DP-ADMM: ADMM-Based Distributed Learning With Differential Privacy

Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong

***Abstract*— Alternating direction method of multipliers (ADMM) is a widely used tool for machine learning in distributed settings where a machine learning model is trained over distributed data sources through an interactive process of local computation and message passing. Such an iterative process could cause privacy concerns of data owners. The goal of this paper is to provide differential privacy for ADMM-based distributed machine learning. Prior approaches on differentially private ADMM exhibit low utility under high privacy guarantee and assume the objective functions of the learning problems to be smooth and strongly convex. To address these concerns, we propose a novel differentially private ADMM-based distributed learning algorithm called DP-ADMM, which combines an approximate augmented Lagrangian function with time-varying Gaussian noise addition in the iterative process to achieve higher utility for general objective functions under the same differential privacy guarantee. We also apply the moments accountant method to analyze the end-to-end privacy loss. The theoretical analysis shows that the DP-ADMM can be applied to a wider class of distributed learning problems, is provably convergent, and offers an explicit utility-privacy tradeoff. To our knowledge, this is the first paper to provide explicit convergence and utility properties for differentially private ADMM-based distributed learning algorithms. The evaluation results demonstrate that our approach can achieve good convergence and model accuracy under high end-to-end differential privacy guarantee.**

***Index Terms*— Machine learning, ADMM, distributed algorithms, privacy, differential privacy, and moments accountant.**

I. Introduction

DISTRIBUTED machine learning is a widely adopted approach due to the high demand of large-scale and distributed data processing. It allows multiple entities to keep their datasets unexposed, and meanwhile to collaborate in a common learning objective (usually formulated as a regularized empirical risk minimization problem) by iterative local computation and message passing. Therefore, distributed machine learning helps to reduce computational burden and improves both robustness and scalability of data processing. As pointed out in recent studies [1], [2], existing approaches to decentralizing an optimization problem mainly consist of subgradient-based algorithms [3], [4], alternating direction method of multipliers (ADMM) based algorithms [5]–[8], and composite of sub-gradient descent and ADMM [9]. It has been shown that ADMM-based algorithms can converge at the rate of $O(1/t)$ while subgradient-based algorithms typically converge at the rate of $O(1/\sqrt{t})$, where t is the number of iterations [10]. Therefore, ADMM has become a popular method for designing distributed versions of a machine learning algorithm [5], [8], [11], and our work focuses on ADMM-based distributed learning.

With ADMM, the learning problem is divided into several sub-problems solved by agents independently and locally, and only intermediate parameters need to be shared. However, the iterative process of ADMM involves privacy leakage, and the adversary can obtain the sensitive information from the shared model parameters as shown in [12], [13]. Thus, we aim to limit the privacy leakage during the iterative process of ADMM using differential privacy. Differential privacy is a widely used privacy definition [14]–[16] and can be guaranteed in ADMM through adding noise to the exchanged messages. However, in existing studies on ADMM-based distributed learning with differential privacy [1], [2], [17]–[19], noise addition would disrupt the learning process and severely degrade the performance of the trained model, especially when large noise is needed to provide high privacy protection. Besides, their privacy-preserving algorithms only apply to the learning problems with both smoothness and strongly convexity assumptions about the objective functions. Such weaknesses and limitations motivate us to explore further in this area.

In this paper, we mainly focus on using ADMM to enable distributed learning while guaranteeing differential privacy, and propose a novel differentially private ADMM-based distributed learning algorithm called DP-ADMM, which has good convergence properties, low computational cost, and an explicit and improved utility-privacy tradeoff, and can be applied to a wide class of distributed learning problems. The key algorithmic feature of DP-ADMM is the combination of an approximate augmented Lagrangian function and time-varying Gaussian noise addition in the iterative process,

Manuscript received August 30, 2018; revised March 16, 2019 and May 29, 2019; accepted July 9, 2019. Date of publication July 25, 2019; date of current version October 8, 2019. The work of R. Hu and Y. Gong was supported by the National Science Foundation under Grant CNS-1850523. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Damien Vergnaud. *(Correponding author: Yanmin Gong.)*

Z. Huang is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: zonghao.huang@okstate.edu).

R. Hu and Y. Gong are with the Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: rui.hu@my.utsa.edu; yanmin.gong@utsa.edu).

Y. Guo is with the Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: yuanxiong.guo@utsa.edu).

E. Chan-Tin is with the Department of Computer Science, Loyola University Chicago, Chicago, IL 60660 USA (e-mail: chantin@cs.luc.edu).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org provided by the authors. The material includes detailed algebraic manipulations of proofs. Contact yanmin.gong@utsa.edu for further questions about this work.

Digital Object Identifier 10.1109/TIFS.2019.2931068

which enables the algorithm to be noise-resilient and provably convergent. The moments accountant method [20] is used to analyze the end-to-end privacy guarantee of DP-ADMM. We also rigorously analyze the convergence rate and utility bound of our approach. To our knowledge, this is the first paper to provide explicit convergence and utility properties for differentially private ADMM-based distributed learning algorithms.

The main contributions of this paper are summarized as follows:

1) We design a novel differentially private ADMM-based distributed learning algorithm called DP-ADMM, which combines an approximate augmented Lagrangian function with time-varying Gaussian noise addition in the iterative process to achieve higher utility for more general objective functions than prior works under the same differential privacy guarantee.
2) Different from previous studies providing only differential privacy guarantee for each iteration, we use the moments accountant method to analyze the total privacy loss and provide a tight end-to-end differential privacy guarantee for DP-ADMM.
3) We provide rigorous convergence and utility analysis of the proposed DP-ADMM. To our knowledge, this is the first paper to provide explicit convergence and utility properties for differentially private ADMM-based distributed learning algorithms.
4) We conduct extensive simulations based on real-world datasets to validate the effectiveness of DP-ADMM in distributed learning settings.

The rest of the paper is organized as follows. In Section II, we present our problem statement. In Section III, we describe a differentially private standard ADMM-based algorithm and propose our DP-ADMM. In Section IV and Section V, we theoretically analyze our privacy guarantee and convergence and utility properties of DP-ADMM, respectively. The numerical results of DP-ADMM based on real-world datasets are shown in Section VI. Section VII discusses the related work, and Section VIII concludes the paper.

II. Problem Statement

In this section, we first introduce the problem setting. Then we present the standard ADMM-based distributed learning algorithm and discuss the associated privacy concern. A summary of notations used in this paper is listed in Table I.

A. Problem Setting

We consider a set of agents $[n] := \{1, \ldots, n\}$ and a central aggregator. Each agent $i \in [n]$ has a private training dataset $\mathcal{D}_i := \{(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}) : \forall j \in [m_i]\}$, where m_i is the number of training samples in the dataset $\mathcal{D}_i$, $\boldsymbol{a}_{i,j} \in \mathbb{R}^d$ is the d-dimensional data feature vector of the j-th training sample, and $\boldsymbol{b}_{i,j} \in \mathbb{R}^p$ is the corresponding p-dimensional data label. In this paper, we consider a star network topology where each agent can communicate with the central aggregator and the aggregator is responsible for message passing and aggregation. Note that our approach can be generalized to other network topologies where agents are connected with their neighbors without a central aggregator, as discussed in [1], [2], [17].

TABLE I
List of Notations

$\boldsymbol{a}_{i,j}$	Data feature vector
$\boldsymbol{b}_{i,j}$	Data label
$\ell(\cdot)$	Loss function
$R(\cdot)$	Regularizer function
λ	Regularizer parameter
$\ell'(\cdot)$	Subgradient of loss function
$R'(\cdot)$	Subgradient of regularizer
$\nabla\ell(\cdot)$	Gradient of loss function
$\nabla R(\cdot)$	Gradient of regularizer
$\boldsymbol{w}$	Global machine learning model
$\boldsymbol{w}_i$	Local learning model from agent i
$\boldsymbol{\gamma}_i$	Dual variable from agent i
ρ	Penalty parameter
$\mathcal{L}_\rho(\cdot)$	Augmented Lagrangian function
$\hat{\mathcal{L}}_{\rho,k}(\cdot)$	Approximate augmented Lagrangian function
$\boldsymbol{w}_i^k$	Primal variable from agent i in k-th iteration
$\tilde{\boldsymbol{w}}_i^k$	Noisy version of $\boldsymbol{w}_i^k$ after perturbation
$\boldsymbol{\gamma}_i^k$	Dual variable from agent i in k-th iteration
$\boldsymbol{w}^k$	Global variable in k-th iteration
$\boldsymbol{\xi}_i^k$	Sampled noise from agent i in k-th iteration
σ_i^2	Constant variance of Gaussian mechanism
η_i^k	Time-varying step size in k-th iteration
$\sigma_{i,k}^2$	Time-varying variance of Gaussian mechanism

The goal of our problem is to train a supervised learning model on the aggregated dataset $\{\mathcal{D}_i\}_{i\in[n]}$, which enables predicting a label for any new data feature vector. The learning objective can be formulated as the following regularized empirical risk minimization problem:

$$\min_{\boldsymbol{w}} \quad \sum_{i=1}^{n}\sum_{j=1}^{m_i}\frac{1}{m_i}\ell(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \boldsymbol{w}) + \lambda R(\boldsymbol{w}), \tag{1}$$

where $\boldsymbol{w} \in \mathbb{R}^{d\times p}$ is the trained machine learning model, $\ell(\cdot) : \mathbb{R}^d \times \mathbb{R}^p \times \mathbb{R}^{d\times p} \to \mathbb{R}$ is the loss function used to measure the quality of the trained model, $R(\cdot)$ refers to the regularizer function introduced to prevent overfitting, and $\lambda > 0$ is the regularizer parameter controlling the impact of regularizer. Note that the problem formulation (1) can represent a wide range of machine learning tasks by choosing different loss functions. For instance, the loss function of binary logistic regression is:

$$\ell(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \boldsymbol{w}) = \ln\left(1 + \exp(-\boldsymbol{b}_{i,j}\boldsymbol{w}^\intercal \boldsymbol{a}_{i,j})\right), \tag{2}$$

and the loss function of multi-class logistic regression is:

$$\ell(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \boldsymbol{w}) = \sum_{h=1}^{p}\boldsymbol{b}_{i,j}^{(h)}\ln\left(\frac{\sum_{l=1}^{p}\exp(\boldsymbol{w}^{(l)\intercal}\boldsymbol{a}_{i,j})}{\exp(\boldsymbol{w}^{(h)\intercal}\boldsymbol{a}_{i,j})}\right). \tag{3}$$

In this paper, we assume that the loss function $\ell(\cdot)$ and the regularizer function $R(\cdot)$ are both convex but not necessarily smooth. Throughout this paper, we use $\ell'(\cdot)$ and $R'(\cdot)$ to denote the sub-gradient of $\ell(\cdot)$ and $R(\cdot)$ respectively. When we consider smooth functions, we use $\nabla\ell(\cdot)$ and $\nabla R(\cdot)$ instead.

B. ADMM-Based Distributed Learning Algorithm

To apply ADMM, we re-formulate the problem (1) as:

$$\min_{\{\boldsymbol{w}_i\}_{i\in[n]}} \quad \sum_{i=1}^{n}\Big(\sum_{j=1}^{m_i}\frac{1}{m_i}\ell(\boldsymbol{a}_{i,j},\boldsymbol{b}_{i,j},\boldsymbol{w}_i)+\frac{\lambda}{n}R(\boldsymbol{w}_i)\Big), \tag{4a}$$

$$\text{s.t.} \quad \boldsymbol{w}_i=\boldsymbol{w}, i=1,\ldots,n, \tag{4b}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{d\times p}$ is the local model, and $\boldsymbol{w} \in \mathbb{R}^{d\times p}$ is the global one. The objective function (4a) is decoupled and each agent only needs to minimize the sub-problem associated with its dataset. Constraints (4b) enforce that all the local models reach consensus finally.

In standard ADMM, the augmented Lagrangian function associated with the problem (4) is:

$$\mathcal{L}_\rho(\boldsymbol{w},\{\boldsymbol{w}_i\}_{i\in[n]},\{\boldsymbol{\gamma}_i\}_{i\in[n]})=\sum_{i=1}^{n}\mathcal{L}_{\rho,i}(\boldsymbol{w}_i,\boldsymbol{w},\boldsymbol{\gamma}_i), \tag{5}$$

where

$$\mathcal{L}_{\rho,i}(\boldsymbol{w}_i,\boldsymbol{w},\boldsymbol{\gamma}_i)=\sum_{j=1}^{m_i}\frac{1}{m_i}\ell(\boldsymbol{a}_{i,j},\boldsymbol{b}_{i,j},\boldsymbol{w}_i)+\frac{\lambda}{n}R(\boldsymbol{w}_i) -\langle\boldsymbol{\gamma}_i,\boldsymbol{w}_i-\boldsymbol{w}\rangle+\frac{\rho}{2}\|\boldsymbol{w}_i-\boldsymbol{w}\|^2. \tag{6}$$

In (6), $\{\boldsymbol{\gamma}_i\}_{i\in[n]} \in \mathbb{R}^{d\times p\times n}$ are the dual variables associated with constraints (4b) and $\rho > 0$ is the penalty parameter. The standard ADMM solves the problem (4) in a Gauss-Seidel manner by minimizing (5) w.r.t. $\{\boldsymbol{w}_i\}_{i\in[n]}$ and $\boldsymbol{w}$ alternatively followed by a dual update of $\{\boldsymbol{\gamma}_i\}_{i\in[n]}$. The ADMM-based distributed algorithm is shown in Algorithm 1.

Algorithm 1 ADMM-Based Distributed Algorithm

Initialize $\boldsymbol{w}^0$, $\{\boldsymbol{w}_i^0\}_{i\in[n]}$, and $\{\boldsymbol{\gamma}_i^0\}_{i\in[n]}$;
for $k=1,2,\ldots,t$ **do**
  for $i=1,2,\ldots,n$ **do**
    $\boldsymbol{w}_i^k \leftarrow \arg\min_{\boldsymbol{w}_i}\mathcal{L}_{\rho,i}(\boldsymbol{w}_i,\boldsymbol{w}^{k-1},\boldsymbol{\gamma}_i^{k-1})$;
  end for
  $\boldsymbol{w}^k \leftarrow \frac{1}{n}\sum_{i=1}^n \boldsymbol{w}_i^k - \frac{1}{n}\sum_{i=1}^n \boldsymbol{\gamma}_i^{k-1}/\rho$;
  for $i=1,2,\ldots,n$ **do**
    $\boldsymbol{\gamma}_i^k \leftarrow \boldsymbol{\gamma}_i^{k-1}-\rho(\boldsymbol{w}_i^k-\boldsymbol{w}^k)$.
  end for
end for

C. Privacy Concern

In Algorithm 1, the intermediate parameters $\{\boldsymbol{w}_i^k\}_{i\in[n],k\in[t]}$ need to be shared with the aggregator, which may reveal the agents' private information as demonstrated by model inversion attacks [13]. Thus, we need to develop privacy-preserving methods to control such information leakage. The main goal of this paper is to provide privacy protection against inference attacks from an adversary, who tries to infer sensitive information about the agents' private datasets from the shared messages. We assume that the adversary can neither intrude into the local datasets nor have access to the datasets directly. The adversary could be an outsider who eavesdrops the shared messages, or the honest-but-curious aggregator who follows the protocol honestly but tends to infer the sensitive information. We do not assume any trusted third party, thus a privacy-preserving mechanism should be applied locally by each agent to provide privacy protection.

In order to provide privacy guarantee against such attacks, we define our privacy model formally by the notion of differential privacy [14]. Specifically, we adopt the (ϵ,δ)-differential privacy defined as follows:

Definition 1 ((ϵ,δ)-Differential Privacy): A randomized mechanism $\mathcal{M}$ is (ϵ,δ)-differentially private if for any two neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ differing in only one tuple, and for any subsets of outputs $\mathcal{O}\subseteq$ range($\mathcal{M}$):

$$\Pr[\mathcal{M}(\mathcal{D})\in\mathcal{O}]\le e^\epsilon\cdot\Pr[\mathcal{M}(\mathcal{D}')\in\mathcal{O}]+\delta, \tag{7}$$

which means, with probability of at least $1-\delta$, the ratio of the probability distributions for two neighboring datasets is bounded by e^ϵ.

In Definition 1, the parameters δ and ϵ are privacy budgets indicating the strength of privacy protection from the mechanism. Smaller ϵ or δ indicates better privacy protection. Gaussian mechanism is a common randomization method used to guarantee (ϵ,δ)-differential privacy, where noise sampled from normal distribution is added to the output. In this paper, we use $\mathcal{MN}_{d,p}(0,\sigma^2\mathbf{I}_d,\sigma^2\mathbf{I}_p)$ to denote the matrix normal distribution with variance σ^2.

III. ADMM With Differential Privacy

In this section, we achieve differential privacy under the framework of ADMM. First, we introduce an intuitive method by directly combining standard ADMM and primal variable perturbation (PVP) and discuss the weaknesses of this method. Then we propose our new approach to achieving differential privacy in ADMM with an improved utility-privacy tradeoff.

Algorithm 2 ADMM With PVP

Initialize $\boldsymbol{w}^0$, $\{\boldsymbol{w}_i^0\}_{i\in[n]}$, and $\{\boldsymbol{\gamma}_i^0\}_{i\in[n]}$.
for $k=1,2,\ldots,t$ **do**
  for $i=1,2,\ldots,n$ **do**
    $\boldsymbol{w}_i^k \leftarrow \arg\min_{\boldsymbol{w}_i}\mathcal{L}_{\rho,i}(\boldsymbol{w}_i,\boldsymbol{w}^{k-1},\boldsymbol{\gamma}_i^{k-1})$.
    $\tilde{\boldsymbol{w}}_i^k \leftarrow \boldsymbol{w}_i^k+\mathcal{MN}_{d,p}(0,\sigma_i^2\mathbf{I}_d,\sigma_i^2\mathbf{I}_p)$.
  end for
  $\boldsymbol{w}^k \leftarrow \frac{1}{n}\sum_{i=1}^n \tilde{\boldsymbol{w}}_i^k - \frac{1}{n}\sum_{i=1}^n \boldsymbol{\gamma}_i^{k-1}/\rho$.
  for $i=1,2,\ldots,n$ **do**
    $\boldsymbol{\gamma}_i^k \leftarrow \boldsymbol{\gamma}_i^{k-1}-\rho(\tilde{\boldsymbol{w}}_i^k-\boldsymbol{w}^k)$.
  end for
end for

A. ADMM With Primal Variable Perturbation (PVP)

As described in Section II, we need to use a local privacy-preserving mechanism in order to guarantee (ϵ,δ)-differential privacy for each agent. An intuitive way to achieve this goal is to combine the primal variable perturbation mechanism (PVP) and standard ADMM directly as proposed in [17]. Specifically, as given in Algorithm 2, at the k-th iteration, after obtaining the local primal variable $\boldsymbol{w}_i^k$, we apply Gaussian mechanism

Algorithm 3 DP-ADMM

Initialize $\boldsymbol{w}^0$, $\{\tilde{\boldsymbol{w}}_i^0\}_{i\in[n]}$, and $\{\boldsymbol{\gamma}_i^0\}_{i\in[n]}$.
for $k = 1, 2, \ldots, t$ **do**
  for $i = 1, 2, \ldots, n$ **do**
    $\boldsymbol{w}_i^k \leftarrow \arg\min_{\boldsymbol{w}_i} \ \hat{\mathcal{L}}_{\rho,k,i}(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k-1}, \boldsymbol{w}^{k-1}, \boldsymbol{\gamma}_i^{k-1})$.
    $\boldsymbol{\xi}_i^k \leftarrow \mathcal{MN}_{d,p}(0, \sigma_{i,k}^2\mathbf{I}_d, \sigma_{i,k}^2\mathbf{I}_p)$.
    $\tilde{\boldsymbol{w}}_i^k \leftarrow \boldsymbol{w}_i^k + \boldsymbol{\xi}_i^k$.
  end for
  $\boldsymbol{w}^k \leftarrow \frac{1}{n}\sum_{i=1}^n \tilde{\boldsymbol{w}}_i^k - \frac{1}{n}\sum_{i=1}^n \boldsymbol{\gamma}_i^{k-1}/\rho$.
  for $i = 1, 2, \ldots, n$ **do**
    $\boldsymbol{\gamma}_i^k \leftarrow \boldsymbol{\gamma}_i^{k-1} - \rho(\tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^k)$.
  end for
end for

with a pre-defined variance σ_i^2 to perturb it and share the noisy primal variable $\tilde{\boldsymbol{w}}_i^k$, which can guarantee differential privacy. According to [21], [22], by assuming the smoothness of loss function $l(\cdot)$ and regularizer function $R(\cdot)$, strongly convexity of regularizer $R(\cdot)$, and the bounded l_2 norm of the derivative of loss function by c_1, the l_2 sensitivity of $\boldsymbol{w}_i^k$ update function in standard ADMM is $2\ c_1/\big(m_i(\lambda/n+\rho)\big)$ as proved in Appendix A in Supplementary Material. Therefore, the noise magnitude $\sigma_i = 2\ c_1\sqrt{2\ln(1.25/\delta)}/\big((\lambda/n+\rho)m_i\epsilon\big)$ can achieve (ϵ, δ)-differential privacy in each iteration.

However, the added noise from the perturbation mechanism would disrupt the learning process, break the convergence property of the iterative process, and lead to a trained model with poor performance. This is especially the case when the privacy budget is small. Specifically, when the iteration number k is large, the trained model would keep changing dramatically due to the existence of large noise. Besides, the above perturbation method can only be applied when the objective function is smooth and the regularizer is strongly convex [17], [22]. In order to address such problems, we need to consider an alternative way to preserving differential privacy of ADMM-based distributed learning algorithms.

B. Our Approach

Our approach is inspired by the intuition that it is not necessary to solve the problem up to a very high precision in each iteration in order to guarantee the overall convergence. In our approach, instead of using the exact augmented Lagrangian function, we employ its first-order approximation with a scalar l_2-norm prox-function. Here we define:

$$\begin{aligned}
&\hat{\mathcal{L}}_{\rho,k,i}(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k-1}, \boldsymbol{w}, \boldsymbol{\gamma}_i) \\
&= \sum_{j=1}^{m_i}\frac{1}{m_i}\ell(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \tilde{\boldsymbol{w}}_i^{k-1}) + \frac{\lambda}{n}R(\tilde{\boldsymbol{w}}_i^{k-1}) \\
&\quad + \Big\langle \sum_{j=1}^{m_i}\frac{1}{m_i}\ell'(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \tilde{\boldsymbol{w}}_i^{k-1}) + \frac{\lambda}{n}R'(\tilde{\boldsymbol{w}}_i^{k-1}), \boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{k-1}\Big\rangle \\
&\quad - \big\langle \boldsymbol{\gamma}_i, \boldsymbol{w}_i - \boldsymbol{w}\big\rangle + \frac{\rho}{2}\|\boldsymbol{w}_i - \boldsymbol{w}\|^2 + \frac{\|\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{k-1}\|^2}{2\eta_i^k}, \qquad (8)
\end{aligned}$$

where $\eta_i^k \in \mathbb{R}$ is the time-varying step size, and it decreases as the iteration number k increases.

The proposed approximate augmented Lagrangian function used in our approach is defined by:

$$\begin{aligned}
&\hat{\mathcal{L}}_{\rho,k}(\{\boldsymbol{w}_i\}_{i\in[n]}, \{\tilde{\boldsymbol{w}}_i^{k-1}\}_{i\in[n]}, \boldsymbol{w}, \{\boldsymbol{\gamma}_i\}_{i\in[n]}) \\
&\qquad = \sum_{i=1}^n \hat{\mathcal{L}}_{\rho,k,i}(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k-1}, \boldsymbol{w}, \boldsymbol{\gamma}_i). \qquad (9)
\end{aligned}$$

Our approach minimizes (9) in a Gauss-Seidel manner and adds zero-mean Gaussian noise with time-varying variance $\sigma_{i,k}^2$ that decreases as the iteration number k increases.

The resulting ADMM steps that provide differential privacy are as follows:

$$\boldsymbol{w}_i^k = \arg\min_{\boldsymbol{w}_i} \ \hat{\mathcal{L}}_{\rho,k,i}(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k-1}, \boldsymbol{w}^{k-1}, \boldsymbol{\gamma}_i^{k-1}), \qquad (10a)$$

$$\tilde{\boldsymbol{w}}_i^k = \boldsymbol{w}_i^k + \mathcal{MN}_{d,p}(0, \sigma_{i,k}^2\mathbf{I}_d, \sigma_{i,k}^2\mathbf{I}_p), \qquad (10b)$$

$$\boldsymbol{w}^k = \frac{1}{n}\sum_{i=1}^n \tilde{\boldsymbol{w}}_i^k - \frac{1}{n}\sum_{i=1}^n \boldsymbol{\gamma}_i^{k-1}/\rho, \qquad (10c)$$

$$\boldsymbol{\gamma}_i^k = \boldsymbol{\gamma}_i^{k-1} - \rho(\tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^k), \qquad (10d)$$

where (10c) is computed at the aggregator while (10a), (10b) and (10d) are performed at each agent.

The details are given in Algorithm 3. The central aggregator firstly initializes the global variable $\boldsymbol{w}^0$, and the agents also initialize their noisy primal variables $\{\tilde{\boldsymbol{w}}_i^0\}_{i\in[n]}$ and dual variables $\{\boldsymbol{\gamma}_i^0\}_{i\in[n]}$. At the beginning of each iteration k, each agent i first samples a zero-mean Gaussian noise $\boldsymbol{\xi}_i^k$ with variance $\sigma_{i,k}^2$ and updates the noisy primal variable $\tilde{\boldsymbol{w}}_i^k$ based on (10a) and (10b). Then the aggregator receives the noisy primal variables $\{\tilde{\boldsymbol{w}}_i^k\}_{i\in[n]}$ and the dual variables $\{\boldsymbol{\gamma}_i^{k-1}\}_{i\in[n]}$ from the agents, and uses them to update the global variable $\boldsymbol{w}^k$ according to (10c). After that, agents receive the updated global variable $\boldsymbol{w}^k$ from the aggregator and continue to update the dual variables $\{\boldsymbol{\gamma}_i^k\}_{i\in[n]}$ by (10d). The iterative process will continue until reaching t iterations.

Algorithm 3 is different from Algorithm 2 in three aspects. Firstly, the approximate augmented Lagrangian function used in this approach replaces the objective function with its first-order approximation at $\tilde{\boldsymbol{w}}_i^{k-1}$, which is similar to the stochastic mirror descent [23]. This approximation enforces the smoothness of the Lagrangian function and makes it easy to solve (10a). Even when the objective function is non-smooth, we can still get a closed-form solution to (10a), which achieves fast computation. More importantly, this approximation can lead to a bounded l_2 sensitivity in differential privacy guarantee without the limitation that the objective function should be smooth and strongly convex. Thus our approach can be applied to any convex problems. We demonstrate this in Section IV.

Secondly, similar to linearized ADMM [24], [25], there is an l_2-norm prox-function $\|\boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{k-1}\|^2$ but scaled by $1/2\eta_i^k$ added in (8), where the step size η_i^k decreases when the iteration number k increases. Such additional part can guarantee the consistency between the updated model $\boldsymbol{w}_i^k$ and the previous one, especially when k is large. Thus, as k increases, the updated model would change more smoothly. Note that the time-varying step-size η_i^k is significant for the overall convergence guarantee. In Section V, we will define η_i^k and show its importance in algorithmic convergence.

Lastly, the variance $\sigma_{i,k}^2$ of Gaussian mechanism used in Algorithm 3 is time-varying rather than constant as adopted in prior studies [20]. It decreases when the iteration number k increases. The motivation of using Gaussian mechanism with time-varying variance is to mitigate the negative effect from noise and guarantee the convergence property of our approach. As explained before, the added noise would disrupt the learning process. By using the Gaussian mechanism with time-varying variance, the added noise will decrease when the iteration number k increases. Therefore, the negative affect from the added noise will be mitigated, enabling the updates to be stable. In Section IV, we would define the magnitude of time-varying variance $\sigma_{i,k}^2$ to achieve differential privacy.

IV. Privacy Guarantee

In this section, we analyze the privacy guarantee of the proposed DP-ADMM. In DP-ADMM, the shared messages $\{\tilde{\boldsymbol{w}}_i^k\}_{k\in[t]}$ may reveal the sensitive information of agent i, which has been discussed in Section II. Thus, we need to demonstrate that DP-ADMM guarantees differential privacy with outputs $\{\tilde{\boldsymbol{w}}_i^k\}_{k\in[t]}$. We first estimate the l_2 norm sensitivity of $\boldsymbol{w}_i^k$ update function, then analyze the privacy leakage from the shared primal variable $\tilde{\boldsymbol{w}}_i^k$ in each iteration, and finally compute the end-to-end differential privacy guarantee across t iterations using the moments accountant method. Here we use $\boldsymbol{w}_{i,\mathcal{D}_i}^k$ and $\boldsymbol{w}_{i,\mathcal{D}_i'}^k$ to denote the local primal variables updated from two neighboring datasets $\mathcal{D}_i$ and $\mathcal{D}_i'$.

A. L_2-Norm Sensitivity

In our approach, we apply Gaussian mechanism to add noise whose magnitude is calibrated by the l_2-norm sensitivity. Note that compared with Algorithm 2 and prior works [1], [2], [17], the derivation of the sensitivity in our proposed algorithm does not require the assumption of smoothness and strong convexity of the objective function due to the first-order approximation used in the approximate augmented Lagrangian function.

Lemma 1: Assume that $\|\ell'(\cdot)\| \le c_1$. The l_2-norm sensitivity of local primal variable $\boldsymbol{w}_i^k$ update function is given by:

$$\max_{\mathcal{D}_i,\mathcal{D}_i'} \|\boldsymbol{w}_{i,\mathcal{D}_i}^k - \boldsymbol{w}_{i,\mathcal{D}_i'}^k\| = \frac{2c_1}{m_i(\rho + 1/\eta_i^k)}. \tag{11}$$

Proof: Since $\hat{\mathcal{L}}_{\rho,k,i}(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k-1}, \boldsymbol{w}^{k-1}, \boldsymbol{\gamma}_i^{k-1})$ in the first step of DP-ADMM (10a) is a quadratic function w.r.t. $\boldsymbol{w}_i$ and therefore convex, we could obtain that:

$$\begin{aligned}\boldsymbol{w}_{i,\mathcal{D}_i}^k = \Bigg(&- \sum_{j=1}^{m_i} \frac{1}{m_i}\ell'(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \tilde{\boldsymbol{w}}_i^{k-1}) - \frac{\lambda}{n}R'(\tilde{\boldsymbol{w}}_i^{k-1}) \\ &+ \boldsymbol{\gamma}_i^{k-1} + \rho\boldsymbol{w}^{k-1} + \frac{\tilde{\boldsymbol{w}}_i^{k-1}}{\eta_i^k}\Bigg)\Big(\rho + 1/\eta_i^k\Big)^{-1}, \end{aligned} \tag{12a}$$

$$\begin{aligned}\boldsymbol{w}_{i,\mathcal{D}_i'}^k = \Bigg(&- \sum_{j=1}^{m_i-1} \frac{1}{m_i}\ell'(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \tilde{\boldsymbol{w}}_i^{k-1}) \\ &- \frac{1}{m_i}\ell'(a_{i,m_i}', b_{i,m_i}', \tilde{\boldsymbol{w}}_i^{k-1}) - \frac{\lambda}{n}R'(\tilde{\boldsymbol{w}}_i^{k-1}) \\ &+ \boldsymbol{\gamma}_i^{k-1} + \rho\boldsymbol{w}^{k-1} + \frac{\tilde{\boldsymbol{w}}_i^{k-1}}{\eta_i^k}\Bigg)\Big(\rho + 1/\eta_i^k\Big)^{-1}, \end{aligned} \tag{12b}$$

by computing the derivative of (8) with inputs $\boldsymbol{w}^{k-1}$ and $\boldsymbol{\gamma}_i^{k-1}$ and letting $\nabla\hat{\mathcal{L}}_{\rho,k,i}(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i^{k-1}, \boldsymbol{w}^{k-1}, \boldsymbol{\gamma}_i^{k-1})$ to be 0.

With $\boldsymbol{w}_{i,\mathcal{D}_i}^k$ and $\boldsymbol{w}_{i,\mathcal{D}_i'}^k$ calculated by (12a) and (12b) respectively, the l_2-norm sensitivity of primal variable $\boldsymbol{w}_i^k$ update function is defined by:

$$\begin{aligned}&\max_{\mathcal{D}_i,\mathcal{D}_i'} \|\boldsymbol{w}_{i,\mathcal{D}_i}^k - \boldsymbol{w}_{i,\mathcal{D}_i'}^k\| \\ &= \max_{\mathcal{D}_i,\mathcal{D}_i'} \frac{\big\|\ell'(\boldsymbol{a}_{i,m_i}, \boldsymbol{b}_{i,m_i}, \tilde{\boldsymbol{w}}_i^{k-1}) - \ell'(\boldsymbol{a}_{i,m_i}', \boldsymbol{b}_{i,m_i}', \tilde{\boldsymbol{w}}_i^{k-1})\big\|}{m_i(\rho + 1/\eta_i^k)}. \end{aligned} \tag{13}$$

Since $\|\ell'(\cdot)\|$ is bounded by c_1, the sensitivity of $\boldsymbol{w}_i^k$ update function is given by $2c_1/\big(m_i(\rho + 1/\eta_i^k)\big)$. □

Lemma 1 shows that the sensitivity of $\boldsymbol{w}_i^k$ update function in our approach is affected by the time-varying η_i^k. When we set η_i^k to decrease with increasing k, the sensitivity becomes smaller with larger k, then the noise added would be smaller when ϵ is fixed. Thus, the updates would be stable in spite of the existence of the noise.

B. (ϵ, δ)-Differential Privacy Guarantee

In this section, we prove that each iteration of Algorithm 3 guarantees (ϵ, δ)-differential privacy.

Theorem 1: Assume that $\|\ell'(\cdot)\| \le c_1$. Let $\epsilon \in (0, 1]$ be arbitrary and $\boldsymbol{\xi}_i^k$ be the noise sampled from Gaussian mechanism with variance $\sigma_{i,k}^2$ where

$$\sigma_{i,k} = \frac{2c_1\sqrt{2\ln(1.25/\delta)}}{m_i\epsilon(\rho + 1/\eta_i^k)}. \tag{14}$$

Each iteration of DP-ADMM guarantees (ϵ, δ)-differential privacy. Specifically, for any neighboring datasets $\mathcal{D}_i$ and $\mathcal{D}_i'$, for any output $\tilde{\boldsymbol{w}}_i^k$, the following inequality always holds:

$$\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i] \le e^\epsilon \cdot \Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i'] + \delta. \tag{15}$$

Proof: The privacy loss from $\tilde{\boldsymbol{w}}_i^k$ is calculated as

$$\left|\ln\frac{\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i]}{\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i']}\right| = \left|\ln\frac{\Pr[\tilde{w}_i^{k(h,l)}|\mathcal{D}_i]}{\Pr[\tilde{w}_i^{k(h,l)}|\mathcal{D}_i']}\right| = \left|\ln\frac{\Pr[\xi_i^{k(h,l)}]}{\Pr[\xi_i^{k,'(h,l)}]}\right|, \tag{16}$$

where $\xi_i^{k(h,l)}$ and $\xi_i^{k,'(h,l)}$ are the (h,l)-entry of $\boldsymbol{\xi}_i^k$ and $\boldsymbol{\xi}_i^{k,'}$, and are sampled from $\mathcal{N}(0, \sigma_{i,k}^2)$. This leads to:

$$\begin{aligned}&\left|\ln\frac{\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i]}{\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i']}\right| \\ &= \Big|\frac{1}{2\sigma_{i,k}^2}\big(\big\|\xi_i^{k(h,l)}\big\|^2 - \big\|\xi_i^{k,'(h,l)}\big\|^2\big)\Big| \\ &= \Big|\frac{1}{2\sigma_{i,k}^2}\big(2\xi_i^{k(h,l)}\|w_{i,\mathcal{D}_i}^{k(h,l)} - w_{i,\mathcal{D}_i'}^{k(h,l)}\| + \|w_{i,\mathcal{D}_i}^{k(h,l)} - w_{i,\mathcal{D}_i'}^{k(h,l)}\|^2\big)\Big|. \end{aligned} \tag{17}$$

Since $\|\ell'(\cdot)\| \le c_1$, according to Lemma 1, we have $\|w_{i,\mathcal{D}_i}^{k(h,l)} - w_{i,\mathcal{D}_i'}^{k(h,l)}\| < \|\boldsymbol{w}_{i,\mathcal{D}_i}^k - \boldsymbol{w}_{i,\mathcal{D}_i'}^k\| \le 2c_1/\big(m_i(\rho + 1/\eta_i^k)\big)$. Thus, by letting $\sigma_{i,k} = 2\,c_1\sqrt{2\ln(1.25/\delta)}/\big(m_i\epsilon(\rho+1/\eta_i^k)\big)$, we have

$$\left|\ln\frac{\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i]}{\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i']}\right| \le \left|\frac{\xi_i^{k(h,l)}m_i(\rho + 1/\eta_i^k) + c_1}{4\ln(1.25/\delta)c_1/\epsilon^2}\right|. \tag{18}$$

When $|\xi_i^{k(h,l)}| \leq \big(4\ln(1.25/\delta)c_1/\epsilon - c_1\big)/\big(\epsilon m_i(\rho + 1/\eta_i^k)\big)$, $\big|\ln\big(\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i]/\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i']\big)\big|$ is bounded by ϵ. Next, we need to prove that $\Pr\big[|\xi_i^{k(h,l)}| > \big(4\ln(1.25/\delta)c_1/\epsilon - c_1\big)/\big(\epsilon m_i(\rho + 1/\eta_i^k)\big)\big] \leq \delta$, which requires $\Pr\big[\xi_i^{k(h,l)} > \big(4\ln(1.25/\delta)c_1/\epsilon - c_1\big)/\big(\epsilon m_i(\rho + 1/\eta_i^k)\big)\big] \leq \delta/2$. According to the tail bound of normal distribution $\mathcal{N}(0, \sigma_{i,k}^2)$, we have

$$\Pr\big[\xi_i^{k(h,l)} > r\big] \leq \frac{\sigma_{i,k}}{r\sqrt{2\pi}} e^{-r^2/2\sigma_{i,k}^2}. \tag{19}$$

By letting $r = \big(4\ln(1.25/\delta)c_1/\epsilon - c_1\big)/\big(\epsilon m_i(\rho + 1/\eta_i^k)\big)$ in the above inequality, we have:

$$\Pr\left[\xi_i^{k(h,l)} > \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^k)}\right] \leq \frac{2\sqrt{2\ln(1.25/\delta)}}{(4\ln(1.25/\delta) - \epsilon)\sqrt{2\pi}} \exp\left(-\frac{(4\ln(1.25/\delta) - \epsilon)^2}{8\ln(1.25/\delta)}\right). \tag{20}$$

When δ is small (≤ 0.01) and let $\epsilon \leq 1$, we have

$$\frac{2\sqrt{2\ln(1.25/\delta)}}{(4\ln(1.25/\delta) - \epsilon)\sqrt{2\pi}} < \frac{1}{\sqrt{2\pi}}, \tag{21}$$

and

$$-\frac{\big(4\ln(1.25/\delta) - \epsilon\big)^2}{8\ln(1.25/\delta)} < \ln(\sqrt{2\pi}\frac{\delta}{2}). \tag{22}$$

As a result, we have:

$$\Pr\left[\xi_i^{k(h,l)} > \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^k)}\right] < \frac{\delta}{2}. \tag{23}$$

So far we have proved that $\Pr\big[\xi_i^{k(h,l)} > \big(4\ln(1.25/\delta)c_1/\epsilon - c_1\big)/\big(\epsilon m_i(\rho + 1/\eta_i^k)\big)\big] \leq \delta/2$, thus we can prove that $\Pr\big[|\xi_i^{k(h,l)}| > \big(4\ln(1.25/\delta)c_1/\epsilon - c_1\big)/\big(\epsilon m_i(\rho + 1/\eta_i^k)\big)\big] \leq \delta$. We define:

$$\mathbb{A}_1 = \{\xi_i^{k(h,l)} : |\xi_i^{k(h,l)}| \leq \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^k)}\}, \tag{24a}$$

$$\mathbb{A}_2 = \{\xi_i^{k(h,l)} : |\xi_i^{k(h,l)}| > \frac{4\ln(1.25/\delta)c_1/\epsilon - c_1}{m_i(\rho + 1/\eta_i^k)}\}. \tag{24b}$$

Therefore, we obtain the result:

$$\begin{aligned}\Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i] &= \Pr[w_{i,\mathcal{D}_i}^{k(h,l)} + \xi_i^{k(h,l)} : \xi_i^{k(h,l)} \in \mathbb{A}_1] \\ &\quad + \Pr[w_{i,\mathcal{D}_i}^{k(h,l)} + \xi_i^{k(h,l)} : \xi_i^{k(h,l)} \in \mathbb{A}_2] \\ &< e^{\epsilon} \cdot \Pr[\tilde{\boldsymbol{w}}_i^k|\mathcal{D}_i'] + \delta,\end{aligned} \tag{25}$$

which proves that each iteration of DP-ADMM guarantees (ϵ, δ)-differential privacy. □

C. Total Privacy Leakage

We have proved that each iteration of the proposed algorithm is (ϵ, δ)-differentially private. Here we focus on the total privacy leakage of our algorithm. Since Algorithm 3 is a t-fold adaptive algorithm, we follow prior studies [20], [26] and use the moments accountant method to analyze the total privacy leakage.

Theorem 2 (Advanced Composition Theorem): Assume $\|\ell'(\cdot)\| \leq c_1$. Let $\epsilon \in (0, 1]$ be arbitrary and $\boldsymbol{\xi}_i^k$ be sampled from Gaussian mechanism with variance $\sigma_{i,k}^2$ where

$$\sigma_{i,k} = \frac{2c_1\sqrt{2\ln(1.25/\delta)}}{m_i\epsilon(\rho + 1/\eta_i^k)}. \tag{26}$$

Then Algorithm 3 guarantees $(\bar{\epsilon}, \delta)$-differential privacy, where $\bar{\epsilon} = c_0\sqrt{t}\epsilon$ for some constant c_0.

Proof: See Appendix B in Supplementary Material. □

V. Convergence Analysis

In this section, we analyze the convergence of the proposed DP-ADMM. Let $\boldsymbol{w}^*$ denote the optimal solution of problem (4), and c_w denote $\|\boldsymbol{w}^*\|$. Firstly, we analyze the convergence property based on the general assumption that the objective function is convex and non-smooth. Secondly, we refine the convergence property under a stricter assumption that the objective function is convex and smooth.

We define the following notations to be used for the analysis:

$$f_i(\boldsymbol{w}_i) = \sum_{j=1}^{m_i} \frac{1}{m_i}\ell(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \boldsymbol{w}_i) + \frac{\lambda}{n}R(\boldsymbol{w}_i),$$

$$\bar{\boldsymbol{w}}^t = \frac{1}{t}\sum_{k=1}^{t}\boldsymbol{w}^k, \quad \bar{\boldsymbol{\gamma}}_i^t = \frac{1}{t}\sum_{k=1}^{t}\boldsymbol{\gamma}_i^k, \quad \bar{\boldsymbol{w}}_i^t = \frac{1}{t}\sum_{k=0}^{t-1}\tilde{\boldsymbol{w}}_i^k,$$

$$\boldsymbol{u}_i^k = \begin{bmatrix}\tilde{\boldsymbol{w}}_i^k \\ \boldsymbol{w}^k \\ \boldsymbol{\gamma}_i^k\end{bmatrix}, \quad \boldsymbol{u}_i = \begin{bmatrix}\boldsymbol{w}_i \\ \boldsymbol{w} \\ \boldsymbol{\gamma}_i\end{bmatrix}, \quad F(\boldsymbol{u}_i^k) = \begin{bmatrix}-\boldsymbol{\gamma}_i^k \\ \boldsymbol{\gamma}_i^k \\ \tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^k\end{bmatrix}.$$

We show that DP-ADMM achieves an $O(1/\sqrt{t})$ rate of convergence in terms of both the objective value and the constraint violation: $\sum_{i=1}^n \big(f_i(\bar{\boldsymbol{w}}_i^t) - f_i(\boldsymbol{w}^*) + \beta\|\bar{\boldsymbol{w}}_i^t - \bar{\boldsymbol{w}}^t\|\big)$, where $\sum_{i=1}^n \big(f_i(\bar{\boldsymbol{w}}_i^t) - f_i(\boldsymbol{w}^*)\big)$ represents the distance between the current objective value and the optimal value while $\sum_{i=1}^n \beta\|\bar{\boldsymbol{w}}_i^t - \bar{\boldsymbol{w}}^t\|$ measures the difference between the local model and the global one. Therefore, when we have $\sum_{i=1}^n \big(f_i(\bar{\boldsymbol{w}}_i^t) - f_i(\boldsymbol{w}^*) + \beta\|\bar{\boldsymbol{w}}_i^t - \bar{\boldsymbol{w}}^t\|\big) = 0$, our training result converges to the optimal one and all local models reach consensus.

A. Non-Smooth Convex Objective Function

In this section, we analyze the convergence when the objective function is convex but non-smooth. We firstly analyze a single iteration of our algorithm in Lemma 2 and then give the convergence result of DP-ADMM in Theorem 3.

Lemma 2: Assume $\ell(\cdot)$ and $R(\cdot)$ are convex. For any $k \geq 1$, we have:

$$\begin{aligned}&\sum_{i=1}^n \Big(f_i(\tilde{\boldsymbol{w}}_i^{k-1}) - f_i(\boldsymbol{w}_i) + (\boldsymbol{u}_i^k - \boldsymbol{u}_i)^{\mathsf{T}}F(\boldsymbol{u}_i^k)\Big) \\ &\leq \sum_{i=1}^n \Big(\frac{\eta_i^k}{2}\big\|f_i'(\tilde{\boldsymbol{w}}_i^{k-1}) - (\rho + 1/\eta_i^k)\boldsymbol{\xi}_i^k\big\|^2 - \frac{\rho}{2}\|\boldsymbol{w}_i - \boldsymbol{w}^k\|^2 \\ &\quad + \frac{\rho}{2}\|\boldsymbol{w}_i - \boldsymbol{w}^{k-1}\|^2 - \big(\rho + 1/\eta_i^k\big)\big\langle\boldsymbol{\xi}_i^k, \boldsymbol{w}_i - \tilde{\boldsymbol{w}}_i^{k-1}\big\rangle\end{aligned}$$

$$+\frac{1}{2\eta_i^k}\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{k-1}\|^2-\frac{1}{2\eta_i^k}\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{k}\|^2 +\frac{1}{2\rho}\|\boldsymbol{\gamma}_i-\boldsymbol{\gamma}_i^{k-1}\|^2-\frac{1}{2\rho}\|\boldsymbol{\gamma}_i-\boldsymbol{\gamma}_i^{k}\|^2\Big). \quad (28)$$

Proof: See Appendix D in Supplementary Material. □

Based on Lemma 2, we give the following convergence theorem.

Theorem 3: Assume $\ell(\cdot)$ and $R(\cdot)$ are convex, $\|\ell'(\cdot)\|\le c_1$, and $\|R'(\cdot)\|\le c_2$. Let

$$\eta_i^k=\frac{c_w}{\sqrt{2k}}\left((c_1+\lambda c_2/n)^2+\frac{8dpc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}\right)^{-\frac{1}{2}}. \quad (29)$$

Define

$$M_1(\epsilon,\delta)=\sum_{i=1}^{n}c_w\sqrt{2(c_1+\lambda c_2/n)^2+\frac{16dpc_1^2\ln(1.25/\delta)}{m_i^2\epsilon^2}}, \quad (30)$$

and

$$M_2=\frac{n(\rho c_w^2+\beta^2/\rho)}{2}. \quad (31)$$

For any $t\ge 1$ and β, we have:

$$\mathbb{E}\Big[\sum_{i=1}^{n}\Big(f_i(\bar{\boldsymbol{w}}_i^t)-f_i(\boldsymbol{w}^*)+\beta\|\bar{\boldsymbol{w}}_i^t-\bar{\boldsymbol{w}}^t\|\Big)\Big]\le\frac{M_1(\epsilon,\delta)}{\sqrt{t}}+\frac{M_2}{t}. \quad (32)$$

Proof: See Appendix E in Supplementary Material. □

Theorem 3 shows an explicit utility-privacy trade-off of our approach: when privacy guarantee is weaker (larger ϵ and δ), our approach has better utility. In addition, it demonstrates that our algorithm converges at a rate of $O(1/\sqrt{t})$.

B. Smooth Convex Objective Function

In this section, we refine Theorem 3 under a stricter assumption that $\ell(\cdot)$ and $R(\cdot)$ are both convex and smooth. Here, we replace the definition of $\bar{\boldsymbol{w}}_i^t$: $\bar{\boldsymbol{w}}_i^t=\frac{1}{t}\sum_{k=0}^{t-1}\tilde{\boldsymbol{w}}_i^k$ by $\bar{\boldsymbol{w}}_i^t=\frac{1}{t}\sum_{k=1}^{t}\tilde{\boldsymbol{w}}_i^k$. Similar to Section V-A, we first focus on a single iteration and then give the final convergence result.

Lemma 3: Assume $\ell(\cdot)$ and $R(\cdot)$ are convex and smooth, $\|\nabla^2\ell(\cdot)\|\le c_3$, and $\|\nabla^2\ R(\cdot)\|\le c_4$. For any $k\ge 1$, we have:

$$\sum_{i=1}^{n}\Big(f_i(\tilde{\boldsymbol{w}}_i^k)-f_i(\boldsymbol{w}_i)+(\boldsymbol{u}_i^k-\boldsymbol{u}_i)^{\mathsf{T}}F(\boldsymbol{u}_i^k)\Big) \le\sum_{i=1}^{n}\Big(\frac{\big(\rho+1/\eta_i^k\big)^2}{2/\eta_i^k-2(c_3+\lambda c_4/n)}\|\boldsymbol{\xi}_i^k\|^2-\frac{1}{2\eta_i^k}\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^k\|^2 +\frac{1}{2\eta_i^k}\|\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{k-1}\|^2-\big(\rho+1/\eta_i^k\big)\big\langle\boldsymbol{\xi}_i^k,\boldsymbol{w}_i-\tilde{\boldsymbol{w}}_i^{k-1}\big\rangle +\frac{\rho}{2}\|\boldsymbol{w}_i-\boldsymbol{w}^{k-1}\|^2-\frac{\rho}{2}\|\boldsymbol{w}_i-\boldsymbol{w}^{k}\|^2 +\frac{1}{2\rho}\|\boldsymbol{\gamma}_i-\boldsymbol{\gamma}_i^{k-1}\|^2-\frac{1}{2\rho}\|\boldsymbol{\gamma}_i-\boldsymbol{\gamma}_i^{k}\|^2\Big). \quad (33)$$

Proof: See Appendix F in Supplementary Material. □

Based on Lemma 3, we give the following theorem.

Theorem 4: Assume $\ell(\cdot)$ and $R(\cdot)$ are convex and smooth, $\|\nabla^2\ell(\cdot)\|\le c_3$, and $\|\nabla^2\ R(\cdot)\|\le c_4$. Let

$$\eta_i^k=\left(c_3+\lambda c_4/n+\frac{4c_1\sqrt{dpk\ln(1.25/\delta)}}{m_i\epsilon c_w}\right)^{-1}. \quad (34)$$

Define

$$M_3(\epsilon,\delta)=\sum_{i=1}^{n}\frac{4c_wc_1\sqrt{dp\ln(1.25/\delta)}}{m_i\epsilon}, \quad (35)$$

and

$$M_4=\frac{nc_w^2(c_3+\lambda c_4/n+\rho)+n\beta^2/\rho}{2}. \quad (36)$$

For any $t\ge 1$ and β, we have:

$$\mathbb{E}\Big[\sum_{i=1}^{n}\Big(f_i(\bar{\boldsymbol{w}}_i^t)-f_i(\boldsymbol{w}^*)+\beta\|\bar{\boldsymbol{w}}_i^t-\bar{\boldsymbol{w}}^t\|\Big)\Big]\le\frac{M_3(\epsilon,\delta)}{\sqrt{t}}+\frac{M_4}{t}. \quad (37)$$

Proof: See Appendix G in Supplementary Material. □

Theorem 4 also shows an explicit relation between the privacy budget (i.e., ϵ and δ) and the utility of our approach with smoothness, and demonstrates that the result from our algorithm converges to the optimal result at a rate of $O(1/\sqrt{t})$.

VI. Performance Evaluation

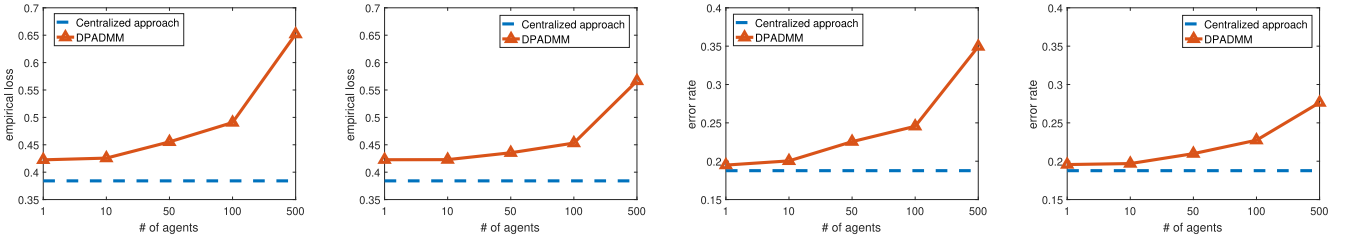
In this section, we evaluate the performance of DP-ADMM with both non-smooth objectives and smooth objectives by considering logistic regression problems with l_1-norm and l_2-norm regularizers, respectively.

A. Dataset

We evaluate our approach on a real-world dataset: Adult dataset [27] from UCI Machine Learning Repository. Adult dataset includes 48, 842 instances. Each instance has 14 attributes such as age, sex, education, occupation, marital status, and native country, and is associated with a label representing whether the income is above \$50, 000 or not. Before the simulation, we firstly preprocess the data by removing all the instances with missing values, converting the categorical attributes into binary vectors, normalizing columns to guarantee the maximum value of each column is 1, normalizing rows to enforce their l_2 norm to be less than 1, and converting the labels $\{>50k, <50k\}$ into $\{+1,-1\}$. After this, we obtain 45, 222 entries each with a 104-dimensional feature vector ($d=104$) and a 1-dimensional label belonging to $\{+1,-1\}$ ($p=1$). In each simulation, we sample 40, 000 instances for training, and the remaining 5, 222 instances for testing. In the training process, we divide the training data into n groups randomly, and thus each group contains $40000/n$ data points ($m_i=40000/n$).

B. Baseline Algorithms

We compare our DP-ADMM (Algorithm 3) with five baseline algorithms: (1) non-private centralized approach, (2) ADMM algorithm (Algorithm 1), (3) ADMM algorithm with PVP (Algorithm 2), (4) ADMM with dual variable perturbation (DVP) in [17], and (5) differentially private stochastic gradient descent (DPSGD) in [20] for distributed settings. We evaluate the accuracy and effectiveness of our approach by comparing it with the five baseline algorithms.



(a) $\epsilon = 0.05, \bar{\epsilon} = 0.5009, \delta = 10^{-3}$ (b) $\epsilon = 0.1, \bar{\epsilon} = 1.0193, \delta = 10^{-3}$ (c) $\epsilon = 0.05, \bar{\epsilon} = 0.5009, \delta = 10^{-3}$ (d) $\epsilon = 0.1, \bar{\epsilon} = 1.0193, \delta = 10^{-3}$

Fig. 1. Impact of distributed data source number on DP-ADMM (l_1-regularized logistic regression).

TABLE II
COMPUTATION TIME (100 ITERATIONS)

	ADMM	PVP	DVP	DPADMM
$\epsilon = 0.01$	67.242s	102.282s	59.743s	6.937s
$\epsilon = 0.05$	67.242s	78.798s	65.935s	5.322s
$\epsilon = 0.1$	67.242s	79.013s	69.855s	5.218s

C. Setup

We set up the simulation by MATLAB in an Intel(R) Core(TM) 3.40 GHz computer with 16 GB RAM. In the simulation, we set the total iteration number $t = 100$ and the penalty parameter $\rho = 0.1$, and choose the optimal regularizer parameter λ/n to be 10^{-6} by 10-cross-validation in non-private setting. In DPSGD, we set the optimal learning rate to be 0.1 and the sampling ratio to be 1. We focus on the settings with strong privacy guarantee and thus we set privacy budget per iteration $\epsilon = \{0.01, 0.05, 0.1, 0.2\}$ and $\delta = \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, and use moments accountant method to obtain the corresponding total privacy loss $\bar{\epsilon}$. In each simulation, we run it for 10 times to get averaged result.

D. Evaluations

We consider logistic regression problem in a distributed setting and evaluate our approach for logistic regression problems with l_1-norm and l_2-norm regularizers respectively, in terms of convergence, accuracy, and computation cost. The loss function of binary logistic regression is defined by (2). The convergence properties are evaluated with respect to the augmented objective value, which measures the loss as well as the constraint penalty and is defined as $\sum_{i=1}^{n}\left(f_i(\tilde{\boldsymbol{w}}_i^k) + \rho\|\bar{\boldsymbol{w}}_i^k - \bar{\boldsymbol{w}}^k\|\right)$. We evaluate the accuracy by empirical loss $\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m_i}\frac{1}{m_i}\ell(\boldsymbol{a}_{i,j}, \boldsymbol{b}_{i,j}, \tilde{\boldsymbol{w}}_i^k)$, and classification error rate. We measure the computation cost using the running time of training.

E. L_1-Regularized Logistic Regression

We obtain the DP-ADMM steps for l_1 regularized logistic regression by:

$$\boldsymbol{w}_i^k = \left(\frac{1}{m_i}\sum_{j=1}^{m_i}\frac{\boldsymbol{b}_{i,j}\boldsymbol{a}_{i,j}}{1+\exp(\boldsymbol{b}_{i,j}\tilde{\boldsymbol{w}}_i^{k-1\intercal}\boldsymbol{a}_{i,j})} - \frac{\lambda}{n}\mathrm{sgn}(\tilde{\boldsymbol{w}}_i^{k-1}) + \boldsymbol{\gamma}_i^{k-1} + \rho\boldsymbol{w}^{k-1} + \tilde{\boldsymbol{w}}_i^{k-1}/\eta_i^k\right)\left(\rho + 1/\eta_i^k\right)^{-1}, \quad (38a)$$

$$\tilde{\boldsymbol{w}}_i^k = \boldsymbol{w}_i^k + \mathcal{MN}_{d,p}(0, \sigma_{i,k}^2\mathbf{I}_d, \sigma_{i,k}^2\mathbf{I}_p), \quad (38b)$$

$$\boldsymbol{w}^k = \frac{1}{n}\sum_{i=1}^{n}\tilde{\boldsymbol{w}}_i^k - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\gamma}_i^{k-1}/\rho, \quad (38c)$$

$$\boldsymbol{\gamma}_i^k = \boldsymbol{\gamma}_i^{k-1} - \rho\left(\tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^k\right), \quad (38d)$$

where $\mathrm{sgn}(\cdot)$ is the sign function.

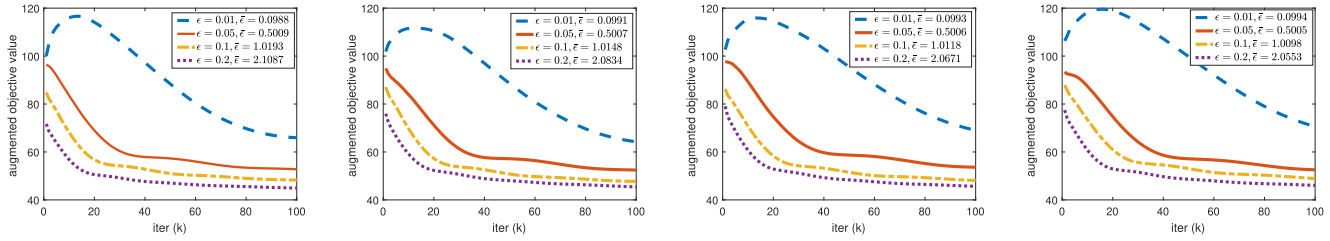
Since the l_1 regularized objective function is convex but non-smooth, we apply Theorem 3 to set η_i^k. Since we enforce $\|\ell'(\cdot)\| \leq 1$ by data preprocessing, and we have $\|R'(\cdot)\| \leq \sqrt{dp}$ ($d = 104$ and $p = 1$), we set $c_1 = 1$ and $c_2 = \sqrt{104}$. We obtain $\boldsymbol{w}^*$ by pre-training and set c_w to be 23. According to Theorem 3, we set η_i^k to be $23\big(2k(1+10^{-6}\sqrt{104}/n)^2 + 1664k\ln(1.25/\delta)/\big(m_i^2\epsilon^2\big)\big)^{-\frac{1}{2}}$.

Since PVP and DVP cannot be applied when the objective function is non-smooth, we only compare our approach with ADMM and DPSGD in this section. We first investigate the performance of our approach with different numbers of distributed data sources and compare it with the centralized approach. Figure 1 shows that the accuracy of our training model would decrease if we consider larger number of data sources. Since the size of local dataset is smaller for larger number of agents, more noise should be introduced to guarantee the same level of differential privacy, thus degrading the performance of the trained model. This is consistent with Theorem 1 that the noise magnitude is scaled by $1/m_i$. In following simulations, we consider the case when the number of agents n equals 100. Figure 2 demonstrates the convergence properties of our approach by showing how the augmented objective value converges for different ϵ and δ. It shows that our approach with larger ϵ and larger δ has better convergence, which is consistent with Theorem 3. Finally, we evaluate the accuracy of our approach by empirical loss and classification error rate by comparing with ADMM and DPSGD. Figure 3 shows our approach outperforms DPSGD due to the faster convergence property, demonstrating the advantage of ADMM framework. In addition, Figure 3 shows the privacy-utility trade-off of our approach. When privacy leakage increases (larger ϵ and larger δ), our approach achieves better utility.

F. L_2-Regularized Logistic Regression

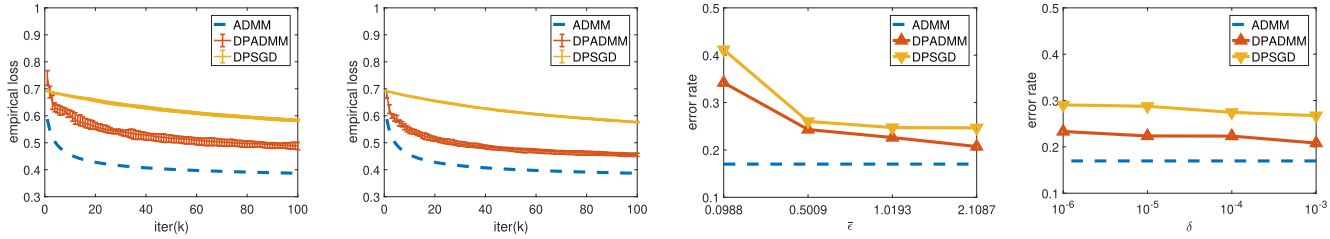
The DP-ADMM steps for l_2 regularized logistic regression are described as follows:

$$\boldsymbol{w}_i^k = \left(\frac{1}{m_i}\sum_{j=1}^{m_i}\frac{\boldsymbol{b}_{i,j}\boldsymbol{a}_{i,j}}{1+\exp(\boldsymbol{b}_{i,j}\tilde{\boldsymbol{w}}_i^{k-1\intercal}\boldsymbol{a}_{i,j})} - \frac{\lambda}{n}\tilde{\boldsymbol{w}}_i^{k-1} + \boldsymbol{\gamma}_i^{k-1} + \rho\boldsymbol{w}^{k-1} + \tilde{\boldsymbol{w}}_i^{k-1}/\eta_i^k\right)\left(\rho + 1/\eta_i^k\right)^{-1}, \quad (39a)$$



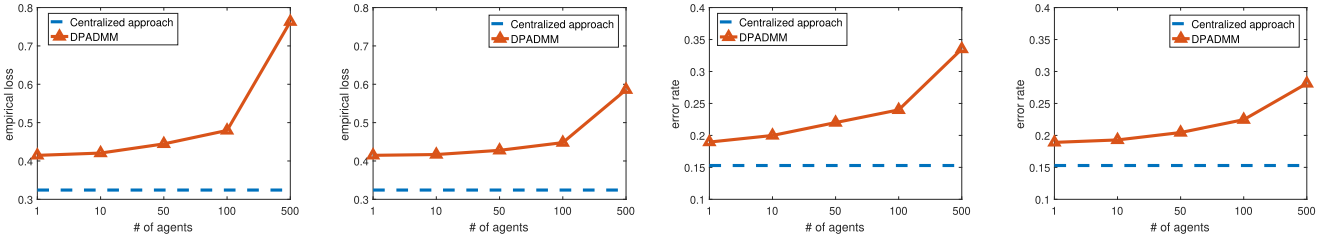
(a) $\delta = 10^{-3}$ (b) $\delta = 10^{-4}$ (c) $\delta = 10^{-5}$ (d) $\delta = 10^{-6}$

Fig. 2. Convergence properties of DP-ADMM (l_1-regularized logistic regression).



(a) $\epsilon = 0.05$, $\bar{\epsilon} = 0.5009$, $\delta = 10^{-3}$ (b) $\epsilon = 0.1$, $\bar{\epsilon} = 1.0193$, $\delta = 10^{-3}$ (c) $\delta = 10^{-3}$ (d) $\epsilon = 0.1$

Fig. 3. Accuracy comparison in empirical loss and classification error rate (l_1-regularized logistic regression).



(a) $\epsilon = 0.05, \bar{\epsilon} = 0.5009, \delta = 10^{-3}$ (b) $\epsilon = 0.1, \bar{\epsilon} = 1.0193, \delta = 10^{-3}$ (c) $\epsilon = 0.05, \bar{\epsilon} = 0.5009, \delta = 10^{-3}$ (d) $\epsilon = 0.1, \bar{\epsilon} = 1.0193, \delta = 10^{-3}$

Fig. 4. Impact of distributed data source number on DP-ADMM (l_2-regularized logistic regression).

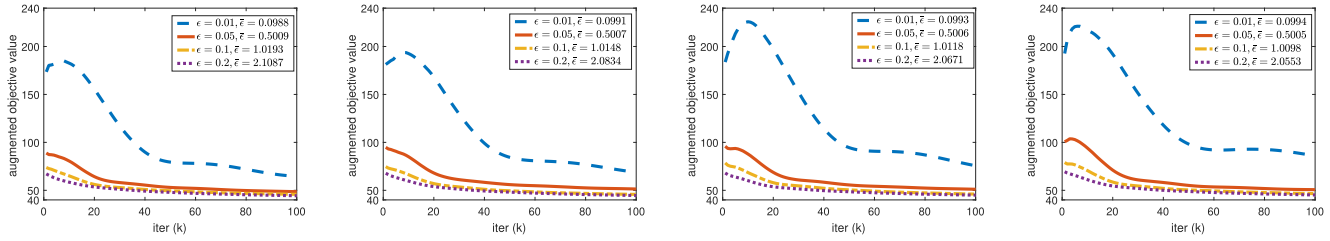
$$\tilde{\boldsymbol{w}}_i^k = \boldsymbol{w}_i^k + \mathcal{MN}_{d,p}(0, \sigma_{i,k}^2 \mathbf{I}_d, \sigma_{i,k}^2 \mathbf{I}_p), \tag{39b}$$

$$\boldsymbol{w}^k = \frac{1}{n}\sum_{i=1}^{n} \tilde{\boldsymbol{w}}_i^k - \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\gamma}_i^{k-1}/\rho, \tag{39c}$$

$$\boldsymbol{\gamma}_i^k = \boldsymbol{\gamma}_i^{k-1} - \rho\left(\tilde{\boldsymbol{w}}_i^k - \boldsymbol{w}^k\right). \tag{39d}$$

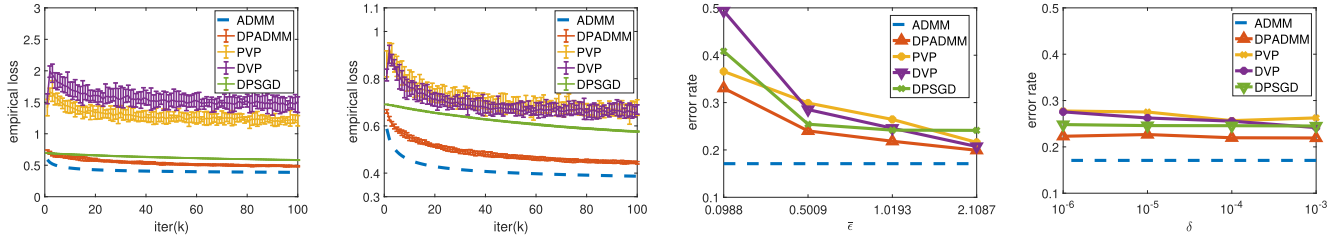
Here the l_2 regularized objective function is convex and smooth, thus we apply Theorem 4 to set η_i^k. Since we have $\|\nabla^2 R(\cdot)\| \leq 1$, and we enforce $\|\nabla \ell(\cdot)\| \leq 1$ and $\|\nabla^2 \ell(\cdot)\| \leq 0.25$ by data preprocessing, thus we set $c_1 = 1$, $c_3 = 0.25$, and $c_4 = 1$. We obtain the optimal solution $\boldsymbol{w}^*$ by pre-training, and set c_w to be 89. According to Theorem 4, we set η_i^k to be $\left(0.25 + 10^{-6} + 2\sqrt{416k\ln(1.25/\delta)}/\left(89\ m_i\epsilon\right)\right)^{-1}$.

We fist investigate the performance of our approach under the settings with different numbers of distributed data sources and Figure 4 depicts the corresponding accuracy changes (accuracy decreases with increasing number of agents). Since the total data size is fixed, when we consider a larger number of agents, the size of local dataset is smaller, so the training model has lower accuracy due to more added noise for the same level of privacy guarantee. In the following simulations, we focus on the case where the number of agents is 100. Next, we show the convergence properties of our approach. Figure 5 demonstrates that under weaker privacy guarantee (larger ϵ and larger δ), our approach has better convergence, which is consistent with Theorem 4. We evaluate the accuracy of our approach by comparing it with ADMM, PVP, DVP, and DPSGD on empirical loss and classification error rate. Figure 6 shows that our approach outperforms PVP, DVP, and DPSGD. Specifically, ADMM has fast convergence but is sensitive to noise. Thus the methods directly perturbing intermediate results in ADMM (PVP and DVP) have poor performance. Gradient-based method (DPSGD) has good noise-resilience property but converges slowly. Our approach is based on ADMM framework, and combines the approximate augmented Lagrangian function with time-varying Gaussian noise addition to achieve higher utility. Furthermore, the results in Figure 6 also show the utility-privacy trade-off of our approach: larger ϵ and larger δ indicating weaker privacy guarantee would result in better utility. Finally, we show the advantage of our approach in computation cost by running time. Table II gives the comparison and shows that DP-ADMM has much less computation cost than all three ADMM baseline algorithms, which is resulted from the first-order approximation used in our approach enabling updates with closed-form solutions.



(a) $\delta = 10^{-3}$ (b) $\delta = 10^{-4}$ (c) $\delta = 10^{-5}$ (d) $\delta = 10^{-6}$

Fig. 5. Convergence properties of DP-ADMM (l_2-regularized logistic regression).



(a) $\epsilon = 0.05$, $\bar{\epsilon} = 0.5009$, $\delta = 10^{-3}$ (b) $\epsilon = 0.1$, $\bar{\epsilon} = 1.0193$, $\delta = 10^{-3}$ (c) $\delta = 10^{-3}$ (d) $\epsilon = 0.1$

Fig. 6. Accuracy comparison in empirical loss and classification error rate (l_2-regularized logistic regression).

VII. Related Work

The existing literature related to our work could be categorized by: privacy-preserving empirical risk minimization, privacy-preserving distributed learning, and variants of ADMM.

A. Privacy-Preserving Empirical Risk Minimization

There have been tremendous research efforts on privacy-preserving empirical risk minimization [22], [28]–[30]. Most of them focus on a centralized setting where sensitive data is collected and stored centrally, thus the privacy leakage comes from the final released trained model. Chaudhuri *et al.* [22] propose two perturbation methods: output perturbation and objective perturbation to guarantee ϵ-differential privacy. Bassily *et al.* [28] provide a systematic investigation of differentially private algorithms for convex empirical risk minimization and propose efficient algorithms with tighter error bound. Wang *et al.* [29] focus on a more general problem: non-convex problem, and propose a faster algorithm based on a proximal stochastic gradient method. Thakurta and Smith [30] explore the stability of model selection problems, and propose two differentially private algorithms based on perturbation stability and subsampling stability respectively.

B. Privacy-Preserving Distributed Learning

Preserving privacy in distributed learning is challenging due to frequent information exchange in the iterative process. Recently, much works have been done to develop privacy-preserving distributed learning algorithms. Some of them employ cryptography-based methods in the protocol to hide the private information [31]–[34]. A recent work [33] uses partially homomorphic cryptography in ADMM-based distributed learning to preserve data privacy but the proposed approach cannot protect the information leakage of the private user data from the final learned models. In contrast, our approach provides differential privacy in the final trained machine learning models. Among the works on distributed learning with differential privacy, most of them focus on subgradient-based algorithms [35]–[38] and only a few works consider ADMM-based methods [1], [2], [17]–[19]. Zhang and Zhu [17] propose two perturbation methods: primal perturbation and dual perturbation to guarantee dynamic differential privacy in ADMM-based distributed learning. Zhang et al. [1] propose to perturb the penalty parameter of ADMM to guarantee differential privacy. Zhang *et al.* [2] propose recycled ADMM with differential privacy guarantee where the results from odd iterations could be re-utilized by the even iterations, and thus half of updates incur no privacy leakage. Guo and Gong [18] preserve differential privacy in the asynchronous ADMM algorithm. We design an ADMM-based distributed learning scheme with differential privacy which uses approximate augmented Lagrangian function for all iterations and adaptively changes the variance of added Gaussian noise in each iteration. We also use moments accountant method to analyze the total privacy loss to better estimate the trade-off between the data privacy and utility. We are the first to analyze rigorously the convergence rate and utility performance of ADMM with differential privacy.

C. Variants of ADMM

Some variants of ADMM have been proposed recently for applicability to more generous problems. Linearized ADMM [24], [25] replaces the quadratic function in the augmented Lagrangian function with a linearized approximation and thus provides a better way to solve subproblems without closed-form solutions. Stochastic ADMM [39], [40] considers stochastic and composite objective functions caused by natural uncertainties in observations. Our DP-ADMM algorithm inherits the features of linearized ADMM and stochastic ADMM, and guarantees strong differential privacy with good utility and low computation cost.

VIII. Conclusion

In this paper, we have proposed an improved ADMM-based differentially private distributed learning algorithm, DP-ADMM, for a class of learning problems that can be formulated as convex regularized empirical risk minimization. By designing an approximate augmented Lagrangian function and Gaussian mechanism with time-varying variance, our novel approach is noise-resilient, convergent and computation-efficient, especially under high privacy guarantee. We have also applied the moments accountant method to analyze the end-to-end privacy loss of the proposed iterative algorithm. The theoretical convergence guarantee and utility bound of our approach are derived. The evaluations on real-world datasets have demonstrated the effectiveness of our approach in the setting under high privacy guarantee.

References

[1] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," 2018, *arXiv:1806.02246*. [Online]. Available: https://arxiv.org/abs/1806.02246

[2] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *Proc. 56th Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2018, pp. 959–965.

[3] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 4177–4184.

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

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[6] Q. Ling and A. Ribeiro, "Decentralized linearized alternating direction method of multipliers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 5447–5451.

[7] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.

[8] R. Zhang and J. T. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2014, pp. 1701–1709.

[9] P. Bianchi, W. Hachem, and F. Iutzeler, "A stochastic primal-dual algorithm for distributed asynchronous composite optimization," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2014, pp. 732–736.

[10] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 5445–5450.

[11] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2718–2723, May 2013.

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

[13] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1322–1333.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. Theory Cryptogr. Conf.* Springer, 2006, pp. 265–284.

[15] Z. Huang and Y. Gong, "Differential location privacy for crowdsourced spectrum sensing," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 1–9.

[16] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 591–606, Jul./Aug. 2018.

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

[18] Y. Guo and Y. Gong, "Practical collaborative learning for crowdsensing in the Internet of Things with differential privacy," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May/Jun. 2018, pp. 1–9.

[19] 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. SecureComm*, EAI, Orlando, FL, USA, 2019.

[20] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 308–318.

[21] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.

[22] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *J. Mach. Learn. Res.*, vol. 12, pp. 1069–1109, Mar. 2011.

[23] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, 2009.

[24] J. Yang and X. Yuan, "Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization," *Math. Comput.*, vol. 82, no. 281, pp. 301–329, 2012.

[25] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 612–620.

[26] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Found. Symp. (CSF)*, Aug. 2017, pp. 263–275.

[27] A. Asuncion and D. J. Newman. "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2007. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Adult

[28] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci.*, Oct. 2014, pp. 464–473.

[29] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2722–2731.

[30] A. Thakurta and A. Smith, "Differentially private model selection via stability argumentsand the robustness of the lasso," in *Proc. Conf. Learn. Theory*, Jun. 2013, pp. 819–850.

[31] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct./Nov. 2017, pp. 1175–1191.

[32] Q. Wang, S. Hu, M. Du, J. Wang, and K. Ren, "Learning privately: Privacy-preserving canonical correlation analysis for cross-media retrieval," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[33] C. Zhang, M. Ahmad, and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 565–580, Mar. 2019.

[34] Y. Gong, Y. Fang, and Y. Guo, "Privacy-preserving collaborative learning for mobile health monitoring," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[35] A. Bellet, R. Guerraoui, M. Taziki, and M. Tommasi, "Personalized and private peer-to-peer machine learning," 2017, *arXiv:1705.08435*. [Online]. Available: https://arxiv.org/abs/1705.08435

[36] S. Han, U. Topcu, and G. J. Pappas, "Differentially private distributed constrained optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 1, pp. 50–64, Jan. 2017.

[37] M. Hale and M. Egerstedt, "Differentially private cloud-based multi-agent optimization with constraints," 2017, *arXiv:1708.08422*. [Online]. Available: https://arxiv.org/abs/1708.08422

[38] Z. Huang, S. Mitra, and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distrib. Comput. Netw.*, Jan. 2015, Art. no. 4.

[39] H. Ouyang, N. He, L. Tran, and A. Gray, "Stochastic alternating direction method of multipliers," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2013, pp. 80–88.

[40] S. Azadi and S. Sra, "Towards an optimal stochastic alternating direction method of multipliers," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2014, pp. 620–628.

[41] S. Shalev-Shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. dissertation, Hebrew Univ. Jerusalem, Jerusalem, Israel, Jul. 2007.