# Privacy-Preserved Distributed Learning with Zeroth-Order Optimization

Cristiano Gratton, *Member, IEEE,* Naveen K. D. Venkategowda, *Member, IEEE,* Reza Arablouei, and Stefan Werner, *Senior Member, IEEE*

***Abstract*—We develop a privacy-preserving distributed algorithm to minimize a regularized empirical risk function when the first-order information is not available and data is distributed over a multi-agent network. We employ a zeroth-order method to minimize the associated augmented Lagrangian function in the primal domain using the alternating direction method of multipliers (ADMM). We show that the proposed algorithm, named distributed zeroth-order ADMM (D-ZOA), has intrinsic privacy-preserving properties. Most existing privacy-preserving distributed optimization/estimation algorithms exploit some perturbation mechanism to preserve privacy, which comes at the cost of reduced accuracy. Contrarily, by analyzing the inherent randomness due to the use of a zeroth-order method, we show that D-ZOA is intrinsically endowed with $(\epsilon, \delta)-$differential privacy. In addition, we employ the moments accountant method to show that the total privacy leakage of D-ZOA grows sublinearly with the number of ADMM iterations. D-ZOA outperforms the existing differentially-private approaches in terms of accuracy while yielding similar privacy guarantee. We prove that D-ZOA reaches a neighborhood of the optimal solution whose size depends on the privacy parameter. The convergence analysis also reveals a practically important trade-off between privacy and accuracy. Simulation results verify the desirable privacy-preserving properties of D-ZOA and its superiority over the state-of-the-art algorithms as well as its network-wide convergence.**

***Index Terms*—Alternating direction method of multipliers, differential privacy, distributed optimization, zeroth-order optimization methods.**

## I. Introduction

PERFORMING learning tasks at a central processing hub in a large distributed network may be prohibitive due to computation/communication costs. Collecting all data at a central hub may also create a single point of failure. Therefore, it is important to develop algorithms that are capable of processing the data gathered by agents dispersed over a distributed network [2]–[10]. Such distributed solutions are highly demanded in many of today's optimization problems pertaining to statistics [2]–[4], signal processing [5]–[7], and control [8]–[10].

This work was partly supported by the Research Council of Noway. A conference precursor of this work appears in the *Proceedings of the European Signal Processing Conference*, Amsterdam, NL, January 2021 [1].

C. Gratton and S. Werner are with the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway (email:cristiano.gratton@ntnu.no; stefan.werner@ntnu.no).

N. K. D. Venkategowda is with the Department of Science and Technology, Linköping University, 601 74, Norrköping, Sweden (email: naveen.venkategowda@liu.se ).

R. Arablouei is with the Commonwealth Scientific and Industrial Research Organisation, Pullenvale QLD 4069, Australia (email:reza.arablouei@csiro.au).

Moreover, in some real-world problems, obtaining first-order information is hard due to non-smooth objectives [2], [8], [9] or lack of any complete objective function, e.g., in bandit optimization [11], in simulation-based optimization [12], or in adversarial black-box machine learning [13]. This motivates the use of zeroth-order methods, which only use the values of the objective functions to approximate their gradients [14]–[16].

However, the communications between neighboring agents in a distributed network may lead to privacy violation issues. An adversary may infer sensitive data of one or more agents by sniffing the communicated information. The adversary can be either a curious member of the network or an eavesdropper. Therefore, it is important to develop privacy-preserving methods that allow distributed processing of data without revealing private information. Differential privacy provides privacy protection against adversarial attacks by ensuring minimal change in the outcome of the algorithm regardless of whether or not a single individual's data is taken into account.

There have been several works developing privacy-preserving algorithms for distributed convex optimization [17]–[28]. The work in [17] proposes two differentially private distributed algorithms that are based on the alternating direction method of multipliers (ADMM). The algorithms in [17] are obtained by perturbing the dual and the primal variable, respectively. However, in both algorithms, the privacy leakage of an agent is bounded only at a single iteration and an adversary might exploit knowledge available from all iterations to infer sensitive information. This shortcoming is mitigated in [18]–[21]. The works in [18], [19] develop ADMM-based differentially private algorithms with improved accuracy. The work in [20] employs the ADMM to develop a distributed algorithm where the primal variable is perturbed by adding a Gaussian noise with diminishing variance to ensure zero-concentrated differential privacy enabling higher accuracy compared to the common $(\epsilon, \delta)$-differential privacy. The work in [21] develops a stochastic ADMM-based distributed algorithm that further enhances the accuracy while ensuring differential privacy. The authors of [22]–[24] propose differentially-private distributed algorithms that utilize the projected-gradient-descent method for handling constraints. The differentially private distributed algorithm proposed in [25] is based on perturbing the local objective functions. However, the algorithms in [17]–[25], [27] offer distributed solutions only for problems with smooth objective functions.

The work in [26] addresses problems with non-smooth objective functions by employing a first-order approximation

of the augmented Lagrangian with a scalar $l_2$-norm proximity operator. However, this algorithm is not fully distributed since it requires a central coordinator to average all the perturbed primal variable updates over the network at every iteration. All the above-mentioned algorithms in [17]–[26] require some modifications through deliberately perturbing either the local estimates or the objective functions. This compromises the performance of the algorithm by degrading its accuracy especially when large amount of noise is required to provide high privacy levels. The work in [28] considers privacy-preserving properties that are intrinsic, i.e., they do not require any change in the algorithm but are associated with the algorithm's inherent properties. However, the approach taken in [28] considers a privacy metric based on the topology of the communication graph. Therefore, none of the existing algorithms are able to offer fully-distributed solutions that are intrinsically capable of ensuring differential privacy.

### A. Contributions

In this paper, we develop a fully-distributed differentially-private algorithm to solve a class of regularized empirical risk minimization (ERM) problems when first-order information is unavailable or hard to obtain. We utilize the ADMM for distributed optimization and a zeroth-order method, called the two-point stochastic gradient algorithm [29], to minimize the augmented Lagrangian function in the ADMM's primal update step. The proposed algorithm, called distributed zeroth-order ADMM (D-ZOA), is fully distributed in the sense that each agent of the network communicates only with its immediate neighbors and no central coordination is necessary. No communication among agents is required throughout the inner loop.

The privacy-preserving properties of the proposed D-ZOA algorithm are intrinsic. To substantiate this novel finding, we model the primal variable at each agent as the sum of an exact (unperturbed) value and a random perturbation. This enables us to address the challenging problem of approximating the distribution of the primal variable and verify that the stochasticity inherent to the employed zeroth-order method can adequately make D-ZOA differentially private. To this end, we find a suitable approximation for the probability distribution of the primal variable. Subsequently, we show that the inherent randomness in D-ZOA enables it to preserve $(\epsilon, \delta)$-differential privacy. Utilizing the moments accountant method [30], we also show that the total privacy leakage over all iterations grows sublinearly with the number of ADMM iterations. This is particularly important as we observe that, with any similar level of privacy, the optimization accuracy of D-ZOA is higher compared to the existing privacy-preserving approaches, which perturb the variables exchanged among the network agents by adding noise.

We prove that D-ZOA reaches a neighborhood of the optimal solution, i.e., a near-optimal solution, and the size of the neighborhood is determined by the privacy parameter. This gives an explicit privacy-accuracy trade-off where a stronger privacy guarantee corresponds to a lower accuracy. Through numerical simulations, we show that D-ZOA is competitive with the state-of-the-art zeroth-order-based optimization algorithms even though they are designed for centralized processing. We also verify numerically that the entries of the zeroth-order stochastic gradient are normally distributed by illustrating the associated histograms and (quantile-quantile) QQ plots. Simulation results also demonstrate that, with any given level of required privacy guarantee, D-ZOA outperforms existing privacy-preserving algorithms in terms of accuracy. To the best of our knowledge, this is the first work on distributed non-smooth optimization that is capable of exploiting the inherent randomness due to the use of a zeroth-order method and enjoy the ensuing intrinsic $(\epsilon, \delta)$-differential privacy.

### B. Paper Organization

The rest of the paper is organized as follows. In Section II, we describe the system model and formulate the distributed ERM problem when first-order information is not available. In Section III, we describe our proposed D-ZOA algorithm. In Section IV, we explain the privacy issues associated with distributed learning and we propose a solution to the very difficult problem of characterizing the distribution of the inherent randomness. Hence, we present the intrinsic privacy-preserving properties of the proposed D-ZOA algorithm by showing that the privacy leakage of each agent at any iteration is bounded and the total privacy leakage grows sublinearly with the number of ADMM iterations. In Section V, we prove the convergence of D-ZOA by confirming that both inner and outer loops of the algorithm converge. We provide some simulation results in Section VI and draw conclusions in Section VII.

### C. Mathematical Notations

The set of natural and real numbers are denoted by $\mathbb{N}$ and $\mathbb{R}$, respectively. The set of positive real numbers is denoted by $\mathbb{R}_+$. Scalars, column vectors, and matrices are respectively denoted by lowercase, bold lowercase, and bold uppercase letters. The operators $(\cdot)^\mathsf{T}$, $\det(\cdot)$, and $\mathrm{tr}(\cdot)$ denote transpose, determinant, and trace of a matrix, respectively. $\|\cdot\|$ represents the Euclidean norm of its vector argument. $\mathbf{I}_n$ is an identity matrix of size $n$, $\mathbf{0}_n$ is an $n \times 1$ vector with all zeros entries, $\mathbf{0}_{n\times p} = \mathbf{0}_n\mathbf{0}_p^\mathsf{T}$, and $|\cdot|$ denotes the cardinality if its argument is a set. The statistical expectation and covariance operators are represented by $\mathbb{E}[\cdot]$ and $\mathrm{cov}[\cdot]$, respectively. The notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. For a positive semidefinite matrix $\mathbf{X}$, $\lambda_{\min}(\mathbf{X})$ and $\lambda_{\max}(\mathbf{X})$ denote the nonzero smallest and largest eigenvalues of $\mathbf{X}$, respectively. For a vector $\mathbf{x} \in \mathbb{R}^n$ and a matrix $\mathbf{A}$, $\|\mathbf{x}\|_\mathbf{A}^2$ denotes the quadratic form $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}$. For a function $f$, its subgradient and subdifferential are denoted by $f'$ and $\partial f$, respectively.

## II. System Model

We consider a network with $K \in \mathbb{N}$ agents and $E \in \mathbb{N}$ edges modeled as an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V} = \{1, \ldots, K\}$ corresponds to the agents and the set $\mathcal{E}$ represents the bidirectional communication links between the pairs of neighboring agents. Edge $e_{kl} = (k, l) \in \mathcal{E}$

indicates that agent $k$ and $l$ are neighbors. Agent $k \in \mathcal{V}$ can communicate only with the agents in its neighborhood $\mathcal{V}_k = \{l \in \mathcal{V} \,|\, (k,l) \in \mathcal{E}\}$.

Each agent $k \in \mathcal{V}$ has a private dataset

$$\mathcal{D}_k = \Big\{(\mathbf{X}_k, \mathbf{y}_k) : \mathbf{X}_k = [\mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \ldots, \mathbf{x}_{k,N_k}]^\mathsf{T} \in \mathbb{R}^{N_k \times P}, \; \mathbf{y}_k = [y_{k,1}, y_{k,2}, \ldots, y_{k,N_k}]^\mathsf{T} \in \mathbb{R}^{N_k}\Big\}$$

where $N_k$ is the number of data samples collected at agent $k$ and $P$ is the number of features in each sample.

We consider the problem of estimating a parameter of interest $\boldsymbol{\beta} \in \mathbb{R}^P$ that relates the value of an output measurement stored in the response vector $\mathbf{y}_k$ to input measurements collected in the corresponding row of the local matrix $\mathbf{X}_k$. The associated supervised learning problem can be cast as a regularized ERM expressed by

$$\min_{\boldsymbol{\beta}} \sum_{k=1}^{K} \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \boldsymbol{\beta}) + \eta R(\boldsymbol{\beta}) \tag{1}$$

where $\ell : \mathbb{R}^P \to \mathbb{R}$ is the loss function, $R : \mathbb{R}^P \to \mathbb{R}$ is the regularizer function, and $\eta > 0$ is the regularization parameter. The ERM problem pertains to several applications in machine learning, e.g., linear regression [2], support vector machine [31], and logistic regression [20], [26]. We assume that the loss function $\ell(\cdot)$ and the regularizer function $R(\cdot)$ are both closed and convex but at least one of them is *non-smooth*. Let us denote the optimal solution of (1) by $\boldsymbol{\beta}^c$.

## III. Non-Smooth Distributed Learning

We first discuss the consensus-based reformulation of the problem that allows its distributed solution through an iterative process consisting of two nested loops. Then, we describe the ADMM procedure that forms the outer loop and the zeroth-order two-point stochastic gradient algorithm that constitutes the inner loop solving the ADMM primal update step. Finally, we discuss the related computational complexity.

### A. Consensus-Based Reformulation

To solve (1) in a distributed manner, we reformulate it as the following constrained minimization problem

$$\begin{aligned} \min_{\{\boldsymbol{\beta}_k\}} \quad & \sum_{k=1}^{K} \Big( \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \boldsymbol{\beta}_k) + \frac{\eta}{K} R(\boldsymbol{\beta}_k) \Big) \\ \text{s.t.} \quad & \boldsymbol{\beta}_k = \boldsymbol{\beta}_l, \quad l \in \mathcal{V}_k, \quad \forall k \in \mathcal{V} \end{aligned} \tag{2}$$

where $\{\boldsymbol{\beta}_k\}_{k=1}^K$ are the primal variables representing local copies of $\boldsymbol{\beta}$ at the agents. The equality constraints impose consensus across each agent's neighborhood $\mathcal{V}_k$. To solve (2) collaboratively and in a fully-distributed manner, we utilize the ADMM [7]. For this purpose, we rewrite (2) as

$$\begin{aligned} \min_{\{\boldsymbol{\beta}_k\}} \quad & \sum_{k=1}^{K} \Big( \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \boldsymbol{\beta}_k) + \frac{\eta}{K} R(\boldsymbol{\beta}_k) \Big) \\ \text{s.t.} \quad & \boldsymbol{\beta}_k = \mathbf{z}_k^l, \; \boldsymbol{\beta}_l = \mathbf{z}_k^l, \quad l \in \mathcal{V}_k, \quad \forall k \in \mathcal{V} \end{aligned} \tag{3}$$

where $\{\mathbf{z}_k^l\}_{k \in \mathcal{V}, l \in \mathcal{V}_k}$ are the auxiliary variables yielding an alternative but equivalent representation of the constraints in (2). They help decouple $\boldsymbol{\beta}_k$ in the constraints and facilitate the derivation of the local recursions before being eventually eliminated. Solving (3) via the ADMM requires an iterative process that is described in the next subsection.

### B. Zeroth-Order-Based Distributed ADMM Algorithm

To solve (3) by employing the ADMM, we generate the augmented Lagrangian function by associating the Lagrange multipliers $\{\bar{\boldsymbol{\gamma}}_k^l\}_{l \in \mathcal{V}_k}$, $\{\tilde{\boldsymbol{\gamma}}_k^l\}_{l \in \mathcal{V}_k}$ with the constraints in (3). Following the steps outlined in [7], the ADMM iterations to solve (3) in a distributed manner are given by

$$\boldsymbol{\beta}_k^{(m)} = \arg\min_{\boldsymbol{\beta}_k} \mathcal{F}_k(\boldsymbol{\beta}_k) \tag{4}$$

$$\boldsymbol{\gamma}_k^{(m)} = \boldsymbol{\gamma}_k^{(m-1)} + \rho \sum_{l \in \mathcal{V}_k} \Big( \boldsymbol{\beta}_k^{(m)} - \boldsymbol{\beta}_l^{(m)} \Big). \tag{5}$$

where

$$\begin{aligned} \mathcal{F}_k(\boldsymbol{\beta}_k) =& f_k(\boldsymbol{\beta}_k) + h_k(\boldsymbol{\beta}_k), \\ f_k(\boldsymbol{\beta}_k) =& \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \boldsymbol{\beta}_k) + \frac{\eta}{K} R(\boldsymbol{\beta}_k), \\ h_k(\boldsymbol{\beta}_k) =& \boldsymbol{\beta}_k^\mathsf{T} \boldsymbol{\gamma}_k^{(m-1)} + \rho \sum_{l \in \mathcal{V}_k} \Big\| \boldsymbol{\beta}_k - \frac{\boldsymbol{\beta}_k^{(m-1)} + \boldsymbol{\beta}_l^{(m-1)}}{2} \Big\|^2, \\ \boldsymbol{\gamma}_k^{(m)} =& 2 \sum_{l \in \mathcal{V}_k} \bar{\boldsymbol{\gamma}}_k^{l(m)}, \end{aligned} \tag{6}$$

$m$ is the iteration index, and all initial values $\{\boldsymbol{\beta}_k^{(0)}\}_{k \in \mathcal{V}}$, $\{\boldsymbol{\gamma}_k^{(0)}\}_{k \in \mathcal{V}}$ are set to zero. Note that the update equations in (4) and (5) can be implemented in a fully-distributed fashion since they involve only the variables available within every agent's neighborhood.

Since the objective function in (4) is assumed to be non-smooth, the corresponding minimization problem cannot be solved using any first-order method. To overcome this, we use a zeroth-order method as in [1]. We utilize the two-point stochastic-gradient algorithm that has been proposed in [29] for optimizing general non-smooth functions. More specifically, we use the stochastic mirror descent method with the proximal function $\frac{1}{2}\|\cdot\|$ and the gradient estimator at point $\boldsymbol{\beta}_k$ given by

$$\begin{aligned} \Gamma(\boldsymbol{\beta}_k, \boldsymbol{\gamma}_k^{(m-1)}, u_1, u_2, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = u_2^{-1} [\mathcal{F}_k(\boldsymbol{\beta}_k + u_1 \boldsymbol{\nu}_1 \\ + u_2 \boldsymbol{\nu}_2, \boldsymbol{\gamma}_k^{(m-1)}) - \mathcal{F}_k(\boldsymbol{\beta}_k + u_1 \boldsymbol{\nu}_1, \boldsymbol{\gamma}_k^{(m-1)})] \boldsymbol{\nu}_2 \end{aligned} \tag{7}$$

where $u_1 > 0$ and $u_2 > 0$ are smoothing constants and $\boldsymbol{\nu}_1$, $\boldsymbol{\nu}_2$ are independent zero-mean Gaussian random vectors with the covariance matrix $\mathbf{I}_P$, i.e., $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2 \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$.

The two-point stochastic-gradient algorithm consists of two randomization steps where the second step is aimed at preventing the perturbation vector $\boldsymbol{\nu}_2$ from being close to a point of non-smoothness [29]. This algorithm entails an iterative procedure that consists of three steps at each iteration $t$. First, $J \in \mathbb{N}$ independent random vectors $\{\boldsymbol{\nu}_{1,t}^{j,k}\}_{j=1}^J$ and $\{\boldsymbol{\nu}_{2,t}^{j,k}\}_{j=1}^J$

are sampled from $\mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$. Second, a $k$-local stochastic gradient $\mathbf{g}_k^{(t)}$ is computed as

$$\mathbf{g}_k^{(t)} = \frac{1}{J}\sum_{j=1}^{J} \mathbf{g}_{j,k}^{(t)} \tag{8}$$

where

$$\mathbf{g}_{j,k}^{(t)} = \Gamma(\tilde{\boldsymbol{\beta}}_k^{(t)}, \boldsymbol{\gamma}_k^{(m-1)}, u_{1,t}, u_{2,t}, \boldsymbol{\nu}_{1,t}^{j,k}, \boldsymbol{\nu}_{2,t}^{j,k}),$$

$\tilde{\boldsymbol{\beta}}_k^{(t)}$ is the $t$th iterate of the two-point stochastic-gradient algorithm with the initial value $\tilde{\boldsymbol{\beta}}_k^{(0)} = \mathbf{0}$ and $\{u_{1,t}\}_{t=1}^{\infty}$ and $\{u_{2,t}\}_{t=1}^{\infty}$ are two non-increasing sequences of positive parameters such that $u_{2,t} \leq u_{1,t}/2$. Finally, $\tilde{\boldsymbol{\beta}}_k^{(t)}$ is updated as

$$\tilde{\boldsymbol{\beta}}_k^{(t)} = \tilde{\boldsymbol{\beta}}_k^{(t-1)} - \alpha_t \mathbf{g}_k^{(t)} \tag{9}$$

where $\alpha_t$ is a time-varying step-size. The step-size is computed as

$$\alpha_t = \left(L\sqrt{tP\log(2P)}\right)^{-1} \alpha_0 R$$

where $\alpha_0$ is an appropriate initial step-size and $R$ is an upper bound on the distance between the minimizer $\boldsymbol{\beta}_k^*$ to (4) and the first iterate $\tilde{\boldsymbol{\beta}}_k^{(1)}$ as per [29].

We use multiple independent random samples $\{\boldsymbol{\nu}_{1,t}^{j,k}\}_{j=1}^{J}$ and $\{\boldsymbol{\nu}_{2,t}^{j,k}\}_{j=1}^{J}$ to obtain a more accurate estimate of the gradient $\mathbf{g}_k^{(t)}$ as remarked in [29]. Furthermore, no communication among agents is needed in the inner loop.

The proposed algorithm, D-ZOA, is summarized in Algorithm 1.

### C. Computational Complexity

Solving D-ZOA's inner loop, i.e., the minimization in (4), requires multiple evaluations of the function $\mathcal{F}_k(\cdot)$. The computational requirement at each agent and each ADMM outer loop iteration depends on the local objective function $f_k$. Let us indicate the number of computations required by D-ZOA to carry out one iteration of the inner loop at agent $k$ and the number of iterations of the inner loop by $m_k$ and $T$, respectively. Hence, the total number of computations required by D-ZOA at agent $k$ and each ADMM outer loop iteration is $\mathcal{O}(Tm_k)$. However, the cost of transmission/communication among the neighboring agents does not depend on $T$ or $m_k$ since the inner loop does not require any communication among agents.

## IV. Intrinsic Differential Privacy Guarantee

In this section, we consider the privacy concerns associated with distributed learning and reveal that the inherent randomness due to the use of a zeroth-order method is sufficient for the proposed D-ZOA algorithm to preserve $(\epsilon, \delta)$-differential privacy. First, we present the attack model along with the definition of the attacker. Second, we propose our solution to the challenging problem of characterizing the randomness inherent to the algorithm. Subsequently, we assess the $l_2$-norm sensitivity of the primal variable and compute the covariance that the primal variable is required to have so that the privacy leakage of a single iteration of D-ZOA is bounded at each agent. Finally, we prove that the total privacy leakage over all iterations grows sublinearly with the number of ADMM iterations.

**Algorithm 1** Distributed Zeroth-Order ADMM (D-ZOA)

At all agents $k \in \mathcal{V}$, initialize $\boldsymbol{\beta}_k^{(0)} = \mathbf{0}$, $\boldsymbol{\gamma}_k^{(0)} = \mathbf{0}$, and locally run
**for** $m = 1, 2, \ldots, M$ **do**
  Share $\boldsymbol{\beta}_k^{(m-1)}$ with neighbors in $\mathcal{V}_k$
  Update $\boldsymbol{\gamma}_k^{(m)}$ as in (5)
  Initialize $\tilde{\boldsymbol{\beta}}_k^{(0)} = \mathbf{0}$
  **for** $t = 1, 2, \ldots, T$ **do**
    Draw independent $\{\boldsymbol{\nu}_{1,t}^{j,k}\}_{j=1}^{J}, \{\boldsymbol{\nu}_{2,t}^{j,k}\}_{j=1}^{J} \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$
    Set $u_{1,t} = u_1/t$, $u_{2,t} = u_1/(Pt)^2$
    Compute $\mathbf{g}_k^{(t)}$ as in (8) and (7)
    Update $\tilde{\boldsymbol{\beta}}_k^{(t)} = \tilde{\boldsymbol{\beta}}_k^{(t-1)} - \alpha_t \mathbf{g}_k^{(t)}$
  **end for**
  Update $\boldsymbol{\beta}_k^{(m)} = \tilde{\boldsymbol{\beta}}_k^{(T)}$
**end for**

### A. Attack Model and Privacy Concerns

In Algorithm 1, the data stored at each agent, $\mathbf{X}_k$ and $\mathbf{y}_k$, is not shared with any other agent. However, the local estimates $\{\boldsymbol{\beta}_k^{(m)}\}_{k\in\mathcal{V}}$ are exchanged within the local neighborhoods. Therefore, the risk of privacy breach still exists as it has been shown by the model inversion attacks [32].

In this paper, we consider the following attack model. We assume that the adversary is able to access the local estimates $\{\boldsymbol{\beta}_k^{(m)}\}_{k\in\mathcal{V}}$ that are exchanged throughout the intermediate ADMM iterations as well as the final output. The adversary can be either a honest-but-curious member of the network or an external eavesdropper. The adversary's goal is to infer sensitive data of one or more agents by sniffing the communicated information $\{\boldsymbol{\beta}_k^{(m)}\}_{k\in\mathcal{V}}$.

We show that D-ZOA guarantees $(\epsilon, \delta)$-differential privacy as per the below definition since it is intrinsically resistant to such inference attacks.

**Definition 1.** A randomized algorithm $\mathcal{M}$ is $(\epsilon, \delta)$-differentially private if for any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in only one data sample and for any subset of outputs $\mathcal{O} \subseteq \text{range}(\mathcal{M})$, we have

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq e^{\epsilon}\Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta. \tag{10}$$

This means the ratio of the probability distributions of $\mathcal{M}(\mathcal{D})$ and $\mathcal{M}(\mathcal{D}')$ is bounded by $e^{\epsilon}$.

In Definition 1, $\epsilon$ and $\delta$ are privacy parameters indicating the level of privacy preservation ensured by a differentially private algorithm. A better privacy preservation is achieved with smaller $\epsilon$ or $\delta$. On the other hand, low privacy guarantee corresponds to higher values of $\epsilon$, i.e., close to 1. Therefore, it is reasonable to assume that $\epsilon \in (0, 1]$ as in [20], [33].

In the next subsection, we find an approximate distribution of the primal variable, which is needed to prove that (10) holds for our proposed D-ZOA.

### B. Primal Variable Distribution

Due to the stochasticity inherent to the zeroth-order method, its employment for the ADMM primal update produces a perturbed estimate. Therefore, the solution in the primal update step in (4) at agent $k$ using D-ZOA can be modeled as

$$\boldsymbol{\beta}_k^{(m)} = \breve{\boldsymbol{\beta}}_k^{(m)} + \boldsymbol{\xi}_k^{(m)} \tag{11}$$

where $\breve{\boldsymbol{\beta}}_k^{(m)} \in \mathbb{R}^P$ is the exact ADMM primal update and $\boldsymbol{\xi}_k^{(m)} \in \mathbb{R}^P$ is a random variable representing the perturbation. As in [34], the optimality condition for $\breve{\boldsymbol{\beta}}_k^{(m)}$ is given by

$$\begin{aligned}\mathbf{0} \in &\partial f_k(\breve{\beta}_k^{(m)}) + \boldsymbol{\gamma}_k^{(m-1)} + 2\rho|\mathcal{V}_k|\breve{\boldsymbol{\beta}}^{(m)} \\ &- \rho|\mathcal{V}_k|\boldsymbol{\beta}_k^{(m-1)} - \rho\sum_{l\in\mathcal{V}_k}\boldsymbol{\beta}_l^{(m-1)}.\end{aligned} \tag{12}$$

Hence, for any subgradient $f_k'(\breve{\boldsymbol{\beta}}_k^{(m)}) \in \partial f_k(\breve{\boldsymbol{\beta}}_k^{(m)})$, we have

$$\begin{aligned}f_k'(\breve{\beta}_k^{(m)}) = &-\boldsymbol{\gamma}_k^{(m-1)} - 2\rho|\mathcal{V}_k|\breve{\boldsymbol{\beta}}^{(m)} \\ &+ \rho|\mathcal{V}_k|\boldsymbol{\beta}_k^{(m-1)} + \rho\sum_{l\in\mathcal{V}_k}\boldsymbol{\beta}_l^{(m-1)}.\end{aligned} \tag{13}$$

The model (11) represents an implicit primal variable perturbation that can be contrasted with the explicit primal variable perturbation used in [17], [20]. Using (11) and the primal update equation in (13), the ADMM primal update step in (4) can be expressed as

$$\begin{aligned}\breve{\boldsymbol{\beta}}_k^{(m)} = &-\frac{1}{2\rho|\mathcal{V}_k|}f_k'(\breve{\boldsymbol{\beta}}_k^{(m)}) + \frac{1}{2|\mathcal{V}_k|}\Big(|\mathcal{V}_k|\boldsymbol{\beta}_k^{(m-1)} \\ &+ \sum_{l\in\mathcal{V}_k}\boldsymbol{\beta}_l^{(m-1)}\Big) - \frac{1}{2\rho|\mathcal{V}_k|}\boldsymbol{\gamma}_k^{(m-1)}\end{aligned} \tag{14}$$

$$\boldsymbol{\beta}_k^{(m)} = \breve{\boldsymbol{\beta}}_k^{(m)} + \boldsymbol{\xi}_k^{(m)} \tag{15}$$

where $\breve{\boldsymbol{\beta}}_k$ is the local exact primal update at agent $k$ and $\boldsymbol{\xi}_k$ is the local perturbation of $\breve{\boldsymbol{\beta}}_k$ at agent $k$.

To prove that the inherent randomness due to employing a zeroth-order method makes D-ZOA differentially private, we need the knowledge of the probability distribution of the primal variable $\boldsymbol{\beta}_k^{(m)}$. To approximate the probability distribution, in view of (9) and the fact that $\boldsymbol{\beta}_k^{(0)} = \mathbf{0}$, we unfold $\boldsymbol{\beta}_k^{(m)}$ as $\boldsymbol{\beta}_k^{(m)} = -\sum_{t=1}^{T}\alpha_t\mathbf{g}_k^{(t)}$. The stochastic gradient $\mathbf{g}_k^{(t)}$ is the average of $J$ independent random samples $\{\mathbf{g}_{j,k}^{(t)}\}_{j=1}^{J}$ that are functions of the random values $\{\boldsymbol{\nu}_{1,t}^{j,k}\}_{j=1}^{J}$ and $\{\boldsymbol{\nu}_{2,t}^{j,k}\}_{j=1}^{J}$ drawn from the same normal distribution. Therefore, we assume that $\mathbf{g}_k^{(t)}$ is normally distributed with the mean $\boldsymbol{\mu}_k^{(t)}$ and the finite covariance matrix $\boldsymbol{\Psi}_k^{(t)}$, i.e., $\mathbf{g}_k^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Psi}_k^{(t)})$. Thus, the probability distribution of $\boldsymbol{\beta}_k^{(m)}$ is given by the following lemma.

**Lemma 1.** *Given* $\mathbf{g}_k^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Psi}_k^{(t)})$*, the distribution of* $\boldsymbol{\beta}_k^{(m)}$ *is*

$$\boldsymbol{\beta}_k^{(m)} \sim \mathcal{N}\Big(\breve{\boldsymbol{\beta}}_k^{(m)}, \frac{1}{J}\sum_{t=1}^{T}\alpha_t^2\boldsymbol{\Psi}_k^{(t)}\Big). \tag{16}$$

*Proof.* See Appendix A.

In the next subsection, we find an explicit expression for the covariance of $\boldsymbol{\beta}_k^{(m)}$.

The assumption of normal distribution for $\mathbf{g}_k^{(t)}$ is a natural one as $\mathbf{g}_k^{(t)}$ is the average of stochastic variable vectors $\{\mathbf{g}_{j,k}^{(t)}\}_{j=1}^{J}$, which are themselves functions of normally-distributed random variable vectors $\{\boldsymbol{\nu}_{1,t}^{j,k}\}_{j=1}^{J}$ and $\{\boldsymbol{\nu}_{2,t}^{j,k}\}_{j=1}^{J}$. We provide some numerical experiments to explicitly verify this assumption in Section VI.

The assumption is necessary to make the problem of deriving theoretical differential privacy guarantees for the proposed algorithm tractable. Note that our analysis based on this and other assumptions does not result in any deterministic guarantee but yields a probabilistic statement for privacy guarantee by setting a bound on a ratio of probabilities relevant in the concept of differential privacy. Therefore, we do not require perfectly accurate evaluations of the parameters, variables, or statistical models involved in the analysis. Nonetheless, we are cognizant that the reliability of the results highly depends on the accuracy of the underlying assumptions and approximations. Our simulation results in Section VI implicitly corroborate the veracity of our assumptions.

### C. Covariance of the Primal Variable

In this subsection, we derive an explicit expression for the covariance of the primal variable $\boldsymbol{\beta}_k^{(m)}$. This is needed to show that the privacy leakage of any iteration of D-ZOA is bounded at all agents.

To make the problem more tractable, we assume that the entries of the random vector $\boldsymbol{\beta}_k^{(m)}$ are independent of each other and have the same variance [17], [26], [33]. Let us denote the variance of every entry of $\boldsymbol{\xi}_k^{(m)}$ by $\sigma_k^2$. Therefore, in view of Lemma 1, we have

$$\sigma_k^2 = \frac{1}{JP}\sum_{t=1}^{T}\alpha_t^2\mathrm{tr}\left(\boldsymbol{\Psi}_k^{(t)}\right),$$

which can be computed as per the following lemma.

**Lemma 2.** *There exists a constant* $c$ *such that*

$$\sigma_k^2 = \frac{c\alpha_0^2R^2}{JP\log(2P)}\Big(s_1(1+\log P)+s_2\Big) - \frac{4\left\|\boldsymbol{\beta}^c\right\|^2}{TJP}. \tag{17}$$

*where* $s_1 = \sum_{t=1}^{T}t^{-1}$*,* $s_2 = \sum_{t=1}^{T}t^{-1.5}$*, and* $\boldsymbol{\beta}^c$ *is the optimal solution.*

*Proof.* See Appendix B.

In [29], it is shown that $c = 0.5$ is suitable when $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are sampled from a multivariate normal distribution. Note that $s_1$ and $s_2$ grow slowly with $T$. Hence, even for a very large $T$, $s_1$ and $s_2$ have reasonable values. For example, with $T = 2.5\times10^8$, we have $\sum_{t=1}^{T}t^{-1} < 20$. A large $T$ will increase the computational complexity according to the discussions of Section III.D.

### D. $l_2$-norm Sensitivity

In this subsection, we estimate the $l_2$-norm sensitivity of $\breve{\boldsymbol{\beta}}_k^{(m)}$. The $l_2$-norm sensitivity calibrates the magnitude of

the noise by which $\breve{\boldsymbol{\beta}}_k^{(m)}$ has to be perturbed to preserve privacy. Unlike the existing privacy-preserving methods where the noise is added to the output of the algorithm [17], [20], [26], [33], [35], [36], in D-ZOA, the noise is inherent.

We introduce the following assumption that is widely used in the literature, see, e.g., [17], [26], [33].

*Assumption 1:* There exists a constant $c_1$ such that $\|\ell'(\cdot)\| \leq c_1$ where $\ell(\cdot)$ is the loss function defined in Section II.

Similar to the classical methods of differential privacy analysis, e.g., [26], [33], we first define the $l_2$ norm sensitivity. Subsequently, we estimate the $l_2$-norm sensitivity of $\boldsymbol{\beta}_k^{(m)}$.

**Definition 2.** The $l_2$-norm sensitivity of $\breve{\boldsymbol{\beta}}_k^{(m)}$ is defined as

$$\Delta_{k,2} = \max_{\mathcal{D}_k, \mathcal{D}_k'} \left\| \breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k}^{(m)} - \breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k'}^{(m)} \right\| \tag{18}$$

where $\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k}^{(m)}$ and $\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k'}^{(m)}$ denote the local primal variables for two neighboring datasets $\mathcal{D}_k$ and $\mathcal{D}_k'$ differing in only one data sample, i.e., one row of $\mathbf{X}_k$ and the corresponding entry of $\mathbf{y}_k$.

The $l_2$-norm sensitivity of $\breve{\boldsymbol{\beta}}_k^{(m)}$ is an upper bound on $\Delta_{k,2}$ and is computed as in the following lemma.

**Lemma 3.** *Under Assumption 1, the $l_2$-norm sensitivity of $\breve{\boldsymbol{\beta}}_k^{(m)}$ is given by*

$$\Delta_{k,2} = \frac{c_1}{\rho |\mathcal{V}_k| N_k}. \tag{19}$$

*Proof.* See Appendix C.

### *E. Intrinsic $(\epsilon, \delta)$-Differential Privacy Guarantee*

In this subsection, we reveal that the immanent stochasticity imparted by the embedded zeroth-order method makes D-ZOA $(\epsilon, \delta)$-differentially private. We provide an expression relating the primal variable variance, $\sigma_k$, to the privacy parameters $\epsilon$ and $\delta$ as well as an expression for $\epsilon$ relating it to the relevant algorithmic parameters.

We first prove that Algorithm 1 is $(\epsilon, \delta)$-differentially private at each iteration providing a relationship between $\sigma_k$ and $\epsilon$, $\delta$.

**Theorem 1.** *Let $\epsilon \in (0, 1]$ and*

$$\sigma_k = \frac{c_1 \sqrt{2.1 \log(1.25/\delta)}}{\rho |\mathcal{V}_k| N_k \epsilon}. \tag{20}$$

*Under Assumption 1, at each iteration of D-ZOA, $(\epsilon, \delta)$-differential privacy is guaranteed. Specifically, for any neighboring datasets $\mathcal{D}_k$ and $\mathcal{D}_k'$ and any output $\boldsymbol{\beta}_k^{(m)}$, the following inequality holds:*

$$Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k}^{(m)}] \leq e^{\epsilon} Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k'}^{(m)}] + \delta. \tag{21}$$

*Proof.* See Appendix D.

Theorem 1 shows that the primal variable variance is inversely proportional to the privacy parameter $\epsilon$. This implies that a higher variance leads to a smaller $\epsilon$ and higher privacy guarantee. A smaller $\epsilon$ means that the ratio of the probability distributions of $\boldsymbol{\beta}_{k,\mathcal{D}_k}^{(m)}$ and $\boldsymbol{\beta}_{k,\mathcal{D}_k'}^{(m)}$ is smaller and consequently less information is available to any sniffing/spoofing adversary through $\boldsymbol{\beta}_k$ hence the improved privacy [17].

We then introduce the following corollary.

**Corollary 1.** *If $\{\mathbf{g}_{j,k}^{(t)}\}_{j=1}^{J}$ are i.i.d. with $\mathbf{g}_{j,k}^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_k^{(t)}, \Psi_k^{(t)})$, and Assumption 1 holds, we have*

$$\epsilon = \frac{c_1}{\rho |\mathcal{V}_k| N_k} \sqrt{\frac{2.1 JP \log(\frac{1.25}{\delta})}{\frac{cR^2 \alpha_0^2}{\log(2P)} \left(s_1(1 + \log P) + s_2\right) - \frac{4 \|\boldsymbol{\beta}^c\|^2}{T}}}. \tag{22}$$

*Proof.* The proof follows from equating the expressions for $\sigma_k$ in Lemma 2, (17), and Theorem 1, (20), and solving for $\epsilon$. □

The equation (22) shows how the intrinsic privacy preserving property of D-ZOA is affected by various involved parameters. For example, a smaller $J$ results in a smaller $\epsilon$. This is consistent with the fact that a smaller $J$ leads to a higher variance, which yields a higher privacy guarantee due to the inherent randomness brought about by using a zeroth-order method in the inner loop.

The denominator of the second factor in (22) is required to be positive for the factor to be real. We ensure this by setting

$$T > \frac{4 \|\boldsymbol{\beta}^c\|^2 \log(2P)}{cR^2 \alpha_0^2 (s_1(1 + \log(P)) + s_2)}. \tag{23}$$

### *F. Total Privacy Leakage*

In this subsection, we consider the total privacy leakage of the proposed D-ZOA algorithm. Since D-ZOA is an $M$-fold adaptive algorithm, we utilize the results of [30] together with the moments accountant method to evaluate its total privacy leakage. The main result is summarized in the following theorem.

**Theorem 2.** *Let $\epsilon \in (0, 1]$ and*

$$\sigma_k = \frac{c_1 \sqrt{2.1 \log(1.25/\delta)}}{\rho |\mathcal{V}_k| N_k \epsilon}. \tag{24}$$

*Under Assumption 1, Algorithm 1 guarantees $(\bar{\epsilon}, \delta)$-differential privacy where*

$$\bar{\epsilon} = \epsilon \sqrt{\frac{M \log(1/\delta)}{1.05 \log(1.25/\delta)}}. \tag{25}$$

*Proof.* The proof is obtained by using the log moments of the privacy loss and their linear composability in the same way as in [26, Theorem 2]. □

## V. Convergence Analysis and Privacy-Accuracy Trade-off

We establish the convergence of D-ZOA to a near-optimal solution by corroborating that both inner and outer loops of the algorithm converge.

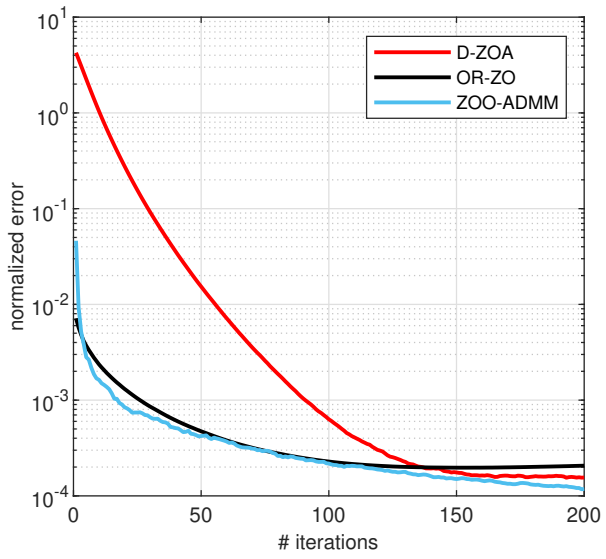



Fig. 1. The normalized errors of D-ZOA, OR-ZO [29], and ZOO-ADMM [14] versus the iteration number.

### A. Convergence of the Inner Loop

The convergence of the inner loop can be verified following [29, Theorem 2], i.e., it can be shown that, if $\mathcal{F}_k(\cdot)$ is Lipschitz-continuous with the Lipschitz constant $L$, there exists a constant $c$ such that, for each $T$ representing a fixed number of inner-loop iterations, the following inequality holds:

$$\begin{aligned}&\mathbb{E}[\mathcal{F}_k(\hat{\boldsymbol{\beta}}_k^{(T)}) - \mathcal{F}_k(\boldsymbol{\beta}_k^*)]\\ \leq& c\frac{RL\sqrt{P}}{\sqrt{T}}\Big(\max\{\alpha_0, \alpha_0^{-1}\}\sqrt{\log(2P)} + \frac{u_1\log(2T)}{\sqrt{T}}\Big)\end{aligned} \tag{26}$$

where $\boldsymbol{\beta}_k^*$ is the minimizer to (4) and

$$\hat{\boldsymbol{\beta}}_k^{(T)} = \frac{1}{T}\sum_{t=1}^{T}\tilde{\boldsymbol{\beta}}_k^{(t)}.$$

The inequality in (26) implies that the two-point stochastic gradient algorithm constituting the inner loop converges at a rate of $\mathcal{O}(\sqrt{P/T})$. In [29], it is shown that $c = 0.5$ is suitable when $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ are sampled from a normal distribution. The function $\mathcal{F}_k(\cdot)$ is the sum of $f_k(\cdot)$, which is assumed to be closed and convex, and $h_k(\cdot)$ that is also both closed and convex since it is a positive definite quadratic function. Hence, the function $\mathcal{F}_k(\cdot)$ is closed and convex in addition to being Lipschitz-continuous [37]. Therefore the convergence result in (26) follow from [29].

### B. Convergence of the Outer Loop

The convergence of the outer loop can be proven by verifying the convergence of a fully-distributed ADMM with perturbed primal updates. To present the convergence result, we rewrite (3) in the matrix form. By defining $\mathbf{w} \in \mathbb{R}^{KP}$ concatenating all $\boldsymbol{\beta}_k$ and $\mathbf{z} \in \mathbb{R}^{2EP}$ concatenating all $\mathbf{z}_k^l$, (3) can be written as

$$\begin{aligned}\min_{\mathbf{w},\mathbf{z}} \quad & f(\mathbf{w})\\ \text{s.t.} \quad & \mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{z} = \mathbf{0}\end{aligned} \tag{27}$$

where

$$f(\mathbf{w}) = \sum_{k=1}^{K} f_k(\boldsymbol{\beta}_k),$$

$\mathbf{A} = [\mathbf{A}_1^\mathsf{T}, \mathbf{A}_2^\mathsf{T}]^\mathsf{T}$, and $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{2EP\times KP}$ are both composed of $2E \times K$ blocks of $P \times P$ matrices. If $(k,l) \in \mathcal{E}$ and $\mathbf{z}_k^l$ is the $q$th block of $\mathbf{z}$, then the $(q,k)$th block of $\mathbf{A}_1$ and the $(q,l)$th block of $\mathbf{A}_2$ are the identity matrix $\mathbf{I}_P$. Otherwise, the corresponding blocks are $\mathbf{0}_{P\times P}$. Furthermore, we have

$$\mathbf{B} = [-\mathbf{I}_{2EP}, -\mathbf{I}_{2EP}]^\mathsf{T}.$$

To facilitate the representation, we also define the following matrices

$$\begin{aligned}\mathbf{M}_+ &= \mathbf{B}_1^\mathsf{T} + \mathbf{B}_2^\mathsf{T}\\ \mathbf{M}_- &= \mathbf{B}_1^\mathsf{T} - \mathbf{B}_2^\mathsf{T}\\ \mathbf{L}_+ &= 0.5\mathbf{M}_+\mathbf{M}_+^\mathsf{T}\\ \mathbf{L}_- &= 0.5\mathbf{M}_-\mathbf{M}_-^\mathsf{T}\\ \mathbf{H} &= 0.5(\mathbf{L}_+ + \mathbf{L}_-)\\ \mathbf{Q} &= \sqrt{0.5\mathbf{L}_-}.\end{aligned}$$

We construct the auxiliary sequence

$$\mathbf{r}^{(m)} = \sum_{s=0}^{m}\mathbf{Q}\mathbf{w}^{(s)}$$

and define the auxiliary vector $\mathbf{q}^{(m)}$ and the auxiliary matrix $\mathbf{G}$ as

$$\mathbf{q}^{(m)} = \begin{bmatrix}\mathbf{r}^{(m)}\\ \mathbf{w}^{(m)}\end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix}\rho\mathbf{I}_P & \mathbf{0}_{P\times P}\\ \mathbf{0}_{P\times P} & \rho\frac{\mathbf{L}_+}{2}\end{bmatrix}. \tag{28}$$

The convergence results of [38], [20], and [39] can now be adapted to D-ZOA as per the following theorem that also provides an explicit privacy-accuracy trade-off.

**Theorem 3.** *For any $M > 0$, we have*

$$\begin{aligned}&\mathbb{E}[f(\hat{\mathbf{w}}^{(M)}) - f(\mathbf{w}^*)]\\ &\leq \frac{\|\mathbf{q}^{(0)} - \mathbf{q}\|_{\mathbf{G}}^2}{M} + \frac{2.1c_1^2P\log(1.25/\delta)\lambda_{max}^2(\mathbf{L}_+)}{2\rho|\mathcal{V}_k|^2N_k^2\epsilon^2\lambda_{min}(\mathbf{L}_-)}\end{aligned} \tag{29}$$

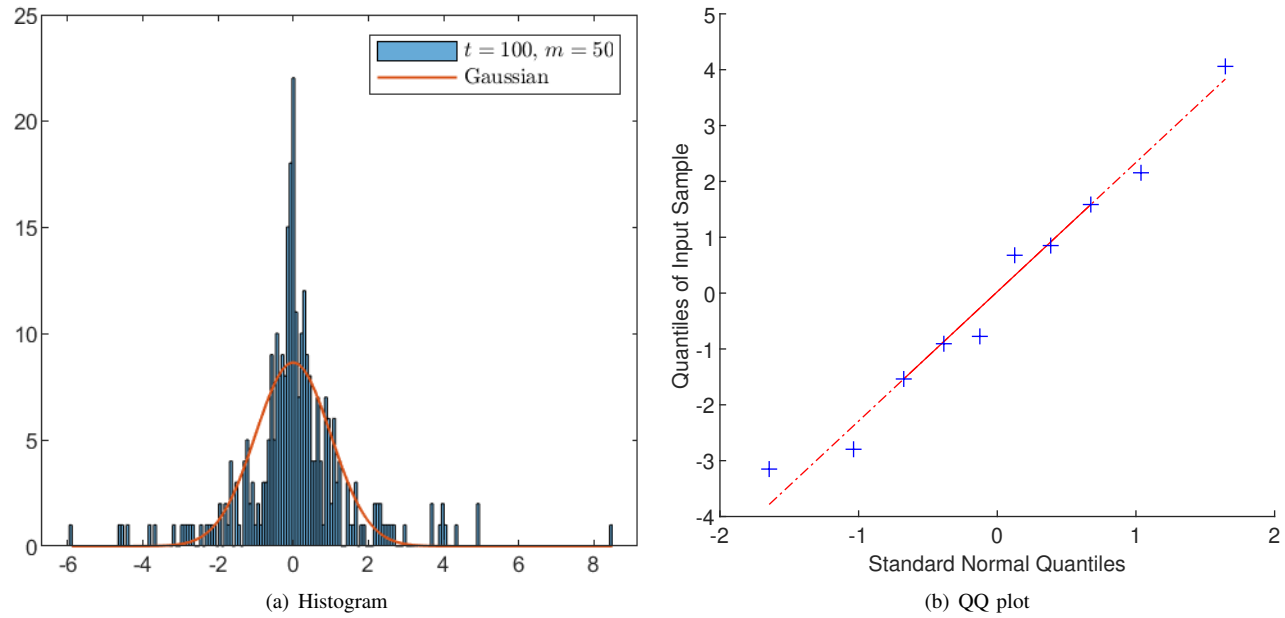
*where $\mathbf{q} = [\mathbf{r}^\mathsf{T}, (\mathbf{w}^*)^\mathsf{T}]^\mathsf{T}$ and*

$$\hat{\mathbf{w}}^{(M)} = \frac{1}{M}\sum_{m=1}^{M}\breve{\mathbf{w}}^{(m)}.$$

*Proof.* See Appendix E. □

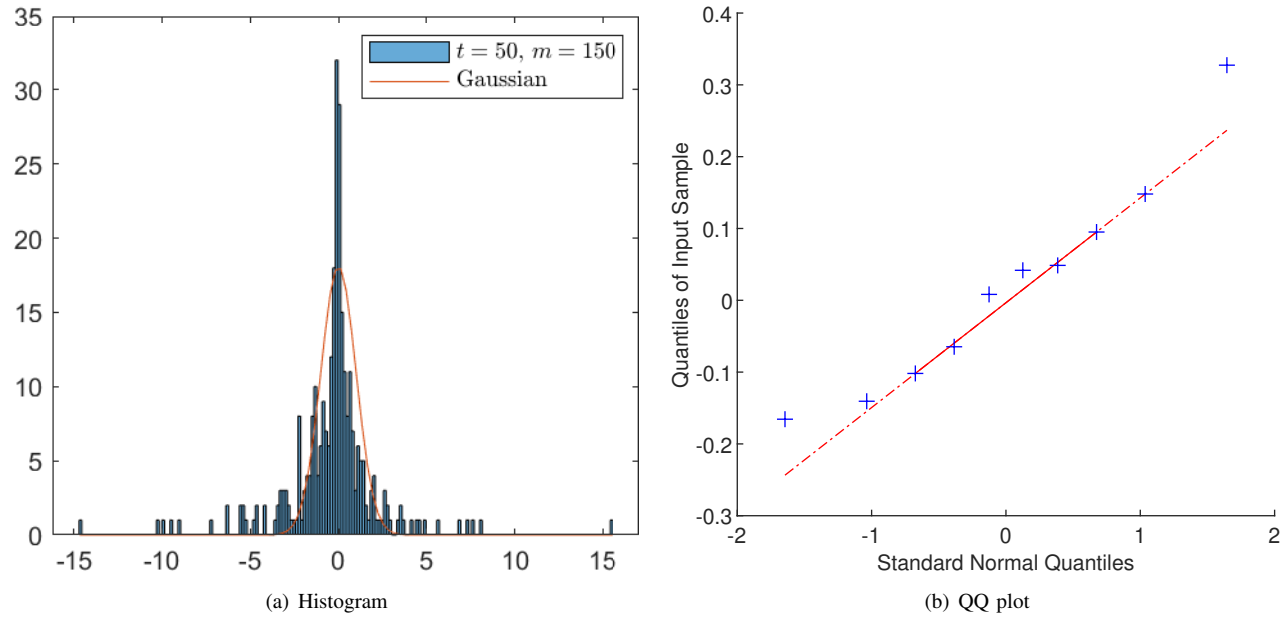
Theorem 3 shows that D-ZOA reaches a neighborhood of the optimal (centralized) solution with the size of the neighborhood determined by the privacy-parameter $\epsilon$. This discloses a privacy-accuracy trade-off offered by D-ZOA. When the privacy guarantee is stronger (smaller $\epsilon$ and $\delta$), the accuracy is lower.

Note that we do not need to solve the minimization problem in the ADMM primal update step of the outer loop with high accuracy [26], [34]. We perform the ADMM primal update step in the outer loop after obtaining an inexact solution to (4). Therefore, we select the number of inner loop iterations $T$ as the minimum number of iterations satisfying (23) and




(a) Histogram (b) QQ plot

Fig. 2. The histogram and the QQ plot of $\mathbf{g}_k^{(t)}$ at agent 2, the inner loop iteration $t = 100$, and the outer loop iteration $m = 50$.




(a) Histogram (b) QQ plot

Fig. 3. The histogram and the QQ plot of $\mathbf{g}_k^{(t)}$ at agent 2, the inner loop iteration $t = 50$, and the outer loop iteration $m = 150$.

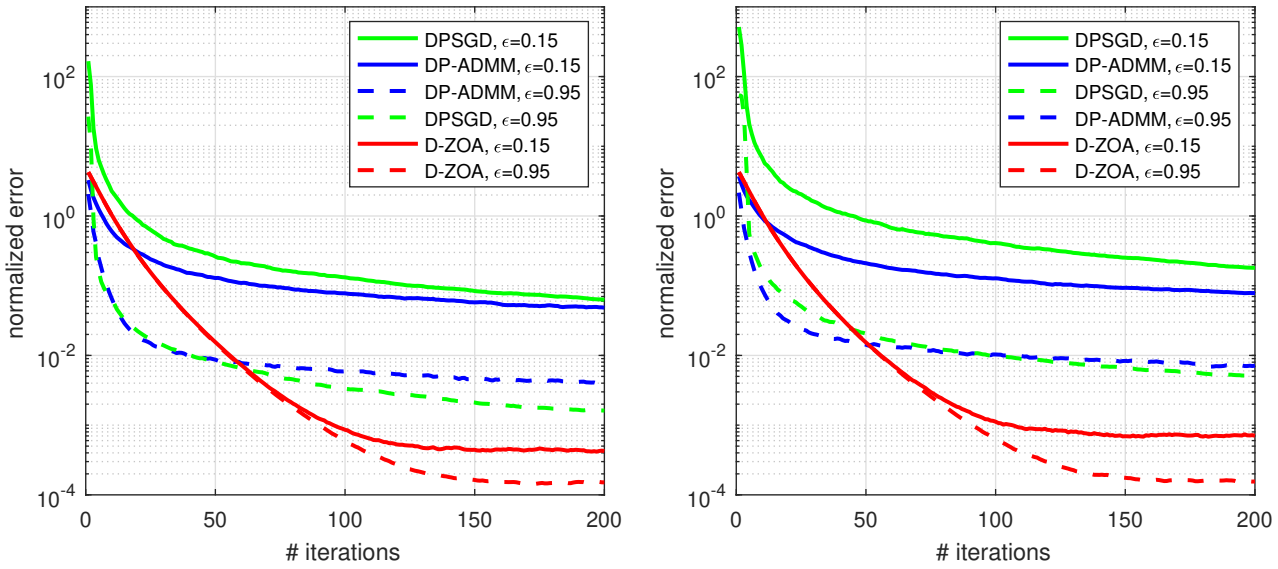
entailing an accuracy that is sufficient to ensure convergence of ADMM according to [29]. In fact, $T$ should be chosen as low as possible within the above-mentioned constraints to minimize the computational complexity according to the discussions of Section III.D.

## VI. Simulations

In this section, we present some simulation examples to evaluate the performance of the proposed D-ZOA algorithm in comparison with the most relevant state-of-the-art algorithms as well as the privacy-accuracy trade-off offered by the proposed algorithm.

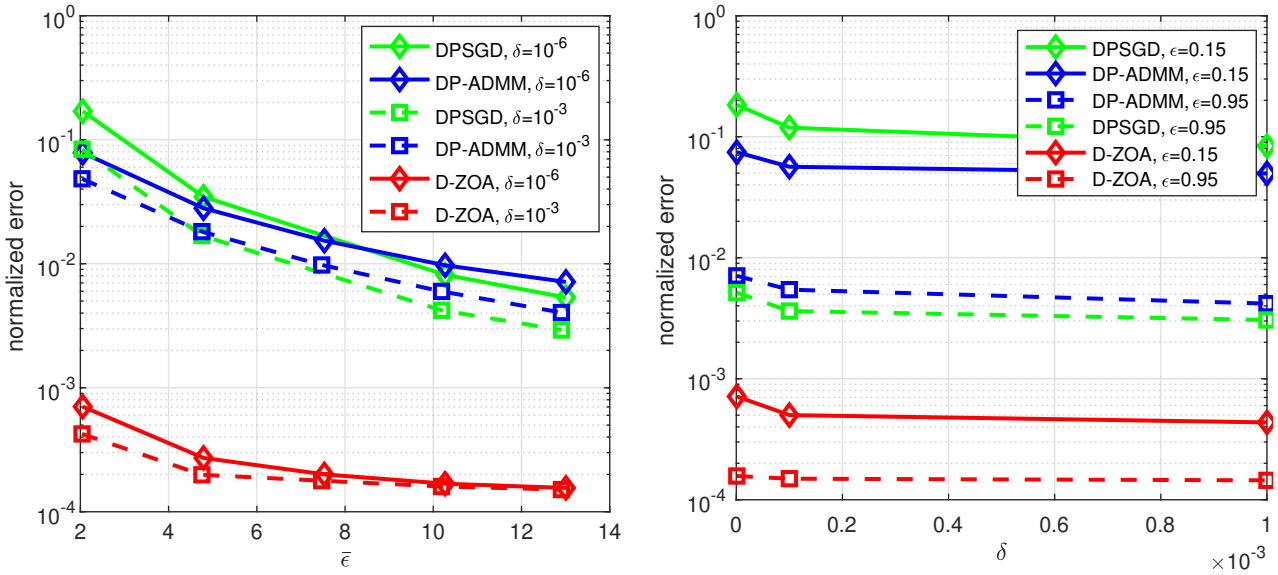
We first benchmark D-ZOA against two popular existing algorithms for zeroth-order-based optimization, which have originally been designed for centralized settings, i.e., those proposed in [14] and [29] and called zeroth-order online ADMM (ZOO-ADMM) and optimal-rate zeroth-order (OR-ZO) algorithm, respectively. Then, we illustrate two sets of example histograms and QQ plots to verify that the entries of the gradient vector $\mathbf{g}_k^{(t)}$ are normally distributed.

Next, we benchmark D-ZOA against some existing baseline differentially-private algorithms: the ADMM algorithm with primal variable perturbation (PVP) proposed in [17], [26], the ADMM with dual variable perturbation (DVP) proposed




(a) $\delta = 10^{-3}$ (b) $\delta = 10^{-6}$

Fig. 4. Normalized error of DPSGD, DP-ADMM, and D-ZOA for two values of $\epsilon$ and fixed $\delta$ for ERM with $\ell_1$-norm regularization.




(a) $\delta = 10^{-6}$ and $\delta = 10^{-3}$ (b) $\epsilon = 0.15$ and $\epsilon = 0.95$

Fig. 5. Privacy-accuracy trade-off of DPSGD, DP-ADMM, and D-ZOA for ERM with $\ell_1$-norm regularization.

in [17], the ADMM-based differentially private distributed algorithm called DP-ADMM and proposed in [26], and the distributed subgradient method proposed in [9] that is customized to include differential privacy (DPSG). Note that DP-ADMM is the only existing privacy-preserving distributed algorithm for non-smooth objectives.

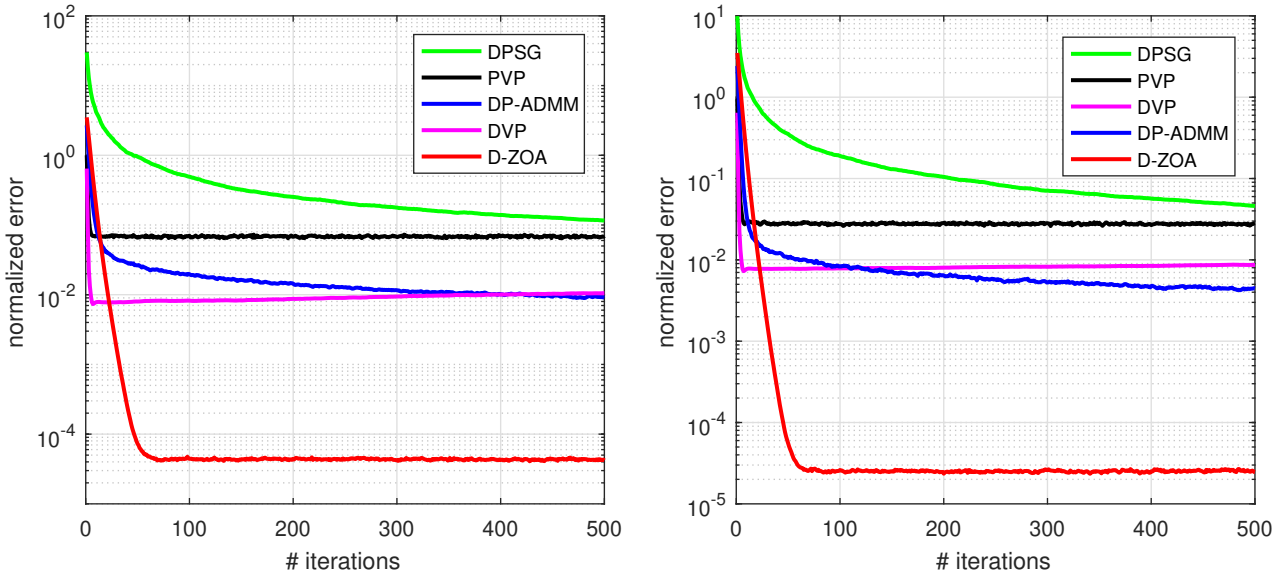
As for the applications, we consider a distributed version of the empirical risk minimization problem with an $\ell_1$-norm regularization (lasso penalty) and an $\ell_2$-norm regularization (ridge penalty) [40].

The network-wide observations are represented by a design matrix $\mathbf{X} \in \mathbb{R}^{N\times P}$ and a response vector $\mathbf{y} \in \mathbb{R}^{N\times 1}$ where $N$ is the number of data samples and $P$ is the number of features in each sample. The matrix $\mathbf{X}$ consists of $K$ submatrices $\mathbf{X}_k$, i.e., $\mathbf{X} = [\mathbf{X}_1^\mathsf{T}, \mathbf{X}_2^\mathsf{T}, \ldots, \mathbf{X}_K^\mathsf{T}]^\mathsf{T}$, and the vector $\mathbf{y}$ consists of $K$ subvectors $\mathbf{y}_k$, i.e., $\mathbf{y} = \left[\mathbf{y}_1^\mathsf{T}, \mathbf{y}_2^\mathsf{T}, \ldots, \mathbf{y}_K^\mathsf{T}\right]^\mathsf{T}$, as the data is distributed among the agents and each agent $k$ holds its respective $\mathbf{X}_k \in \mathbb{R}^{N_k\times P}$ and $\mathbf{y}_k \in \mathbb{R}^{N_k\times 1}$ where $N = \sum_{k=1}^{K} N_k$. The parameter vector that establishes a linear regression between $\mathbf{X}$ and $\mathbf{y}$ is $\boldsymbol{\beta} \in \mathbb{R}^{P\times 1}$. In the centralized approach, a lasso estimate of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta}^c = \arg\min_{\boldsymbol{\beta}} \{\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2 + \eta \|\boldsymbol{\beta}\|_1\} \tag{30}$$

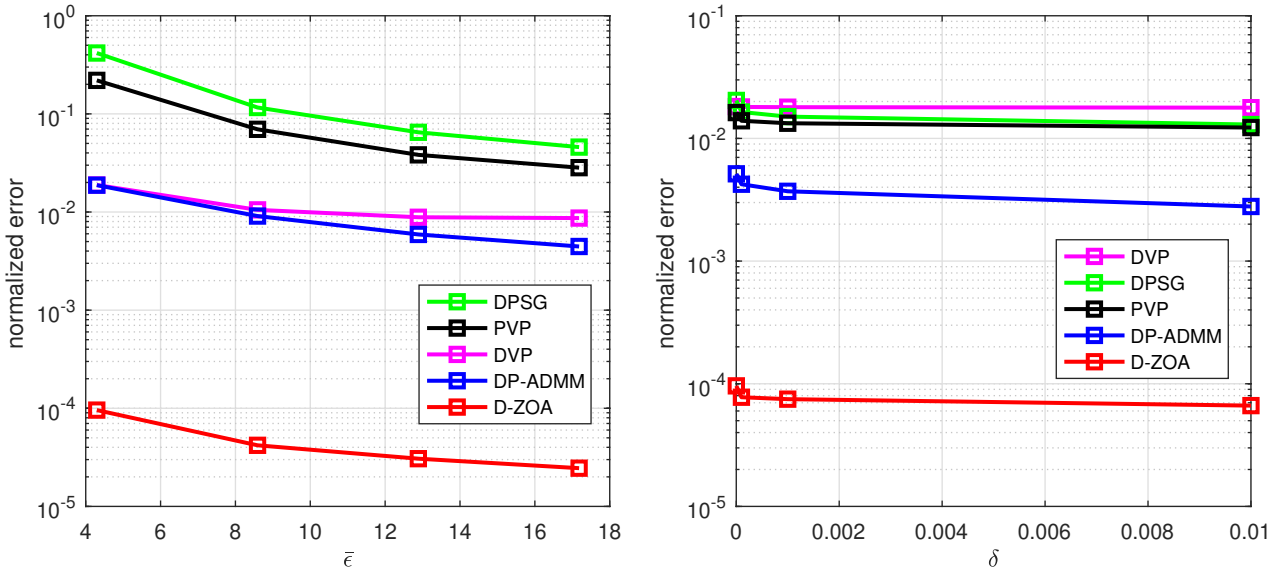
while a ridge estimate of $\boldsymbol{\beta}$ is given by

$$\boldsymbol{\beta}^c = \arg\min_{\boldsymbol{\beta}} \{\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2 + \eta \|\boldsymbol{\beta}\|^2\}. \tag{31}$$




(a) $\epsilon = 0.40$ and $\delta = 10^{-3}$ (b) $\epsilon = 0.80$ and $\delta = 10^{-3}$

Fig. 6. Normalized error of DPSGD, PVP, DP-ADMM, DVP, and D-ZOA for two values of $\epsilon$ and fixed $\delta$ for ERM with $\ell_2$-norm regularization.




(a) $\delta = 10^{-3}$ (b) $\epsilon = 0.95$

Fig. 7. Privacy-accuracy trade-off of DPSGD, PVP, DP-ADMM, DVP, and D-ZOA for ERM with $\ell_2$-norm regularization.

In the distributed setting, we solve problem (2) with

$$\sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \boldsymbol{\beta}_k) = \|\mathbf{X}_k \boldsymbol{\beta}_k - \mathbf{y}_k\|^2 \tag{32}$$

and $R(\boldsymbol{\beta}_k) = \|\boldsymbol{\beta}_k\|_1$ for the lasso penalty. For the ridge penalty, we have $R(\boldsymbol{\beta}_k) = \|\boldsymbol{\beta}_k\|^2$.

We assess the performance of the D-ZOA algorithm over a network of $K = 5$ agents with edge set $\mathcal{E} = \{e_{12}, e_{14}, e_{23}, e_{34}, e_{45}\}$. The number of samples at each agent is set to $N_k = 20\ \forall k \in \mathcal{V}$ and the total number of samples is $N = 100$. The number of features in each sample is $P = 10$. For each agent $k \in \mathcal{V}$, we create a $2P \times P$ local observation matrix $\mathbf{X}_k$ whose entries are i.i.d. zero-mean unit-variance Gaussian random variables. The response vector $\mathbf{y}$ is synthesized as $\mathbf{y} = \mathbf{X}\boldsymbol{\omega} + \boldsymbol{\psi}$ where $\boldsymbol{\omega} \in \mathbb{R}^P$ and $\boldsymbol{\psi} \in \mathbb{R}^M$ are random vectors with distributions $\mathcal{N}(\mathbf{0}, \mathbf{I}_P)$ and $\mathcal{N}(\mathbf{0}, 0.1\mathbf{I}_N)$, respectively. The data are preprocessed by normalizing the columns of $\mathbf{X}$ to guarantee that the maximum value of each column is 1 and by normalizing the rows to enforce their $l_2$-norm to be less than 1 as in [26]. This is motivated by the need for homogeneous scaling of the features. Therefore, we have $c_1 = 1$. The regularization parameter is set to $\eta = 1$ and the penalty parameter is set to $\rho = 4$. The number of iterations of the ADMM outer loop is set to 200. For the inner loop, the number of iterations is set to 100 and the smoothing constant $u_1$ to 1. We set $\alpha_0 = 0.54$ according to [41] and

calculate $J$ from equation (22) by fixing $\epsilon$, solving for $J$ and rounding the solution to the nearest integer. Performance of D-ZOA is evaluated using the normalized error between the centralized solutions $\boldsymbol{\beta}^c$ as per (31) and the local estimates. It is defined as $\sum_{k=1}^{K} \|\boldsymbol{\beta}_k - \boldsymbol{\beta}^c\|^2 / \|\boldsymbol{\beta}^c\|^2$ where $\boldsymbol{\beta}_k$ denotes the local estimate at agent $k$. The centralized solution $\boldsymbol{\beta}^c$ is computed using the convex optimization toolbox CVX [42]. Results are obtained by averaging over 100 independent trials.

In Fig. 1, we compare the performance of the proposed D-ZOA algorithm with that of two existing zeroth-order-based algorithms, i.e., those proposed in [14] and [29]. The simulation results show that the steady-state normalized error of D-ZOA is comparable to those of these algorithms, even though they are designed for centralized processing. The centralized algorithms converge faster than Z-DOA since, unlike our fully-distributed D-ZOA, they have all data concentrated at a central processing hub and do not rely on diffusing information across the network by sharing the local estimates within each agent's neighborhood. Note that the notion of iteration is essentially different for each algorithm whose learning curve is shown in Fig. 1. Thus, we provide the learning curve plots in Fig. 1 only to examine how D-ZOA performs in comparison with the existing zeroth-order optimization algorithms notwithstanding the underlying fundamental differences.

In Figs. 2 and 3, we provide two sets of histograms and QQ plots for an arbitrary entry of the stochastic gradient vector $\mathbf{g}_k^{(t)}$, i.e., the one corresponding to agent 2, and different inner and outer loop iterations, i.e., $t = 100$, $t = 50$, and $m = 50$, $m = 150$. The plots help us verify that the entries of $\mathbf{g}_k^{(t)}$ are normally distributed hence attest to the validity of our related assumption in Section IV-B. Meanwhile, we made similar observations with other entries of $\mathbf{g}_k^{(t)}$ and iteration numbers. However, due to space limitation, we only provide Figs. 2 and 3 as examples.

We first benchmark our D-ZOA in the case of the ERM with the lasso penalty. Since PVP and DVP cannot be employed when the objective function is non-smooth, we benchmark our algorithm only with DP-ADMM and DPSGD similar to [26]. In Fig. 4, we plot the normalized error versus the outer loop iteration index for D-ZOA, DP-ADMM, and DPSGD. The plots show that all algorithms converge for two different values of $\epsilon$ and $\delta$. In all plots, accuracy improves as $\epsilon$ increases. This is consistent with both Theorem 3 and [26, Theorem 3]. The hyper-parameters in DP-ADMM and DPSGD are tuned to achieve the best accuracy and convergence rate. However, D-ZOA has higher accuracy than DP-ADMM and DPSGD.

In Fig. 5, we illustrate the privacy-accuracy trade-off for D-ZOA, DP-ADMM, and DPSGD. The figures show that D-ZOA, DP-ADMM, and DPSGD achieve higher accuracy with larger $\epsilon$ and $\delta$. In Fig. 5(a), we show the normalized error versus the privacy parameter $\bar{\epsilon}$ as given in (25) for $\delta = 10^{-6}$ and $\delta = 10^{-3}$. We observe that D-ZOA outperforms both DP-ADMM and DPSGD in terms of accuracy likely due to its intrinsic privacy-preserving properties. Fig. 5(b) also attests to the superiority of D-ZOA over DP-ADMM and DPSGD when $\epsilon = 0.15$ and $\epsilon = 0.95$ and $\delta$ varies between $10^{-6}$ and $10^{-2}$.

We also evaluate the performance of the D-ZOA algorithm in comparison with the considered benchmark algorithms for the ERM with the ridge penalty. In Fig. 6, we plot the normalized error versus the outer loop iteration index for D-ZOA, DP-ADMM, DPSGD, PVP, and DVP. The plots show that D-ZOA outperforms all other considered algorithms in terms of accuracy for different values of $\epsilon$.

In Fig. 7, we demonstrate the privacy-accuracy trade-off for D-ZOA, DP-ADMM, DPSGD, PVP, and DVP. As expected, smaller values of the privacy parameters $\epsilon$ and $\delta$ lead to lower accuracy. However, D-ZOA outperforms all the other approaches in terms of accuracy due to its intrinsic privacy-preserving properties.

In the considered applications of ERM with lasso and ridge penalty, we make the following observations regarding the complexity-accuracy trade-off of D-ZOA and the baseline algorithms. D-ZOA outperforms all the baseline algorithms in terms of accuracy by roughly two orders of magnitude. However, D-ZOA has a relatively high computational complexity due to its inner loop that is run at every agent $k$ and every ADMM iteration. Since the number of arithmetic operations required to evaluate the objective function is $\mathcal{O}(PN_k)$, calculation of (8) needs $\mathcal{O}(JPN_k)$ operations and, therefore, D-ZOA requires $\mathcal{O}(TJPN_k)$ operations to perform (4). The baseline algorithms have the following computational complexities: $\mathcal{O}(P^2N_k)$ for DP-ADMM and DPSGD, and $\mathcal{O}(P^2N_k + P^3)$ for PVP and DVP.

## VII. Conclusion

We proposed an intrinsically privacy-preserving consensus-based algorithm for solving a class of distributed regularized ERM problems where first-order information is hard or even impossible to obtain. We recast the original problem into an equivalent constrained optimization problem whose structure is suitable for distributed implementation via ADMM. We employed a zeroth-order method, known as the two-point stochastic-gradient algorithm, to minimize the augmented Lagrangian in the primal update step. We proved that the inherent randomness due to employing the zeroth-order method can adequately make the D-ZOA algorithm intrinsically privacy-preserving. In addition, we used the moments accountant method to show that the total privacy leakage of D-ZOA grows sublinearly with the number of ADMM iterations. We verified the convergence of D-ZOA to a near-optimal solution as well as studying its privacy-preserving properties through both theoretical analysis and numerical simulations.

## Appendix A
## Proof of Lemma 1

*Proof.* We prove this lemma in two steps. First, we prove that $\mathbb{E}[\boldsymbol{\beta}_k^{(m)}] = \breve{\boldsymbol{\beta}}_k^{(m)}$. Then, we calculate the covariance of $\boldsymbol{\beta}_k^{(m)}$.

We prove that $\mathbb{E}[\boldsymbol{\beta}_k^{(m)}] = \breve{\boldsymbol{\beta}}_k^{(m)}$ by induction over $m$.

*Base case:* Since $\boldsymbol{\beta}_k^{(0)} = \breve{\boldsymbol{\beta}}_k^{(0)} = \mathbf{0}$, we have $\mathbb{E}[\boldsymbol{\beta}_k^{(0)}] = \breve{\boldsymbol{\beta}}_k^{(0)}$.

*Induction step:* We assume that $\mathbb{E}[\boldsymbol{\beta}_k^{(m-1)}] = \breve{\boldsymbol{\beta}}_k^{(m-1)}$ as the induction hypothesis. Considering (14) and (11), we have

$$\begin{aligned}\mathbb{E}[\boldsymbol{\beta}_k^{(m)}] &= \mathbb{E}[\breve{\boldsymbol{\beta}}_k^{(m)}] + \mathbb{E}[\boldsymbol{\xi}_k^{(m)}] \\ &= -\frac{1}{2\rho|\mathcal{V}_k|} f_k'(\breve{\boldsymbol{\beta}}_k^{(m)}) + \frac{1}{2|\mathcal{V}_k|}\Big(|\mathcal{V}_k|\breve{\boldsymbol{\beta}}_k^{(m-1)} \\ &\quad + \sum_{l\in\mathcal{V}_k} \breve{\boldsymbol{\beta}}_l^{(m-1)}\Big) - \frac{1}{2\rho|\mathcal{V}_k|}\boldsymbol{\gamma}_k^{(m-1)} + \mathbb{E}[\boldsymbol{\xi}_k^{(m)}] \\ &= -\frac{1}{2\rho|\mathcal{V}_k|}\mathbb{E}[f_k'(\breve{\boldsymbol{\beta}}_k^{(m)})] \\ &\quad + \frac{1}{2|\mathcal{V}_k|}\Big(|\mathcal{V}_k|\mathbb{E}[\boldsymbol{\beta}_k^{(m-1)}] + \sum_{l\in\mathcal{V}_k}\mathbb{E}[\boldsymbol{\beta}_l^{(m-1)}]\Big) \\ &\quad - \frac{1}{2\rho|\mathcal{V}_k|}\mathbb{E}[\boldsymbol{\gamma}_k^{(m-1)}] + \mathbb{E}[\boldsymbol{\xi}_k^{(m)}] \\ &= \mathbb{E}[\boldsymbol{\beta}_k^{(m)}] + \mathbb{E}[\boldsymbol{\xi}_k^{(m)}],\end{aligned}$$

which implies that $\mathbb{E}[\boldsymbol{\xi}_k^{(m)}] = \mathbf{0}$. Therefore, $\mathbb{E}[\boldsymbol{\beta}_k^{(m)}] = \breve{\boldsymbol{\beta}}_k^{(m)}$.

Since we have $\mathbf{g}_k^{(t)} \sim \mathcal{N}(\boldsymbol{\mu}_k^{(t)}, \boldsymbol{\Psi}_k^{(t)})$ and $\boldsymbol{\beta}_k^{(m)} = -\sum_{t=1}^T \alpha_t \mathbf{g}_k^{(t)}$, in view of the additive property of the normal distribution, $\boldsymbol{\beta}_k^{(m)}$ is normally distributed with the mean $\breve{\boldsymbol{\beta}}_k^{(m)}$ and the covariance

$$\mathrm{cov}[\boldsymbol{\beta}_k^{(m)}] = \frac{1}{J}\sum_{t=1}^T \alpha_t^2 \boldsymbol{\Psi}_k^{(t)}. \tag{33}$$

□

## Appendix B
## Proof of Lemma 2

*Proof.* It is easy to verify that

$$\mathrm{tr}(\boldsymbol{\Psi}_k^{(t)}) = \mathbb{E}\left[\left\|\mathbf{g}_{j,k}^{(t)}\right\|^2\right] - \left\|\boldsymbol{\mu}_k^{(t)}\right\|^2. \tag{34}$$

By [29, Lemma 2], there exists a constant $c$ such that

$$\mathbb{E}\left[\left\|\mathbf{g}_{j,k}^{(t)}\right\|^2\right] \le cL^2P\Big(\sqrt{\frac{u_{2,t}}{u_{1,t}}}P + 1 + \log(P)\Big). \tag{35}$$

Since $u_{2,t}/u_{1,t} = P^{-2}t^{-1}$, we have

$$\mathbb{E}\left[\left\|\mathbf{g}_{j,k}^{(t)}\right\|^2\right] \le cL^2P\Big(\frac{1}{\sqrt{t}} + 1 + \log(P)\Big). \tag{36}$$

In addition, from $\boldsymbol{\beta}_k^{(m)} = -\sum_{t=1}^T \alpha_t \mathbf{g}_k^{(t)}$ and (11), we have

$$\breve{\boldsymbol{\beta}}_k^{(m)} = -\sum_{t=1}^T \alpha_t \boldsymbol{\mu}_k^{(t)}. \tag{37}$$

Taking the Euclidean norm of both sides in (37) and using the triangle inequality, we have

$$\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\| = \left\|-\sum_{t=1}^T \alpha_t \boldsymbol{\mu}_k^{(t)}\right\| \le \sum_{t=1}^T |\alpha_t| \left\|\boldsymbol{\mu}_k^{(t)}\right\|. \tag{38}$$

Squaring both sides of (38) and using the Cauchy-Schwarz inequality, we get

$$\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\|^2 \le \Big(\sum_{t=1}^T |\alpha_t| \left\|\boldsymbol{\mu}_k^{(t)}\right\|\Big)^2 \le T\sum_{t=1}^T |\alpha_t|^2 \left\|\boldsymbol{\mu}_k^{(t)}\right\|^2 \tag{39}$$

and consequently

$$-\frac{1}{JP}\sum_{t=1}^T \alpha_t^2 \left\|\boldsymbol{\mu}_k^{(t)}\right\|^2 \le -\frac{1}{TJP}\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\|^2. \tag{40}$$

Using (36), (40), and the definition of $\alpha_t$ after (9), we have

$$\begin{aligned}&\frac{1}{JP}\sum_{t=1}^{T-1}\alpha_t^2 \mathrm{tr}(\boldsymbol{\Psi}_k^{(t)}) \\ \le&\frac{1}{JP}\sum_{t=1}^{T-1}\alpha_t^2 cL^2P\Big(\frac{1}{\sqrt{t}} + 1 + \log(P)\Big) \\ &- \frac{1}{JP}\sum_{t=1}^{T-1}\alpha_t^2\left\|\boldsymbol{\mu}_k^{(t)}\right\|^2 \\ =&\frac{1}{JP}\frac{c\alpha_0^2R^2}{\log(2P)}\Big(\sum_{t=1}^T\frac{1}{t\sqrt{t}} + (1+\log(P))\sum_{t=1}^T\frac{1}{t}\Big) \\ &- \frac{1}{TJP}\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\|^2.\end{aligned} \tag{41}$$

Defining $s_1 = \sum_{t=1}^{T-1} t^{-1}$ and $s_2 = \sum_{t=1}^{T-1} t^{-1.5}$, (41) simplifies to

$$\begin{aligned}&\frac{1}{JP}\sum_{t=1}^T \alpha_t^2 \mathrm{tr}(\boldsymbol{\Psi}_k^{(t)}) \\ &\le \frac{c\alpha_0^2R^2}{JP\log(2P)}\Big(s_1(1+\log(P)) + s_2\Big) - \frac{\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\|^2}{TJP}.\end{aligned} \tag{42}$$

Considering that the algorithm converges as proven in Section V, i.e., $\breve{\boldsymbol{\beta}}_k^{(m)} \to \boldsymbol{\beta}^c$ as $m \to \infty$, $\breve{\boldsymbol{\beta}}_k^{(0)} = \mathbf{0}$, and the triangle inequality, for $m > 0$ we have

$$\left|\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\| - \|\boldsymbol{\beta}^c\|\right| \le \left\|\breve{\boldsymbol{\beta}}_k^{(m)} - \boldsymbol{\beta}^c\right\| \le \|\boldsymbol{\beta}^c\|, \tag{43}$$

which implies $\left\|\breve{\boldsymbol{\beta}}_k^{(m)}\right\| \le 2\|\boldsymbol{\beta}^c\|$. Therefore, we obtain

$$\begin{aligned}&\frac{1}{JP}\sum_{t=1}^T \alpha_t^2 \mathrm{tr}(\boldsymbol{\Psi}_k^{(t)}) \\ &= \frac{c\alpha_0^2R^2}{JP\log(2P)}\Big(s_1(1+\log(P)) + s_2\Big) - \frac{4\|\boldsymbol{\beta}^c\|^2}{TJP}.\end{aligned} \tag{44}$$

□

and consequently (17).

## APPENDIX C
## PROOF OF LEMMA 3

*Proof.* From the adopted exact primal update equation (14), we obtain

$$\begin{aligned}\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k}^{(m)} =& -\frac{0.5}{\rho|\mathcal{V}_k|}\Big(\frac{1}{N_k}\sum_{j=1}^{N_k}\ell'(\mathbf{x}_{k,j},y_{k,j};\breve{\boldsymbol{\beta}}_k)+\boldsymbol{\gamma}_k^{(m-1)}\Big)\\ &+\frac{0.5}{|\mathcal{V}_k|}\Big(\breve{\boldsymbol{\beta}}_k^{(m-1)}+\sum_{l\in\mathcal{V}_k}\breve{\boldsymbol{\beta}}_l^{(m-1)}+\frac{\eta R'(\breve{\boldsymbol{\beta}}_k)}{\rho K}\Big)\\ \breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k'}^{(m)} =& -\frac{0.5}{\rho|\mathcal{V}_k|}\Big(\frac{1}{N_k}\sum_{j=1}^{N_k-1}\ell'(\mathbf{x}_{k,j},y_{k,j};\breve{\boldsymbol{\beta}}_k)+\boldsymbol{\gamma}_k^{(m-1)}\\ &+\frac{1}{N_k}\ell'(\mathbf{x}'_{k,N_k},y'_{k,N_k};\breve{\boldsymbol{\beta}}_k)\Big)\\ &+\frac{0.5}{|\mathcal{V}_k|}\Big(\breve{\boldsymbol{\beta}}_k^{(m-1)}+\sum_{l\in\mathcal{V}_k}\breve{\boldsymbol{\beta}}_l^{(m-1)}+\frac{\eta R'(\breve{\boldsymbol{\beta}}_k)}{\rho K}\Big).\end{aligned} \tag{45}$$

Using Assumption 1, the quantity $\left\|\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k}^{(m)}-\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k'}^{(m)}\right\|$ is upper bounded as follows

$$\begin{aligned}&\left\|\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k}^{(m)}-\breve{\boldsymbol{\beta}}_{k,\mathcal{D}_k'}^{(m)}\right\|\\ &=\frac{\left\|\ell'(\mathbf{x}'_{k,N_k},y'_{k,N_k};\breve{\boldsymbol{\beta}}_k)-\ell'(\mathbf{x}_{k,N_k},y_{k,N_k};\breve{\boldsymbol{\beta}}_k)\right\|}{2\rho|\mathcal{V}_k|N_k}\\ &\le\frac{c_1}{\rho|\mathcal{V}_k|N_k}.\end{aligned} \tag{46}$$

□

## APPENDIX D
## PROOF OF THEOREM 1

*Proof.* The privacy loss due to sharing $\boldsymbol{\beta}_k^{(m)}$ is calculated as

$$\left|\log\frac{\Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k}^{(m)}]}{\Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k'}^{(m)}]}\right|=\left|\log\frac{\Pr[\boldsymbol{\xi}_{k,\mathcal{D}_k}^{(m)}]}{\Pr[\boldsymbol{\xi}_{k,\mathcal{D}_k'}^{(m)}]}\right|=\left|\log\frac{\Pr[\xi_{s,k,\mathcal{D}_k}^{(m)}]}{\Pr[\xi_{s,k,\mathcal{D}_k'}^{(m)}]}\right| \tag{47}$$

where the first equality holds since the Jacobian matrix of the linear transformation from $\boldsymbol{\beta}_k^{(m)}$ to $\boldsymbol{\xi}_k^{(m)}$ is the identity matrix and the second equality holds as the entries of $\boldsymbol{\xi}_k^{(m)}$, denoted by $\xi_{s,k}^{(m)}$, are independent of each other, for any entry $s$.

Using the triangle inequality, Lemma 3, and substituting $\sigma_k$ in the resulting expression, we obtain

$$\left|\log\frac{\Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k}^{(m)}]}{\Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k'}^{(m)}]}\right|\le\frac{\rho|\mathcal{V}_k|N_k\epsilon^2}{2.1c_1\log(1.25/\delta)}\left|\xi_{s,k}^{(m)}+\frac{c_1}{2\rho|\mathcal{V}_k|N_k}\right|. \tag{48}$$

When

$$|\xi_{s,k}^{(m)}|\le\frac{c_1}{\rho|\mathcal{V}_k|N_k}\left(2.1\epsilon^{-1}\log(1.25/\delta)-0.5\right),$$

the privacy loss is bounded by $\epsilon$. Hence, let us define

$$r=\frac{c_1}{\rho|\mathcal{V}_k|N_k}\left(2.1\epsilon^{-1}\log(1.25/\delta)-0.5\right).$$

Subsequently, we need to prove that

$$\Pr[|\xi_{s,k}^{(m)}|>r]\le\delta$$

or equivalently

$$\Pr[\xi_{s,k}^{(m)}>r]\le 0.5\delta.$$

Using the tail bound of the normal distribution $\mathcal{N}(0,\sigma_k^2)$ [35], we obtain

$$\Pr[\xi_{s,k}^{(m)}>r]\le\frac{\sigma_k}{r\sqrt{2\pi}}\exp\Big(-\frac{r^2}{2\sigma_k^2}\Big). \tag{49}$$

Since $\delta$ is assumed to be small ($\le 0.01$) and $\epsilon\le 1$, we have $\sigma_k<r$ and $-r^2<2\sigma_k^2\log(0.5\sqrt{2\pi}\delta)$. Therefore, $\Pr[\xi_{s,k}^{(m)}>r]<0.5\delta$, which implies $\Pr[|\xi_{s,k}^{(m)}|>r]\le\delta$. By defining

$$\mathbb{A}_1=\{\xi_{s,k}^{(m)}:|\xi_{s,k}^{(m)}|\le r\},\quad \mathbb{A}_2=\{\xi_{s,k}^{(m)}:|\xi_{s,k}^{(m)}|>r\},$$

we have

$$\begin{aligned}\Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k}^{(m)}]&=\Pr[\breve{\beta}_{s,k,\mathcal{D}_k}^{(m)}+\xi_{s,k}^{(m)}:\xi_{s,k}^{(m)}\in\mathbb{A}_1]\\ &\quad+\Pr[\breve{\beta}_{s,k,\mathcal{D}_k}^{(m)}+\xi_{s,k}^{(m)}:\xi_{s,k}^{(m)}\in\mathbb{A}_2]\\ &<e^{\epsilon}\Pr[\boldsymbol{\beta}_{k,\mathcal{D}_k'}^{(m)}]+\delta,\end{aligned} \tag{50}$$

which concludes the proof by showing that, at each iteration of D-ZOA, $(\epsilon,\delta)$-differential privacy is guaranteed. □

## APPENDIX E
## PROOF OF THEOREM 3

*Proof.* In virtue of [38, Lemma 1 and Lemma 2], $\breve{\mathbf{w}}^{(m)}$ satisfies the following equation

$$\frac{f'(\breve{\mathbf{w}}^{(m)})}{\rho}=2\mathbf{H}\boldsymbol{\xi}^{(m)}-2\mathbf{Q}\mathbf{r}^{(m)}-\mathbf{L}_+(\mathbf{w}^{(m)}-\mathbf{w}^{(m-1)}). \tag{51}$$

Therefore, by using (51), [38, Lemma 3, Lemma 4 and Lemma 5], and the steps in the proof of [39, Theorem 1], we can show that, for any $\mathbf{r}\in\mathbb{R}^{KP}$ and $m>0$, we have

$$\begin{aligned}&\frac{f(\breve{\mathbf{w}}^{(m)})-f(\mathbf{w}^*)}{\rho}+2\mathbf{r}^{\mathsf{T}}\mathbf{Q}\breve{\mathbf{w}}^{(m)}\\ \le&\frac{\left\|\mathbf{q}^{(m-1)}-\mathbf{q}\right\|_{\mathbf{G}}^2}{\rho}-\frac{\left\|\mathbf{q}^{(m)}-\mathbf{q}\right\|_{\mathbf{G}}^2}{\rho}\\ &-\left\|\mathbf{Q}\breve{\mathbf{w}}^{(m)}\right\|^2-\left\|\mathbf{Q}\boldsymbol{\xi}^{(m)}\right\|^2+2(\boldsymbol{\xi}^{(m)})^{\mathsf{T}}\mathbf{Q}(\mathbf{r}^{(m)}-\mathbf{r})\\ &+2\Big(\frac{\mathbf{L}_+}{2}(\breve{\mathbf{w}}^{(m)}-\mathbf{w}^*)\Big)^{\mathsf{T}}(\mathbf{w}^{(m-1)}-\breve{\mathbf{w}}^{(m-1)})\end{aligned} \tag{52}$$

where $\mathbf{q}=[\mathbf{r}^{\mathsf{T}},(\mathbf{w}^*)^{\mathsf{T}}]^{\mathsf{T}}$.

For any symmetric matrix $\mathbf{X}\in\mathbb{R}^{P\times P}$ and vector $\mathbf{y}\in\mathbb{R}^P$, we have

$$\|\mathbf{y}\|^2\lambda_{\min}(\mathbf{X})\le\mathbf{y}^{\mathsf{T}}\mathbf{X}\mathbf{y}\le\|\mathbf{y}\|^2\lambda_{\max}(\mathbf{X})$$

and, for any $\mathbf{a},\mathbf{b}\in\mathbb{R}^P$ and $\tau\in\mathbb{R}_+$, we have

$$2\mathbf{a}^{\mathsf{T}}\mathbf{b}\le\tau^{-1}\|\mathbf{a}\|^2+\tau\|\mathbf{b}\|^2.$$

Therefore, (52) yields

$$\begin{aligned}&\frac{f(\breve{\mathbf{w}}^{(m)})-f(\mathbf{w}^*)}{\rho}+2\mathbf{r}^\mathsf{T}\mathbf{Q}\breve{\mathbf{w}}^{(m)}\\ \leq&\frac{\left\|\mathbf{q}^{(m-1)}-\mathbf{q}\right\|_\mathbf{G}^2}{\rho}-\frac{\left\|\mathbf{q}^{(m)}-\mathbf{q}\right\|_\mathbf{G}^2}{\rho}-\left\|\mathbf{Q}\boldsymbol{\xi}^{(m)}\right\|^2\\ &-\frac{\lambda_{\min}(\mathbf{L}_-)}{2}\left\|\breve{\mathbf{w}}^{(m)}-\mathbf{w}^*\right\|^2+\frac{1}{\tau}\left\|\frac{\mathbf{L}_+}{2}(\breve{\mathbf{w}}^{(m)}-\mathbf{w}^*)\right\|^2\\ &+\tau\left\|\mathbf{w}^{(m-1)}-\breve{\mathbf{w}}^{(m-1)}\right\|^2+2(\boldsymbol{\xi}^{(m)})^\mathsf{T}\mathbf{Q}(\mathbf{r}^{(m)}-\mathbf{r}).\end{aligned}\tag{53}$$

By setting

$$\tau=\frac{\lambda_{\max}^2(\mathbf{L}_+)}{2\lambda_{\min}(\mathbf{L}_-)},$$

(53) leads to

$$\begin{aligned}&\frac{f(\breve{\mathbf{w}}^{(m)})-f(\mathbf{w}^*)}{\rho}+2\mathbf{r}^\mathsf{T}\mathbf{Q}\breve{\mathbf{w}}^{(m)}\\ \leq&\frac{\left\|\mathbf{q}^{(m-1)}-\mathbf{q}\right\|_\mathbf{G}^2}{\rho}-\frac{\left\|\mathbf{q}^{(m)}-\mathbf{q}\right\|_\mathbf{G}^2}{\rho}\\ &+\frac{\lambda_{\max}^2(\mathbf{L}_+)}{2\lambda_{\min}(\mathbf{L}_-)}\left\|\boldsymbol{\xi}^{(m-1)}\right\|^2+2(\boldsymbol{\xi}^{(m)})^\mathsf{T}\mathbf{Q}(\mathbf{r}^{(m)}-\mathbf{r}).\end{aligned}\tag{54}$$

Setting $\mathbf{r}=\mathbf{0}_P$ and summing both sides of (54) over $m=1$ to $M$ gives

$$\begin{aligned}&\frac{1}{\rho}\sum_{m=1}^M(f(\breve{\mathbf{w}}^{(m)})-f(\mathbf{w}^*))\leq\frac{1}{\rho}\left\|\mathbf{q}^{(0)}-\mathbf{q}\right\|_\mathbf{G}^2\\ &+\sum_{m=1}^M\frac{\lambda_{\max}^2(\mathbf{L}_+)}{2\lambda_{\min}(\mathbf{L}_-)}\left\|\boldsymbol{\xi}^{(m-1)}\right\|^2+2(\boldsymbol{\xi}^{(m)})^\mathsf{T}\mathbf{Q}\mathbf{r}^{(m)}.\end{aligned}\tag{55}$$

Using Jensen's inequality [43], (17), (20), and applying the expectation operator to both sides of (55), we obtain

$$\begin{aligned}&\mathbb{E}[f(\hat{\mathbf{w}}^{(M)})-f(\mathbf{w}^*)]\\ \leq&\frac{1}{M}\left\|\mathbf{q}^{(0)}-\mathbf{q}\right\|_\mathbf{G}^2+\frac{\rho\lambda_{\max}^2(\mathbf{L}_+)}{2M\lambda_{\min}(\mathbf{L}_-)}\sum_{m=1}^M\mathbb{E}\left[\left\|\boldsymbol{\xi}^{(m-1)}\right\|^2\right]\\ \leq&\frac{\left\|\mathbf{q}^{(0)}-\mathbf{q}\right\|_\mathbf{G}^2}{M}+\frac{2.1c_1^2P\log(1.25/\delta)\lambda_{\max}^2(\mathbf{L}_+)}{2\rho|\mathcal{V}_k|^2N_k^2\epsilon^2\lambda_{\min}(\mathbf{L}_-)}\end{aligned}\tag{56}$$

where

$$\hat{\mathbf{w}}^{(M)}=\frac{1}{M}\sum_{m=1}^M\breve{\mathbf{w}}^{(m)}.$$

□

## REFERENCES


[1] C. Gratton, N. K. D. Venkategowda, R. Arablouei, and S. Werner, "Distributed learning with non-smooth objective functions," in *Proc. European Speech and Signal Processing Conference*, Jan. 2021.

[2] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[3] C. Gratton, N. K. D. Venkategowda, R. Arablouei, and S. Werner, "Consensus-based distributed total least-squares estimation using parametric semidefinite programming," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2019, pp. 5227–5231.

[4] ——, "Distributed ridge regression with feature partitioning," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Oct. 2018.

[5] J. Akhtar and K. Rajawat, "Distributed sequential estimation in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 1, pp. 86–100, Jan. 2018.

[6] N. K. D. Venkategowda and S. Werner, "Privacy-preserving distributed precoder design for decentralized estimation," in *Proc. IEEE Global Conference on Signal and Information Processing*, Nov. 2018.

[7] G. B. Giannakis, Q. Ling, G. Mateos, and I. D. Schizas, *Splitting Methods in Communication, Imaging, Science, and Engineering*, ser. Scientific Computation, R. Glowinski, S. J. Osher, and W. Yin, Eds. Cham: Springer International Publishing, 2016.

[8] D. Hajinezhad, M. Hong, and A. Garcia, "ZONE: Zeroth-order nonconvex multiagent optimization over networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 3995–4010, Oct. 2019.

[9] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[10] S. P. Talebi and S. Werner, "Distributed Kalman filtering and control through embedded average consensus information fusion," *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4396–4403, Mar. 2019.

[11] A. Agarwal, O. Dekel, and L. Xiao, "Optimal algorithms for online convex optimization with multi-point bandit feedback," in *Proc. 23rd Annual Conference on Learning Theory*, Jun. 2010, pp. 28–40.

[12] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Wiley, 2003.

[13] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop on Artificial Intelligence and Security*, Nov. 2017, pp. 15–26.

[14] S. Liu, J. Chen, P.-Y. Chen, and A. Hero, "Zeroth-order online alternating direction method of multipliers: convergence analysis and applications," in *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 84, Apr. 2018, pp. 288–297.

[15] F. Huang, S. Gao, S. Chen, and H. Huang, "Zeroth-order stochastic alternating direction method of multipliers for nonconvex nonsmooth optimization," in *Proc. 28th International Joint Conference on Artificial Intelligence*, S. Kraus, Ed., 2019, pp. 2549–2555.

[16] S. Liu, P. Y. Chen, B. Kailkhura, G. Zhang, A. O. Hero III, and P. K. Varshney, "A primer on zeroth-order optimization in signal processing and machine learning: principals, recent advances, and applications," *IEEE Signal Processing Magazine*, vol. 37, no. 5, pp. 43–54, 2020.

[17] T. Zhang and Q. Zhu, "Dynamic differential privacy for ADMM-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, Jan. 2017.

[18] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *Proc. 56th Annual Allerton Conference on Communication, Control, and Computing*, Oct. 2018, pp. 959–965.

[19] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proc. 35th International Conference on Machine Learning*, vol. 80, Jul. 2018, pp. 5796–5805.

[20] J. Ding, Y. Gong, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," 2019. [Online]. Available: http://arxiv.org/abs/1901.02094

[21] J. Ding, S. M. Errapotu, H. Zhang, Y. Gong, M. Pan, and Z. Han, "Stochastic ADMM based distributed machine learning with differential privacy," in *Proc. 15th SecureComm*, Oct. 2019, pp. 257–277.

[22] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. 2015 International Conference on Distributed Computing and Networking*, 2015.

[23] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 50–64, 2017.

[24] M. T. Hale and M. Egerstedty, "Differentially private cloud-based multi-agent optimization with constraints," in *Proc. 2015 American Control Conference*, Jul. 2015, pp. 1235–1240.

[25] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via functional perturbation," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 395–408, 2018.

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

[27] F. Farokhi, N. Wu, D. Smith, and M. A. Kaafar, "The cost of privacy in asynchronous differentially-private machine learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2118–2129, 2021.

[28] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2013.

[29] J. C. Duchi, M. I. Jordan, M. J. Wainwright, and A. Wibisono, "Optimal rates for zero-order convex optimization: the power of two function evaluations," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2788–2806, May 2015.

[30] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[31] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *Journal of Machine Learning Research*, vol. 11, pp. 1663–1707, Aug. 2010.

[32] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 2015 ACM SIGSAC Conference on Computer and Communications Security*, Oct. 2015, pp. 1322–1333.

[33] Y. Hu, P. Liu, L. Kong, and D. Niu, "Learning privately over distributed features: an ADMM sharing approach," 2019. [Online]. Available: http://arxiv.org/abs/1907.07735

[34] Z. Han, M. Hong, and D. Wang, *Signal processing and networking for big data applications*. Cambridge University Press, 2017.

[35] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, Aug. 2014.

[36] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Third Conference on Theory of Cryptography*. Springer-Verlag, 2006, pp. 265–284.

[37] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[38] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust federated learning using ADMM in the presence of data falsifying byzantines," 2017. [Online]. Available: http://arxiv.org/abs/1710.05241

[39] ——, "Robust decentralized learning using ADMM with unreliable agents," 2018. [Online]. Available: http://arxiv.org/abs/1710.05241

[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2010.

[41] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multi-agent optimization," 2020. [Online]. Available: https://arxiv.org/abs/1908.11444

[42] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, 2014.

[43] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*, 4th ed. McGraw Hill, 2002.


**Cristiano Gratton** received both the B.Sc. and M.Sc. degree in Mathematics from the University of Udine (Italy) in 2014 and 2017, respectively. He is pursuing the Ph.D. degree at the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway. His research interests are around distributed optimization and private data analysis.

**Naveen K. D. Venkategowda** Naveen K. D. Venkategowda (S'12–M'17) received the B.E. degree in electronics and communication engineering from Bangalore University, Bengaluru, India, in 2008, and the Ph.D. degree in electrical engineering from Indian Institute of Technology, Kanpur, India, in 2016. He is currently an Universitetslektor at the Department of Science and Technology, Linköping University, Sweden. From Oct. 2017 to Feb. 2021, he was postdoctoral researcher at the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway. He was a Research Professor at the School of Electrical Engineering, Korea University, South Korea from Aug. 2016 to Sep. 2017. He was a recipient of the TCS Research Fellowship (2011-15) from TCS for graduate studies in computing sciences and the ERCIM Alain Bensoussan Fellowship in 2017.

**Reza Arablouei** received the Ph.D. degree in telecommunications engineering from the Institute for Telecommunications Research, University of South Australia, Mawson Lakes, SA, Australia, in 2013. He was a Research Fellow with the University of South Australia from 2013 to 2015. He is currently a Senior Research Scientist with the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Pullenvale, QLD, Australia. His research interests include signal processing and machine learning on embedded systems.

**Stefan Werner** (SM'07) received the M.Sc. degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998, and the D.Sc. degree (Hons.) in electrical engineering from the Signal Processing Laboratory, Helsinki University of Technology, Espoo, Finland, in 2002. He is currently a Professor at the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), and Director of IoT@NTNU. He is also an Adjunct Professor with Aalto University in Finland and an Adjunct Senior Research Fellow with the Institute for Telecommunications Research, University of South Australia. He was a visiting Melchor Professor with the University of Notre Dame during summer 2019 and held an Academy Research Fellowship, funded by the Academy of Finland, from 2009 to 2014. His research interests include adaptive and statistical signal processing, signal processing for communications, and security and privacy in cyberphysical systems. He is a member of the editorial boards for the EURASIP Journal of Signal Processing and the IEEE Transactions on Signal and Information Processing over Networks.