_{i=1}^{m}\|x_i^k - \bar{x}^k\|\\ &\leq \frac{L}{\sqrt{m}}\|x^k - \bar{x}^k \otimes \mathbf{1}\|\end{aligned} \tag{23}$$

where the second to last inequality used the Lipschitz continuous assumption of the gradients in Assumption 1 and the last inequality used the inequality $\sum_{i=1}^m a_i \leq \sqrt{m\sum_i^m (a_i)^2}$.

From (15), we have

$$\begin{aligned}
&\|x^{k+1} - \bar{x}^{k+1} \otimes \mathbf{1}\| \\
&\leq \eta \|x^k - \bar{x}^k \otimes \mathbf{1}\| + \eta\lambda^k \|g^k + N^k\| \\
&\leq \eta^2 \|x^{k-1} - \bar{x}^{k-1} \otimes \mathbf{1}\| + \eta^2 \lambda^{k-1} \|g^{k-1} + N^{k-1}\| \\
&\quad + \eta\lambda^k \|g^k + N^k\| \\
&\quad \vdots \\
&\leq \eta^{k+1} \|x^0 - \bar{x}^0 \otimes \mathbf{1}\| + \sum_{l=0}^{k} \eta^{k+1-l} \lambda^l \|g^l + N^l\|
\end{aligned} \tag{24}$$

which leads to

$$\begin{aligned}
\sum_{k=0}^{\infty} \lambda^k q(k, x^k) \leq & \frac{L}{\sqrt{m}} \sum_{k=0}^{\infty} \eta^k \lambda^k \|x^0 - \bar{x}^0 \otimes \mathbf{1}\| \\
& + \frac{L}{\sqrt{m}} \sum_{k=0}^{\infty} \lambda^k \sum_{l=0}^{k-1} \eta^{k-l} \lambda^l \|g^l + N^l\|
\end{aligned} \tag{25}$$

Since λ^k is non-increasing, we have

$$\sum_{k=0}^{\infty} \lambda^k \eta^k \leq \lambda^0 \sum_{k=0}^{\infty} \eta^k = \lambda^0 \frac{1}{1-\eta} < \infty$$

which further means that the first item on the right hand side of (25) is finite.

For the second term on the right hand side of (25), since $\{\lambda^k\}$ is a non-increasing sequence, we always have $\lambda^k \leq \lambda^\ell$ for $\ell \leq k$ and hence

$$\sum_{k=0}^{\infty} \lambda^k \sum_{l=0}^{k-1} \eta^{k-l} \lambda^l \leq \sum_{k=0}^{\infty} \sum_{l=0}^{k-1} \eta^{k-l} (\lambda^l)^2$$

Noticing that $(\lambda^l)^2$ is summable and η resides in the interval (0, 1), we have that $\sum_{k=0}^{\infty} \lambda^k \sum_{l=0}^{k-1} \eta^{k-l} \lambda^l$ is finite according to Lemma 3 in the Appendix. Further using Assumption 2 that g^k is always bounded and the observation that N^k is finite almost surely [19], we have that the right hand side of (25) is finite almost surely. Therefore, we can conclude that $\sum_{k=0}^{\infty} \lambda^k q(x^k, \bar{x}^k)$ is finite almost surely.

In summary, under the conditions of the theorem, all conditions in Lemma 1 are satisfied. Therefore, we can conclude that $\bar{x}^k$ will converge to a point in the union of saddles and minima or the boundary of the union, and hence all x_i^k will converge to the same point in the union of saddles and minima or the boundary of the union. □

4.3 Avoidance of saddle points

As we discussed in Sec. I, avoiding saddle points is a central challenge for first-order based optimization methods. Recently some advances have been reported on avoiding saddles in nonconvex optimization (see e.g., [16,23]). However, these results are all for centralized optimization. Given that in decentralized optimization generally the local objective functions of individual agents may have saddle points different from those of the aggregated objective function, and inter-agent interactions also complicate the evolution of local optimization variables, it is unclear if the results for the centralized case can immediately be generalized to the decentralized optimization problem. Therefore, in this section, we systematically address the saddle avoidance problem by leveraging reported results on centralized optimization.

Inspired by the results in [23], we also use coupling sequence to address the problem of saddle escaping:

Definition 2 *Given an optimization algorithm*

$$x^{k+1} = x^k - \lambda^k (\nabla f(x^k) + \xi^k(x^k) + w^k) \tag{26}$$

where w^k is Gaussian noise with an identity covariance matrix and $\xi^k(x^k)$ is some function of the state x^k, we call $\{x'^k\}$ and $\{x''^k\}$ coupling sequences starting from a strict saddle point x_0 if the following three conditions are satisfied:

(1) both sequences start from x_0;
(2) both are obtained as separate runs of the optimization algorithm under the same $\xi^k(x^k)$;
(3) both are obtained under w'^k and w''^k that are only different in the e_1^T direction, i.e., $e_1^T w'^k = -e_1^T w''^k$, where e_1 denotes the eigenvector associated with the minimum eigenvalue of the Hessian matrix at the saddle point x_0.

The results for the centralized optimization in [23] show that for a strict saddle x_0, if with a positive probability, the magnitude of the projected $\xi^k(x^k)$ on $x'^k - x''^k$ is less than half of the magnitude of the projected w^k on $x'^k - x''^k$, then the algorithm in (26) can effectively avoid the saddle point:

Lemma 2 *[23] For the stochastic approximation process $x^{k+1} = x^k - a(\nabla f(x^k) + \xi^k(x^k) + w^k)$ where w^k is Gaussian noise with an identity covariance matrix, suppose that $\{x'^k\}$ and $\{x''^k\}$ are coupling sequences starting from a strict saddle point x_0. If with a positive probability, the magnitude of the projected $\xi^k(x^k)$ on $x'^k - x''^k$ is less than half of the magnitude of projected w^k on $x'^k - x''^k$ (the dynamics of the difference $x'^k - x''^k$ is dominated by w^k), then for any given probability $0 < \mu < 1$, x^k will escape the saddle x_0 with probability at least $1 - \mu$ after at most $\mathcal{O}(\frac{\log(\frac{1}{\mu})}{a})$ iterates with a sufficiently small constant stepsize $a \leq \frac{1}{\ell}$.*

According to Theorem 1, under Assumption 4, all x_i^k will converge to the mean state $\bar{x}^k$ almost surely, so we only

need to consider if $\bar{x}^k$ can avoid saddle points. Given that the dynamics of $\bar{x}^k$ is governed by (22), from Lemma 2, if we can prove that the difference between two coupling sequences initiating from a saddle point is governed by $\frac{\sum_{i=1}^m n_i^k}{m}$ rather than $q(k,x^k) = \nabla f(\bar{x}^k) - \frac{\sum_{i=1}^m g_i^k}{m}$, then $\bar{x}^k$ will escape the saddle point, meaning that all x_i^k will escape the saddle point. More specifically, we can prove the following result:

Theorem 3 *Under Assumption 1, Assumption 2, and Assumption 4, the differentially-private decentralized nonconvex optimization algorithm in Algorithm 2 avoids saddle points with probability at least* $1-\mu$ *for any* $0<\mu<1$ *under the following stepsize strategy:*

(1) constant and small enough $\lambda^k \leq \frac{1}{\ell}$ *for the first* $\mathcal{O}(\frac{\log(\frac{1}{\mu})}{\lambda})$ *iterates; and then*
(2) diminishing λ^k *satisfying the non-summable but square summable condition.*

Proof: We first study the dynamics of the difference between two coupling sequences $\bar{x}'^k$ and $\bar{x}''^k$. Since the evolution of $\bar{x}^k$ is governed by

$$\bar{x}^{k+1} = \bar{x}^k - \lambda^k \nabla f(\bar{x}^k) + \lambda^k q(k,x^k) - \lambda^k \frac{\sum_{i=1}^m n_i^k}{m}$$

we can represent the dynamics of the two coupling sequences x'^k and x''^k as

$$\begin{aligned} x'^{k+1} &= x'^k - \lambda^k \nabla f(x'^k) + \lambda^k q(k,x'^k) - \lambda^k \xi'^k \\ x''^{k+1} &= x''^k - \lambda^k \nabla f(x''^k) + \lambda^k q(k,x''^k) - \lambda^k \xi''^k \end{aligned}$$

where ξ'^k and ξ''^k represent the corresponding aggregated Gaussian noise $\frac{\sum_{i=1}^m n_i^k}{m}$ differing in only the e_1 direction.

Therefore, the difference $\tilde{x}^k \triangleq \xi'^k - \xi''^k$ has the following dynamics

$$\begin{aligned} \tilde{x}^{k+1} =& \tilde{x}^k - \lambda^k(\nabla f(x'^k) - \nabla f(x''^k)) \\ &+ \lambda^k(q(k,x'^k) - q(k,x''^k)) - \lambda^k(\xi'^k - \xi''^k) \end{aligned} \tag{27}$$

Let $\tilde{\xi}^k \triangleq \xi'^k - \xi''^k$ and $\Delta^k \triangleq \int_0^1 \nabla^2 f(\phi x'^k - (1-\phi)x''^k)d\phi$. Noticing that $\nabla f(x'^k) - \nabla f(x''^k) = \int_0^1 \nabla^2 f(\phi x'^k - (1-\phi)x''^k)d\phi(x'^k - x''^k)$, we can rewrite (27) as

$$\tilde{x}^{k+1} = (I - \lambda^k \Delta^k)\tilde{x}^k + \lambda^k(q(k,x'^k) - q(k,x''^k)) - \lambda^k \tilde{\xi}^k$$

From (23) and (24), we have

$$\begin{aligned} \|q(k,x'^k)\| &\leq \frac{L}{\sqrt{m}}\|x^k - \bar{x}^k \otimes \mathbf{1}\| \\ &\leq \frac{L\eta^k}{\sqrt{m}}\|x^0 - \bar{x}^0 \otimes \mathbf{1}\| \\ &+ \frac{L}{\sqrt{m}}\sum_{l=0}^{k}\eta^{k-l}\lambda^l\|g^l + N^l\| \end{aligned}$$

Since g^l is bounded according to Assumption 2 (denote the upper bound as G), and under a positive probability, N^l is less than some positive T, with a positive probability, $\|g^l + N^l\|$ is less than $G+T$, which means that we have the following relationship with a positive probability under a constant stepsize λ:

$$\|q(k,x'^k)\| \leq \frac{L\eta^k}{\sqrt{m}}\|x^0 - \bar{x}^0 \otimes \mathbf{1}\| + \frac{L\lambda}{\sqrt{m}}\frac{1-\eta^k}{1-\eta}(G+T)$$

Similarly, we have

$$\|q(k,x''^k)\| \leq \frac{L\eta^k}{\sqrt{m}}\|x^0 - \bar{x}^0 \otimes \mathbf{1}\| + \frac{L\lambda}{\sqrt{m}}\frac{1-\eta^k}{1-\eta}(G+T)$$

with a positive probability. Therefore, with a positive probability, we have

$$\begin{aligned} \|q(k,x'^k) - q(k,x''^k)\| \leq& \frac{2L\eta^k}{\sqrt{m}}\|x^0 - \bar{x}^0 \otimes \mathbf{1}\| \\ &+ \frac{2L\lambda}{\sqrt{m}}\frac{1-\eta^k}{1-\eta}(G+T) \end{aligned}$$

In the mean time, given that $\tilde{\xi}^k$ is also Gaussian, we have that with a positive probability,

$$\|\tilde{\xi}^k\| > T$$

holds.

Therefore, for λ sufficiently small, we can always have $\|\tilde{\xi}^k\| > 2\|q(k,x'^k) - q(k,x''^k)\|$ with a positive probability, implying that with a positive probability, $\tilde{\xi}^k$ dominates the dynamics of $x'^k - x''^k$. Then according to Lemma 2, we have that $\bar{x}^k$ can avoid the saddle with at least probability $1-\mu$ for any $0<\mu<1$. Further using the result from Theorem 1 that all x_i^k converge to $\bar{x}^k$ almost surely yields that all x_i^k can avoid the saddle with at least probability $1-\mu$ for any $0<\mu<1$. □

5 Privacy Analysis

In this section, we prove that Algorithm 2 can provide rigorous differential privacy for data samples s_{ij}, individual agents' gradients g_i^k, and intermediate optimization

variables x_i^k. By intermediate optimization variables, we mean the evolution of optimization variables x_i^k before convergence is achieved. Note that after convergence, it is the exact objective of decentralized optimization to have all agents arrive at the same optimization variable (and hence know each other's value). We first analyze the achieved privacy strength for data samples.

According to differential privacy, the minimum of noise variance required to achieve (ϵ, δ)-differential privacy for data samples is determined by the sensitivity function

$$S_{s,i} = \sup_{\|s_{i,p} - s_{i,q}\|_1 \leq 1} \|M_i^k(s_{i,p}) - M_i^k(s_{i,q})\|_1 \qquad (28)$$

where $M_i^k \triangleq x_i^k - \lambda^k g_i^k$. Note that we replaced $v_{ji}^k = w_{ji}(x_i^k - \lambda^k g_i^k)$ with M_i^k to calculate the sensitivity function because the coefficient w_{ji} is publicly known. Further notice that $g_i^k = \nabla f_i(x_i^k)$ holds and in the above sensitivity function, changing one data sample from $s_{i,p}$ to $s_{i,q}$ only affects one cost function $\ell_i(\cdot, \cdot)$, and thus according to Assumption 1 and the relationship between $\ell_i(\cdot, \cdot)$ and $f_i(\cdot)$ in (2), one can obtain that $S_i = \frac{\nu \lambda^k}{n_i}$. Then making use of the standard result in differential privacy, we have the following theorem for privacy protection on data samples:

Theorem 4 *For any $\epsilon, \delta \in (0, 1)$, at iteration k, the noise n_i^k can ensure (ϵ, δ)-differential privacy for agent i's every data sample $s_{i,q}$ when the variance σ^2 satisfies*

$$\sigma^2 \geq 2\nu^2 (\lambda^k)^2 \frac{\ln(1.25/\delta)}{n_i^2 \epsilon^2} \qquad (29)$$

Proof: According to [13], a Gaussian noise of variance $\sigma^2 \geq 2\frac{\ln(1.25/\delta)(S_f)^2}{\epsilon^2}$ can achieve (ϵ, δ)-differential privacy for any $\epsilon, \delta \in (0, 1)$ where S_f denotes the sensitivity function. Thus the proof will be completed by incorporating the sensitivity function value of $S_i = \frac{\nu \lambda_i^k}{n_i}$ obtained just above the statement of the theorem. □

Remark 4 *Note that there are infinitely many (ϵ, δ) pairs that satisfy (29) in Theorem 4.*

Note that not only does the differential-privacy noise n_i^k added to agent i enable privacy protection for agent i's data samples, but it also provides differential-privacy protection for agent i's gradient. The sensitivity function for agent i's gradient is given by

$$S_{g,i} = \sup_{\|g_i^k - g'^k_i\|_1 \leq 1} \|M_i^k(g_i^k) - M_i^k(g'^k_i)\|_1 \qquad (30)$$

where $M_i^k \triangleq x_i^k - \lambda^k g_i^k$.

It can be shown that $S_{g,i} = \lambda^k$, and hence we have the following theorem:

Theorem 5 *For any $\epsilon, \delta \in (0, 1)$, at iteration k, the noise n_i^k can also ensure (ϵ, δ)-differential privacy for agent i's gradient g_i^k when the variance σ^2 satisfies*

$$\sigma^2 \geq 2(\lambda^k)^2 \frac{\ln(1.25/\delta)}{\epsilon^2}$$

Proof: The result follows from a similar line of argument as in the proof of Theorem 4. □

From Theorem 4, we can see that to achieve a fixed level of differential privacy for data samples, the required noise level decreases with an increase in the number of data samples n_i. To the contrary, Theorem 5 shows that the required noise level for differential privacy of gradients is not affected by the number of samples, which is understandable since the gradient of an agent is always computed from all data samples of the agent in gradient descent algorithms.

The same noise n_i^k also provides privacy protection for optimization variables. In fact, we can see that the amount of noise applied on x_i^k is $\lambda^k n_i^k$ in shared information. And the variance of this noise is $(\lambda^k)^2 \sigma^2$. Since it can be verified that the sensitivity function for x_i^k is

$$S_{x,i} = \sup_{\|x_i^k - x'^k_i\|_1 \leq 1} \|M_i^k(x_i^k) - M_i^k(x'^k_i)\|_1 = 1 \qquad (31)$$

where $M_i^k \triangleq x_i^k - \lambda^k g_i^k$, we can obtain that the same noise n_i^k also enables the following differential privacy for individual agents' optimization variables:

Theorem 6 *At iteration k, the noise n_i^k also ensures (ϵ, δ)-differential privacy for agent i's optimization variable x_i^k for any $\epsilon, \delta \in (0, 1)$ when the variance σ^2 satisfies $\sigma^2 \geq 2\frac{\ln(1.25/\delta)}{(\lambda^k)^2 \epsilon^2}$.*

Proof: The result follows from a similar line of argument as in the proof of Theorem 4. □

Remark 5 *Note that different from the enabled privacy for data samples and gradients, under a fixed noise variance σ, the strength of enabled (ϵ, δ)-differential privacy for x_i^k decreases with a decrease in λ^k. When λ^k tends to zero, the strength of enabled privacy for x_i^k will decrease to zero. However, note that this is acceptable since the purpose of decentralized optimization is for all agents to learn the same optimum value for the optimization variable cooperatively.*

Remark 6 *Also note that in all the above results on sensitivity and differential privacy, we have assumed that the*

adversary knows the underlying algorithm and can observe every shared message in the network, namely, the adversary can launch both the honest-but-curious attack and the eavesdropping attack discussed in Sec. 2.3. If the adversary is weaker in the sense that it can only launch the honest-cut-curious attack (can only observe messages shared on some but not all links), then the sensitivity of agents whose messages are unaccessible to the adversary will be smaller (and even zero), and hence these agents will have a stronger privacy protection against the adversary. The same conclusion can be drawn for the case where the adversary can only launch the eavesdropping attack (does not know the underlying algorithm).

6 Numerical Experiments

In this section, we evaluate the performance of the proposed decentralized optimization algorithm using numerical experiments in decentralized nonconvex optimization applications. More specifically, we evaluate the performance of the proposed algorithm using two application scenarios, one in decentralized estimation and the other in Independent Component Analysis, a popular dimension reduction tool in statistical machine learning and signal processing [21].

6.1 Decentralized estimation based numerical experiments

We consider a canonical decentralized estimation problem where a network of m sensors collectively estimate an unknown parameter $\theta \in \mathbb{R}^d$. More specifically, we assume that each sensor i has a measurement of the parameter, $Y_i = M\theta + w_i$, where $M \in \mathbb{R}^{s\times d}$ is the measurement matrix of agent i and w_i is measurement noise. Then the estimation of parameter θ can be solved using the optimization problem formulated as (1), with each $f_i(\theta)$ given as

$$f_i(\theta) = \|Y_i - M\theta\|^2 + \kappa\|\theta\|^3$$

Here κ is a regularization parameter, which will be chosen to have some desired properties for $f_i(\cdot)$.

It can be verified that when κ is a positive number, $f_i(\cdot)$ will be a convex function. In order to have a nonconvex objective function so as to test and evaluate the performance of our algorithm in nonconvex optimization, we set κ as $\kappa = -0.1$. We took M and Y_i $(1 \le i \le 5)$ as follows

$$M = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{bmatrix}, \quad Y_i = i \times \begin{bmatrix} 1/3 \\ 2/3 \\ 0 \end{bmatrix}$$

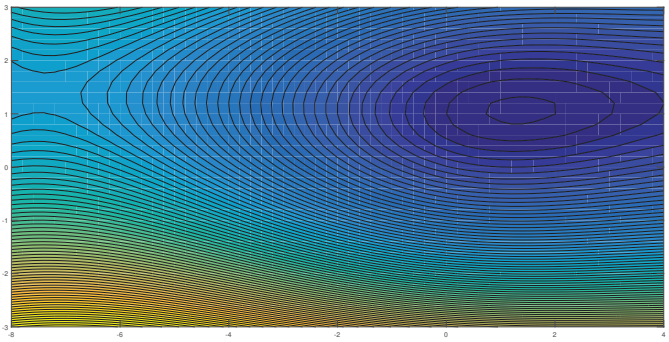


Fig. 1. A two-dimensional contour graph of $f(\theta)$ in $[-8, 4] \times [-3, 3]$.

Using the relationship

$$\nabla f_i = -2M^T Y_i + 2M^T M\theta + 3\kappa\|\theta\|\theta$$

and

$$\nabla^2 f_i = 2M^T M - 3\kappa\|\theta\| I_2 + 3\kappa\frac{1}{\|\theta\|}\theta\theta^T$$

one can obtain that the aggregated function $f(\theta) = \sum_{i=1}^5 \frac{f_i(\theta)}{5}$ has a local minimum at $\theta = \begin{bmatrix} 1.3478, \\ 1.0690 \end{bmatrix}$ and a saddle point at $\theta = \begin{bmatrix} -7.4336 \\ 1.3959 \end{bmatrix}$.

To facilitate numerical experiments, we focus on the region $\theta \in [-8, 4] \times [-3, 3]$. Outside this region we manipulate $f_i(\theta)$ to make it increase linearly with $\|\theta\|$ with continuous and smooth connection on the boundary of $[-8, 4] \times [-3, 3]$. By doing so, our optimization problem has one minimum at $\theta = \begin{bmatrix} 1.3478, \\ 1.0690 \end{bmatrix}$ (and no other local minimum) and one saddle point at $\theta = \begin{bmatrix} -7.4336 \\ 1.3959 \end{bmatrix}$, and it can be verified that the saddle point is a strict saddle point. Please see Fig. 1 for a two-dimensional contour graph of $f(\theta)$ on $[-8, 4] \times [-3, 3]$.

We considered a network of five agents interacting on a graph depicted in Fig. 2. In the numerical experiments, we set the stepsize as $\lambda^k = 0.02$ for $k \le 500$ and switched it to $\lambda_i = \frac{1}{k}$ for $k > 500$. w_i was uniformly distributed in $[0, 1]$. We added Gaussian noise with zero mean and variance $\sigma = 0.5$ in the gradients for the purpose of both privacy protection and global convergence. We first initialized the optimization variables randomly to check if the algorithm can guarantee consensus in decentralized optimization in the presence of differential-privacy noise. The evolution of all agents' optimization variables is illustrated in Fig. 3, which confirms that the algorithm

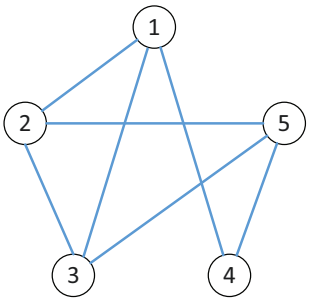


Fig. 2. The interaction topology of the network.

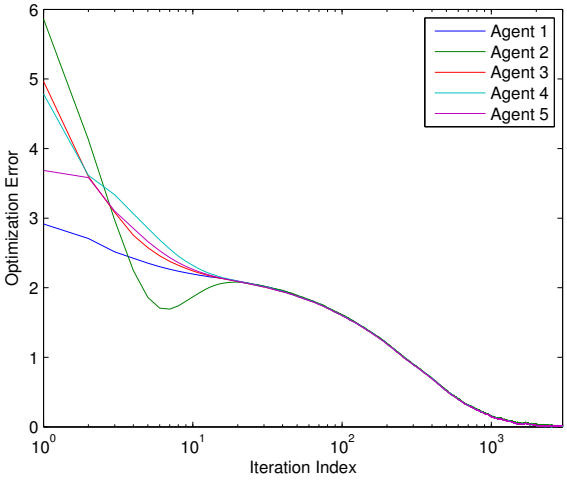


Fig. 3. Evolution of all agents' optimization errors when initialized with random values.

can indeed ensure all agents to converge to the same state under differential-privacy noise. To evaluate the performance of our algorithm with regard to avoiding the saddle point, we also initialized all agents on the saddle point, i.e., $x_i^0 = \begin{bmatrix} -7.4336 \\ 1.3959 \end{bmatrix}$ for all $1 \leq i \leq 5$. Clearly, without the differential-privacy noise, all states would be trapped at the saddle point. In contrast, the differential-privacy noise avoided the saddle point and ensured the convergence of all agents to the optimal value, as illustrated in Fig. 4, which corroborates the theoretical results in Theorem 3. We have also evaluated the influence of the magnitude of σ on the optimization error. The results are summarized in Table I, where each data point was the average of 100 runs. It can be seen that the optimization error increases with an increase in the noise magnitude σ.

6.2 Independent Component Analysis based numerical experiments

Independent Component Analysis (ICA) is widely used in signal processing and statistical machine learning to reduce the dimension of data [21]. Modeling the data

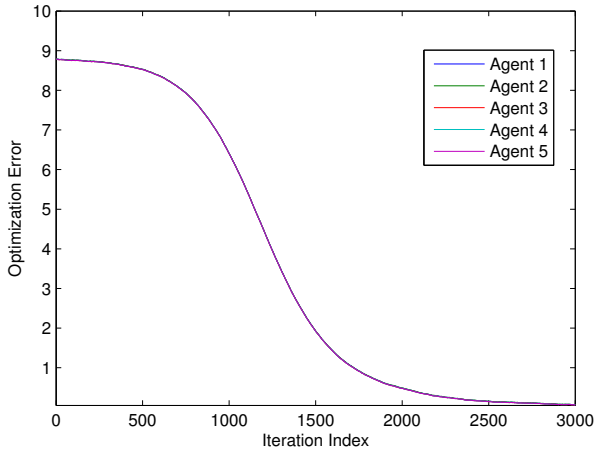


Fig. 4. Evolution of all agents' optimization errors when every agent was initialized from the saddle point.

vector as $Y = AZ$ with $A \in \mathbb{R}^{d\times d}$ an orthonormal matrix and $Z \in \mathbb{R}^d$ a non-Gaussian random sample of d independent entries, the objective of ICA is to recover one of multiple columns of A from independent observations $Y \in \mathbb{R}^d$. In standard practice, the Y vector should be whitened to have zero mean and an identity covariance matrix. Furthermore, in standard practice, the elements of Z are usually assumed to have a fourth moment $\mu \neq 3$, and the d columns of A are usually denoted as $\mathbf{a}_1$, $\mathbf{a}_2$ $\cdots$, $\mathbf{a}_d$. Under these conditions, ICA is usually cast as the following optimization problem:

$$\min_{\|u\|=1} -\text{sign}(\mu - 3) \cdot \mathbb{E}\left[(u^T Y)^4\right]$$

The saddles of the above optimization problem include (but not limited to) all $u^* = d^{-1/2}(\pm 1, \cdots, \pm 1)$, all of which satisfy the strict-saddle condition [26]. We implemented our algorithm to solve the above optimization problem. (It is worth noting that the equality constraint $\|u\| = 1$ can be handled by using the method of Lagrange multipliers [16].) In the implementation, we set $d = 10$. We also generated 800 random samples Y by randomly selecting each entry of Z from a uniform distribution in $\{-1, 1\}$. The 800 samples were evenly distributed among the five agents (each agent had 160 samples). We set the stepsize as $\lambda^k = 0.003$ for $k \leq 100$ and switched it to $\lambda_i = \frac{3}{10k}$ for $k > 100$. The network interaction topology is still the same as in Fig. 2. We evaluated the performance of our algorithm under different variances σ of the differential-privacy noise. In each case, we ran the algorithm for 100 times. In each of the 100 runs, we randomly set A to a random orthonormal matrix and randomly selected x_i^0. The evolution of the maximal reconstruction error of the first column of A, $\mathbf{a_1}$, among all five agents is shown in Fig. 5. It can be seen that after adding differential-privacy noise, our algorithm obtains comparable or even slightly better con-

Table 1
Final optimization error under different σ at $k = 3000$

	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$	$\sigma = 0.6$
Average optimization error	0.048	0.058	0.064	0.070	0.078	0.091

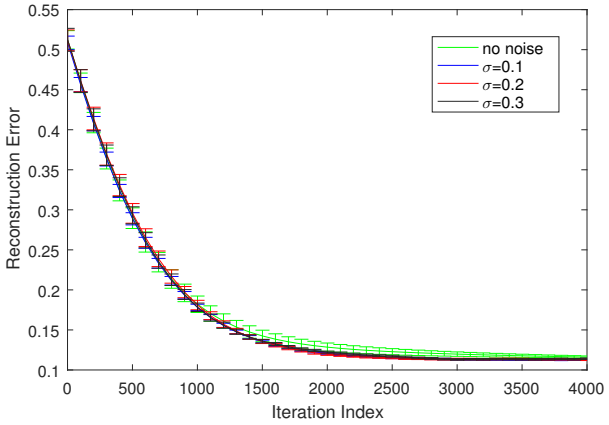


Fig. 5. Evolution of the maximum reconstruction error among all agents in the ICA application.

vergence accuracy and speed compared with the noise-free case. This is understandable since our algorithm can always avoid saddle points, whereas the noise-free case was trapped at saddle points in some of the 100 runs. Note that since the initial conditions for the 100 runs were randomly selected, the noise-free case is not always trapped at saddle points.

7 Conclusions

This paper has proposed an algorithm for decentralized nonconvex optimization that can achieve rigorous differential privacy with guaranteed convergence to a minimum point. By leveraging diminishing stepsizes, the algorithm avoids sacrificing provable convergence for differential privacy, which is a common problem with existing differential-privacy based algorithms for decentralized optimization. Besides enabling privacy protection for data samples and gradients, the approach also achieves privacy protection for optimization variables until the algorithm converges. More interestingly, we have proved that our algorithm has guaranteed saddle avoidance in a polylogrithmic number of iterations. The guarantee on differential privacy, algorithmic convergence, and saddle-avoidance simultaneously has not been reported in decentralized optimization literature. Note that since in decentralized optimization individual agents may have saddle points different from those of the centralized counterpart, the saddle-avoidance result obtained for decentralized optimization is highly nontrivial compared with existing results for centralized optimization. Numerical experiments for both a decentralized estimation problem and an independent component analysis problem confirm the effectiveness of the proposed algorithm.

Appendix A

Lemma 3 *[35] Let $\{\gamma^k\}$ be a scalar sequence. If $\gamma^k \geq 0$ for all k, $\sum_{k=0}^{\infty} \gamma^k < \infty$, and $0 < \beta < 1$, then $\sum_{k=0}^{\infty}(\sum_{\ell=0}^{k}(\beta)^{k-\ell}\gamma^{\ell}) < \infty$.*

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32:15479–15488, 2019.

[3] Juan Andrés Bazerque and Georgios B Giannakis. Distributed spectrum sensing for cognitive radio networks by exploiting sparsity. *IEEE Transactions on Signal Processing*, 58(3):1847–1862, 2009.

[4] Pascal Bianchi and Jérémie Jakubowicz. Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization. *IEEE Transactions on Automatic Control*, 58(2):391–405, 2012.

[5] Nicolas Boumal, Vlad Voroninski, and Afonso Bandeira. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. *Advances in Neural Information Processing Systems*, 29:2757–2765, 2016.

[6] Xuanyu Cao and Tamer Başar. Decentralized online convex optimization with event-triggered communications. *IEEE Transactions on Signal Processing*, 69:284–299, 2020.

[7] Xuanyu Cao and Tamer Başar. Decentralized online convex optimization based on signs of relative states. *Automatica*, 129:109676, 2021.

[8] Frank E Curtis, Daniel P Robinson, and Mohammadreza Samadi. A trust region algorithm with a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$ for nonconvex optimization. *Mathematical Programming*, 162(1-2):1–32, 2017.

[9] Amir Daneshmand, Gesualdo Scutari, and Vyacheslav Kungurtsev. Second-order guarantees of distributed gradient algorithms. *SIAM Journal on Optimization*, 30(4):3029–3068, 2020.

[10] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pages 1155–1164. PMLR, 2018.

[11] Paolo Di Lorenzo and Gesualdo Scutari. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[12] Simon Du, Chi Jin, Jason Lee, Michael Jordan, Barnabas Póczos, and Aarti Singh. Gradient descent can take exponential time to escape saddle points. In *Advances in Neural Information Processing Systems*, pages 1068–1078, 2017.

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

[14] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476. PMLR, 2018.

[15] Véronique Gayrard, Anton Bovier, Michael Eckhoff, and Markus Klein. Metastability in reversible diffusion processes I: Sharp asymptotics for capacities and exit times. *Journal of the European Mathematical Society*, 6(4):399–424, 2004.

[16] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points: online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842. PMLR, 2015.

[17] Rong Ge, Chi Jin, and Yi Zheng. No spurious local minima in nonconvex low rank problems: A unified geometric analysis. In *International Conference on Machine Learning*, pages 1233–1242. PMLR, 2017.

[18] Oded Goldreich. *Foundations of Cryptography: volume 2, Basic Applications*. Cambridge University Press, 2001.

[19] Peter Hall and Christopher Heyde. *Martingale Limit Theory and Its Application*. Academic Press, 2014.

[20] Zhenqi Huang, Sayan Mitra, and Nitin Vaidya. Differentially private distributed optimization. In *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, pages 1–10, 2015.

[21] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[22] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. *Advances in Neural Information Processing Systems*, 30, 2017.

[23] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M Kakade, and Michael I Jordan. On nonconvex optimization for machine learning: Gradients, stochasticity, and saddle points. *Journal of the ACM (JACM)*, 68(2):1–29, 2021.

[24] Anastasia Koloskova, Sebastian Stich, and Martin Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *International Conference on Machine Learning*, pages 3478–3487. PMLR, 2019.

[25] Jason D Lee, Max Simchowitz, Michael I Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In *Conference on learning theory*, pages 1246–1257. PMLR, 2016.

[26] Chris Junchi Li, Zhaoran Wang, and Han Liu. Online ICA: Understanding global dynamics of nonconvex optimization via diffusion processes. In *Advances in Neural Information Processing Systems*, pages 4967–4975, 2016.

[27] Peng Lin, Wei Ren, and Yongduan Song. Distributed multi-agent optimization subject to nonidentical constraints and communication delays. *Automatica*, 65:120–131, 2016.

[28] Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in Neural Information Processing Systems*, 24, 2011.

[29] Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[30] Yurii Nesterov. Squared functional systems and optimization problems. In *High Performance Optimization*, pages 405–440. Springer, 2000.

[31] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[32] Mikhail Borisovich Nevelson, Rafail Zalmanovich Khasminskii, and Rafail Zalmanovich Khasminskii. *Stochastic Approximation and Recursive Estimation*, volume 47. Amer Mathematical Society, 1976.

[33] Guannan Qu and Na Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, 2017.

[34] Robin L Raffard, Claire J Tomlin, and Stephen P Boyd. Distributed optimization for cooperative agents: Application to formation flight. In *2004 43rd IEEE Conference on Decision and Control (CDC)*, volume 3, pages 2453–2459. IEEE, 2004.

[35] S Sundhar Ram, Angelia Nedić, and Venugopal V Veeravalli. Distributed stochastic subgradient projection algorithms for convex optimization. *Journal of Optimization Theory and Applications*, 147(3):516–545, 2010.

[36] Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–1761, 2014.

[37] Ju Sun, Qing Qu, and John Wright. Complete dictionary recovery over the sphere I: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, 2016.

[38] Brian Swenson, Soummya Kar, H Vincent Poor, and José MF Moura. Annealing for distributed global optimization. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3018–3025. IEEE, 2019.

[39] Brian Swenson, Soummya Kar, H Vincent Poor, José MF Moura, and Aaron Jaech. Distributed gradient methods for nonconvex optimization: Local and global convergence guarantees. *arXiv preprint arXiv:2003.10309*, 2020.

[40] Zhenheng Tang, Shaohuai Shi, Xiaowen Chu, Wei Wang, and Bo Li. Communication-efficient distributed deep learning: A comprehensive survey. *arXiv preprint arXiv:2003.06307*, 2020.

[41] Tatiana Tatarenko and Behrouz Touri. Non-convex distributed optimization. *IEEE Transactions on Automatic Control*, 62(8):3744–3757, 2017.

[42] Konstantinos I Tsianos, Sean Lawlor, and Michael G Rabbat. Consensus-based distributed optimization: Practical issues and applications in large-scale machine learning. In *Proceedings of the 50th annual Allerton Conference on Communication, Control, and Computing*, pages 1543–1550. IEEE, 2012.

[43] John N. Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

[44] Georgios Tychogiorgos, Athanasios Gkelias, and Kin K Leung. A non-convex distributed optimization framework and its application to wireless ad-hoc networks. *IEEE Transactions on Wireless Communications*, 12(9):4286–4296, 2013.

[45] Hoi-To Wai, Jean Lafond, Anna Scaglione, and Eric Moulines. Decentralized Frank–Wolfe algorithm for convex and nonconvex problems. *IEEE Transactions on Automatic Control*, 62(11):5522–5537, 2017.

[46] Yongqiang Wang and Tamer Başar. Gradient-tracking based distributed optimization with guaranteed optimality under noisy information sharing. *IEEE Transactions on Automatic Control*, 2022.

[47] Yongqiang Wang and Tamer Başar. Quantization enabled privacy protection in decentralized stochastic optimization. *IEEE Transactions on Automatic Control*, 2022.

[48] Yongqiang Wang and Angelia Nedic. Tailoring gradient methods for differentially-private distributed optimization. *arXiv preprint arXiv:2202.01113*, 2022.

[49] Yongqiang Wang and H Vincent Poor. Decentralized stochastic optimization with inherent privacy protection. *IEEE Transactions on Automatic Control*, 2022.

[50] Ermin Wei, Asuman Ozdaglar, and Ali Jadbabaie. A distributed Newton method for network utility maximization–I: Algorithm. *IEEE Transactions on Automatic Control*, 58(9):2162–2175, 2013.

[51] Feng Yan, Shreyas Sundaram, SVN Vishwanathan, and Yuan Qi. Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2483–2493, 2012.

[52] Tao Yang, Xinlei Yi, Junfeng Wu, Ye Yuan, Di Wu, Ziyang Meng, Yiguang Hong, Hong Wang, Zongli Lin, and Karl H Johansson. A survey of distributed optimization. *Annual Reviews in Control*, 47:278–305, 2019.

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

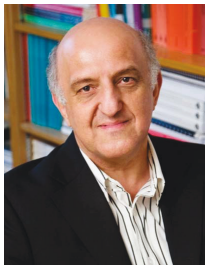
[54] Chunlei Zhang, Muaz Ahmad, and Yongqiang Wang. ADMM based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, 14(3):565–580, 2019.

[55] Chunlei Zhang and Yongqiang Wang. Distributed event localization via alternating direction method of multipliers. *IEEE Transactions on Mobile Computing*, 17(2):348–361, 2017.

[56] Kaiqing Zhang, Bin Hu, and Tamer Başar. Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit regularization and global convergence. In *Learning for Dynamics and Control*, pages 179–190. PMLR, 2020.

[57] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, pages 14774–14784, 2019.

Yongqiang Wang was born in Shandong, China. He received dual B.S. degrees in electrical engineering & automation and computer science & technology from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2004, and the M.Sc. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2009. From 2007-2008, he was with the University of Duisburg-Essen, Germany, as a visiting student. He was a Project Scientist at the University of California, Santa Barbara 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 protection. He currently serves as an associate editor for *IEEE Transactions on Automatic Control* and *IEEE Transactions on Control of Network Systems*.



Tamer Başar has been with the University of Illinois Urbana-Champaign since 1981, where he is currently Swanlund Endowed Chair Emeritus and Center for Advanced Study (CAS) Professor Emeritus of Electrical and Computer Engineering, with also affiliations with the Coordinated Science Laboratory, Information Trust Institute, and Mechanical Science and Engineering. At Illinois, he has also served as Director of CAS (2014-2020), Interim Dean of Engineering (2018), and Interim Director of the Beckman Institute (2008-2010). He received B.S.E.E. from Robert College, Istanbul, and M.S., M.Phil, and Ph.D. from Yale University, from which he received in 2021 the Wilbur Cross Medal. He is a member of the US National Academy of Engineering, and Fellow of IEEE, IFAC, and SIAM. He has served as presidents of IEEE CSS (Control Systems Society), ISDG (International Society of Dynamic Games), and AACC (American Automatic Control Council). He has received several awards and recognitions over the years, including the highest awards of IEEE CSS, IFAC, AACC, and ISDG, the IEEE Control Systems Award, and a number of international honorary doctorates and professorships. He has around 1000 publications in systems, control, communications, optimization, networks, and dynamic games, including books on non-cooperative dynamic game theory, robust control, network security, wireless and communication networks, and stochastic networked control. He was the Editor-in-Chief of Automatica between 2004 and 2014, and is currently editor of several book series. His current research interests include stochastic teams, games, and networks; multi-agent systems and learning; data-driven distributed optimization; epidemics modeling and control over networks; security and trust; energy systems; and cyber-physical systems.