Privacy-Preserving Distributed 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. Unlike the existing privacy-preserving methods based on the ADMM where the primal or the dual variables are perturbed with noise, the inherent randomness due to the use of a zeroth-order method endows D-ZOA with intrinsic differential privacy. By analyzing the perturbation of the primal variable, we show that the privacy leakage of the proposed D-ZOA algorithm is bounded. In addition, we employ the moments accountant method to show that the total privacy leakage grows sublinearly with the number of ADMM iterations. D-ZOA outperforms the existing differentially private approaches in terms of accuracy while yielding the same privacy guarantee. We prove that D-ZOA converges to the optimal solution at a rate of $O(1/M)$ where $M$ is the number of ADMM iterations. 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 a state-of-the-art algorithm as well as its network-wide convergence to the optimal solution.

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

I. INTRODUCTION

W ith the recent advances in technology, large amounts of data are gathered by numerous sensors scattered over large geographical areas. Performing learning tasks at a central processing hub in a large distributed network can 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]. In this context, each agent has access only to the information of its local objective function while the agents aim to collaboratively optimize the aggregate of the local objective functions. 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].

However, the communications between neighboring agents 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 agent’s data is taken into account.

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. For example, in bandit optimization [11], an adversary generates a sequence of loss functions and the goal is to minimize such sequence that is only available at some agents. In addition, in simulation-based optimization, the objective is available only using repeated simulations [12] while, in adversarial black-box machine learning models, only the function values are given [13]. This motivates the use of zeroth-order methods, which only use the values of the objective functions to approximate their gradients.

A. Related Work

There have been several works developing privacy-preserving algorithms for distributed convex optimization [14]–[21]. The work in [14] proposes two differentially private distributed algorithms that are based on the alternating direction method of multipliers (ADMM). The algorithms in [14] 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 [15]–[18]. The works in [15], [16] develop ADMM-based differentially private algorithms with improved accuracy. The work in [17] 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 [18] develops
a stochastic ADMM-based distributed algorithm that further enhances the accuracy while ensuring differential privacy. The authors of [19]–[21] propose differentially-private distributed algorithms that utilize the projected-gradient-descent method for handling constraints. The differentially private distributed algorithm proposed in [22] is based on perturbing the local objective functions.

All the above-mentioned algorithms offer distributed solutions only for problems with smooth objective functions. The work in [23] 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.

Most existing algorithms 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 [24] 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 [24] considers a privacy metric based on the topology of the communication graph.

B. 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 [25], 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. It also only requires the objective function values to solve the underlying ERM problem while respecting privacy.

The privacy-preserving properties of the proposed D-ZOA algorithm are intrinsic. To prove this, we model the primal variable at each agent as the sum of an exact (unperturbed) value and a random perturbation. This enables us to approximate the distribution of the primal variable and verify that the stochasticity inherent to the employed zeroth-order method makes D-ZOA differential private. To this end, we prove that the privacy leakage of a single iteration of D-ZOA at each agent is bounded. Utilizing the moments accountant method [26], we also show that the total privacy leakage over all iterations grows sublinearly with the number of ADMM iterations. This is particularly significant as we observe that 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 the convergence rate of D-ZOA is $O(1/M)$ where $M$ is the number of iterations of the ADMM outer loop. The convergence rate of the inner loop is $O(\sqrt{P/T})$ where $P$ is the number of features and $T$ is the number of iterations of the inner loop. More importantly, no communication among agents is required throughout the inner loop. The convergence analysis also reveals an explicit privacy-accuracy trade-off where a stronger privacy guarantee corresponds to a lower accuracy.

Simulation results demonstrate that, with any given level of required privacy guarantee, D-ZOA outperforms an existing ADMM-based related algorithm (DP-ADMM) presented in [23]. This algorithm has been developed for distributed optimization of non-smooth objective functions and achieves differential privacy by adding noise to the primal variables [23].

C. 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 and explain the privacy issues associated with distributed learning. In Section IV 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.

D. 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 $(\cdot)^{\top}$, $\det(\cdot)$, and $\text{tr}(\cdot)$ 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, $\mathbf{0}_{n \times p} = \mathbf{0}_n \mathbf{0}_p^\top$, and $\| \cdot \|$ denotes the cardinality if its argument is a set. The statistical expectation and covariance operators are represented by $\mathbb{E} [\cdot]$ and $\text{cov} [\cdot]$, respectively. The notation $\mathcal{N}(\mu, \Sigma)$ denotes normal distribution with mean $\mu$ and covariance matrix $\Sigma$. For a positive semidefinite matrix $\mathbf{X}$, $\lambda_{\text{min}}(\mathbf{X})$ and $\lambda_{\text{max}}(\mathbf{X})$ denote the nonzero smallest and largest eigenvalues of $\mathbf{X}$, respectively. For a vector $\mathbf{x} \in \mathbb{R}^n$ and a matrix $\mathbf{A}$, $\| \mathbf{x} \|^2_\mathbf{A}$ denotes the quadratic form $\mathbf{x}^\top \mathbf{A} \mathbf{x}$.

II. System Model

We consider a network with $K \in \mathbb{N}$ agents and $\mathcal{E} \in \mathbb{N}$ edges modeled as an undirected graph $G(K, \mathcal{E})$ where the vertex set $K = \{1, \ldots, K\}$ corresponds to the agents and the set $\mathcal{E}$ represents the bidirectional communication links between the pairs of neighboring agents. Agent $k \in K$ can communicate only with the agents in its neighborhood $\mathcal{N}_k$. By convention, the set $\mathcal{N}_k$ includes the agent $k$ as well.
Each agent $k \in \mathcal{K}$ has a private dataset
\[ \mathcal{D}_k = \{ (\mathbf{x}_k, \mathbf{y}_k) : \mathbf{x}_k = [x_{k,1}, x_{k,2}, \ldots, x_{k,N_k}]^T \in \mathbb{R}^{N_k \times P}, \mathbf{y}_k = [y_{k,1}, y_{k,2}, \ldots, y_{k,N_k}]^T \in \mathbb{R}^{N_k} \} \]
where $N_k$ is the number of data samples collected at the agent $k$ and $P$ is the number of features in each sample.

We consider the problem of estimating a parameter of interest $\beta \in \mathbb{R}^P$ 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
\[
\min_{\beta} \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \beta) + \frac{\eta}{K} R(\beta), \tag{1}
\]
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 [27], and logistic regression [17, 23]. We assume that the loss function $\ell(\cdot)$ and the regularizer function $R(\cdot)$ are both convex but at least one of them is non-smooth. Let us denote the optimal solution of (1) by $\beta^*$. 

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 privacy matters.

A. Consensus-Based Reformulation

To solve (1) in a distributed manner, we reformulate it as the following constrained minimization problem
\[
\min_{\{\beta_k\}} \sum_{k=1}^{K} \left( \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \beta_k) + \frac{\eta}{K} R(\beta_k) \right) \tag{2}
\]
subject to $\beta_k = \beta_l$, $l \in \mathcal{N}_k$, $\forall k \in \mathcal{K}$ where $\mathcal{V} = \{\beta_k\}_{k=1}^{K}$ are the primal variables representing local copies of $\beta$ at the agents. The equality constraints impose consensus across each agent’s neighborhood $\mathcal{N}_k$. To solve (2) collaboratively and in a fully-distributed manner, we utilize the ADMM [7]. For this purpose, we rewrite (2) as
\[
\min_{\{\beta_k\}} \sum_{k=1}^{K} \left( \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \beta_k) + \frac{\eta}{K} R(\beta_k) \right) \tag{3}
\]
subject to $\beta_k = \beta_l'$, $l \in \mathcal{N}_k$, $\forall k \in \mathcal{K}$
where $\mathcal{Z} = \{z_k'\}_{k \in \mathcal{K}, l \in \mathcal{N}_k}$ are the auxiliary variables yielding an alternative but equivalent representation of the constraints in (2). They help decouple $\beta_k$ in the constraints and facilitate the derivation of the local recursions before being eventually eliminated.

To apply the ADMM, we rewrite (3) in the matrix form. By defining $\mathbf{w} \in \mathbb{R}^{KP}$ concatenating all $\beta_k$ and $\mathbf{z} \in \mathbb{R}^{2EP}$ concatenating all $z_k'$, (3) can be written as
\[
\min_{\mathbf{w}, \mathbf{z}} f(\mathbf{w}) \quad \text{s.t.} \quad \mathbf{A w} + \mathbf{B z} = \mathbf{0} \tag{4}
\]
where
\[
f(\mathbf{w}) = \sum_{k=1}^{K} f_k(\beta_k), \quad f_k(\beta_k) = \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(\mathbf{x}_{k,j}, y_{k,j}; \beta_k) + \frac{\eta}{K} R(\beta_k),
\]
$\mathbf{A} = [\mathbf{A}_1^T, \mathbf{A}_2^T]^T$, 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 $z_l$ 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 $0_{P \times P}$. Furthermore, we have
\[
\mathbf{B} = [-\mathbf{I}_{2EP}, -\mathbf{I}_{2EP}]^T.
\]
To facilitate the representation, we also define the following matrices
\[
\mathbf{M}_+ = \mathbf{A}_1^T + \mathbf{A}_2^T, \quad \mathbf{M}_- = \mathbf{A}_1^T - \mathbf{A}_2^T, \quad \mathbf{L}_+ = 0.5 \mathbf{M}_+ \mathbf{M}_+^T, \quad \mathbf{L}_- = 0.5 \mathbf{M}_- \mathbf{M}_-^T, \quad \mathbf{H} = 0.5 (\mathbf{L}_+ + \mathbf{L}_-), \quad \mathbf{Q} = \sqrt{0.5 \mathbf{L}_-}.
\]
Solving (4) via the ADMM requires a procedure that is described in the next subsection.

B. Distributed ADMM Algorithm

To solve the minimization problem (4) in a distributed manner, we employ the ADMM [28]. The augmented Lagrangian function associated with (4) is given by
\[
\mathcal{L}_\rho(\mathbf{w}, \mathbf{z}, \lambda) = f(\mathbf{w}) + \lambda^T (\mathbf{A w} + \mathbf{B z}) + \frac{\rho}{2} \| \mathbf{A w} + \mathbf{B z} \|^2 \tag{5}
\]
where $\lambda \in \mathbb{R}^{4EP}$ is the Lagrange multiplier and $\rho > 0$ is a penalty parameter.

The ADMM entails an iterative procedure consisting of three steps at each iteration $m$ as
\[
\mathbf{w}^{(m)} = \arg \min_{\mathbf{w}} \mathcal{L}_\rho(\mathbf{w}, \mathbf{z}^{(m-1)}, \lambda^{(m-1)}) \quad \mathbf{z}^{(m)} = \arg \min_{\mathbf{z}} \mathcal{L}_\rho(\mathbf{w}^{(m)}, \mathbf{z}, \lambda^{(m-1)}) \tag{6}
\]
\[
\lambda^{(m)} = \lambda^{(m-1)} + \rho (\mathbf{A} \mathbf{w}^{(m)} + \mathbf{B} \mathbf{z}^{(m)}) \quad \gamma^{(m)} = \gamma^{(m-1)} + \rho \mathbf{L}_- \mathbf{w}^{(m)} \tag{7}
\]
In [28], it is shown that, considering $\lambda = [\lambda_1^T, \lambda_2^T]^T$ with $\lambda_1, \lambda_2 \in \mathbb{R}^{2EP}$ and $\gamma = \mathbf{M}_- \lambda_1$, the ADMM algorithm steps in (5) reduce to
\[
\mathbf{w}^{(m)} = \arg \min_{\mathbf{w}} \mathcal{F}(\mathbf{w}, \mathbf{w}^{(m-1)}, \gamma^{(m-1)}) \quad \gamma^{(m)} = \gamma^{(m-1)} + \rho \mathbf{L}_- \mathbf{w}^{(m)} \tag{8}
\]
where
\[ F(w, w^{(m-1)}, \gamma^{(m-1)}) = f(w) + w^T \gamma^{(m-1)} + \rho w^T H w - \rho w^T L_+ w^{(m-1)} \]
and the initial values of \( w \) and \( \gamma \) are set to zero. Note that the update equations (7) and (8) are distributed among the agents as \( w = [\beta_1, \beta_2, \ldots, \beta_K]^T \) and \( \gamma = [\gamma_1, \gamma_2, \ldots, \gamma_K]^T \) where \( \gamma_k = 2 \sum_{i \in N_k} \gamma_{ik} \)
is the local Lagrange multiplier at agent \( k \) associated with the constraints in (3) (7).

Since the objective function in (7) is assumed to be non-smooth, it cannot be solved using any first-order method. To overcome this, we use a zeroth-order method described in the next subsection.

C. Zeroth-Order Method

To solve (7) employing a zeroth-order method, we make the following assumptions that are common in the zeroth-order optimization literature, see, e.g., [8, 25, 29].

Assumption 1: The function \( F(\cdot) \) is closed.

Assumption 2: The function \( F(\cdot) \) is Lipschitz-continuous with the Lipschitz constant \( L \).

In zeroth-order optimization, \( F(\cdot) \) is also required to be convex [8, 25, 29]. However, \( F(\cdot) \) is the sum of \( f(\cdot) \), which are assumed to be convex, the linear terms \( w^T \gamma^{(m-1)} \) and \(-\rho w^T L_+ w^{(m-1)} \), and the quadratic term \( \rho w^T H w \), which is convex since \( H \) is positive semi-definite for being the sum of the positive semi-definite matrices \( L_+ \) and \( L_- \). Therefore, \( F(\cdot) \) is convex as it is the sum of convex functions [30].

We utilize the two-point stochastic-gradient algorithm that has been proposed in [25] for optimizing general non-smooth functions. More specifically, we use the stochastic mirror descent method with the proximal function \( \frac{1}{2} \| \| \) and the gradient estimator at point \( w \) given by
\[ \Gamma(w, \gamma^{(m-1)}, u_1, u_2, \nu_1, \nu_2) = u_2^{-1} F(w + u_1 \nu_1 + u_2 \nu_2, \gamma^{(m-1)}) - F(w + u_1 \nu_1, \gamma^{(m-1)}) \nu_2 \quad (9) \]
where \( u_1 > 0 \) and \( u_2 > 0 \) are smoothing constants and \( \nu_1, \nu_2 \) are independent zero-mean Gaussian random vectors with the covariance matrix \( I_{KP} \), i.e., \( \nu_1, \nu_2 \sim \mathcal{N}(0, I_{KP}) \).

The two-point stochastic-gradient algorithm consists of two randomization steps where the second step is aimed at preventing the perturbation vector \( \nu_2 \) from being close to a point of non-smoothness [25]. This algorithm entails an iterative procedure that consists of three steps at each iteration \( t \). First, \( J \in \mathbb{N} \) independent random vectors \( \{\nu_{1,t}\}_{j=1}^J \) and \( \{\nu_{2,t}\}_{j=1}^J \) are sampled from \( \mathcal{N}(0, I_{KP}) \). Second, a stochastic gradient \( g_k^{(t)} \) is calculated as
\[ g_k^{(t)} = \frac{1}{J} \sum_{j=1}^J g_{j,k}^{(t)} \quad (10) \]
where
\[ g_{j,k}^{(t)} = \Gamma(w^{(t)}, \gamma^{(m-1)}, u_{1,t}, u_{2,t}, \nu_{1,t,j}, \nu_{2,t,j}) \]

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

At all agents \( k \in K \), initialize \( \beta_k^{(0)} = 0 \), \( \gamma_k^{(0)} = 0 \), and locally run
for \( m = 1, 2, \ldots, M \) do
Share \( \beta_k^{(m-1)} \) with neighbors in \( N_k \)
Update \( \gamma_k^{(m)} \) as in (20)
Initialize \( \beta_k^{(0)} = 0 \)
for \( t = 1, 2, \ldots, T \) do
Draw independent \( \{\nu_{1,t,j}\}_{j=1}^J, \{\nu_{2,t,j}\}_{j=1}^J \sim \mathcal{N}(0, I_P) \)
Set \( u_{1,t} = u_{1,t}/t, u_{2,t} = u_2/(Pt)^2 \)
Compute \( g_k^{(t)} \) as in (12) and (9)
Update \( \beta_k^{(t)} = \beta_k^{(t-1)} - \alpha_t g_k^{(t)} \)
end for
\end{algorithm}

\( w^{(t)} \) is the \( t \)th iterate of the two-point stochastic-gradient algorithm with the initial value \( w^{(0)} = 0 \) 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, \( w^{(t)} \) is updated as
\[ w^{(t)} = w^{(t-1)} - \alpha_t g_k^{(t)} \quad (11) \]
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 a minimizer \( w^* \) to (7) and the first iterate \( w^{(1)} \) as per [25].

Locally at agent \( k \), the appropriate subvectors \( \{\nu_{1,t,j}\}_{j=1}^J \) and \( \{\nu_{2,t,j}\}_{j=1}^J \) of, respectively, \( \{\nu_{1,t,j}\}_{j=1}^J \) and \( \{\nu_{2,t,j}\}_{j=1}^J \) are utilized. These subvectors are also independent of each other and have independent entries. Subsequently, a \( k \)-local stochastic gradient \( g_k^{(t)} \) is computed as
\[ g_k^{(t)} = \frac{1}{J} \sum_{j=1}^J g_{j,k}^{(t)} \quad (12) \]
where
\[ g_{j,k}^{(t)} = \Gamma(w^{(t)}, \gamma^{(m-1)}, u_{1,t}, u_{2,t}, \nu_{1,t,j}, \nu_{2,t,j}) \]

Hence, \( \beta_k^{(t)} \) is updated as
\[ \beta_k^{(t)} = \beta_k^{(t-1)} - \alpha_t g_k^{(t)} \quad (13) \]
We use multiple independent random samples \( \{\nu_{1,t,j}\}_{j=1}^J \) and \( \{\nu_{2,t,j}\}_{j=1}^J \) to obtain a more accurate estimate of the gradient \( g_k^{(t)} \) as remarked in [25].

Furthermore, no communication among agents is needed in the inner loop. The update equations in (11) and (13) can be implemented in a fully-distributed fashion since they involve only the variables available within every agent’s neighborhood. The proposed algorithm, D-ZOA, is summarized in Algorithm 1.
D. Privacy

In Algorithm\textsuperscript{1} the data stored at each agent, $X_k$ and $y_k$, is not shared with any other agent. However, the local estimates $\{\beta_k^{(m)}\}_{k \in \mathcal{K}}$ are exchanged within the local neighborhoods. Therefore, the risk of privacy breach still exists as it has been shown by model inversion attacks \textsuperscript{[31].} We assume that the adversary cannot access the local data, but is able to access the exchanged local estimates $\{\beta_k^{(m)}\}_{k \in \mathcal{K}}$. The adversary can be either a member of the network or an external eavesdropper. We show that D-ZOA guarantees $(\epsilon, \delta)$-differential privacy as per 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 $D$ and $D'$ differing in only one data sample and for any subset of outputs $\mathcal{O} \subseteq \text{range}(\mathcal{M})$, we have

$$\Pr[\mathcal{M}(D) \in \mathcal{O}] \leq e^\epsilon \Pr[\mathcal{M}(D') \in \mathcal{O}] + \delta. \quad (14)$$

This means the ratio of the probability distributions of $\mathcal{M}(D)$ and $\mathcal{M}(D')$ 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 [17], [32].

IV. INTRINSIC DIFFERENTIAL PRIVACY GUARANTEE

Employing the zero-order method for the ADMM primal update produces a perturbed (inexact) estimate. Therefore, the solution in the primal update step in (7) using D-ZOA can be modeled as

$$\tilde{w}^{(m)} = \tilde{w}^{(m)} + \xi^{(m)} \quad (15)$$

where $\tilde{w}^{(m)} \in \mathbb{R}^P$ is the exact ADMM primal update and $\xi^{(m)} \in \mathbb{R}^P$ is a random variable representing the perturbation. The oracle $\tilde{w}^{(m)}$ satisfies the equation

$$\nabla f(\tilde{w}^{(m)}) + \gamma^{(m-1)} + 2\rho H\tilde{w}^{(m)} = \rho L + w^{(m-1)} \quad (16)$$

where $\nabla f$ is the hypothetical exact gradient of $f$. Equation (16) is obtained by computing $\nabla f(\tilde{w}^{(m-1)}, \gamma^{(m-1)})$ and equating it to zero \textsuperscript{[33], pg. 741}. The model (15) represents an implicit primal variable perturbation that can be contrasted with the explicit primal variable perturbation used in [14], [17].

Using (15), the primal and dual update equations in (16) and (8) can be expressed as

$$\begin{align*}
\tilde{w}^{(m)} &= \frac{1}{2\rho}H^{-1}\nabla f(\tilde{w}^{(m)}) - \frac{1}{2\rho}H^{-1}\gamma^{(m-1)} \\
&\quad + \frac{1}{2}H^{-1}L + w^{(m-1)} \\
\tilde{w}^{(m)} &= \tilde{w}^{(m)} + \xi^{(m)} \\
\gamma^{(m)} &= \gamma^{(m-1)} + \rho L_\ast w^{(m)}.
\end{align*} \quad (17)$$

Note that (15) also applies to all agents as

$$\tilde{w}^{(m)} = [\tilde{\beta}_1^{(m)^T}, \tilde{\beta}_2^{(m)^T}, \ldots, \tilde{\beta}_K^{(m)^T}]^T$$

$$\xi^{(m)} = [\xi_1^{(m)^T}, \xi_2^{(m)^T}, \ldots, \xi_K^{(m)^T}]^T$$

where $\tilde{\beta}_k$ is the local exact primal update at agent $k$ and $\xi_k$ is the local perturbation of $\tilde{\beta}_k$ at agent $k$. Recalling the definitions of $H_k$, $L_+$, and $L_-$, the update equations in (17) entail the following local update equations at the $k$th agent:

$$\beta_k^{(m)} = -\frac{1}{2\rho|N_k|}\nabla f_k(\tilde{\beta}_k^{(m)}) + \frac{1}{2|N_k|}\gamma^{(m-1)} + \sum_{i \in N_k} \beta_i^{(m-1)} - \frac{1}{2\rho|N_k|}\gamma_k^{(m-1)} \quad (18)$$

$$\beta_k^{(m)} = \tilde{\beta}_k^{(m)} + \xi_k^{(m)} \quad (19)$$

$$\gamma_k^{(m)} = \gamma^{(m-1)} + \rho \sum_{i \in N_k} (\beta_k^{(m)} - \beta_i^{(m)}) \quad (20)$$

To prove that D-ZOA is differentially private, we require the probability distribution of the primal variable $\beta_k^{(m)}$. In the next subsection, we find an approximate distribution of the primal variable.

A. Primal Variable Distribution

To approximate its pdf, in view of (13) and the fact that $\beta_k^{(0)} = 0$, we unfold $\beta_k^{(m)}$ as

$$\beta_k^{(m)} = -\sum_{t=1}^T \alpha_t g_k^{(t)}.$$ $\text{The stochastic gradient } \gamma^{(t)} \text{ is the average of } J \text{ independent random samples } g_{j,k}^{(t)} \text{ that are functions of the random values } \{v_{j,k}^{(t)}\}_{j=1}^J \text{ and } \{\nu_{j,k}^{(t)}\}_{j=1}^J \text{ drawn from the same distribution. Therefore, it is realistic to assume that } \{g_{j,k}^{(t)}\}_{j=1}^J \text{ are independent and identically distributed (i.i.d.) with a common mean } \mu_k^{(t)} \text{ and a finite covariance matrix } \Psi_k^{(t)}. \text{ Thus, the probability distribution of } \beta_k^{(m)} \text{ is given by the following lemma.}$

Lemma 1. If $\{g_{j,k}^{(t)}\}_{j=1}^J$ are i.i.d. and $J$ is sufficiently large, then $\beta_k^{(m)}$ is distributed as

$$\beta_k^{(m)} \sim \mathcal{N}(\tilde{\beta}_k^{(m)} : \frac{1}{J} \sum_{t=1}^T \alpha_t^2 \Psi_k^{(t)}). \quad (21)$$

Proof. See Appendix A.

Since we are interested in exploring the privacy properties of the proposed D-ZOA and, thereby, the best privacy guarantee that it offers, we consider a conservative upper-bound for the covariance matrices $\Psi_k^{(t)}$. This is motivated by the direct connection between the algorithmic privacy guarantee and the extent of the randomness involved in the algorithm where a higher variance corresponds to a higher privacy guarantee. Such a relationship between variance and privacy is also shown in the existing works on differential privacy, see, e.g., [17], [23].
To make the problem more tractable, we assume that the entries of the random vector $\mathbf{\beta}_k^{(m)}$ are independent of each other and have the same variance $[14], [23], [32]$. Let us denote the variance of every entry of $\mathbf{\xi}_k^{(m)}$ by $\sigma_k^2$. Therefore, in view of Lemma 1, we have

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

which is in turn upper-bounded as per the following lemma.

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

$$\frac{1}{JP} \sum_{t=1}^{T} \alpha_t^2 \text{tr} \left( \mathbf{\Psi}_k^{(t)} \right) \leq \frac{c \sigma_0^2 R^2}{JP \log(2P)} \left( s_1 (1 + \log(P)) + s_2 \right) - 4 \left\| \mathbf{\beta}_k^{(m)} \right\|^2 TJP,$$

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

**Proof.** See Appendix B.

In [25], it is shown that $c = 0.5$ is suitable when $\nu_1$ and $\nu_2$ are sampled from a multivariate normal distribution.

### B. l_2-Norm Sensitivity

In this subsection, we estimate the l_2-norm sensitivity of $\mathbf{\beta}_k^{(m)}$. The l_2 norm sensitivity calibrates the magnitude of the noise by which $\mathbf{\beta}_k^{(m)}$ has to be perturbed to preserve privacy. Unlike the existing privacy-preserving methods where the noise is added to the output of the algorithm [14], [17], [23], [32], in D-ZOA, the noise is inherent.

In addition to Assumptions 1 and 2, we introduce another assumption that is widely used in the literature, see, e.g., [14], [23], [32].

**Assumption 3:** There exists a constant $c_1$ such that $\| \nabla \ell(\cdot) \| \leq c_1$ where $\ell(\cdot)$ is the loss function defined in Section II.

Similar to the classical methods of differential privacy analysis, e.g., [23], [32], we first define the l_2 norm sensitivity. Subsequently, we estimate the l_2-norm sensitivity of $\mathbf{\beta}_k^{(m)}$.

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

$$\Delta_{k,2} = \max_{D_k, D'_k} \left\| \mathbf{\beta}_k^{(m)}_{D_k, D'_k} - \mathbf{\beta}_k^{(m)}_{D'_k, D_k} \right\|,$$

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

The l_2-norm sensitivity of $\mathbf{\beta}_k^{(m)}$ is an upper bound on $\left\| \mathbf{\beta}_k^{(m)}_{D_k, D'_k} - \mathbf{\beta}_k^{(m)}_{D'_k, D_k} \right\|$ and is computed as in the following lemma.

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

$$\Delta_{k,2} = \frac{c_1}{\rho |N_k| N_k}.$$

**Proof.** See Appendix C.

In the next subsection, we present our main result proving that the proposed D-ZOA algorithm is differentially private.

### C. Intrinsic (ε, δ)-Differential Privacy Guarantee

In this section, we prove that, at each iteration of Algorithm [11] (ε, δ)-differential privacy is guaranteed.

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

$$\sigma_k = \frac{c_1}{\rho |N_k| N_k} \sqrt{\frac{2.1 \log(1.25/\delta)}{\epsilon TJP}}.$$

Under Assumption 3, at each iteration of D-ZOA, (ε, δ)-differential privacy is guaranteed. Specifically, for any neighboring datasets $D_k$ and $D'_k$ and any output $\mathbf{\beta}_k^{(m)}$, the following inequality holds:

$$\Pr[\beta_k^{(m)}] \leq e^\epsilon \Pr[\beta_k^{(m)}] + \delta.$$

**Proof.** See Appendix D.

Theorem 1 also shows that the variance of the inherent noise is inversely proportional to the privacy parameter $\epsilon$. This implies that a higher variance leads to a smaller $\epsilon$ and higher privacy guarantee. In fact, a smaller $\epsilon$ implies that the ratio of the probability distributions of $\mathbf{\beta}_k^{(m)}_{D_k}$ and $\mathbf{\beta}_k^{(m)}_{D'_k}$ is smaller, which means less information is available to a sniffing/spoofing adversary through $\mathbf{\beta}_k$ hence the improved privacy [14].

The following corollary shows the connection between the privacy parameter $\epsilon$ and the number of samples $J$ in the most private case of the variance $\sigma_k$ approaching its upper bound.

**Corollary 1.** If $\{\mathbf{g}_j^{(t)}\}_{j=1}^J$ are i.i.d., $J$ is sufficiently large, and Assumption 3 holds, we have

$$\epsilon = \frac{c_1}{\rho |N_k| N_k} \sqrt{\frac{2.1 JP \log(1.25/\delta)}{\epsilon}} \times \left( \frac{c_1}{\rho |N_k| N_k} \sqrt{\frac{2.1 JP \log(1.25/\delta)}{\epsilon}} \right)^{-\frac{1}{2}}.$$

**Proof.** The proof follows from equating the expression for $\sigma_k$ in Theorem 1, (25), and the expression for the trace of the covariance of $\mathbf{\beta}_k^{(m)}$ in Lemma 1 where the trace of $\mathbf{\Psi}_k^{(t)}$ has been replaced by the upper-bound derived in Lemma 2. Solving the resultant equation for $\epsilon$ yields (27).

Corollary 1 provides the highest possible level of privacy guarantee that can be ensured by D-ZOA due to its inherent randomness brought about by using a zeroth-order method in the inner loop.

### D. 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 [26] 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
\[
\sigma_k = \frac{c_1 \sqrt{2.1 \log(1.25/\delta)}}{\rho |N_k| \epsilon}.
\] (28)

Under Assumption 3, Algorithm [7] guarantees $(\epsilon, \delta)$-differential privacy where
\[
\bar{\epsilon} = c_2 \sqrt{\frac{M \log(1/\delta)}{1.05 \log(1.25/\delta)}}.
\] (29)

Proof. The proof is obtained by using the log moments of the privacy loss and their linear composability in the same way as in [23, Theorem 2]. □

V. CONVERGENCE ANALYSIS

The convergence of D-ZOA to the centralized solution is established by corroborating that both inner and outer loops of the algorithm converge. The convergence of the inner loop can be verified following [25, Theorem 2], i.e., it can be shown that, under Assumptions 1 and 2, there exists a constant $c$ such that, for each $T$ representing a fixed number of inner-loop iterations, the following inequality holds:
\[
\mathbb{E}[\mathcal{F}(\hat{w}(T)) - \mathcal{F}(w^*)] \leq c R \sqrt{T} \left( \max\{\alpha_0, \alpha_0^{-1}\} \sqrt{\log(2P)} + \frac{u_1 \log(2T)}{\sqrt{T}} \right)
\] (30)

where
\[
\hat{w}(T) = \frac{1}{T} \sum_{t=1}^{T} w(t).
\]

In [25], it is shown that $c = 0.5$ is suitable when $\nu_1$ and $\nu_2$ are sampled from a normal distribution. The convergence of the outer loop can be proven by verifying the convergence of a fully distributed ADMM with inexact primal updates.

To present the convergence result, we construct the auxiliary sequence
\[
r^{(m)} = \sum_{s=0}^{m} Q_w(s)
\]
and define the auxiliary vector $q^{(m)}$ and the auxiliary matrix $G$ as
\[
q^{(m)} = \left[ r^{(m)}, w^{(m)} \right], G = \begin{bmatrix} \rho I_P & 0_{P \times P} \ \ 0_{P \times P} & \rho \mathbf{L}_w \end{bmatrix}.
\] (31)

The convergence results of [36], [17], and [37] can now be adapted to D-ZOA as per the following theorem that also provides an explicit privacy-accuracy trade-off.

Theorem 3. If $f(\cdot)$ is convex, $\{\theta_{j,k}\}_{j=1}^{J}$ are i.i.d., $J$ is sufficiently large, and Assumption 3 holds, for any $M > 0$, we have
\[
\mathbb{E}[f(\hat{w}(M)) - f(w^*)] \leq \frac{\|q^{(0)} - q\|^2}{M} + \frac{2.1c_2^2 P \log(1.25/\delta)}{2}\lambda_{\max}(\mathbf{L}_w) \sum_{k=1}^{K} N_k^2 \epsilon^2 \lambda_{\min}(\mathbf{L}_w)
\] (32)

where $q = [r^T, (w^*)^T]^T$ and
\[
\hat{w}(M) = \frac{1}{M} \sum_{m=1}^{M} \hat{w}(m).
\]

Proof. See Appendix E. □

Theorem 3 reveals a privacy-accuracy trade-off offered by D-ZOA. When the privacy guarantee is stronger (smaller $\epsilon$ and $\delta$), the accuracy is lower. It also shows that D-ZOA converges at a rate of $O(1/M)$ where $M$ is the number of iterations of the ADMM outer loop.

VI. SIMULATIONS

In this section, we present some simulated examples to evaluate the performance and the privacy-accuracy trade-off of the proposed D-ZOA algorithm. We benchmark the performance and the privacy-accuracy trade-off of D-ZOA against an existing differentially-private ADMM-based algorithm, called DP-ADMM and proposed in [23]. DP-ADMM is suitable for learning problems with non-smooth objective functions. It is a differentially-private algorithm that is not fully-distributed since it needs a central coordinator to average the dual variable and the perturbed primal variable over the network at every iteration. As for the application, we consider a distributed version of the empirical risk minimization problem defined by lasso [38].

The network-wide observations are represented by a design matrix $X \in \mathbb{R}^{N \times P}$ and a response vector $y \in \mathbb{R}^{N \times 1}$ where $N$ is the number of data samples and $P$ is the number of features in each sample. The matrix $X$ consists of $K$ submatrices $X_k$, i.e., $X = [X_1, X_2, \ldots, X_K]^T$, and the vector $y$ consists of $K$ subvectors $y_k$, i.e., $y = [y_1^T, y_2^T, \ldots, y_K^T]^T$, as the data is distributed among the agents and each agent $k$ holds its respective $X_k \in \mathbb{R}^{N_k \times P}$ and $y_k \in \mathbb{R}^{N_k \times 1}$ where $N = \sum_{k=1}^{K} N_k$. The parameter vector that establishes a linear regression between $X$ and $y$ is $\beta \in \mathbb{R}^{P \times 1}$. In the centralized approach, a lasso estimate of $\beta$ is given by
\[
\beta^c = \arg\min_{\beta} \{\|X \beta - y\|^2 + \eta \|\beta\|_1\}.
\] (33)

In the distributed setting, we solve problem (2) with
\[
\sum_{j=1}^{N_k} \ell(x_{k,j}, y_{k,j}; \beta_k) = \|X_k \beta_k - y_k\|^2
\] (34)

\[
R(\beta_k) = \|\beta_k\|_1.
\]

We assess the performance of the D-ZOA algorithm over a network of $K = 5$ agents with the topology as shown in Fig. 1. Topology of the considered multi-agent network.
Fig. 1. The number of samples at each agent is set to $N_k = 20 \ \forall k \in K$ and the total number of samples is $N = 100$. The number of features in each sample is $P = 10$. For each agent $k \in K$, we create a $2P \times P$ local observation matrix $X_k$ whose entries are i.i.d. zero-mean unit-variance Gaussian random variables. The response vector $y$ is synthesized as

$$y = X\omega + \psi$$

where $\omega \in \mathbb{R}^P$ and $\psi \in \mathbb{R}^M$ are random vectors with distributions $\mathcal{N}(0, I_P)$ and $\mathcal{N}(0, 0.1I_N)$, respectively. The data are preprocessed by normalizing the columns of $X$ 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 [23]. 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. To apply the central limit theorem, $J$ needs to be sufficiently large, i.e., $J \geq 30$. We set $J = 30$ and calculate $\alpha_0$ according to (27).

Performance of D-ZOA is evaluated using the normalized error between the centralized solutions $\beta^c$ as per (33) and the local estimates. It is defined as $\sum_{k=1}^{\mathcal{K}} \| \beta_k - \beta^c \|^2 / \| \beta^c \|^2$ where $\beta_k$ denotes the local estimate at agent $k$. The centralized solution $\beta^c$ is computed using the convex optimization toolbox CVX.

Fig. 2. Normalized error of DP-ADMM and D-ZOA for two values of $\epsilon$ and fixed $\delta$.

Fig. 3. Privacy-accuracy trade-off.
Results are obtained by averaging over 100 independent trials.

In Fig. 2, we plot the normalized error versus the outer loop iteration index for both D-ZOA and DP-ADMM. The plots show that both algorithms converge to the centralized solution for two different values of $\epsilon$ and $\delta$. In all plots, accuracy improves as $\epsilon$ increases. This is consistent with both Theorem 3 and [23] Theorem 3. The hyper-parameters in DP-ADMM are tuned to achieve the best accuracy and convergence rate. The faster convergence of DP-ADMM is mainly due to its extra centralized processing. However, D-ZOA has higher accuracy than DP-ADMM.

In Fig. 3, we illustrate the privacy-accuracy trade-off for both D-ZOA and DP-ADMM. The figures show that D-ZOA and DP-ADMM achieve higher accuracy with larger $\epsilon$ and $\delta$. In Fig. 3(a), we show the normalized error versus the privacy parameter $\bar{\epsilon}$ as given in [28] for $\delta = 10^{-6}$ and $\delta = 10^{-3}$. We observe that D-ZOA outperforms DP-ADMM in terms of accuracy likely due to its intrinsic privacy-preserving properties. Fig. 3(b) also attests to the superiority of D-ZOA over DP-ADMM when $\epsilon = 0.15$ and $\epsilon = 0.95$ and $\delta$ varies between $10^{-6}$ and $10^{-2}$.

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 makes 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 D-ZOA iterations. We verified the convergence of D-ZOA to the 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 $E[\beta_k] = \bar{\beta}_k$. Second, we calculate the covariance of $\beta_k$.

We prove that $E[\beta_k] = \bar{\beta}_k$ based on induction over $m$.

**Base case:** Since $\beta_k(0) = \bar{\beta}_k(0) = 0$, we have $E[\beta_k(0)] = \bar{\beta}_k(0)$.

**Induction step:** We assume that $E[\beta_k(m-1)] = \bar{\beta}_k(m-1)$ as the induction hypothesis. Considering (18) and (15), we have

$$E[\beta_k(m)] = E[\beta_k(m) + E[\xi_k[m]]$$

$$= \beta_k(m) + E[\xi_k[m]]$$

$$= -\frac{1}{\rho(N_e)} \nabla f(\beta_k) + \frac{1}{2\rho(N_e)} \left( |N_e| \bar{\gamma}_k(m-1) + \sum_{l \in N_e} \bar{\gamma}_l(m-1) - \frac{1}{2\rho} |N_e| \bar{\gamma}_k(m-1) + E[\xi_k(m)]$$

$$= -\frac{1}{2\rho(N_e)} E[\nabla f_k(\beta_k)]$$

$$+ \frac{1}{2\rho(N_e)} \left( |N_e| \bar{\gamma}_k(m-1) + \sum_{l \in N_e} E[\gamma_l(m-1)] \right)$$

$$- \frac{1}{2\rho(N_e)} E[\gamma_k(m-1)] + E[\xi_k(m)]$$

$$= E[\beta_k(m)] + E[\xi_k(m)]$$

which implies that $E[\xi_k(m)] = 0$. Therefore, $E[\beta_k(m)] = \bar{\beta}_k(m)$.

Since the sequence $\{g_t\}_{t=1}^{T}$ is i.i.d. and $T$ is sufficiently large, thanks to the central limit theorem [40], we have

$$E[\beta_k(m)] = \frac{1}{T} \sum_{t=1}^{T} \beta_t \sim \mathcal{N}(\mu_t, \frac{1}{T})$$

In view of the additive property of the normal distribution and recalling that $\beta_k(m) = -\sum_{t=1}^{T} \alpha_t \beta_t$, we have

$$\text{cov}[\beta_k(m)] = \frac{1}{T} \sum_{t=1}^{T} \alpha_t^2 \Psi_k(t)$$

We also know $E[\beta_k(m)] = \bar{\beta}_k(m)$. Therefore, $\beta_k(m)$ is distributed as

$$\beta_k(m) \sim \mathcal{N}(\bar{\beta}_k(m), \frac{1}{T} \sum_{t=1}^{T} \alpha_t^2 \Psi_k(t))$$

APPENDIX B

PROOF OF LEMMA 2

Proof. It is easy to verify that

$$\text{tr}(\Psi(t)) = \text{tr}(E[g_{j,k}(t)g_{j,k}(t)^T] - E[g_{j,k}(t)]E[g_{j,k}(t)^T])$$

$$= \text{tr}(E[g_{j,k}(t)g_{j,k}(t)^T]) - \|E[g_{j,k}(t)]\|^2$$

$$= \text{tr}(E[g_{j,k}(t)g_{j,k}(t)^T]) - \|E[\mu(t)]\|^2$$

$$= E[\|g_{j,k}(t)\|^2] - \|E[\mu(t)]\|^2.$$ (37)

By [25] Lemma 2, there exists a constant $c$ such that

$$E[\|g_{j,k}(t)\|^2] \leq cL^2 P\left( \frac{\sqrt{u_{2,t}^2 u_{1,t}}}{u_{1,t}} P + 1 + \log(P) \right).$$

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

$$E[\|g_{j,k}(t)\|^2] \leq cL^2 P\left( \frac{1}{\sqrt{t}} + 1 + \log(P) \right).$$ (38)
In addition, from $\beta_k^{(m)} = -\sum_{t=1}^T \alpha_t g_k^{(t)}$ and (15), we have

$$\tilde{\beta}_k^{(m)} = -\sum_{t=1}^T \alpha_t \mu_k^{(t)}.$$  

(40)

Taking the Euclidean norm of both sides in (40) and using the triangle inequality, we have

$$\|\tilde{\beta}_k^{(m)}\| = \left\| -\sum_{t=1}^T \alpha_t \mu_k^{(t)} \right\| \leq \sum_{t=1}^T |\alpha_t| \|\mu_k^{(t)}\|.$$  

(41)

Squaring both sides of (41) and using the Cauchy-Schwarz inequality, we get

$$\|\tilde{\beta}_k^{(m)}\|^2 \leq \sum_{t=1}^T |\alpha_t|^2 \|\mu_k^{(t)}\|^2$$  

(42)

and consequently

$$-\frac{1}{JP} \sum_{t=1}^T \alpha_t^2 \|\mu_k^{(t)}\|^2 \leq -\frac{1}{TJP} \|\tilde{\beta}_k^{(m)}\|^2.$$  

(43)

Using (39), (43), and the definition of $\alpha_t$ after (11), we have

$$\frac{1}{JP} \sum_{t=1}^{T-1} \alpha_t^2 \text{tr}(\Psi_k^{(t)})$$

$$\leq \frac{1}{JP} \sum_{t=1}^{T-1} \alpha_t^2 c \kappa^2 \eta \left( \frac{1}{\sqrt{k}} + 1 + \log(P) \right)$$

$$- \frac{1}{JP} \sum_{t=1}^{T-1} \alpha_t^2 \|\mu_k^{(t)}\|^2$$

(44)

$$= \frac{c \alpha_0^2 R^2}{JP \log(2P)} \left( \sum_{t=1}^P \frac{1}{\sqrt{t}} + (1 + \log(P)) \sum_{t=1}^T \frac{1}{t} \right)$$

$$- \frac{1}{TJP} \|\tilde{\beta}_k^{(m)}\|^2.$$  

Defining $s_1 = \sum_{t=1}^{T-1} t^{-1}$ and $s_2 = \sum_{t=1}^{T-1} t^{-1.5}$, (44) simplifies to

$$\frac{1}{JP} \sum_{t=1}^T \alpha_t^2 \text{tr}(\Psi_k^{(t)})$$

$$\leq \frac{c \alpha_0^2 R^2}{JP \log(2P)} \left( s_1 (1 + \log(P)) + s_2 \right) - \frac{\|\tilde{\beta}_k^{(m)}\|^2}{TJP}.$$  

(45)

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

$$\|\tilde{\beta}_k^{(m)}\| = \|\beta^c\| \leq \|\tilde{\beta}_k^{(m)} - \beta^c\| \leq \|\beta^c\|,$$  

(46)

which implies $\|\tilde{\beta}_k^{(m)}\| \leq 2 \|\beta^c\|$. Therefore, we obtain

$$\frac{1}{JP} \sum_{t=1}^T \alpha_t^2 \text{tr}(\Psi_k^{(t)})$$

$$\leq \frac{c \alpha_0^2 R^2}{JP \log(2P)} \left( s_1 (1 + \log(P)) + s_2 \right) - \frac{4 \|\beta^c\|^2}{TJP}.$$  

(47)

**Appendix C**

**Proof of Lemma 3**

**Proof.** From the adopted exact primal update equation (18), we obtain

$$\tilde{\beta}_k^{(m)} = -\frac{0.5}{\rho |N_k|} \left( \frac{1}{N_k} \sum_{j=1}^{N_k} \nabla\ell(x_{k,j}, y_{k,j}; \tilde{\beta}_k) + \gamma_k^{(m-1)} \right)$$

$$+ \frac{0.5}{|N_k|} \left( \tilde{\beta}_k^{(m-1)} - \sum_{i \in N_k} \beta_i^{(m-1)} + \eta \nabla R(\tilde{\beta}_k) \right)$$

$$= -\frac{0.5}{\rho |N_k|} \left( \frac{1}{N_k} \sum_{j=1}^{N_k} \nabla\ell(x_{k,j}, y_{k,j}; \tilde{\beta}_k) + \gamma_k^{(m-1)} \right)$$

$$+ \frac{0.5}{|N_k|} \left( \tilde{\beta}_k^{(m-1)} - \sum_{i \in N_k} \beta_i^{(m-1)} + \eta \nabla R(\tilde{\beta}_k) \right).$$

(48)

Using Assumption 3, the quantity $\|\tilde{\beta}_k^{(m)} - \tilde{\beta}_k^{(m)}\|$ is upper bounded as follows

$$\left\| \tilde{\beta}_k^{(m)} - \tilde{\beta}_k^{(m)} \right\| = \left\| \nabla\ell(x_{k,N_k}, y_{k,N_k}; \tilde{\beta}_k) - \nabla\ell(x_{k,N_k}, y_{k,N_k}; \tilde{\beta}_k) \right\|$$

$$\leq \frac{c_1}{\rho |N_k|}.$$  

\[\square\]

**Appendix D**

**Proof of Theorem 1**

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

$$\log \left( \frac{\Pr(\tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)})}{\Pr(\tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)})} \right)$$

(50)

where the equality holds since the Jacobian matrix of the linear transformation from $\beta_k^{(m)}$ to $\xi_k^{(m)}$ is the identity matrix. Furthermore, as the entries of $\xi_k^{(m)}$, denoted by $\xi_{s,k}^{(m)}$, are independent of each other, for any entry $s$, we have

$$\log \left( \frac{\Pr(\xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)})}{\Pr(\xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)})} \right).$$

(51)

Hence, we have

$$\log \left( \frac{\Pr(\tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)})}{\Pr(\tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)}, \tilde{\beta}_k^{(m)})} \right) = \log \left( \frac{\Pr(\xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)})}{\Pr(\xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)}, \xi_{s,k}^{(m)})} \right).$$

(52)
Via the triangle inequality, (52) leads to
\[
\log \frac{\Pr[\beta_{k,D_k}^{(m)}]}{\Pr[\beta_{k,D_k}^{(m)\prime}]} \leq \frac{1}{2\sigma_k^2} \left( 2\xi_{s,k}^{(m)} \frac{\beta_{s,k,D_k}^{(m)}}{\beta_{s,k,D_k}^{(m)\prime}} - \frac{\beta_{s,k,D_k}^{(m)} - \beta_{s,k,D_k}^{(m)\prime}}{2} \right) + \frac{\sigma_k}{\sigma_D} \left( \frac{\beta_{s,k,D_k}^{(m)} - \beta_{s,k,D_k}^{(m)\prime}}{2} \right)^2.
\]  
(53)

Since \(|\nabla \ell(\cdot)| \leq c_1\), using Lemma 3, we have
\[
\left| \frac{\beta_{s,k,D_k}^{(m)} - \beta_{s,k,D_k}^{(m)\prime}}{\sigma_{k}} \right| < \frac{c_1}{\rho |N_k| N_k}.
\]  
(54)

Hence, substituting \(\sigma_k\) in (25) into (53), we obtain
\[
\log \frac{\Pr[\beta_{k,D_k}^{(m)}]}{\Pr[\beta_{k,D_k}^{(m)\prime}]} \leq \frac{\rho |N_k| N_k \epsilon^2}{2.1c_1 \log(1.25/\delta)} \frac{\xi_{s,k}^{(m)}}{2\rho |N_k| N_k} + \frac{c_1}{2\rho |N_k| N_k}.
\]  
(55)

When
\[
|\xi_{s,k}^{(m)}| \leq \frac{c_1}{\rho |N_k| N_k} (2.1e^{-1} \log(1.25/\delta) - 0.5),
\]  
the privacy loss is bounded by \(\epsilon\). Hence, let us define
\[
r = \frac{c_1}{\rho |N_k| N_k} (2.1e^{-1} \log(1.25/\delta) - 0.5).
\]  

Subsequently, we need to prove that
\[
\Pr[|\xi_{s,k}^{(m)}| > r] \leq \delta
\]  
or equivalently
\[
\Pr[|\xi_{s,k}^{(m)}| > r] \leq 0.5\delta.
\]

Using the tail bound of the normal distribution \(\mathcal{N}(0, \sigma^2_k)\), we obtain
\[
\Pr[|\xi_{s,k}^{(m)}| > r] \leq \frac{\sigma_k}{\sqrt{2\pi r^2 \exp\left(-\frac{r^2}{2\sigma_k^2}\right)}}.
\]  
(56)

Since \(\delta\) is assumed to be small (\(\leq 0.01\)) and \(\epsilon \leq 1\), we have
\[
\frac{\sigma_k}{\sqrt{2\pi}} \frac{r^2}{2\sigma_k^2} < \log(0.5\sqrt{2\pi\delta}).
\]

Therefore,
\[
\Pr[|\xi_{s,k}^{(m)}| > r] \leq 0.5\delta,
\]  
which implies
\[
\Pr[|\xi_{s,k}^{(m)}| > r] \leq \delta.
\]

By defining
\[
A_1 = \{ \xi_{s,k}^{(m)} : |\xi_{s,k}^{(m)}| \leq r \}
\]  
and
\[
A_2 = \{ \xi_{s,k}^{(m)} : |\xi_{s,k}^{(m)}| > r \},
\]  
we have
\[
\Pr[\beta_{k,D_k}^{(m)}] = \Pr[\beta_{s,k,D_k}^{(m)\prime} + \xi_{s,k}^{(m)} : |\xi_{s,k}^{(m)}| \leq r] + \Pr[\beta_{s,k,D_k}^{(m)\prime} + \xi_{s,k}^{(m)} : |\xi_{s,k}^{(m)}| > r],
\]  
(57)

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

\section*{Appendix E: Proof of Theorem 3}

\textbf{Proof.} Using the first-order condition for convexity \([30]\), we have
\[
f(\bar{w}^{(m)}) - f(w^*) \leq (\bar{w}^{(m)} - w^*)^T \nabla f(\bar{w}^{(m)}).
\]  
(58)

In addition, in virtue of \([36]\) Lemma 1 and Lemma 2, \(\bar{w}^{(m)}\) satisfies the following equation
\[
\nabla f(\bar{w}^{(m)}) = 2H\bar{\xi}^{(m)} - 2Qr^{(m)} - L_+(\bar{w}^{(m)} - w^{(m-1)}).
\]  
(59)

Therefore, by using \([58], [59], [46]\) Lemma 3, Lemma 4 and Lemma 5, and the steps in the proof of \([37]\) Theorem 1, we can show that, for any \(r \in \mathbb{R}^P\) and \(m > 0\), we have
\[
f(\bar{w}^{(m)}) - f(w^*) + 2r^T Q\bar{w}^{(m)}
\]
\[
\leq (\bar{w}^{(m)} - w^*)^T \left(-L_+(\bar{w}^{(m-1)} - \bar{w}^{(m-1)})
\right.
\]
\[
- L_+ (\bar{w}^{(m-1)} - w^{(m-1)}) - 2Q(r^{(m)} - r)
\]
\[
+ L_+ (\bar{w}^{(m)} - \bar{w}^{(m-1)})
\]
\[
= \left(\|q^{(m-1)} - dl\|^2_G - \|q^{(m)} - q_l\|^2_G - \|q^{(m)} - q^{(m-1)}\|^2_G\right) + \frac{\rho}{\rho} + (\bar{w}^{(m)} - w^*)^T L_+ (\bar{w}^{(m-1)} - \bar{w}^{(m-1)})
\]
\[
+ (\bar{w}^{(m)} - w^*)^T L_- (\bar{w}^{(m)} - \bar{w}^{(m-1)}) + 2(\bar{\xi}^{(m)})^T Q(r^{(m)} - r)
\]
\[
= \left(\|q^{(m)} - dl\|^2_G - \|q^{(m)} - q_l\|^2_G + (\bar{w}^{(m)} - w^*)^T L_+ (w^{(m-1)} - \bar{w}^{(m-1)})
\right)
\]
\[
- 2(\bar{\xi}^{(m)})^T Q(r^{(m)} - r)
\]
\[
\leq \left(\|q^{(m)} - dl\|^2_G - \|q^{(m)} - q_l\|^2_G + 2(\bar{\xi}^{(m)})^T Q(r^{(m)} - r)
\right)
\]
\[
\leq \left(\|q^{(m)} - dl\|^2_G - \|q^{(m)} - q_l\|^2_G + 2(\bar{\xi}^{(m)})^T Q(r^{(m)} - r)
\right)
\]  
(60)

where \(q = [r^T, (w^*)^T]^T\).

For any symmetric matrix \(X \in \mathbb{R}^{P \times P}\) and vector \(y \in \mathbb{R}^P\), we have
\[
\|y\|^2 \lambda_{\min}(X) \leq y^T X y \leq \|y\|^2 \lambda_{\max}(X)
\]  
and, for any \(a, b \in \mathbb{R}^P\) and \(\tau \in \mathbb{R}^+\), we have
\[
2a^T b \leq \tau^{-1} \|a\|^2 + 2\|b\|^2.
\]  

Therefore, \(60\) yields
\[
f(\bar{w}^{(m)}) - f(w^*) + 2r^T Q\bar{w}^{(m)}
\]
\[
\leq \left(\|q^{(m-1)} - dl\|^2_G - \|q^{(m)} - q_l\|^2_G - \|Q\xi^{(m)}\|^2\right) + \frac{\lambda_{\min}(L_-)}{2} \left(\|\bar{w}^{(m)} - w^*\|^2 + \frac{1}{\tau} \|\bar{L}_+(\bar{w}^{(m)} - w^*)\|^2\right)
\]
\[
+ \tau \left(\|w^{(m-1)} - \bar{w}^{(m-1)}\|^2 + 2(\bar{\xi}^{(m)})^T Q(r^{(m)} - r)\right).
\]  
(61)

By setting
\[
\tau = \frac{\lambda_{\max}(L_+)}{2\lambda_{\min}(L_-)},
\]  

\(\square\)
Setting $r = 0_r$ and summing both sides of (62) over $m = 1$ to $M$ gives
\begin{equation}
\frac{1}{M} \sum_{m=1}^{M} (f(\hat{w}(m)) - f(w^*)) \leq \frac{1}{\rho} \left\| q(0) - q \right\|_G^2 + \frac{\lambda_2^2(L_+)}{2\lambda_{\min}(L_-)} \sum_{m=1}^{M} \left\| \xi^{(m-1)} \right\|_2^2 \tag{63}\end{equation}

Using Jensen’s inequality [40], (22), (25), and applying the expectation operator to both sides of (63), we obtain
\begin{equation}
\mathbb{E}[f(\hat{w}(M)) - f(w^*)] \leq \frac{1}{M} \left\| q(0) - q \right\|_G^2 + \frac{\lambda_2^2(L_+)}{2M\lambda_{\min}(L_-)} \sum_{m=1}^{M} \mathbb{E} \left[ \left\| \xi^{(m-1)} \right\|_2^2 \right] \leq \frac{1}{M} \left\| q(0) - q \right\|_G^2 + \frac{\lambda_2^2(L_+)}{2M\lambda_{\min}(L_-)} \sum_{m=1}^{M} \left( \rho \lambda_{\max}(L_+) \right) \leq \frac{1}{M} \left\| q(0) - q \right\|_G^2 + \frac{\lambda_2^2(L_+)}{2M\lambda_{\min}(L_-)} \sum_{m=1}^{M} P \sigma_k^2 \leq \frac{1}{M} \left\| q(0) - q \right\|_G^2 + \frac{2.1c^2 \rho \log(1.25/\delta) \lambda_{\max}(L_+)}{2\rho^2 \left\| N_k^2 - N_k^2 \right\|_2^2 \lambda_{\min}(L_-)} \tag{64}\end{equation}

where
\begin{equation}
\hat{w}(M) = \frac{1}{M} \sum_{m=1}^{M} \hat{w}(m). \tag{61}\end{equation}
[29] Y. Nesterov and V. Spokoiny, “Random gradient-free minimization of convex functions,” Found. Comput. Math., vol. 17, no. 2, pp. 527–566, Apr. 2017.

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

[31] 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.

[32] 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

[33] T. Kailath, A. H. Sayed, and B. Hassibi, “Linear estimation,” 2000.

[34] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” Foundations and Trends in Theoretical Computer Science, vol. 9, no. 34, pp. 211–407, Aug. 2014.

[35] 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.

[36] 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

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

[38] 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.

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

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