Towards Plausible Collaborative Machine Learning: Privacy, Efficiency and Fairness

by

Jiahao Ding

A dissertation submitted to the Department of Electrical and Computer Engineering,

Cullen College of Engineering

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Electrical Engineering

Chair of Committee: Miao Pan

Committee Member: Zhu Han

Committee Member: Hien Van Nguyen

Committee Member: Xin Fu

Committee Member: Yanmin Gong

University of Houston

May 2022

ACKNOWLEDGMENTS

I have been fortunate to complete this dissertation with plenty of help and support from amazing mentors, colleagues, collaborators, and friends. I hereby express my sincere gratitude to all of them. First and foremost, I would like to thank my advisor, Dr. Miao Pan, for his keen insights and plentiful encouragements, for giving me tremendous support, and for giving me a chance as a beginner in the field and nurtured me to become a successful researcher. It has been a great privilege and honor to work and study under his guidance.

I would like to express my deepest appreciation to my dissertation committee members Dr. Zhu Han, Dr. Hien Van Nguyen, Dr. Xin Fu, and Dr. Yanmin Gong, for helpful discussions, the ingenious suggestions and insightful feedback. Their constructive comments significantly improve the quality of this dissertation. I am also grateful to all other researchers I have collaborated with: Dr. Guannan Liang, Dr. Jinbo Bi, Dr. Di Wang, Dr. Xiaohuan Li, Dr. Junyi Wang, Dr. Maoqiang Wu, Dr. Rong Yu, Dr. Mingsong Chen, Dr. Kaiping Xue, Dr. Chi Zhang, Dr. Haijun Zhang, Dr. Yuanxiong Guo, Dr. Haixia Zhang, Dr. Dongfeng Yuan, and Tian Liu. I have been truly honored to work with these excellent researchers.

My gratitude also goes to all of my friends and colleagues in the AI, Networking Technologies and Security Laboratory (ANTS Lab) at UH ECE department, Dr. Jingyi Wang, Dr. Sai Mounika Errapotu, Dr. Debing Wei, Dr. Xinyue Zhang, Dian Shi, Pavana Prakash, Rui Chen, Chenpei Huang, and many others. It has been wonderful to meet and work with you in Houston.

Finally, I would like to thank my parents for their unconditional love and support. Thank you for having my back and always believing in me. None of my achievements would be possible without you. This dissertation is dedicated to them.

ABSTRACT

Nowadays, the development of machine learning shows great potential in a variety of fields, such as retail, healthcare, and insurance. Effective machine learning models can automatically learn useful information from a large amount of data and provide decisions with high average accuracy. Although machine learning has infiltrated into many areas due to its advantages, a vast amount of data has been generated at an ever-increasing rate, which leads to significant computational complexity for data collection and processing via a centralized machine learning approach. Distributed machine learning thus has received huge interest due to its capability of exploiting the collective computing power of edge devices. However, during the learning process, model updates using local private samples and large-scale parameter exchanges among agents impose severe privacy concerns and communication bottlenecks. Moreover, the decisions and predictions offered by the learning models may cause certain fairness concerns among population groups of interest, when the grouping is based on such sensitive attributes as race and gender.

To address those challenges, in this dissertation, we first propose a number of differentially private Alternating Direction Method of Multipliers (ADMM) algorithms that leverage two key ideas to balance the privacy-accuracy tradeoff: (1) adding Gaussian noise with decaying variance to reduce the negative effects of noise addition and maintain the convergence behaviors; and (2) outputting a noisy approximate solution for the perturbed objective to release the shackles of the exact optimal solution during each ADMM iteration to ensure DP. It is shown that our algorithms can significantly improve the privacy-accuracy tradeoff over existing solutions. Second, we develop a differentially private and communication efficient decentralized gradient descent method that will update the local models by integrating DP noise and random quantization operator to simultaneously enforce DP and communication efficiency. Finally, we focus on addressing the discrimination and privacy concerns in classification models by incorporating functional mechanism and decision boundary covariance, a novel measure of decision boundary fairness.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Nowadays, machine learning is increasingly deployed into large-scale distributive systems that can improve the quality of our life, such as smart home security [1] and AI-aided medical diagnosis [2]. With the proliferation of mobile phone devices, a vast amount of data has been generated at an ever-increasing rate, which leads to significant computational complexity for data collection and processing via a centralized machine learning approach. Therefore, collaborative training of a machine learning model among edge computing devices is beneficial and essential in dealing with large scale decentralized learning tasks [3, 4, 5].

Collaborative machine learning is an appealing paradigm to build high-quality ML models. While an individual party may have limited data, it is possible to build improved, high-quality ML models by training on the aggregated data from many parties. For example, in healthcare, a hospital or healthcare firm whose data diversity and quantity are limited due to its small patient base can draw on data from other hospitals and firms to improve the prediction of some disease progression (e.g., diabetes) [6]. This collaboration can be encouraged by a government agency, such as the National Institute of Health in the United States. In precision agriculture, a farmer with limited land area and sensors can combine his collected data with the other farmers to improve the modeling of the effect of various influences (e.g., weather, pest) on his crop yield [7]. Such data sharing also benefits other application domains, including real estate in which a property agency can pool together its limited transactional data with that of the other agencies to improve the prediction of property prices [8].

In the framework of collaborative learning, data providers (agents) collaboratively solve a learning problem, which can be decomposed into several subproblems, via an interactive procedure of local computation and message passing. While collaborative learning has recently drawn significant attention due its decentralized implementation, it faces major challenges in terms of privacy, efficiency and fairness:

Privacy The information exchanges during collaborative learning process raise serious privacy concerns, and the adversary can extract private information from the shared learning

models via various inference attacks, such as (i) attribute attacks [9] which infer sensitive pieces of information (e.g, whether a patient has cancer) given the patient's public record and the ability to query the machine learning model; (ii) membership inference attacks [10] whose goal is to find out if a patient record is in the pool of the data used to train the machine learning model; and (iii) model inversion attacks [11] which attempt to reconstruct the entire patient data given only access to an intermediate layer of the deep network.

Efficiency The machine learning models are becoming more and more complex, which is the current trend in large-scale distributed machine learning. For example, ResNet-152 has 152 layers and 60.2M parameters [12], VGG-19 has 19 layers and 143M parameters [13], while BERT-Large has 24 layers, 16 attention heads and 340M parameters [14]. Since the size of learning model increases, model exchanges among agents become the significant communication bottleneck. Moreover, the computation speed and computational load of local agents vary greatly, where a subset of nodes can be largely delayed in their local computation, which can substantially slow down the overall system efficiency.

Fairness Fairness issues in machine learning has received growing attentions in the machine learning field due to the social inequities and unfair behaviors observed in classification models. Discrimination indicates unfair treatment towards individuals based on the group to which they are perceived to belong. In machine learning, discrimination may be unintentional but have powerful effect on vulnerable groups. For example, a classification model of automated job hiring system is more likely to hire candidates from certain racial or gender groups [15, 16].

In this Section, we will briefly state each problem, and describe the contribution of each work.

1.1 Overview of Dissertation Contributions and Structure

In Section 3, we focus on Alternating Direction Method of Multipliers (ADMM), one of the most popular methods to design collaborative machine learning architectures. This

method applies iterative local computations over local datasets at each agent and computation results exchange between the neighbors. We propose a differentially private robust ADMM algorithm (PR-ADMM) [17] with Gaussian mechanism, and employ two kinds of noise variance decay schemes to carefully adjust the noise addition in the iterative process and utilize a threshold to eliminate the too noisy results from neighbors. From a theoretical point of view, we analyze the convergence rate of PR-ADMM for general convex objectives, which is $\mathcal{O}(1/K)$ with K being the number of iterations. Despite the first work relieving some critical privacy concerns in the iterations of ADMM, differentially private ADMM still confronts many research challenges. For example, the guarantee of differential privacy (DP) relies on the premise that the optimality of each local problem can be perfectly attained in each ADMM iteration, which may never happen in practice. The model trained by DP ADMM may have low prediction accuracy. Thus, we address these concerns by proposing a novel (Improved) Plausible differentially Private ADMM algorithm [18], called PP-ADMM and IPP-ADMM. In PP-ADMM, each agent approximately solves a perturbed optimization problem that is formulated from its local private data in an iteration, and then perturbs the approximate solution with Gaussian noise to provide the DP guarantee. To further improve the model accuracy and convergence, an improved version IPP-ADMM adopts sparse vector technique (SVT) to determine if an agent should update its neighbors with the current perturbed solution. The agent calculates the difference of the current solution from that in the last iteration, and if the difference is larger than a threshold, it passes the solution to neighbors; or otherwise the solution will be discarded. Moreover, we provide a generalization performance analysis of our new algorithm.

In Section 4, we focus on another common algorithm in collaborative learning architectures, distributed gradient descent-type methods. As we mentioned before, during the collaborative learning process, model updates using local private samples and large-scale parameter exchanges among agents impose severe privacy concerns and communication bottleneck. To address these problems, we propose two differentially private (DP) and communication efficient algorithms, called `Q-DPSGD-1` and `Q-DPSGD-2` [19]. In `Q-DPSGD-1`,

each agent first performs local model updates by a DP gradient descent method to provide the DP guarantee and then quantizes the local model before transmitting it to neighbors to improve communication efficiency. In `Q-DPSGD-2`, each agent injects discrete Gaussian noise to enforce DP guarantee after first quantizing the local model. Moreover, we provide convergence analysis for both convex and non-convex loss functions.

Since in machine learning, privacy concerns related to the training data and unfair behaviors of some decisions with regard to certain attributes (e.g., sex, race) are becoming more critical. In Section 5, we focus on how to construct a fair machine learning model while simultaneously providing privacy protection. Specifically, we propose Purely and Approximately Differential private and Fair Classification algorithms [20], called PDFC and ADFC, respectively, by a calibrated functional mechanism, i.e., injecting different amounts of Laplace noise regarding different attributes to the polynomial coefficients of the constrained objective function to ensure ϵ-differential privacy and reduce effects of discrimination.

Finally, we conclude the dissertation and discusses some potential directions for future research in Section 6. Moreover, Furthermore, some other work during my PhD not included in the thesis include two of our published papers [21, 22] and two submitted manuscript [23, 24].

2 Preliminaries

For the privacy-preserving data analysis, the standard privacy metric, Differential privacy (DP) [25, 26], is proposed to measure the privacy risk of each data sample in the dataset, and has already been adopted in many machine learning domains [22, 20, 27, 28, 29]. Basically, under DP framework, privacy protection is guaranteed by limiting the difference of the distribution of the output regardless of the value change of any one sample in the dataset.

Definition 1 ((ϵ, δ)-**DP [25]**). *A randomized mechanism $\mathcal{M}$ satisfies (ϵ, δ)-DP if for any two neighboring datasets D and $\hat{D}$ differing in at most one single data sample, and for any possible output $o \in Range(\mathcal{M})$, we have* $\Pr[\mathcal{M}(D) = o] \leq e^{\epsilon}\Pr[\mathcal{M}(\hat{D}) = o] + \delta$.

Here ϵ, δ are privacy loss parameters that indicate the strength of the privacy protection from the mechanism $\mathcal{M}$. The privacy protection is stronger if they are smaller. The above privacy definition reduces to pure DP when $\delta = 0$ and when $\delta > 0$, it is referred to as approximate DP. We can achieve pure and approximate DP by utilizing two popular approaches called Laplace and Gaussian Mechanism, both of which share the same form $\mathcal{M}(D) = \mathcal{M}_q(D) + \mathbf{u}$, where $\mathcal{M}_q(D)$ is a query function over dataset D, and $\mathbf{u}$ is random noise. We also denote two neighboring datasets D and $\hat{D}$ as $D \sim \hat{D}$, and denote $\text{Lap}(\lambda)$ as a zero-mean Laplace distribution with scale parameter λ.

Definition 2 (**Laplace Mechanism [25]**). *Given any function $\mathcal{M}_q : \mathcal{D} \to \mathbb{R}^d$, the Laplace Mechanism is defined as: $\mathcal{M}_L(D, q, \epsilon) = \mathcal{M}_q(D) + \mathbf{u}$, where $\mathbf{u}$ is drawn from a Laplace distribution* $\text{Lap}(\frac{\Delta_1}{\epsilon})$ *with scale parameter proportional to the L_1-sensitivity of $\mathcal{M}_q$ given as* $\Delta_1 = \sup_{D \sim \hat{D}} \|\mathcal{M}_q(D) - \mathcal{M}_q(\hat{D})\|_1$. *Laplace Mechanism preservers ϵ-differential privacy.*

Definition 3 (**Gaussian Mechanism [25]**). *Given any function $\mathcal{M}_q : \mathcal{D} \to \mathbb{R}^d$, the Gaussian Mechanism is defined as: $\mathcal{M}_G(D, q, \epsilon, \delta) = \mathcal{M}_q(D) + \mathbf{u}$, where $\mathbf{u}$ is drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2 I_d)$ with $\sigma \geq \frac{\sqrt{2\ln(1.25/\delta)}\Delta_2}{\epsilon}$, and Δ_2 is the L_2-sensitivity of function $\mathcal{M}_q$, i.e., $\Delta_2 = \sup_{D \sim \hat{D}} \|\mathcal{M}_q(D) - \mathcal{M}_q(\hat{D})\|_2$. Gaussian Mechanism provides (ϵ, δ)-differential privacy.*

Note that one limitation of Gaussian Mechanism is that privacy budget ϵ should be in $[0, 1]$. To relax the constraint of ϵ, we can use the following Extended Gaussian Mechanism.

Definition 4 (**Extended Gaussian Mechanism [30]**). *Given any function $\mathcal{M}_q : \mathcal{D} \to \mathbb{R}^d$, the Extended Gaussian Mechanism is defined as: $\mathcal{M}_{EG}(D, q, \epsilon, \delta) = \mathcal{M}_q(D) + \mathbf{u}$, where $\mathbf{u}$ is drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2 I_d)$ with $\sigma \geq \frac{\sqrt{2}\Delta_2}{2\epsilon}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}) + \epsilon})$. Extended Gaussian Mechanism provides (ϵ, δ)-differential privacy.*

Next, we introduce a generalization of DP, which is called the zero-concentrated DP (zCDP) [31] that uses the Rényi divergence between $\mathcal{M}(D)$ and $\mathcal{M}(\hat{D})$, which can achieve a much tighter privacy loss bound under multiple privacy mechanisms composition.

Definition 5 (ρ-**zCDP [31]**). *We say that a randomized algorithm $\mathcal{M}$ provides ρ-zCDP, if for all neighboring datasets D and $\hat{D}$ and for all $\tau \in (1, \infty)$, we have $D_\tau(\mathcal{M}(D)\|\mathcal{M}(\hat{D})) \leq \rho\tau$, where $D_\tau(\mathcal{M}(D)\|\mathcal{M}(\hat{D}))$ is the τ-Rényi divergence [1] between the distribution $\mathcal{M}(D)$ and the distribution $\mathcal{M}(\hat{D})$.*

The following lemmas show that the Gaussian mechanism satisfies zCDP, the composition theorem of ρ-zCDP, and the relationship among ρ-zCDP, ϵ-DP, and (ϵ, δ)-DP.

Lemma 1 ([31]). *The Gaussian mechanism with noise $\mathcal{N}(0, \sigma^2)$ satisfies $\Delta_2^2/(2\sigma^2)$-zCDP.*

Lemma 2 (**Serial Composition [31]**). *If randomized mechanisms $\mathcal{M}_1 : \mathcal{D} \to \mathcal{R}_1$ and $\mathcal{M}_2 : \mathcal{D} \to \mathcal{R}_2$ obey ρ_1-zCDP and ρ_2-zCDP, respectively. Then their composition defined as $\mathcal{M}'' : D \to \mathcal{R}_1 \times \mathcal{R}_2$ by $\mathcal{M}'' = (\mathcal{M}_1, \mathcal{M}_2)$ obeys $(\rho_1 + \rho_2)$-zCDP.*

Lemma 3 (**DP to zCDP conversion [31]**). *If a randomized mechanism $\mathcal{M}$ provides ϵ-DP, then $\mathcal{M}$ is $\frac{1}{2}\epsilon^2$-zCDP. Moreover, for $\mathcal{M}$ to satisfy (ϵ, δ)-DP, it suffices to satisfy ρ-zCDP with $\rho = \frac{\epsilon^2}{4\ln(1/\delta)}$.*

Lemma 4 (**zCDP to DP conversion [31]**). *If a randomized mechanism $\mathcal{M}$ obeys ρ-zCDP, then $\mathcal{M}$ obeys $(\rho + 2\sqrt{\rho\ln(1/\delta)}, \delta)$-DP for all $0 < \delta < 1$.*

[1] Definition can be found in [31]

Sparse Vector Technique A powerful approach for achieving DP employs the sparse vector technique (SVT) [32], which takes a sequence of queries with bounded sensitivity Δ against a fixed sensitive dataset and outputs a vector representing whether each answer to the query exceeds a threshold or not. A unique advantage of SVT is that it can output some query answer without paying additional privacy cost. Specifically, as shown in [33], SVT has the following four steps. (i), We first compute a noisy threshold $\hat{\gamma}$ by adding a threshold noise $\text{Lap}(\frac{\Delta}{\epsilon_1})$ to the predefined threshold γ. (ii), We then utilize a noise $v_i \sim \text{Lap}(\frac{2c\Delta}{\epsilon_2})$ to perturb each query q_i. (iii), We compare each noisy query answer $q_i(D) + \nu_i$ with the noisy threshold $\hat{\gamma}$ and respond whether it is higher ($\top$) or lower ($\perp$) than the threshold. (iv), This procedure continues until the number of $\top$'s in the output meets the predefined bound c. According to [33], the SVT algorithm satisfies the ϵ-DP with $\epsilon = \epsilon_1 + \epsilon_2$. In order to analyze the privacy guarantee of SVT under the zCDP framework, we utilize the conversion result in Lemma 3. We can see that SVT satisfies $\frac{1}{2}\epsilon^2$-zCDP.

Next, we introduce a new generalization of DP, called Rényi differential privacy (RDP) [34], which is widely used in stochastic iterative learning algorithms due to the tighter composition and subsample amplification results.

Definition 6 (RDP)**.** *Given any neighboring datasets $D, \hat{D}$ differing by one element, we say that a randomized mechanism $\mathcal{M}$ satisfies (ρ, ϵ)-RDP, if for $\rho > 1, \epsilon > 0$, we have*

$$\mathscr{D}_\rho\big(\mathcal{M}(D)||\mathcal{M}(\hat{D}))\big) := \log \mathbb{E}\big(\mathcal{M}(D)/\mathcal{M}(\hat{D})\big)^\rho/(\rho - 1) \leq \epsilon,$$

where the expectation is taken over $\mathcal{M}(\hat{D})$.

The following lemmas from [34] The following three lemmas are some properties of RDP, which will be used in the proofs of our theorems.

Lemma 5. *The Gaussian Mechanism satisfies $(\rho, \rho\Delta_2^2/(2\sigma^2))$-RDP.*

Lemma 6. *If k randomized mechanisms $\mathcal{M}_i$ for $i \in [k]$, satisfy (ρ, ϵ_i)-RDP, then their composition $\big(\mathcal{M}_1(D), \cdots, \mathcal{M}_k(D)\big)$ satisfies $(\rho, \sum_{i=1}^k \epsilon_i)$-RDP. Moreover, the input of the i-th mechanism can base on the outputs of previous $(i-1)$ mechanisms.*

Lemma 7. *If a randomized mechanism $\mathcal{M}$ satisfies (ρ, ϵ)-RDP, then $\mathcal{M}$ satisfies $(\epsilon + \log(1/\delta)/(\rho - 1), \delta)$-DP for all $\delta \in (0, 1)$.*

Lemma 8 ([35])**.** *Let $\mathcal{M}$ be any randomized algorithm that obeys $(\rho, \epsilon(\rho))$-RDP. Then applying $\mathcal{M}$ on the poisson subsampled dataset as input, it satisfies $(\rho, \epsilon'(\rho))$-RDP. Let γ be the poisson sampling probability and then we have for integer $\rho \geq 2$,*

$$\epsilon'(\rho) \leq \frac{1}{\rho} \log \left\{ (1-\gamma)^{\rho-1}(\rho\gamma - \gamma + 1) + \binom{\rho}{2} \gamma^2 (1-\gamma)^{\rho-2} e^{\epsilon(2)+3} \sum_{l=3}^{\rho} \binom{\rho}{l} (1-\gamma)^{\rho-l} \gamma^l e^{(l-1)\epsilon(l)} \right\}.$$

3 Plausible Differentially Private ADMM Based Distributed Machine Learning

3.1 Introduction

With the rapid development of sensing technologies, the past decade has witnessed an explosive growth in size of generated data. For instance, the Cisco Visual Networking Index predicts that the number of mobile devices will be 11.6 billion by the year 2020 and the data will be generated at each smart phone with an average size of 4.4 gigabytes per month [36]. Because of the ability to exploit the collective computing power of the local computing nodes, distributed machine learning is a promising tool to accommodate such deluge data, especially when data is produced from different locations [37]. Several distributed optimization approaches have been developed to design distributed machine learning architectures such as distributed subgradient descent algorithm [38, 39] and alternating direction method of multipliers (ADMM) [40], among which the ADMM typically achieves a fast convergence rate $\mathcal{O}(1/K)$, where K is the number of iterations [41]. Thus, in this work, we aim to design distributed machine learning algorithm with ADMM.

Under the framework of ADMM, a large scale machine learning problem is divided into several sub-problems solved by a connected network of agents locally over local training data, and the local machine learning models are exchanged among the neighbors. However, as many recent works [42, 43, 44] indicate, the local machine learning models exchanged during the iterative process may result in privacy leakage of the sensitive training data such as medical records or financial data.

To prevent such information leakage, differential privacy [25, 45] has been exploited as a well-defined framework for performing machine learning over sensitive data. Intuitively, it works by injecting random noise to the model parameters so that an adversary with arbitrary background knowledge cannot confidently make any conclusions about whether a data sample is utilized in training a model or not. Many pioneering works have focused on integrating differential privacy with ADMM [42, 43, 44, 46]. In [42], Zhang and Zhu

proposed a dual variable perturbation approach, where the dual variable of each agent at each ADMM iteration is perturbed. This approach can provide dynamic differential privacy, a new privacy framework capturing the distributed and iterative nature of ADMM. However, Zhang and Zhu only imposed a privacy constraint on each iteration and did not give a total privacy loss bound over the entire iterative procedure, which makes it hard to balance the tradeoff between the utility of the proposed algorithm and privacy guarantees. Later, in [43], Zhang et al. developed a penalty perturbation method and gave the total privacy loss of all agents during the entire process. Moreover, in [44], Zhang et al. employed the penalty perturbation method and modified the original ADMM to repeatedly use the existing computational results in order to further reduce the privacy loss. However, privacy analysis provided in above works [42, 43, 44] requires the objective functions of the learning problem are strongly convex. Our privacy analysis only needs to assume that the gradient of the loss function is bounded. Huang et al. in [46] and Ding et al. in [47] also performed privacy analysis under mild conditions of objective functions, whereas their approaches need a central server to average all shared primal variables. Instead of requiring a central server, our approach is implemented in a fully decentralized manner.

In this section, we first present our work on differentially private robust ADMM algorithm (PR-ADMM), which adds Gaussian noise with decaying variance to perturb exchanged variables at each iteration. To reduce the negative effects of noise addition, we propose two noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme*, and we utilize a threshold U to examine whether the results from neighbors are too noisy. However, the guarantee of DP in this work relies on the premise that the optimality of each local problem can be perfectly attained in each iteration during the whole training procedure, which is seldom seen in practice. Further, the trained models exhibit severe degradation in terms of the convergence performance and model accuracy, compared to their non-private versions.

Then, we present our second work on (**I**mproved) **P**lausible differentially **P**rivate **ADMM**

based distributed machine learning algorithm called PP-ADMM and IPP-ADMM, respectively. Instead of requiring each local problem to reach the optimality, PP-ADMM is able to release a noisy approximate solution of the local optimization with Gaussian noise related to the optimization accuracy, while preserving DP. To further improve the utility, we propose an improved version of PP-ADMM, i.e., IPP-ADMM, by exploiting the sparse vector technique (SVT) to check whether the current approximate solution has enough difference from that of the previous iteration. Moreover, the privacy analysis of our algorithms based on the zero-concentrated DP (zCDP) yields a tight privacy loss bound. We analyze the generalization performance of PP-ADMM.

Notations: We denote $\|x\|_2$ as the Euclidean norm of a vector x and $\langle x, y\rangle$ as the inner product of two vectors x and y. Further, given a semidefinite matrix G, $\sqrt{x^T G x}$ represents the G-norm of x, i.e., $\|x\|_G$. We also denote $\phi_{max}(G)$ as the nonzero largest of G and $\phi_{min}(G)$ as the smallest nonzero singular value of G.

3.2 Problem Setting and Preliminaries

In this work, we consider a connected network contains N agents with node set $\mathcal{N} = \{1, \cdots, N\}$, and each agent i has a dataset D_i with $D_i = \{(z_i^n, y_i^n)\}_{n=1}^{|D_i|}$, where $z_i^n \in \mathcal{X}$ is a feature vector and $y_i^n \in \mathcal{Y}$ is a label. The communication among agents can be represented by an undirected graph $G = \{\mathcal{N}, \mathcal{E}\}$, where $\mathcal{E}$ denotes the set of communication links between agents. Note that two agents i and j can communicate with each other only when they are neighbors, i.e., $(i, j) \in \mathcal{E}$. We also denote the set of neighbors of agent i as $\mathcal{V}_i$. The goal is to cooperatively train a classifier $x \in \mathbb{R}^d$ over the union of all local datasets in a decentralized fashion (i.e., no centralized controller) while keeping the privacy for each data sample, which can be formulated as the following Empirical Risk Minimization (ERM) problem,

$$\min_{x \in \mathbb{R}^d} \sum_{i=1}^{N} \frac{1}{|D_i|} \sum_{n=1}^{|D_i|} \mathcal{L}(y_i^n x^T z_i^n) + \hat{\lambda}\mathcal{R}(x), \tag{1}$$

where $\mathcal{L}(\cdot) : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^d \to \mathbb{R}$ stands for a convex loss function with $|\mathcal{L}'(\cdot)| \leq 1$ and $0 < \mathcal{L}''(\cdot) \leq c_1$, $\mathcal{R}(x) : \mathbb{R}^d \to \mathbb{R}$ is a differentiable and 1-strongly convex regularizer to prevent overfitting, and $\hat{\lambda} \geq 0$ refers to a regularizer parameter that controls the impact of regularizer. We assume that each feature vector z_i^n is normalized to $\|z_i^n\|_2 \leq 1$. Note that the formulations of classification in machine learning like logistic regression, or support vector machines, can also be fallen into the framework of ERM.

In order to solve the ERM problem (1) in a decentralized manner, we adopt the simple but efficient optimization method, ADMM. We then in the following subsection review some preliminaries about ADMM algorithm for solving Problem (1). It is easy to see that the ERM problem (1) can be equivalently reformulated as the following consensus form by introducing x_i, that is, the local copy of common classifier x at agent i,

$$\begin{aligned} \min_{\{x_i\},\{\rho_{ij}\}} \quad & \textstyle\sum_{i=1}^{N} f_i(x_i) \\ \text{s.t.} \quad & x_i = \rho_{ij},\ x_j = \rho_{ij},\ i \in \mathcal{N}, j \in \mathcal{V}_i, \end{aligned} \tag{2}$$

where $\{\rho_{ij} | i \in \mathcal{N}, j \in \mathcal{V}_i\}$ is a set of slack variables to enforce all local copies are equal to each other, i.e., $x_1 = x_2 = \cdots, = x_N$, and $f_i(x_i) = \frac{1}{|D_i|} \sum_{n=1}^{|D_i|} \mathcal{L}(y_i^n x_i^T z_i^n) + \frac{\hat{\lambda}}{N} \mathcal{R}(x_i)$. According to Problem (2), each agent i can minimize local function $f_i(x_i)$ over its own private dataset with respect to x_i, under the consensus constraints. In [43], ADMM is employed to optimize Problem (2) in a decentralized fashion. By defining a dual variable λ_i for agent i, and introducing the following notion, $\mathcal{L}_{non}(x_i, D_i) = f_i(x_i) + (2\lambda_i^t)^T x_i + \eta \sum_{j \in \mathcal{V}_i} ||\frac{1}{2}(x_i^t + x_j^t) - x_i||_2^2$, ADMM then has the following iterative updates in the $(t+1)$-th iteration,

$$x_i^{t+1} = \operatorname*{argmin}_{x_i} \ \mathcal{L}_{non}(x_i, D_i) \tag{3}$$

$$\text{and} \ \ \lambda_i^{t+1} = \lambda_i^t + \frac{\eta}{2} \sum_{j \in \mathcal{V}_i} (x_i^{t+1} - x_j^{t+1}), \tag{4}$$

where $\eta > 0$ is a penalty parameter. Note that the reason why the variable ρ_{ij} is not appeared in (3) and (4) is that it can be expressed by using the primal variable x_i, as

shown in [48]. In the iteration $t+1$, each agent $i \in \mathcal{N}$ updates its local x_i^{t+1} via (3) by using its previous results x_i^t and λ_i^t, and the shared local classifiers x_j^t from its neighbors $j \in \mathcal{V}_i$. Next, agent i broadcasts x_i^{t+1} to all its neighboring agents. After obtaining all of its neighboring computation results, each agent updates the dual variable λ_i^{t+1} through (4).

For a clear presentation, according to [49], problem (2) can be written in a matrix form as

$$\begin{aligned} \min_{x,p} \quad & f(x) + g(p) \\ \text{s.t.} \quad & Ax + Bp = 0, \end{aligned} \tag{5}$$

where $x := [x_1, x_2, \cdots, x_N]^T \in \mathbb{R}^{Nd}$, p is a vector concatenating all $\{p_{ij}\}$, $g(p) = 0$ and $A := [A_1; A_2]$ with $A_1, A_2 \in \mathbb{R}^{2Ed \times Nd}$ whose (q,i)-th element $(A_1)_{qi} = 1, (A_2)_{qi} = 1$ and all other elements are zeros if the q-th element of p is p_{ij}. Moreover, $B := [-I_{2Ed}; -I_{2Ed}]$, and aggregated function $f : \mathbb{R}^{Nd} \to \mathbb{R}$ is defined as $f(x) = \sum_{i=1}^{N} f_i(x_i)$.

The augmented Lagrangian function of (5) is given by

$$L_c(x, p, \lambda) = f(x) + \langle Ax + Bp, \lambda \rangle + \frac{\eta}{2} \|Ax + Bp\|_2^2, \tag{6}$$

where $\lambda \in \mathbb{R}^{4Ed}$ is Lagrangian multiplier and η is a positive penalty parameter.

With ADMM algorithm, alternatively, $L_c(x, p, \lambda)$ is minimized in terms of variables x, p and λ. At iteration $t+1$, the updates of ADMM are

$$\nabla f(x^{t+1}) + A^T \lambda^t + \eta A^T (Ax^{t+1} + Bp^t) = 0, \tag{7}$$

$$B^T \lambda^t + \eta B^T (Ax^{t+1} + Bp^{t+1}) = 0, \tag{8}$$

$$\text{and} \quad \lambda^{t+1} - \lambda^t - \eta (Ax^{t+1} + Bp^{t+1}) = 0. \tag{9}$$

If we let $\lambda = [\beta; \gamma]$ with $\beta, \gamma \in \mathbb{R}^{2EN}$, $H_+ = A_1^T + A_2^T$ and $H_- = A_1^T - A_2^T$, the above

ADMM updates can be simplified as

$$\nabla f(x^{t+1}) + \alpha^t + 2\eta M x^{t+1} - \eta L_+ x^t = 0 \tag{10}$$

$$\text{and} \quad \alpha^{t+1} - \alpha^t - \eta L_- x^{t+1} = 0, \tag{11}$$

where $\alpha = H_-\beta \in \mathbb{R}^{Nd}$ is a new Lagrange multiplier, and $M = \frac{1}{2}(L_+ + L_-)$ with $L_+ = \frac{1}{2}H_+H_+^T$ and $L_- = \frac{1}{2}H_-H_-^T$. Note that L_+ and L_- are the extended signless and signed Laplacian matrices of the network.

Remember that $x := [x_1, x_2, \cdots, x_N]^T \in \mathbb{R}^{Nd}$, where x_i is the local classifier of agent i. After simple manipulations, the matrix form of ADMM updates (10) and (11) are translated to the updates of agent i by

$$\nabla f_i(x_i^{t+1}) + \alpha_i^t + 2\eta|\mathcal{V}_i|x_i^{t+1} = \eta\left(|\mathcal{V}_i|x_i^t + \sum_{j\in\mathcal{V}_i} x_j^t\right) \tag{12}$$

$$\text{and} \quad \alpha_i^{t+1} = \alpha_i^t + \eta\left(|\mathcal{V}_i|x_i^{t+1} - \sum_{j\in\mathcal{V}_i} x_j^{t+1}\right), \tag{13}$$

where $\alpha_i \in \mathbb{R}^d$ is the local Lagrange multiplier of agent i and α is the concatenated form of all α_i. At iteration $t+1$, every agent i updates the local x_i^{t+1} through (12) using its previous x_i^t, α_i^t and its neighbors' previous result x_j^k with $j \in \mathcal{V}_i$, and then broadcasts x_i^{t+1} to all its neighboring agents $j \in \mathcal{V}_i$. After collecting all x_j^{t+1} from its neighbors, agent i updates its local multiplier α_i through (13).

3.3 Differentially Private Robust ADMM

In this section, we propose a novel differentially private robust ADMM algorithm (PR-ADMM). Specifically, to provide differential privacy of each training data point, we let individual agent adds Gaussian noise to the local classifiers before sharing to neighboring agents. Moreover, we propose two techniques to mitigate the effects of noise addition and guarantee convergence property. The first technique is that we design two kinds of noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme*,

to carefully adjust the scale of noise in the iterative process. The second technique is to set a threshold U to decide whether the noisy classifiers from neighboring agents introduce too much noise or not. If it is, this agent would not use these noisy classifiers to do ADMM updates. In addition, the privacy framework, dynamic zero-concentrated differential privacy, is utilized to measure the privacy guarantee of PR-ADMM.

The details of PR-ADMM are given in Algorithm 1. First of all, we choose a Gaussian noise variance decay scheme from *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme* to determine the relationship between $(\sigma^2)_i^{t+1}$ and $(\sigma^2)_i^t$. In each iteration, each agent i computes the primal variable x_i^{t+1} by solving the subproblem in (14) over its own dataset D_i (Line 15). Then, each agent i computes the value of $\sum_{k=0}^{t}\|\tilde{x}_i^k-\tilde{x}_j^k\|_2$ for every neighboring agent $j\in\mathcal{V}_i$, which is a good criterion of measuring the deviation from a consensus at current iteration [50]. When $\sum_{k=0}^{t}\|\tilde{x}_i^k-\tilde{x}_j^k\|_2$ is greater than a threshold U, it means that agent j's variable $\tilde{x}_j^t$ is too noisy. Then each agent i will replace $\tilde{x}_j^t$ with its own variable $\tilde{x}_i^t$ to do the primal variable update (Line 15). After that, each agent adds a noise ξ_i^{t+1} drawn from a Gaussian distribution $\mathcal{N}(0,(\sigma^2)_i^{t+1}I_d)$ to perturb the local variable x_i^{t+1}, according to the chosen noise variance decay scheme. Then, the perturbed local variable $\tilde{x}_i^{t+1}$ is sent by agent i to all its neighboring agents $j\in\mathcal{V}_i$. At last, each agent updates the dual variable α_i^{t+1} through (15). The corresponding ADMM iterations are as

$$\nabla f_i(x_i^{t+1})+\alpha_i^t+2\eta|\mathcal{V}_i|x_i^{t+1}=\eta\left(|\mathcal{V}_i|\tilde{x}_i^t+\sum_{j\in\mathcal{V}_i}\tilde{x}_j^t\right) \tag{14}$$

$$\text{and}\quad \alpha_i^{t+1}=\alpha_i^t+\eta\left(|\mathcal{V}_i|\tilde{x}_i^{t+1}-\sum_{j\in\mathcal{V}_i}\tilde{x}_j^{t+1}\right). \tag{15}$$

Now how to set the value of threshold U is important. Since $\sum_{k=0}^{t}\|\tilde{x}_i^k-\tilde{x}_j^k\|_2\le U$ and $\sum_{k=0}^{t}\|Q\tilde{x}^k\|_2=\frac{1}{\sqrt{2}}\sum_{k=0}^{t}\sum_{(i\in\mathcal{N},j\in\mathcal{V}_i)}\|\tilde{x}_i^k-\tilde{x}_j^k\|_2$ with $Q=\sqrt{L_-/2}$, then the value of noisy local deviation statistics $\sum_{t=0}^{T}\|Q\tilde{x}^t\|_2$ is upper bounded by $\sqrt{2}EU$. Note that if we set the threshold U as $U=\hat{U}/(\sqrt{2}E)$, we have $\sum_{t=0}^{T}\|Q\tilde{x}^t\|_2$ is upper bounded by $\hat{U}$, where $\hat{U}$ is the upper bound of the noise-free local deviation statistics $\sum_{t=0}^{T}\|Qx^t\|_2$. The upper bound

Algorithm 1 Differentially Private Robust ADMM

Input: datasets $\{D_i\}_{i=1}^N$; initial variables $x_i^0 \in \mathbb{R}^d$ and $\alpha_i^0 = 0_d$; threshold U; time period T_p, decay rate $R_P \in (0,1)$ and $R_T > 0$; initial variances $(\sigma^2)_i^1$ for all agents i;
Choose noise variance decay scheme (Line 3-7).
if *Periodic Linear Decay Scheme* is chosen **then**
    $(\sigma^2)_i^{t+1} = (\sigma^2)_i^1 \times R_P^{\lfloor t/T_p \rfloor}$.
else
    $(\sigma^2)_i^{t+1} = (\sigma^2)_i^1 \times \frac{1}{R_T t(t+1)}$. //*Iteration-Based Decay Scheme*
end if
for $t = 0, \ldots, T-1$ **do**
    for $i = 1, \cdots, N$ **do**
        if $\sum_{k=0}^t \|\tilde{x}_i^k - \tilde{x}_j^k\|_2 > U$ with $j \in \mathcal{V}_i$ **then**
            Replace $\tilde{x}_j^t$ with $\tilde{x}_i^t$.
        else
            Keep $\tilde{x}_j^t$.
        end if
        Compute x_i^{t+1} by solving $\nabla f_i(x_i^{t+1}) + \alpha_i^t + 2\eta|\mathcal{V}_i|x_i^{t+1} - \eta\left(|\mathcal{V}_i|\tilde{x}_i^t + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^t\right) = 0$.

        Generate noise $\xi_i^{t+1} \sim \mathcal{N}(0, (\sigma^2)_i^{t+1} I_d)$.
        Perturb x_i^{t+1}: $\tilde{x}_i^{t+1} = x_i^{t+1} + \xi_i^{t+1}$.
    end for
    for $i = 1, \cdots, N$ **do**
        Broadcast $\tilde{x}_i^{t+1}$ to all neighbors $j \in \mathcal{V}_i$.
    end for
    for $i = 1, \cdots, N$ **do**
        Compute α_i^{t+1} from
        $\alpha_i^{t+1} = \alpha_i^t + \eta\left(|\mathcal{V}_i|\tilde{x}_i^{t+1} - \sum_{j\in\mathcal{V}_i}\tilde{x}_j^{t+1}\right)$.
    end for
end for
Output: $\{\tilde{x}_i^T\}_{i=1}^N$ for any $i \in \mathcal{N}$.

$\hat{U}$ can be obtained from the following Lemma.

Lemma 9. *If we randomly initialize x^0 and the gradient $\nabla f(x)$ is bounded as $\|\nabla f(x)\|_2 \leq V_2$ and the feasible x is bounded as $\|x\|_2 \leq V_1$, in conventional ADMM (10-11), we have*

$$\sum_{t=0}^{T} \|Qx^t\|_2 \leq \hat{U} = (\phi_{max}(L_+) + 2\phi_{max}(Q))V_1^2 + \frac{2V_2^2}{\eta^2\phi_{min}(L_-)} + 1.$$

Proof. The upper bound of $\sum_{t=0}^T \|Qx^t\|_2$ can be directly derived from Lemma 8 in [51] if we use the inequality $\|a+b\|_2^2 \leq \|a\|_2^2 + \|b\|_2^2$ for all $a, b \in \mathbb{R}^n$ and $\|Qx^0\| \leq \phi_{max}(Q)\|x^0\|$. □

In distributed settings, the output of the algorithm includes all intermediate results generated at every stage of the learning and final result. For this reason, we present the dynamic zero concentrated differential privacy framework to quantify the privacy leakage of ADMM-based algorithms.

Definition 7 (**Dynamic ρ^t-zCDP [52]**). *Consider a connected network $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$ that contains a set of agents/nodes $\mathcal{N} = \{1, \cdots, N\}$ and each agent possesses a training dataset D_i, and $\tilde{D} = \cup_{i\in\mathcal{N}} D_i$. We denote $\mathcal{T}$ is a randomized version of ADMM algorithm with updates (12) and (13). Let $\mathcal{T}_i^k$ be the agent-i-dependent sub-algorithm of $\mathcal{T}$, which corresponds to ADMM update (12) at t-iteration that outputs x_i^t. A randomized algorithm $\mathcal{T}$ gives dynamic ρ_i^t-zCDP if for all datasets D_i and $\hat{D}_i$ differing at most a single record, and for all agents $i \in \mathcal{N}$, and for all t during a learning process, the privacy loss variable of an outcome $o \in Range(\mathcal{T})$, $Z_i^t(o) = \ln \frac{\Pr[x_{i,D_i}^t = o]}{\Pr[x_{i,\hat{D}_i}^t = o]}$ satisfies $\mathbb{E}[e^{(\tau-1)Z_i^t(o)}] \leq e^{(\tau-1)\tau\rho_i^t}$ $\forall \tau \in (1, \infty)$.*

For dynamic zCDP algorithms, the adversaries cannot obtain additional information by observing the intermediate results and final results at each step. Since the added noise may destroy the convergence behavior and lead to poor model performance. It is vital to carefully design and adjust privacy budget allocation for each iteration, i.e., dynamically reducing the noise variance in the iterative process, instead of just adding a noise ξ_i^{t+1} for agent i in iteration $t+1$ [42].

Here we propose two kinds of noise variance decay schemes, which effectively reduce the bad impact of noise and stabilize the convergence property.

Periodic Linear Decay Scheme In a period of time T_p, there is a decay rate $R_P \in (0, 1)$ to describe the decrease of noise variance. The mathematical form is

$$(\sigma^2)_i^{t+1} = (\sigma^2)_i^1 \times R_P^{\lfloor t/T_p \rfloor}, \tag{16}$$

where $(\sigma^2)_i^1$ is the initial noise variance determined by agent i and the value of T_p decides how often to reduce noise variance regards the number of iterations. Without loss of generality,

suppose the total iteration number T is divisible by T_p.

Iteration-Based Decay Scheme In the iteration $t+1$, the noise variance $(\sigma^2)_i^{t+1}$ can be obtained based on the previous noise variance. It has the mathematical form as

$$(\sigma^2)_i^{t+1} = (\sigma^2)_i^1 \times \frac{1}{R_T t(t+1)}, \tag{17}$$

where $R_T > 0$ is decay rate.

Before showing PR-ADMM satisfies dynamic zCDP, we first estimate the sensitivity of the local primal variable x_i^{t+1} as shown in the following lemma.

Lemma 10. *The sensitivity of local primal variable x_i^{t+1}, denoted by Δ_i, is $\frac{V}{\eta|\mathcal{V}_i|}$, where V is the Lipschitz constant of the loss function $\mathcal{L}(\cdot)$, $|\mathcal{V}_i|$ is the number of neighboring agents of agent i, and η is a positive penalty parameter.*

Proof. According to subproblem (12) and definition of sensitivity, we have

$$x_{i,D_i}^{t+1} = -\frac{1}{2\eta|\mathcal{V}_i|}\nabla f_i(x_i^{t+1}, D_i) + \frac{1}{2|\mathcal{V}_i|}\left(|\mathcal{V}_i|\tilde{x}_i^t + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^t\right) - \frac{1}{2\eta|\mathcal{V}_i|}\alpha_i^t$$

$$\text{and} \quad x_{i,\hat{D}_i}^{t+1} = -\frac{1}{2\eta|\mathcal{V}_i|}\nabla f_i(x_i^{t+1}, \hat{D}_i) + \frac{1}{2|\mathcal{V}_i|}\left(|\mathcal{V}_i|\tilde{x}_i^t + \sum_{j\in\mathcal{V}_i}\tilde{x}_j^t\right) - \frac{1}{2\eta|\mathcal{V}_i|}\alpha_i^t,$$

where D_i and $\hat{D}_i$ are two neighboring datasets. Without loss of generality, suppose only the first data sample in D_i and $\hat{D}_i$ is different, say (y_i^1, z_i^1) and $(\hat{y}_i^1, \hat{z}_i^1)$ respectively. Then

by the definitions of sensitivity, we have

$$\begin{aligned}
\Delta_i &= \|x_{i,D_i}^{t+1} - x_{i,\hat{D}_i}^{t+1}\|_2 \\
&= \frac{1}{2\eta|\mathcal{V}_i|}\|\nabla f_i(x_i^{t+1}, D_i) - \nabla f_i(x_i^{t+1}, \hat{D}_i)\|_2 \\
&= \frac{1}{2\eta|\mathcal{V}_i|}\|\frac{1}{|D_i|}\sum_{n=1}^{|D_i|}\nabla\mathcal{L}(y_i^n, z_i^n, x_i^{t+1}) + \frac{\hat{\lambda}}{N}\nabla\mathcal{R}(x_i^{t+1}) \\
&\quad - \frac{1}{|\hat{D}_i|}\sum_{n=1}^{|\hat{D}_i|}\nabla\mathcal{L}(\hat{y}_i^n, \hat{z}_i^n, x_i^{t+1}) - \frac{\hat{\lambda}}{N}\nabla\mathcal{R}(x_i^{t+1})\|_2 \\
&= \frac{1}{2\eta|\mathcal{V}_i|}\|\nabla\mathcal{L}(y_i^1, z_i^1, x_i^{t+1}) - \nabla\mathcal{L}(\hat{y}_i^1, \hat{z}_i^1, x_i^{t+1})\|_2 \\
&\leq \frac{V}{\eta|\mathcal{V}_i|}.
\end{aligned}$$

In the last inequality, we use the fact the Lipschitz constant V is an upper bound of $\|\nabla\mathcal{L}(\cdot)\|_2$. □

The following theorems show the privacy guarantee of PR-ADMM.

Theorem 1. *The PR-ADMM algorithm satisfies the dynamic ρ_i^{t+1}-zCDP, where $\rho_i^{t+1} = \frac{V^2}{2\eta^2|\mathcal{V}_i|^2(\sigma^2)_i^{t+1}}$.*

Proof. The privacy loss variable of $\tilde{x}_i^{t+1}$ on an output o over two neighboring datasets D_i and $\hat{D}_i$ is

$$Z_i^{t+1}(o) = \ln\frac{\Pr[\tilde{x}_{i,D_i}^{t+1} = o]}{\Pr[\tilde{x}_{i,\hat{D}_i}^{t+1} = o]}.$$

Since $\tilde{x}_i^{t+1} = x_i^{t+1} + \xi_i^{t+1}$ and $\xi_i^{t+1} \sim \mathcal{N}(0, (\sigma^2)_i^{t+1}I_d)$, the probability distribution $\tilde{x}_{i,D_i}^{t+1}$ is $\mathcal{N}(x_{i,D_i}^{t+1}, (\sigma^2)_i^{t+1}I_d)$, and the probability distribution of $\tilde{x}_{i,\hat{D}_i}^{t+1}$ is $\mathcal{N}(x_{i,\hat{D}_i}^{t+1}, (\sigma^2)_i^{t+1}I_d)$. According to Lemma 2.5 in [31] and $\forall\tau \in (1,\infty)$, the Rényi divergence is given by

$$D_\tau(\mathcal{N}(x_{i,D_i}^{t+1}, (\sigma^2)_i^{t+1}I_d)\|\mathcal{N}(x_{i,\hat{D}_i}^{t+1}, (\sigma^2)_i^{t+1}I_d)) = \frac{\tau\|x_{i,D_i}^{t+1} - x_{i,\hat{D}_i}^{t+1}\|_2^2}{2(\sigma^2)_i^{t+1}} = \frac{\tau\Delta_i^2}{2(\sigma^2)_i^{t+1}}.$$

Then, we have

$$\begin{aligned}\mathbb{E}[e^{(\tau-1)Z_i^{t+1}(o)}] &\leq e^{(\tau-1)D_\tau(\mathcal{N}(x_{i,D_i}^{t+1},(\sigma^2)_i^{t+1}I_d)\|\mathcal{N}(x_{i,\hat{D}_i}^{t+1},(\sigma^2)_i^{t+1}I_d))} \\ &= e^{(\tau-1)\tau\Delta_i^2/[2(\sigma^2)_i^{t+1}]} \\ &\leq e^{(\tau-1)\tau\frac{V^2}{2\eta^2|\mathcal{V}_i|^2(\sigma^2)_i^{t+1}}} \\ &= e^{(\tau-1)\tau\rho_i^{t+1}}.\end{aligned}$$

Therefore, PR-ADMM provides the dynamic ρ_i^{t+1}-zCDP at each agent i with $\rho_i^{t+1} = \frac{V^2}{2\eta^2|\mathcal{V}_i|^2(\sigma^2)_i^{t+1}}$. □

The parameter ρ_i^{t+1} in Theorem 1 only inspects the privacy loss of one agent in each iteration. However, it does not show the privacy guarantee when an adversary uses the revealed results from all iterations to perform inference. Therefore, the total privacy loss over the entire computational process and the entire network should be calculated.

For two kinds of noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme*, we leverage (ϵ, δ)-differential privacy to derive the total privacy loss as shown in the Theorem 2 and Theorem 3, respectively.

Theorem 2. *For any $R_P \in (0,1)$ and $\delta \in (0,1)$, if Periodic Linear Decay Scheme is chosen, the PR-ADMM algorithm is (ϵ, δ)-differential privacy with $\epsilon = \max_{i\in\mathcal{N}} \rho_i^{total} + 2\sqrt{\rho_i^{total}\ln 1/\delta}$, where*

$$\rho_i^{total} = \frac{V^2}{2\eta^2|\mathcal{V}_i|^2(\sigma^2)_i^1}\left(\frac{T_p(1-R_P^{T/T_p})}{R_P^{T/T_p-1}-R_P^{T/T_p}}-1\right),$$

and T_p is time period, and $R_P \in (0,1)$ is the decay rate, and K is the total number of iterations.

Proof. According to Theorem 1, PR-ADMM satisfies dynamic ρ_i^{t+1}-zCDP. It ensures that each primal variable x_i^{t+1} perturbed by noise drawn from the Gaussian distribution $\mathcal{N}(0,(\sigma^2)_i^{t+1}I_d)$ is ρ_i^{t+1}-zCDP at $t+1$ iteration. By the composition theorem in Lemma 2

and for each agent, PR-ADMM provides $\sum_{t=0}^{T-1} \rho_i^{t+1}$-zCDP. Since *Periodic Linear Decay Scheme* is chosen, together with the result in Theorem 1, we have $\rho_i^{t+1} = \rho_i^1 / R_P^{\lfloor t/T_p \rfloor}$ and PR-ADMM is ρ_i^{total}-zCDP for each agent with

$$\rho_i^{total} = \rho_i^1 \left(\frac{T_p(1 - R_P^{T/T_p})}{R_P^{T/T_p - 1} - R_P^{T/T_p}} - 1 \right).$$

By Lemma 4 and $\forall \delta \in (0,1)$, PR-ADMM satisfies $(\epsilon_i^{total}, \delta)$-differential privacy with $\epsilon_i^{total} = \rho_i^{total} + 2\sqrt{\rho_i^{total} \ln 1/\delta}$. Therefore, considering all of agents, the total privacy loss of PR-ADMM is bounded by (ϵ, δ)-differential privacy with $\epsilon = \max_{i \in \mathcal{N}} \ \epsilon_i^{total}$.

□

Theorem 3. *For any $R_T > 0$ and $\delta \in (0,1)$, if Iteration-Based Decay Scheme is chosen, the PR-ADMM algorithm is (ϵ, δ)-differential privacy with*

$$\epsilon = \max_{i \in \mathcal{N}} \left(\frac{\rho_i^1 (R_T T(T^2-1)+3)}{3} + 2\sqrt{\frac{\rho_i^1 (R_T T(T^2-1)+3) \ln 1/\delta}{3}} \right),$$

where $\rho_i^1 = \frac{V^2}{2\eta^2 |\mathcal{V}_i|^2 (\sigma^2)_i^1}$ and T is the total number of iterations.

Proof. Since *Iteration-Based Decay Scheme* is chosen, together with the result in Theorem 1, we have $\rho_i^{t+1} = R_T t(t+1)\rho_i^1$. Then PR-ADMM is ρ_i^{total}-zCDP for each agent with

$$\rho_i^{total} = \rho_i^1 \frac{R_T T(T^2-1)+3}{3}.$$

By Lemma 4 and $\forall \delta \in (0,1)$, PR-ADMM satisfies $(\epsilon_i^{total}, \delta)$-differential privacy, where

$$\begin{aligned} \epsilon_i^{total} &= \rho_i^{total} + 2\sqrt{\rho_i^{total} \ln 1/\delta} \\ &= \frac{\rho_i^1 (R_T T(T^2-1)+3)}{3} + 2\sqrt{\frac{\rho_i^1 (R_T T(T^2-1)+3) \ln 1/\delta}{3}}. \end{aligned}$$

Therefore, considering all of agents, the total privacy loss of PR-ADMM is bounded by

(ϵ, δ)-differential privacy with

$$\begin{aligned}\epsilon &= \max_{i\in\mathcal{N}} \ \epsilon_i^{total} \\ &= \max_{i\in\mathcal{N}} \ \left(\frac{\rho_i^1(R_T T(T^2-1)+3)}{3} + 2\sqrt{\frac{\rho_i^1(R_T T(T^2-1)+3)\ln 1/\delta}{3}}\right).\end{aligned}$$

□

3.3.1 Convergence Analysis

In this section, we present the convergence analysis of proposed PR-ADMM algorithm for general convex objective functions. The updates of PR-ADMM can be written in matrix forms as

$$\nabla f(x^{t+1}) + \alpha^t + 2\eta M x^{t+1} - \eta L_+ \tilde{x}^t = 0 \tag{18}$$

$$\text{and} \ \ \alpha^{t+1} - \alpha^t - \eta L_- \tilde{x}^{t+1} = 0, \tag{19}$$

where $\tilde{x}^t = x^t + \xi^t$ and $\xi^t \in \mathbb{R}^{Nd}$ is a vector concatenating all noise variables $\{\xi_i^t\}$.

Given the perturbed primal variable $\tilde{x}^t$, two auxiliary sequences r^t and q^t, and a matrix G are defined as follows

$$r^t = \sum_{s=0}^{t} Q\tilde{x}^s, \ q^t = \begin{pmatrix} r^t \\ x^t \end{pmatrix}, \text{ and } \ G = \begin{pmatrix} \eta I & 0 \\ 0 & \eta L_+/2 \end{pmatrix},$$

where $Q = \sqrt{L_-/2}$. Since the network is connected, the Laplcaian matrix L_- is positive semi-definite.

Substituting (19) into (18), we obtain $x^{t+1} = -\frac{M^{-1}\nabla f(x^{t+1})}{2\eta} + \frac{M^{-1}L_+\tilde{x}^t}{2} - \frac{M^{-1}L_-}{2}\sum_{s=0}^{t}\tilde{x}^s$. Based on the auxiliary sequence r^t and the fact $M = (L_- + L_+)/2$, we further have $\frac{\nabla f(x^{t+1})}{\eta} + 2Qr^{t+1} + L_+(\tilde{x}^{t+1} - \tilde{x}^t) = 2M^{-1}\xi^{t+1}$.

Proposition 1. *For any $r \in \mathbb{R}^{Nd}$ and $k > 0$, we have*

$$\frac{f(x^{t+1}) - f(x^*)}{\eta} + \left\langle 2r, Qx^{t+1} \right\rangle \leq \frac{1}{\eta}\left(\|q^t - q^*\|_G^2 - \|q^{t+1} - q^*\|_G^2\right) + \frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^t\|_2^2 + \left\langle \xi^{t+1}, 2Q(r^{t+1} - r)\right\rangle,$$

where x^* is the optimal solution of (5) and $q^* = \begin{pmatrix} r \\ x^* \end{pmatrix}$.

We can now prove the following convergence results of PR-ADMM for the general convex problem.

Theorem 4. *Suppose the objective function $f(x)$ is general convex. In PR-ADMM, if Periodic Linear Decay Scheme is chosen, we have*

$$\mathbb{E}[f(\hat{x}^T) - f(x^*)] \leq \frac{\eta}{T}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) + \frac{\eta}{T}\underbrace{\frac{d\phi_{max}^2(L_+)T_p\sum_{i=1}^N(\sigma^2)_i^1}{2\phi_{min}(L_-)(1-R_P)}}_{\text{Accumulated noise term}}$$

with $0 < R_P < 1$ and time period T_p, where the expectation is taking with respect to the noise and $\hat{x}^T = \frac{1}{T}\sum_{t=1}^T x^t$.

Proof. Summing Proposition 1 from $t = 0$ to $t = T - 1$, we have

$$\begin{aligned}
&\frac{1}{\eta}(\sum_{t=1}^T f(x^t) - f(x^*)) + \left\langle 2r, Qx^t\right\rangle \\
&\leq \sum_{t=1}^T\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{t-1}\|_2^2 + \left\langle \xi^t, 2Q(r^t - r)\right\rangle\right) + \frac{1}{\eta}\|q^0 - q^*\|_G^2 \\
&\leq \sum_{t=1}^T\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{t-1}\|_2^2 + \|2Q\xi^t\|_2\left(\hat{U} + \|r\|_2\right)\right) + \frac{1}{\eta}\|q^0 - q^*\|_G^2 \\
&\leq \sum_{t=1}^T\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{t-1}\|_2^2\right) + \frac{1}{\eta}\|q^0 - q^*\|_G^2 + \sum_{t=1}^T\|2Q\xi^t\|_2\left(\hat{U} + \|r\|_2\right) \\
&\leq \sum_{t=1}^T\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{t-1}\|_2^2\right) + 2\hat{U}\left(\hat{U} + \|r\|_2\right) + \frac{1}{\eta}\|q^0 - q^*\|_G^2,
\end{aligned}$$

where we use the fact that $\sum_{t=1}^{T}\|Q\xi^t\|_2 \leq \hat{U}$. Letting $r=0$, there is

$$\frac{1}{\eta}(\sum_{t=0}^{T} f(x^t) - f(x^*)) \leq \sum_{t=1}^{T}\left(\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\|\xi^{t-1}\|_2^2\right) + 2\hat{U}^2 + \|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2.$$

By taking expectation of above function and using Jensen's inequality and convexity of the functions, we have

$$\begin{aligned}\mathbb{E}[f(\hat{x}^T) - f(x^*)] &\leq \frac{\eta}{T}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) + \frac{\eta}{T}\sum_{t=1}^{T}\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\mathbb{E}\|\xi^{t-1}\|_2^2 \\ &\leq \frac{\eta}{T}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) + \frac{\eta}{T}\frac{d\phi_{max}^2(L_+)T_p\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)(1-R_P)},\end{aligned}$$

where $\hat{x}^T = \frac{1}{T}\sum_{t=1}^{T} x^t$. In the second inequality, we use the sum of infinity terms of geometric sequence. □

Theorem 5. *Suppose the objective function $f(x)$ is general convex. In PR-ADMM, if Iteration-Based Decay Scheme is chosen, we have*

$$\mathbb{E}[f(\hat{x}^T) - f(x^*)] \leq \frac{\eta}{T}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) + \frac{\eta}{T}\underbrace{\frac{d\phi_{max}^2(L_+)\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)R_T}}_{\textit{Accumulated noise term}}$$

with $R_T > 0$, where the expectation is taking with respect to the noise and $\hat{x}^T = \frac{1}{T}\sum_{t=1}^{T} x^t$.

Proof. Similar to the proof of Theorem 4, we have

$$\begin{aligned}\mathbb{E}[f(\hat{x}^T) - f(x^*)] &\leq \frac{\eta}{T}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) + \frac{\eta}{T}\sum_{t=1}^{T}\frac{\phi_{max}^2(L_+)}{2\phi_{min}(L_-)}\mathbb{E}\|\xi^{t-1}\|_2^2 \\ &\leq \frac{\eta}{T}\left(\|Qx^0\|_2^2 + \|x^0 - x^*\|_{L_-/2}^2 + 2\hat{U}^2\right) + \frac{\eta}{T}\frac{d\phi_{max}^2(L_+)\sum_{i=1}^{N}(\sigma^2)_i^1}{2\phi_{min}(L_-)R_T},\end{aligned}$$

where $\hat{x}^T = \frac{1}{T}\sum_{t=1}^{T} x^t$. □

Remark 1. *As we can see from the above two theorems, if the variance of Gaussian noise decays according to Periodic Linear Decay Scheme and Iteration-Based Decay Scheme, then the averaged function value approaches the minimum function value with a convergence rate*

of $\mathcal{O}(1/T)$. Note that the non-private decentralized ADMM in [50] also achieves a $\mathcal{O}(1/T)$ rate for a general convex problem.

3.3.2 Numerical Experiments

In this section, we experimentally evaluate the performance of PR-ADMM under two noise variance decay schemes: *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme* on binary classification tasks. Specifically, our evaluation considers logistic regression as the loss function.

Logistic regression The logistic regression loss function on a data sample (z, y) with $y \in \{+1, -1\}$ is defined as $\mathcal{L}(y_i^n x_i^T z_i^n) = \log(1+\exp(-y_i^n x_i^T z_i^n))$, and the regularizer $\mathcal{R}(x_i) = \|x_i\|_2^2$.

Data preprocessing We also use the Adult dataset from UCI Machine Learning Repository, as in [42, 43, 44]. The dataset consists of 48,842 personal records with, including age, work-class, sex, race, income, etc. Our goal is to predict whether the annual income an individual is more than \$50k or not. We preprocess the data by removing all individuals with missing values. We also normalize the feature vectors such that its l_2 norm is at most 1 while transforming labels of Adult $\{> 50\text{k}, \leq 50\text{k}\}$ to $\{+1, -1\}$.

Baseline algorithms In our experiments, we compare our PR-ADMM algorithm against four benchmark algorithms, namely, DVP, M-ADMM, and R-ADMM and Non-private. The private ADMM algorithm using dual variable perturbation is called DVP [42]. ADMM with a penalty perturbation, proposed in [43], is referred to M-ADMM. Based on the penalty perturbation, R-ADMM with repeatedly using the existing computational results to make updates is proposed in [44]. Furthermore, we denote the non-private ADMM algorithm [49] as Non-private baseline. Finally, we denote our PR-ADMM with *Periodic Linear Decay Scheme* and *Iteration-Based Decay Scheme* as PR-ADMM (Per) and PR-ADMM (Iter), respectively. **Setup** As shown in Figure 1, we consider a bidirectionally connected network with $N = 5$ agents, and each agent is randomly assigned $|D_i| = 8000$ data samples for training. In the testing process, we random sample 1000 instance from the remaining

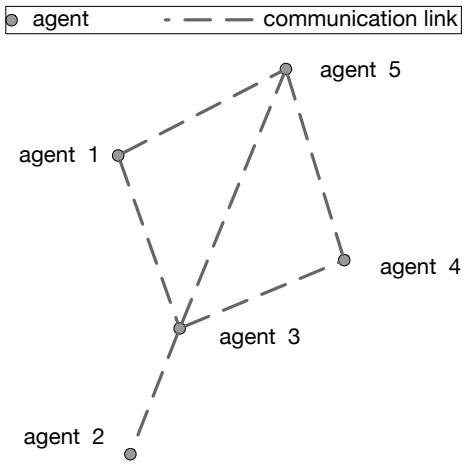
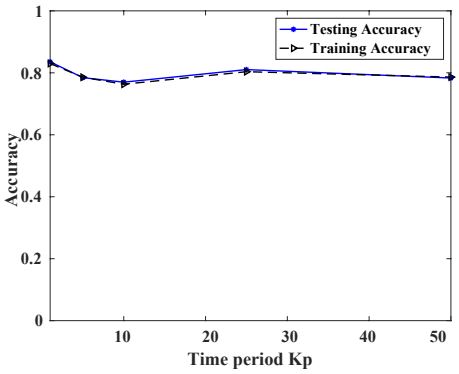
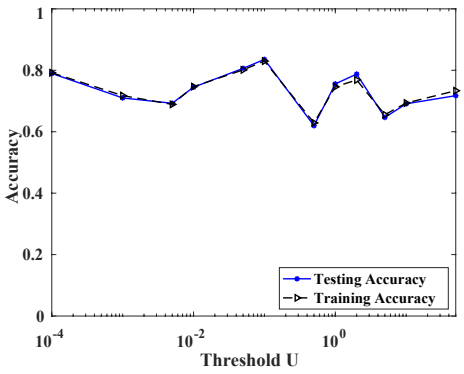


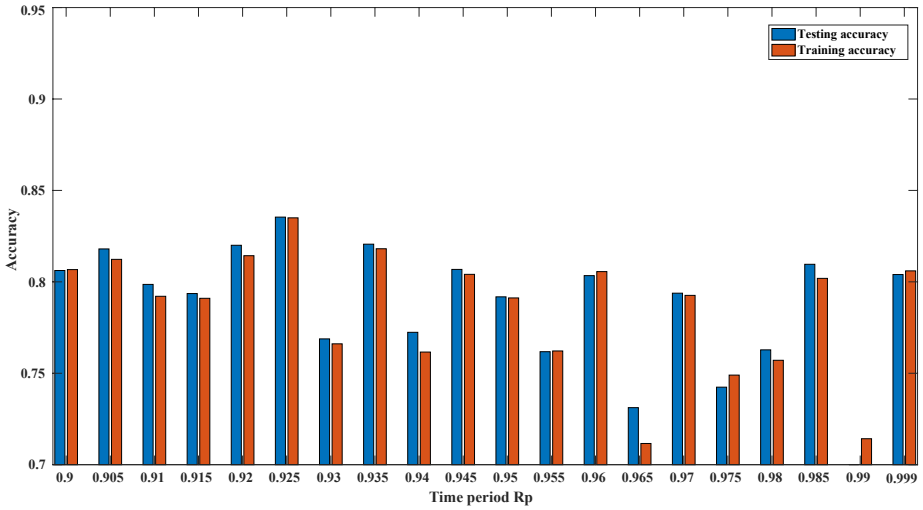
Figure 1: A network with five agents ($N = 5$).



(a) Different values of time period T_p.

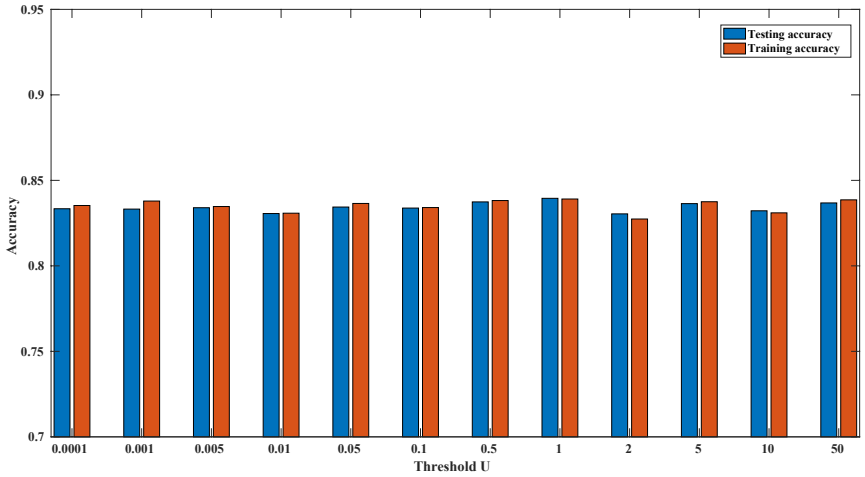


(b) Different values of threshold U (*Periodic Linear Decay*).

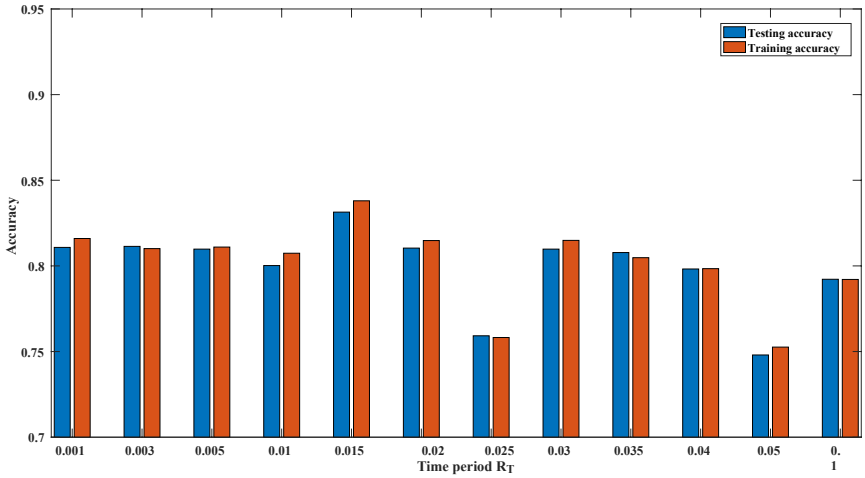


(c) Different values of decay rate R_P.

Figure 2: Hyperparameters of PR-ADMM (*Periodic Linear Decay*).



(a) Different values of threshold U (*Iteration-Based Decay*).



(b) Different values of decay rate R_T.

Figure 3: Hyperparameters of PR-ADMM (*Iteration-Based Decay*).

dataset. We set $\eta = 0.5$ and the total iteration number $T = 50$. For privacy parameters, we consider the total privacy loss $\epsilon = \{0.1, 0.5, 1, 5, 10\}$ and $\delta = 0.0001$.

Evaluation We evaluate the convergence of the algorithms with respect to the average loss defined by $\mathcal{L}_t = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|D_i|}\sum_{n=1}^{|D_i|}\mathcal{L}(y_i^n(x_i^t)^T z_i^n)$. Moreover, the accuracy is measured by classification accuracy defined as follows $Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$. Since each of the baseline algorithms introduces randomness due to noise, we perform 10 independent runs of algorithms and report the mean of accuracy (Testing and Training). Moreover, we also record both the mean and standard deviation of the average loss. The smaller the standard deviation, the more stable of the algorithm.

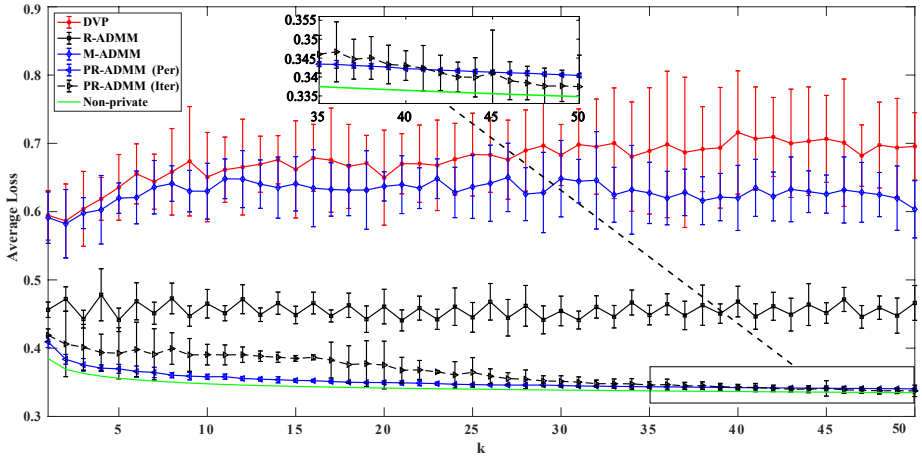


Figure 4: Compare convergence: Total privacy loss $\epsilon = 10$.

As we can see from Algorithm 1, there are some hyperparameters for tuning, such as threshold U, time period T_p, decay rate R_P and R_T. Given the total privacy loss $\epsilon = 10$ and the total iteration number $T = 50$, we manipulate different hyperparameters separately, while keeping the rest unchanged to show their impact on testing/training accuracy[2]. Figure 2(a) shows the impacts of time period T_p on the performance of PR-ADMM with *Periodic Linear Decay Scheme*. The value of time period T_p represents how often to reduce noise variance regards the number of iterations. From the figure, we see that $T_p = 1$ achieves the highest testing/training accuracy. Figure 2(b) illustrates how classification accuracy changes with varying values of U, the threshold U is to eliminate the too noisy results from neighbors. As it was shown in the figure, $U = 0.1$ achieves the best testing/training accuracy. Figure 2(c) describes how classification accuracy changes with varying values of R_P. The parameter R_P controls how fast the noise variance decreases. As it can be seen from the figure, $R_P = 0.925$ is best. To see the impact of decay rate R_T and threshold U on performance of PR-ADMM with *Iteration-Based Decay Scheme*, Figure 3(b) and Figure 3(a) describe how R_T and U affect the testing/training accuracy, respectively. From these figures, we see the testing/training accuracy are highest when $R_T = 0.015$ and $U = 1$.

Figure 4 compares the convergence performance of PR-ADMM algorithm with other

[2]Note that tuning hyperparameters may not be private. In the future work, we can consider differentially private hyperparameter tuning algorithms proposed in [53, 54] to achieve end-to-end differential privacy.

baseline algorithms. Compared to DVP and M-ADMM, R-ADMM indeed improves the privacy-utility tradeoff significantly, i.e., R-ADMM has the low value of average loss, with repeatedly using the existing computational results. However, R-ADMM performs many iterations that do not help decrease the average loss value. As it was shown in the figure, both PR-ADMM (Per) and PR-ADMM (Iter) significantly outperform all other algorithms and get close to the best achievable average loss (Non-private) during the entire iterative process. This is because, with carefully adjusting privacy budgets and setting a threshold to eliminate the too noisy intermediate results, the negative effects of noise addition have been reduced and the convergence behavior of ADMM has maintained.

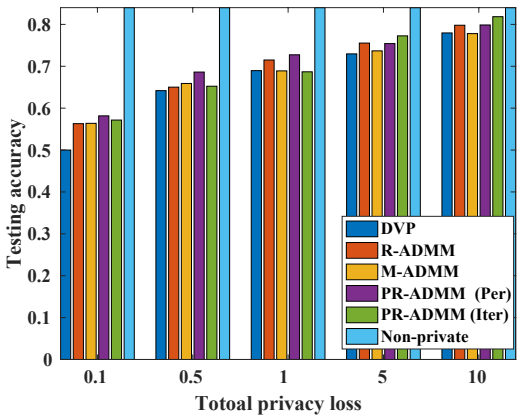


Figure 5: Compare testing accuracy by varying total privacy loss ϵ.

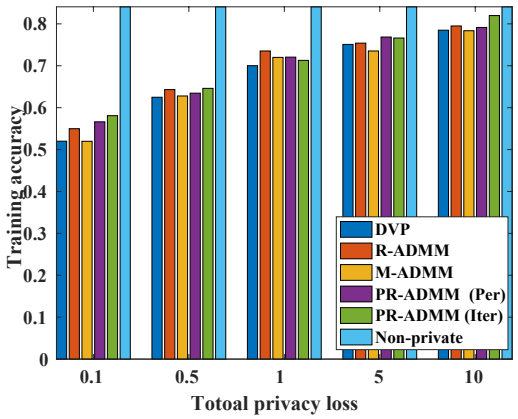


Figure 6: Compare training accuracy by varying total privacy loss ϵ.

Figure 5 and Figure 6 illustrate the testing and training accuracy achieved by each algorithm changes as the value of ϵ increases. We can see that both PR-ADMM (Per) and PR-ADMM (Iter) achieve the competitive testing/training accuracies on a wide range of values for total privacy loss ϵ.

3.4 Plausible Private ADMM

In this section, we will present our plausible differentially private (PP-ADMM) by adding Gaussian noise related to the maximum tolerable gradient norm of perturbed objective in each ADMM iteration, which relaxes the requirement of exact optimal solution as shown in

[55, 43, 44], to provide differential privacy guarantee of each training data sample during the iterative procedure. We also adopt the privacy framework of zCDP to compute much tighter privacy loss estimation of PP-ADMM. In addition, the generalization performance guarantees of PP-ADMM is provided by measuring the number of data samples at each agent to achieve a specific criteria.

Specifically, in each iteration, we perturb the subproblem (3) with the objective perturbation method the same as used in previous studies [55, 43, 44], where a random linear vector $(b_{i1})^T x_i$ is injected to the objective function, and b_{i1} is a random vector drawn from a zero mean Gaussian distribution $\mathcal{N}(0, \sigma_{i1}^2 I_d)$. Consequently the objective function (3) used to update the primal variable x_i^{t+1} becomes the modified function as

$$\mathcal{L}_{per}(x_i, D_i) = f_i(x_i) + (2\lambda_i^t + b_{i1})^T x_i + \eta \sum_{j \in \mathcal{V}_i} ||\frac{1}{2}(x_i^t + x_j^t) - x_i||_2^2, \tag{20}$$

where $f_i(x_i) = \frac{1}{|D_i|} \sum_{n=1}^{|D_i|} \mathcal{L}(y_i^n x_i^T z_i^n) + \frac{\hat{\lambda}}{N} \mathcal{R}(x_i)$. In order to ensure DP guarantee, as pointed out in [55, 43, 44], each agent $i \in \mathcal{N}$ needs to find the optimal solution $\tilde{x}_i^{t+1}$ of the perturbed objective function $\mathcal{L}_{per}(x_i, D_i)$, i.e.,

$$\tilde{x}_i^{t+1} = \underset{x_i}{\operatorname{argmin}} \ \ \mathcal{L}_{per}(x_i, D_i). \tag{21}$$

However, the subproblem (21) may not be easy to solve and obtain an optimal solution in a finite time. For instance, if we choose logistic regression as loss function, the subproblem (21) cannot yield an analytical solution due to the complicated form of logistic regression. Especially when the problem dimension or the number of training samples is large, obtaining optimal solution might not be feasible in every iteration.

Thus, we consider obtaining the approximate solution of perturbed objective function $\mathcal{L}_{per}(x_i, D_i)$ to provide privacy guarantees when the optimal solution is not attainable. Specifically, we approximately solve the perturbed problem until the norm of gradient of $\mathcal{L}_{per}$ is within a pre-defined threshold β. However, due to the limitations of objective

perturbation method [56], releasing this inexact solution leads to the failure of providing DP guarantee. We thus perturb the approximated solution $\hat{x}_i^{t+1}$ with another random noise b_{i2} from Gaussian distribution $\mathcal{N}(0, \sigma_{i2}^2 I_d)$, to "fuzz" the difference between $\hat{x}_i^{t+1}$ and the optimal solution $\tilde{x}_i^{t+1}$. Note that the noise variance σ_{i2}^2 has the parameter β about the maximum tolerable gradient norm, which leads to a trade-off between the gradient norm bound and the difficulty of obtaining an approximate solution within the norm bound.

Algorithm 2 Plausible Private ADMM

Input: datasets $\{D_i\}_{i=1}^N$; initial variables $x_i^0 \in \mathbb{R}^d$ and $\lambda_i^0 = 0_d$; step size η; privacy parameters, ϵ_{i1}, δ_{i1}, ϵ_{i3}, ρ_{i2}; Optimizer $\mathcal{O}(\cdot,\cdot) : \mathcal{F} \times \boldsymbol{\beta} \to \mathbb{R}^d$ ($\mathcal{F}$ is the class of objectives, and $\boldsymbol{\beta}$ is the optimization accuracy, i.e., the gradient norm of objectives); gradient norm threshold $\beta \in \boldsymbol{\beta}$.
Set ϵ_{i1}, δ_{i1}, ϵ_{i3}, $\rho_{i2} > 0$ such that $\epsilon_{i1} > \epsilon_{i3}$.
Set regularizer parameter $\hat{\lambda} \geq \max_i \frac{2.8Nc_1}{(\epsilon_{i1}-\epsilon_{i3})|D_i|}$.
for $t = 0, \ldots, T-1$ **do**
    for $i = 1, \ldots, N$ **do**
        Generate noise $b_{i1} \sim \mathcal{N}(0, \sigma_{i1}^2 I_d)$ with $\sigma_{i1} = 2\sqrt{2\ln(1.25/\delta_{i1})}/(|D_i|\epsilon_{i3})$.
        Construct the perturbed objective function $\mathcal{L}_{per}(x_i, D_i)$ according to (20).
        Compute an approximate solution $\hat{x}_i^{t+1}$: $\hat{x}_i^{t+1} = \mathcal{O}(\mathcal{L}_{per}(x_i, D_i), \beta)$.
        Generate noise $b_{i2} \sim \mathcal{N}(0, \sigma_{i2}^2 I_d)$ with $\sigma_{i2} = \beta/[\sqrt{2\rho_{i2}}(\frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i|)]$.
        Perturb $\hat{x}_i^{t+1}$: $x_i^{t+1} = \hat{x}_i^{t+1} + b_{i2}$.
    end for
    for $i = 1, \ldots, N$ **do**
        Broadcast x_i^{t+1} to all neighbors $j \in \mathcal{V}_i$.
    end for
    for $i = 1, \ldots, N$ **do**
        Update local dual variables λ_i^{t+1} from $\lambda_i^{t+1} = \lambda_i^t + \frac{\eta}{2}\sum_{j\in\mathcal{V}_i}(x_i^{t+1} - x_j^{t+1})$.
    end for
end for

The key steps of PP-ADMM algorithm are summarized in Algorithm 2. The privacy parameters $(\epsilon_{i1}, \delta_{i1})$ are used to perturb the objective function while the parameter ρ_{i2} being used to perturb the approximate solution. Moreover, the parameter ϵ_{i3}, a portion of ϵ_{i1}, is used to scale the noise injected to the objective function, and the remaining privacy budget $(\epsilon_{i1} - \epsilon_{i3})$ is allocated to setting the regularizer parameter. Notice that we also define an **Optimizer** $\mathcal{O}(\cdot,\cdot) : \mathcal{F} \times \boldsymbol{\beta} \to \mathbb{R}^d$, where $\mathcal{F}$ is the class of objectives, and $\boldsymbol{\beta}$ is the optimization accuracy, i.e., the gradient norm of objectives. Each agent i then constructs the perturbed function $\mathcal{L}_{per}(x_i, D_i)$ with a Gaussian random vector b_{i1} and finds an inexact

solution $\hat{x}_i^{t+1}$ where the norm of gradient is lower than β, i.e., $\hat{x}_i^{t+1} = \mathcal{O}(\mathcal{L}_{per}(x_i, D_i), \beta)$.

After that each agent i generates a random Gaussian noise b_{i2} and transmits $x_i^{t+1} = \hat{x}_i + b_{i2}$ to its neighbors $j \in \mathcal{V}_i$. Finally, each agent updates the local dual variables λ_i^{t+1} via $\lambda_i^{t+1} = \lambda_i^t + \frac{\eta}{2} \sum_{j \in \mathcal{V}_i} (x_i^{t+1} - x_j^{t+1})$.

3.4.1 Privacy Analysis

Here, we provide the privacy guarantee of PP-ADMM (Algorithm 2) in the following two theorems. Due to the limited space, we only provide a proof idea of Theorem 6, and the detailed proof can be found in Appendix.

Theorem 6. *The PP-ADMM in Algorithm 2 satisfies ρ_i-zCDP for each agent i with $\rho_i = T(\rho_{i1} + \rho_{i2})$, where $\rho_{i1} = \epsilon_{i1}^2/(4\ln(1/\delta_{i1}))$, and $\rho_{i2} > 0$ is the privacy budget for perturbing the approximate solution.*

Proof Sketch. For achieving ρ_i-zCDP for each agent i at $t+1$ iteration in Algorithm 2, we first divide the output of $t+1$ iteration into two parts. The first part is to obtain the optimal solution $\tilde{x}_i^{t+1}$ of the perturbed objective function $\mathcal{L}_{per}(x_i, D_i)$, and the second part is to obtain the approximate solution with Gaussian noise x_i^{t+1}. We then show obtaining the optimal solution $\tilde{x}_i^{t+1}$ provides ρ_{i1}-zCDP with $\rho_{i1} = \epsilon_{i1}^2/(4\ln(1/\delta_{i1}))$ for the first part, and releasing an approximate solution in the second part is ρ_{i2}-zCDP. By using the composition of zCDP in Lemma 2, we can get releasing the perturbed primal variable x_i^{t+1} at $t+1$ iteration provides $(\rho_{i1} + \rho_{i2})$-zCDP. Considering T iterations, the total privacy loss for each agent i is bounded by $\rho_i = T(\rho_{i1} + \rho_{i2})$. □

We then give the following parallel composition theorem of ρ-zCDP to provides a together characterization of total privacy loss for distributed algorithms.

Lemma 11 (Parallel Composition [57])**.** *Suppose that a mechanism $\mathcal{M}$ consists of a sequence of k adaptive mechanism $\mathcal{M}_1, \cdots, \mathcal{M}_k$ where each $\mathcal{M}_i : \prod_{j=1}^{i-1} \mathcal{R}_j \times \mathcal{D} \to \mathcal{R}_i$ and $\mathcal{M}_i$ satisfies ρ_i-zCDP. Let $\mathbb{D}_1, \mathbb{D}_2, \cdots, \mathbb{D}_k$ be the result of a randomized partition of the input*

domain $\mathbb{D}$. *The mechanism* $\mathcal{M}(D) = (\mathcal{M}_1(D \cap \mathbb{D}_1), \cdots, \mathbb{M}_k(D \cap \mathbb{D}_k))$ *satisfies* $(\max_i \rho_i)$-*zCDP.*

Based on Lemma 11, we can directly obtain the total privacy loss of PP-ADMM given as follows.

Theorem 7. *The PP-ADMM in Algorithm 2 satisfies* ρ*-zCDP with* $\rho = \max_i \rho_i$ *and satisfies* (ϵ, δ)*-DP with* $\epsilon = \rho + 2\sqrt{\rho \ln(1/\delta)}$.

3.4.2 Sample Complexity Analysis

We next measure the generalization performance of PP-ADMM by focusing on the ERM problem given in Section 3.2. We also assume that data samples of each agent i are drawn from a data distribution $\mathcal{P}$. The expected loss of classifier x_i^t at iteration t is defined as

$$\mathbb{L}(x_i^t) = \mathbb{E}_{(x,y)\sim\mathcal{P}}\left(\mathcal{L}(y(x_i^t)^T x)\right).$$

Following the similar analysis in [56, 55], we first introduce a reference classifier x_{ref} with expected loss $\mathbb{L}(x_{ref})$, and we then measure the generalization performance using the number of samples D_i required at each agent to achieve $\mathbb{L}(x_i^t) \leq \mathbb{L}(x_{ref}) + a_{acc}$, where a_{acc} is the generalization error.

PP-ADMM without Noise

Here, we consider the learning performance at all iterations rather than only the final output. Let the intermediate updated classifier $\hat{x}_{i,non}^{t+1}$ at iteration $t+1$ be $\hat{x}_{i,non}^{t+1} = \mathcal{O}(\mathcal{L}_{non}(x_i, D_i), \beta)$.

Note that $\{\hat{x}_{i,non}^{t+1}\}$ is a sequence of non-private classifier without adding perturbations. Let $x^* = \underset{x_i}{\operatorname{argmin}} \ f_i(x_i, D_i)$ be the optimal output of PP-ADMM without Noise. The sequence $\{\hat{x}_{i,non}^{t+1}\}$ is bounded and $\hat{x}_{i,non}^{t+1}$ converges to x^* as $t \to \infty$. Therefore, there exists a constant $\Delta_{i,non}^{t+1}$ at iteration $t+1$ such that $\mathbb{L}(\hat{x}_{i,non}^{t+1}) \leq \mathbb{L}(x^*) + \Delta_{i,non}^{t+1}$. We then have the following result, and the detailed proof can be found in supplemental material.

Theorem 8. *Consider a regularized ERM problem with $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$, and let x_{ref} be the reference classifier for all agents and $\{\hat{x}_{i,non}^{t+1}\}$ be a sequence of outputs of PP-ADMM without adding noise. If the number of samples at agent i satisfies*

$$|D_i| \geq V \max_t \{ \frac{\log(1/\xi)}{\frac{(a_{acc} - \Delta_{i,non}^{t+1})^2}{2\|x_{ref}\|_2^2} - (1+a)\beta} \}$$

for some constant V, then $\hat{x}_{i,non}^{t+1}$ satisfies

$$\Pr\left[\mathbb{L}(\hat{x}_{i,non}^{t+1}) \leq \mathbb{L}(x_{ref}) + a_{acc} \right] \geq 1 - \xi$$

with $a_{acc} \geq \Delta_{i,non}^{t+1}$.

Remark 2. *As we can see from Theorem 8, the number of data samples $|D_i|$ relies on the l_2-norm of reference classifier $\|x_{ref}\|_2^2$ and the parameter β that bounds the optimization accuracy of the non-private intermediate classifier. The results demonstrate that if $|D_i|$ satisfies $|D_i| \geq V \max_t \{ \frac{\log(1/\xi)}{\frac{(a_{acc} - \Delta_{i,non}^{t+1})^2}{2\|x_{ref}\|_2^2} - (1+a)\beta} \}$, each agent's non-private intermediate classifier will have an additional error less than a_{acc} compared to any classifier with $\|x_{ref}\|_2^2$. Moreover, if $\beta = 0$, the result reduces to $|D_i| \geq V \max_t \{ \frac{2\|x_{ref}\|_2^2 \log(1/\xi)}{(a_{acc} - \Delta_{i,non}^{t+1})^2} \}$, the same as given in [55], which shows that the lower optimization accuracy of the non-private intermediate classifier, the more samples required to achieve the same accuracy.*

PP-ADMM

We then show the sample complexity of the PP-ADMM algorithm. Similar to the analysis in PP-ADMM without noise, we also consider bounding the generalization error of the intermediate classifier x_i^{t+1} of each agent i at all iterations. In order to compare the private classifier x_i^{t+1} with a reference classifier x_{ref}, we follow the same strategy used in [55]. We define a new optimization function $f_i^{new}(x_i, D_i) = f_i(x_i, D_i) + {b_{i1}}^T x_i$ and then solving PP-ADMM algorithm is equivalent to solving a new optimization problem, where each agent i's performs local minimization to get $x_i^{t+1} = \mathcal{O}(f_i^{new}(x_i, D_i), \beta) + b_{i2}$. The sequence of outputs $\{x_i^{t+1}\}$ is bounded and x_i^{t+1} converges to a fixed point x_{new}^* as $t \to \infty$. Thus,

there exists a constant $\Delta_{i,new}^{t+1}$ at $t+1$ iteration, such that $\mathbb{L}(x_i^{t+1}) \leq \mathbb{L}(x_{new}^*) + \Delta_{i,new}^{t+1}$. We then give the following result, and the detailed proof can be found in supplemental material.

Theorem 9. *Consider a regularized ERM problem with $\mathcal{R}(x) = \frac{1}{2}\|x\|_2^2$, and let x_{ref} be the reference classifier for all agents and $\{x_i^{t+1}\}$ be a sequence of outputs of PP-ADMM. If the number of samples at agent i satisfies, for some constant V,*

$$|D_i| \geq V \max_t \{ \frac{\log(1/\xi)}{\frac{(a_{acc} - \Delta_{i,new}^{t+1})^2}{2\|x_{ref}\|_2^2} - (1+a)(\beta + \mathcal{H})} \}$$

with $\mathcal{H} = \frac{\sigma_{i2}(a_{acc} - \Delta_{i,new}^{t+1})\sqrt{2d\log\frac{1}{\xi}}}{\|x_{ref}\|_2^2} + 2\sigma_{i1}^2 d \log\frac{1}{\xi}$, then $\hat{x}_{i,new}^{t+1}$ satisfies

$$\Pr\left[\mathbb{L}(x_i^{t+1}) \leq \mathbb{L}(x_{ref}) + a_{acc}\right] \geq 1 - 3\xi$$

with $a_{acc} \geq \Delta_{i,new}^{t+1}$.

Remark 3. *Compared to Theorem 8, we can see that in Theorem 9, the privacy constraints impose an additional term $\mathcal{H}$ with $\mathcal{H} = \sigma_{i2}(a_{acc} - \Delta_{i,new}^{t+1})\sqrt{2d\log\frac{1}{\xi}}/\|x_{ref}\|_2^2 + 2\sigma_{i1}^2 d\log\frac{1}{\xi}$. If both noise variances σ_{i1} and σ_{i2} are equal to zero, the number of required samples $|D_i|$ will reduce to the same result shown in Theorem 8. Moreover, the additional term $\mathcal{H}$ demonstrates that the higher dimension of features, the more added noise to achieve the same accuracy requires more data samples.*

3.5 Improved Plausible Private ADMM

In this section, we present an improved version of PP-ADMM algorithm called Improved Plausible Private ADMM (IPP-ADMM) by leveraging sparse vector technique (SVT) to improve the performance and reduce the communication cost of PP-ADMM. Compared with current differentially private ADMM algorithms [55, 43, 44], although the proposed PP-ADMM algorithm can ensure DP guarantee without requiring the optimal solution during each ADMM iteration, the primal variable is updated using the local data in every iteration and frequently broadcasted to neighboring agents, which leads to the privacy loss

unavoidably accumulating over the iterations, and compromise the accuracy during the whole training procedure.

Hence, we adopt SVT that can output some local computational results without paying any privacy budget, to check whether current approximate solution has a big enough difference from that of previous iteration, where the difference is quantified by a quality function, $f_i(x_i^t) - f_i(\hat{x}_i^{t+1})$, based on the change of the values of local function over the primal variable from previous iteration and current approximate solution. If a sufficient level of difference α is achieved, each agent transmits the current approximate solution with Gaussian noise to its neighbors. Intuitively, if the difference between the current approximate solution $\hat{x}_i^{t+1}$ and the previously transmitted x_i^t is small, then using either one does not help the convergence of the iterative process, which leads to reducing the communication cost.

However, one difficulty in using SVT is that there is no known priori bound on query (i.e., the quality function) $f_i(x_i^t) - f_i(\hat{x}_i)$. To bound the sensitivity of $f_i(x_i^t) - f_i(\hat{x}_i)$, we apply the clipping method to clipping the loss function $\mathcal{L}(\cdot)$. Given a fixed clipping threshold C_{loss}, we compute the value of loss function $\mathcal{L}(\cdot)$ on each local data sample, clip the values at most C_{loss}, and compute the value of $f_i(x_i^t) - f_i(\hat{x}_i)$ based on the clipped values. Note that we denote this loss function clipping procedure as `Clip`.

The complete procedure of IPP-ADMM algorithm for a single agent is shown in Algorithm 3. The privacy parameters ϵ_1 and ϵ_2 are allocated to perturb the quality function and threshold α, respectively. In each iteration, each agent i first constructs the perturbed function $\mathcal{L}_{per}(x_i, D_i)$ with a Gaussian random vector b_{i1} and finds an inexact solution $\hat{x}_i^{t+1}$, where the norm of gradient is lower than β, i.e., $\hat{x}_i^{t+1} = \mathcal{O}(\mathcal{L}_{per}(x_i, D_i), \beta)$. Then each agent apply the clipping method `Clip` to clip the quality function $f_i(x_i^t) - f_i(\hat{x}_i^{t+1})$ with a clipping threshold C_{loss} to limit the sensitivity of quality function. Further, each agent uses SVT to check whether the difference between the approximate solution $\hat{x}_i^{t+1}$ and x_i^t is below a noisy threshold $\hat{\alpha} = \alpha + \text{Lap}(\frac{2\mathbf{c}C_{loss}}{\epsilon_1})$ via a noisy quality function, $\texttt{Clip}\left[f_i(x_i^t) - f_i(\hat{x}_i^{t+1})\right] + \text{Lap}(\frac{4\mathbf{c}C_{loss}}{\epsilon_2})$. If yes, then agent i does not transmit any computational results and let $x_i^{t+1} = x_i^t$; otherwise, each agent i generates a random noise $b_{i2} \sim \mathcal{N}(0, \sigma_{i2}^2 I_d)$ with $\sigma_{i2} = \beta/\sqrt{2\rho_{i2}}(\frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i|)$,

Algorithm 3 Improved Plausible Private ADMM Run by Agent i

1: **Input:** dataset D_i; initial variables $x_i^0 \in \mathbb{R}^d$ and $\lambda_i^0 = 0_d$; threshold, α; Maximum number of primal variables that can be broadcasted, $\mathbf{c}$; loss function clipping threshold C_{loss}; step size η; privacy parameters, ϵ_{i1}, δ_{i1}, ϵ_{i3}, ρ_{i2}, ϵ_1, ϵ_2; Optimizer $\mathcal{O}(\cdot,\cdot) : \mathcal{F}\times\boldsymbol{\beta} \to \mathbb{R}^d$ ($\mathcal{F}$ is the class of objectives, and $\boldsymbol{\beta}$ is the optimization accuracy, i.e., the gradient norm of objectives); gradient norm threshold $\beta \in \boldsymbol{\beta}$.
2: Set ϵ_{i1}, δ_{i1}, ϵ_{i3}, ρ_{i2}, ϵ_1, $\epsilon_2 > 0$ such that $\epsilon_{i1} > \epsilon_{i3}$.
3: Set regularizer parameter $\hat{\lambda} \geq \max_i \frac{2.8Nc_1}{(\epsilon_{i1}-\epsilon_{i3})|D_i|}$.
4: $count_i = 0$.
5: **for** $t = 0, \ldots, T-1$ **do**
6: Generate noise $b_{i1} \sim \mathcal{N}(0, \sigma_{i1}^2 I_d)$ with $\sigma_{i1} = 2\sqrt{2\ln(1.25/\delta_{i1})}/(|D_i|\epsilon_{i1})$.
7: Construct the perturbed objective function $\mathcal{L}_{per}(x_i, D_i)$ according to (20).
8: Compute an approximate solution $\hat{x}_i^{t+1}$: $\hat{x}_i^{t+1} = \mathcal{O}(\mathcal{L}_{per}(x_i, D_i), \beta)$.
9: **if** $\texttt{Clip}\left[f_i(x_i^t) - f_i(\hat{x}_i^{t+1})\right] + \text{Lap}(\frac{4\mathbf{c}C_{loss}}{\epsilon_2}) \geq \alpha + \text{Lap}(\frac{2\mathbf{c}C_{loss}}{\epsilon_1})$ **then**
10: $count_i = count_i + 1$, **Abort** if $count_i > \mathbf{c}$.
11: Generate noise $b_{i2} \sim \mathcal{N}(0, \sigma_{i2}^2 I_d)$ with $\sigma_{i2} = \beta/[\sqrt{2\rho_{i2}}(\frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i|)]$.
12: Perturb $\hat{x}_i^{t+1}$: $x_i^{t+1} = \hat{x}_i^{t+1} + b_{i2}$.
13: Broadcast x_i^{t+1} to all neighbors $j \in \mathcal{V}_i$.
14: **else**
15: Let $x_i^{t+1} = x_i^t$.
16: **end if**
17: **if** x_j^{t+1} is not received from neighbor $j \in \mathcal{V}_i$ **then**
18: Replace x_j^{t+1} with x_j^t.
19: **else**
20: Keep x_j^{t+1}.
21: **end if**
22: Update local dual variables λ_i^{t+1} from $\lambda_i^{t+1} = \lambda_i^t + \frac{\eta}{2}\sum_{j\in\mathcal{V}_i}(x_i^{t+1} - x_j^{t+1})$.
23: **end for**

and transmits $x_i^{t+1} = \hat{x}_i^{t+1} + b_{i2}$ to its neighbors. Moreover, each agent maintains a counter $count_i$ to bound the total number of broadcasts of primal variables during the whole interactive process. If a predefined transmission number $\mathbf{c}(\mathbf{c} \leq T)$ is exceeded, agent i stops transmitting anything even when the condition in Line 7 is satisfied. Hence, if agent i does not receive x_j^{t+1} from any neighbor $j \in \mathcal{V}_i$, then lets $x_j^{t+1} = x_j^t$. Finally, each agent updates the local dual variables λ_i^{t+1} via $\lambda_i^{t+1} = \lambda_i^t + \frac{\eta}{2}\sum_{j\in\mathcal{V}_i}(x_i^{t+1} - x_j^{t+1})$.

3.5.1 Privacy Analysis

We provide the privacy guarantee of IPP-ADMM (Algorithm 3) in following theorem.

Theorem 10. *The IPP-ADMM in Algorithm 3 satisfies ρ_i'-zCDP for each agent i with*

$\rho'_i = \rho'_1 + \mathbf{c}(\rho_{i1} + \rho_{i2})$, *where* $\rho'_1 = \frac{(\epsilon_1+\epsilon_2)^2}{2}$, $\rho_{i1} = \epsilon_{i1}^2/(4\ln(1/\delta_{i1}))$, $\rho_{i2} > 0$ *is the privacy budget for perturbing the approximate solution, and* $\mathbf{c}$ ($\mathbf{c} < T$) *is the maximum number of primal variables that can be broadcasted. Moreover, the total privacy guarantee of IPP-ADMM is* ρ'*-zCDP with* $\rho' = \max_i \rho'_i$.

Proof. For achieving ρ'_i-zCDP for each agent i in Algorithm 3, we first divide the procedure of the algorithm into two parts. The first part is using SVT to compare the noisy threshold and the perturbed query answer (i.e., the value of quality function) to check the quality of the approximate solution obtained in Step 7 of the Algorithm 2. The second part is to share the approximate solution with Gaussian noise, whose value is above the threshold. We prove that DP mechanism used in the first part provides ρ'_1-zCDP (shown in Lemma 13). Moreover, at each iteration, the privacy budget spending on releasing an approximate solution in the second part is $(\rho_{i1} + \rho_{i2})$-zCDP (shown in Theorem 6). Then, using the composition of zCDP in Lemma 2, we obtain the privacy guarantee of IPP-ADMM for each agent i is $\rho_i = \rho_1 + \mathbf{c}(\rho_{i1} + \rho_{i2})$ by considering $\mathbf{c}$ times of broadcasting primal variables. Lastly, we get a total privacy guarantee of IPP-ADMM, i.e., ρ'-zCDP with $\rho' = \max_i \rho'_i$ by adopting the parallel composition in Lemma 11. □

Before presenting the privacy guarantee of the first part, i.e., compare the noisy threshold and the perturbed query answer to check the quality of the approximate solution, we first give the sensitivity of the clipped quality function as follows.

Lemma 12. *Given a clipping threshold* C_{loss} *of the loss function* $\mathcal{L}(\cdot)$*, the sensitivity of quality function* $f_i(x_i^t) - f_i(\hat{x}_i^{t+1})$ *is at most* $2C_{loss}$*, where* $f_i(x_i) = \frac{1}{|D_i|}\sum_{n=1}^{|D_i|}\mathcal{L}(y_i^n x_i^T z_i^n) + \frac{\hat{\lambda}}{N}\mathcal{R}(x_i)$.

Proof. Fix a pair of adjacent datasets D_i and $\hat{D}_i$ and we also assume that only the first data point in D_i and $\hat{D}_i$ are different, i.e., (z_i^1, y_i^1) and $(\hat{x}_i^1, \hat{y}_i^1)$. According to the definition

of L_1-sensitivity, we have

$$\begin{aligned}
\Delta_f = \quad & \|f_i(x_i^t, D_i) - f_i(\hat{x}_i^{t+1}, D_i) - f_i(x_i^t, \hat{D}_i) + f_i(\hat{x}_i^{t+1}, \hat{D}_i)\|_1 \\
= \quad & \|\mathcal{L}(y_i^1(x_i^t)^T z_i^1) - \mathcal{L}(\hat{y}_i^1(x_i^t)^T \hat{x}_i^1) \\
& - (\mathcal{L}(y_i^1(x_i^{t+1})^T z_i^1) - \mathcal{L}(\hat{y}_i^1(x_i^{t+1})^T \hat{x}_i^1))\|_1 \\
\leq \quad & \|\mathcal{L}(y_i^1(x_i^t)^T z_i^1) - \mathcal{L}(\hat{y}_i^1(x_i^t)^T \hat{x}_i^1)\|_1 \\
& + \|\mathcal{L}(y_i^1(x_i^{t+1})^T z_i^1) - \mathcal{L}(\hat{y}_i^1(x_i^{t+1})^T \hat{x}_i^1)\|_1 \\
\leq \quad & 2C_{loss}. \qquad \square
\end{aligned}$$

Then we show the privacy guarantee of the first part in the following lemma.

Lemma 13. *Given the maximum number of primal variables that we can broadcast,* $\mathbf{c}$*, using SVT to check whether the approximate solution is above the threshold* α *provides* ρ_1*-zCDP with* $\rho_1 = \frac{(\epsilon_1+\epsilon_2)^2}{2}$.

Proof. During the whole training process, each agent will receive a stream of queries (i.e., a stream of clipped quality functions $\texttt{Clip}\left[f_i(x_i^t) - f_i(\hat{x}_i^{t+1})\right]$) with sensitivity $2C_{loss}$ and compare them with a noisy threshold $\alpha + \text{Lap}(\frac{2\mathbf{c}C_{loss}}{\epsilon_1})$. According to Theorem 1 in [33], this procedure satisfies $(\epsilon_1 + \epsilon_2)$-DP and by Lemma 3, it also satisfies $\frac{(\epsilon_1+\epsilon_2)^2}{2}$-zCDP. $\square$

3.5.2 Numerical Experiments

Datasets. Experiments are performed on three benchmark datasets[3]: Adult, US, and Brazil. Adult has 48,842 data samples and 41 features, and the label is to predict whether an annual income is more than \$50k or not. US has 40,000 records and 58 features, and the label is to predict whether the annual income of an individual is more than \$25k. BR has 38,000 samples and 53 features, and the goal is to predict whether the monthly income of an individual is more than \$300.

Data preprocessing. We consider the same preprocessing procedure as the method used in [43]. We first normalize each attribute so that the maximum attribute value is 1,

[3] `http://archive.ics.uci.edu/ml/datasets/Adult`, `http://international.ipums.org`

and normalize each sample so its L_2-norm at most 1. As for the label column, we also map it to $\{-1, 1\}$. In each simulation, we randomly sample 35,000 records for training and divide them into N parties, and thus each party includes $35000/N$ data samples (i.e., $|D_i| = 35000/N$). We denote the rest of the data records as testing data.

Baselines. We compare our proposed algorithms against four baseline algorithms: (i) DVP [55], is a dual variable perturbation method, where the dual variable of each agent at each ADMM iteration is perturbed by Gamma noise. (ii) M-ADMM [43], is a penalty perturbation approach, where each agent's penalty variable is perturbed by Gamma noise at each ADMM iteration. (iii) R-ADMM [44], is based on the penalty approach and the re-utilization of previous iteration's results to save the privacy loss. (iv) Non-private (decentralized ADMM without adding noise). Note that the privacy guarantees of DVP, M-ADMM, and R-ADMM hold only when the optimal solution of the perturbed subproblem is obtained in each iteration. In order to have a fair comparison, we adopt the Newton solver to obtain the optimal solution in each iteration. Notice that we also provide sharpened and tight privacy loss of above private ADMM algorithms under the privacy framework of zCDP.

Setup. We adopt logistic loss $\mathcal{L}(y_i^n x_i^T z_i^n) = \log(1+\exp(-y_i^n x_i^T z_i^n))$ as loss function, and the derivative $\mathcal{L}'(\cdot)$ is bounded with $|\mathcal{L}'(\cdot)| \leq 1$ and c_1-Lipschitz with $c_1 = 1/4$. We also let $\mathcal{R}(x_i) = \frac{1}{2}\|x_i\|_2^2$. We evaluate the accuracy by classification error rate over the testing set, defined as $Error\ rate = \frac{Number\ of\ incorrect\ predictions}{Total\ number\ of\ predictions\ made}$ and the convergence of algorithms by the average loss over the training samples, given by $\mathcal{L}_t = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|D_i|}\sum_{n=1}^{|D_i|}\mathcal{L}(y_i^n(x_i^t)^T z_i^n)$. We also report the mean and standard deviation of the average loss. The smaller the average loss, the higher accuracy.

Parameter settings. We consider a randomly generated undirectedly network with $N = 5$ agents and we fix the step size $\eta = 0.5$ and the total iteration number $T = 30$. We also consider the maximum number of primal variables that can be shared, $\mathbf{c} = 15$. Moreover, to maximize the utility of SVT, we follow the ratio between ϵ_1 and ϵ_2 in [33], i.e., $\epsilon_1 : \epsilon_2 = 1 : (2\mathbf{c})^{\frac{2}{3}}$. In all experiments, we set $\delta = 10^{-4}$, and $\epsilon = \{0.5, 1, 1.5, 2, 10\}$.

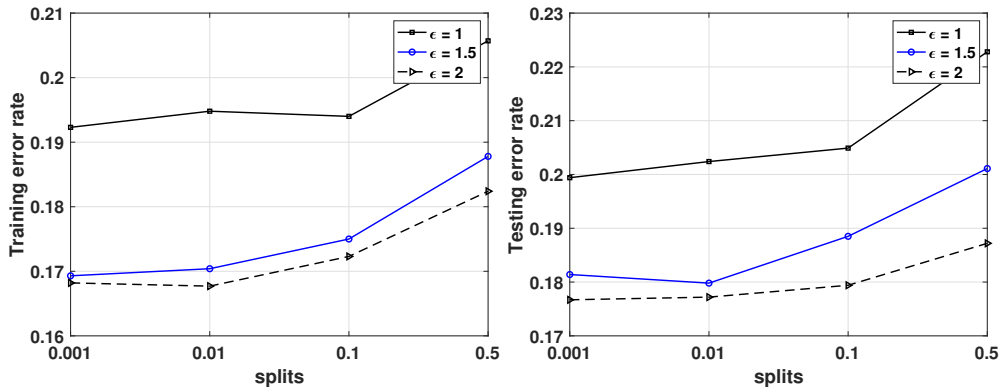


Figure 7: Effects of privacy budget splitting

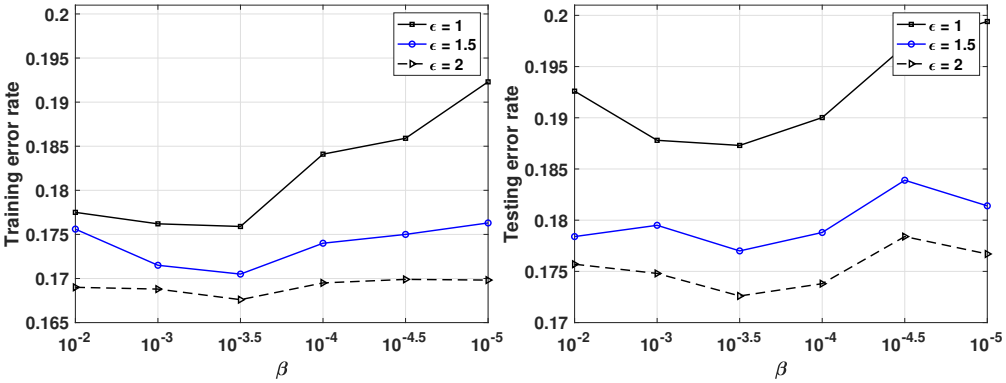


Figure 8: Effects of optimization accuracy β

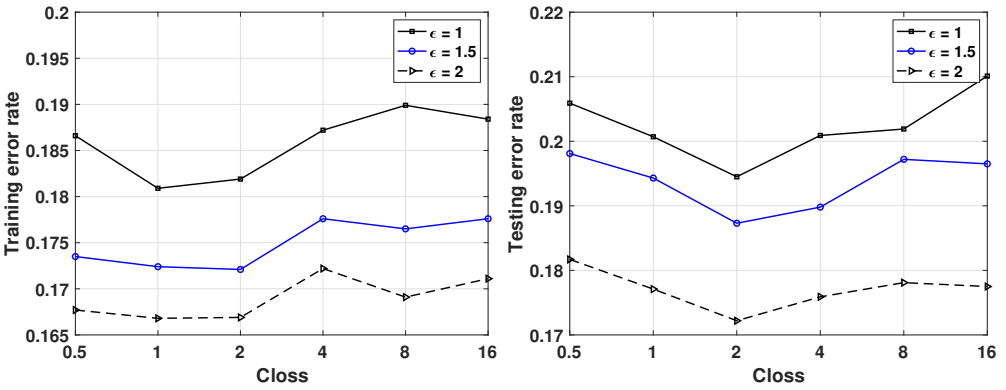


Figure 9: Effects of clipping threshold C_{loss}

Impacts of parameters. In this set of experiments on the Adult dataset, we present the effects of privacy budgets splitting and optimization accuracy (i.e., gradient norm threshold) β on the performance of PP-ADMM, and the loss clipping threshold C_{loss} and the quality function significance threshold α on the performance of IPP-ADMM. Specifically,

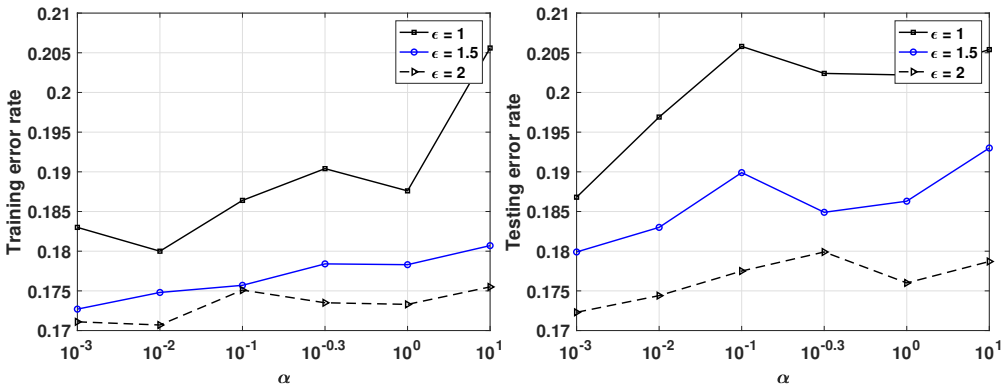


Figure 10: Effects of α

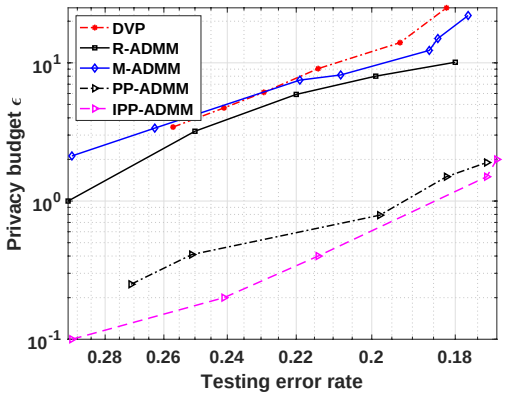


Figure 11: Trade-off between classification error rate and privacy on Adult dataset

we adjust different parameter settings separately, while keeping the rest constant to represent their impacts on training and testing accuracy.

For the privacy budgets splitting of PP-ADMM, we first convert the overall privacy budget parameters (ϵ, δ) to $\rho_{total} = \frac{\epsilon^2}{4\ln(1/\delta)}$. We set $\rho_{i1} = \frac{\rho_{total}}{T} \cdot (1 - splits)$ and $\rho_{i2} = \frac{\rho_{total}}{T} \cdot splits$, where $splits$ denotes the fraction of ρ_{total} allocated to ρ_{i2}. By tuning $splits$, we can find the good trade-off between the privacy budget for perturbing the objective and perturbing the approximate solution. In addition, we compute $\epsilon_{i1} = \rho_{i1} + 2\sqrt{\rho_{i1}\ln(1/\delta_{i1})}$ with $\delta_{i1} = 10^{-4}$, and set $\epsilon_{i3} = 0.99 \cdot \epsilon_{i1}$ to dedicate most of the budget to reduce the amount of noise for perturbing the objective and increase the influence of regularization. Figure 7 shows the effects of privacy budget splitting on the performance of PP-ADMM by setting $\beta = 10^{-6}$. As $splits$ decreases, i.e., allocating less privacy budgets for perturbing the approximate solution, it yields better training and testing accuracy. Thus, we set

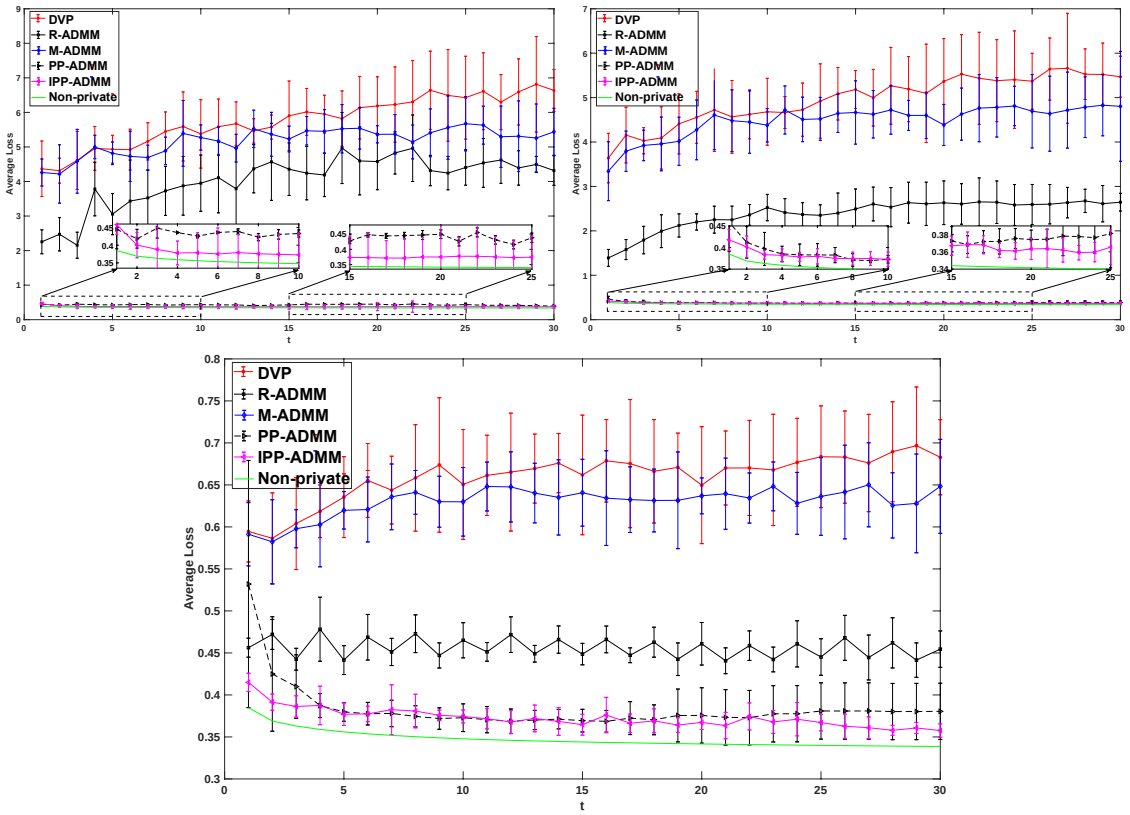
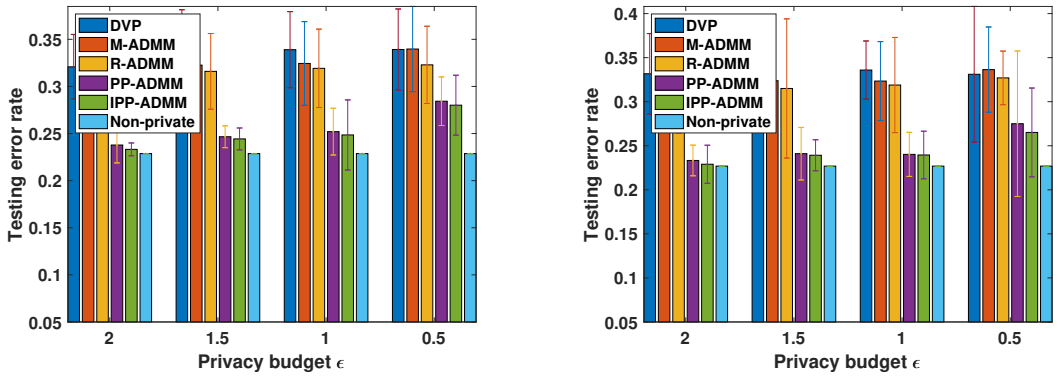


Figure 12: Convergence comparisons on Adult dataset (left: $\epsilon = 1$, middle: $\epsilon = 2$, right: $\epsilon = 10$)



(a) Brazil (b) US

Figure 13: Classification error rate comparisons on Brazil and US datasets.

$splits = 0.001$ to achieve a good trade-off between amount of noise added to the objective and approximate solution.

Figure 8 shows how classification accuracy changes with varying values of β and fixing

$splits = 0.001$. The parameter β controls the optimization accuracy of each iteration of PP-ADMM training process and the amount of noise for perturbing the approximate solution. As it can be observed from the figure, due to randomness of objective introduced by the random noise, when β is too small, solving the noisy objective perfectly in each iteration may not help the final performance. Conversely, when β is too large, large amount of noise is added to perturb the approximate solution, which also leads to performance degradation. In our experiments, we thus fix $\beta = 10^{-3.5}$ that achieves lowest training/testing error rate.

The IPP-ADMM algorithm has two threshold parameters, C_{loss} and α. These two parameters are used to bound the sensitivity of the quality function, and the value of quality function, respectively. If the clipping threshold C_{loss} is set to a small value, it significantly reduces the sensitivity but at the same time it leads much information loss in the estimation of quality function. On the other hand, if C_{loss} is large, the sensitivity becomes large that results in adding too much noise to the estimation. Thus, too large or small values of C_{loss} have a negative effect on employing SVT to check whether the current approximate solution has a big enough difference from that of previous iteration. As we can see from Figure 9, $C_{loss} = 2$ achieves a good trade-off between high information loss and large sensitivity. In Figure 10, we fix the the clipping threshold $C_{loss} = 2$ and vary α from 10^{-3} to 10 to see the effect of α on the performance. Although large value of α may potentially reduce the releasing of low quality approximate solution and reduce the communication cost, we observe that it also leads the learning performance degradation. We then choose $\alpha = 10^{-3}$ in our experiments, which achieves the lowest testing/training error rate.

Performance comparisons. We also present the trade-off between classification error rate and privacy cost in Figure 11, where we measured the privacy costs of all algorithms to obtain some specified testing error rates. Figure 11 illustrates that both of our methods have consistently lower privacy cost than those baselines algorithms. Compared with PP-ADMM, IPP-ADMM further saves more privacy cost due to limiting the number of releasing low-quality computational results. Additionally, we also inspect the convergence performance

(i.e., average loss) of different algorithms under the same budgets, as shown in Fig. 12. We can observe that when budget ϵ decreases from 10 to 1, the average loss values of baseline algorithms increase, which matches the simulation results shown in [43, 44, 17]. Although we also analyze the baseline algorithms using zCDP to provide tight privacy bound, using Gaussian noise instead of Gamma noise might be more beneficial to the performance, which usually has at least $\sqrt{d}$ times improvement of the empirical risk bound [58], where d is the dimension of training model. And our proposed algorithms continues to outperform the baseline algorithms significantly.

Figure 13 compares the accuracy (classification error rate) of different algorithms on Brazil and US. The noise parameter of all algorithms are chosen respectively so that they can achieve the same total privacy loss. As expected, the lower privacy budget, the higher classification error rate. As it was observed in the experiments, our proposed algorithms get close to the best achievable classification error rate for a wide range of total privacy loss considered in the experiments.

3.6 Omitted Proofs

Proof of Theorem 6

Proof. According to (21), we have $\tilde{x}_i^{t+1} = \underset{x_i}{\operatorname{argmin}} \ \mathcal{L}_{per}(x_i, D_i)$. First we will show that for some $o \in \mathbb{R}^d$, we have

$$\frac{\mathrm{pdf}_{D_i}(\tilde{x}_i^{t+1} = o)}{\mathrm{pdf}_{\hat{D}_i}(\tilde{x}_i^{t+1} = o)} \leq e^{\epsilon_{i1}} \ w.p. \ \geq 1 - \delta_{i1}.$$

According to the KKT optimally condition of equation (21), we have

$$B_{i1}^{t+1}(\tilde{x}_i^{t+1}, D_i) = -\nabla f_i(\tilde{x}_i^{t+1}) - 2\lambda_i^t - \eta \sum_{j \in \mathcal{V}_i} (2\tilde{x}_i^{t+1} - x_i^t - x_j^t).$$

Changing variables according to the function inverse theorem [59], we obtain

$$\frac{\mathrm{pdf}_{D_i}(\tilde{x}_i^{t+1}=o)}{\mathrm{pdf}_{\hat{D}_i}(\tilde{x}_i^{t+1}=o)}=\frac{\mathrm{pdf}(B_{i1}^{t+1}(o,D_i);\epsilon_{i1},\delta_{i1})}{\mathrm{pdf}(B_{i1}^{t+1}(o,\hat{D}_i);\epsilon_{i1},\delta_{i1})}\times\frac{|\det(\mathbf{J}(o\to B_{i1}^{t+1}(o,D_i)))|}{|\det(\mathbf{J}(o\to B_{i1}^{t+1}(o,\hat{D}_i)))|},$$

where $\mathbf{J}(o\to B_{i1}^{t+1}(o,D_i))$ is the Jacobian matrix of the mapping from o to $B_{i1}^{t+1}(o,D_i)$ using data D_i.

We will bound the ratios of the densities and the determinants separately.

First, we will show that for $\epsilon_{i3}\leq\epsilon_{i1}$, we have

$$\frac{\mathrm{pdf}(B_{i1}^{t+1}(o,D_i);\epsilon_{i1},\delta_{i1})}{\mathrm{pdf}(B_{i1}^{t+1}(o,\hat{D}_i);\epsilon_{i1},\delta_{i1})}\leq e^{\epsilon_{i3}}\ w.p.\ \geq 1-\delta_{i1},$$

and then we will show that

$$\frac{|\det(\mathbf{J}(o\to B_{i1}^{t+1}(o,D_i)))|}{|\det(\mathbf{J}(o\to B_{i1}^{t+1}(o,\hat{D}_i)))|}\leq e^{\epsilon_{i1}-\epsilon_{i3}}.$$

Consider the first part, without loss of generality, we fix a pair of adjacent datasets D_i and $\hat{D}_i$ and assume that only the first data point in D_i and $\hat{D}_i$ are different, i.e., (z_i^1,y_i^1) and $(\hat{x}_i^1,\hat{y}_i^1)$, respectively. We assume that each feature vector $\|z_i^n\|_2$ is normalized to $\|z_i^n\|_2\leq 1$ and $y_i^n\in\{+1,-1\}$. Then the L_2-sensitivity of $B_{i1}^{t+1}(o,D_i)$ is bounded by

$$\begin{aligned}\|B_{i1}^{t+1}(o,D_i)-B_{i1}^{t+1}(o,\hat{D}_i)\|_2&\leq\|\nabla f_i(o,D_i)-\|\nabla f_i(o,\hat{D}_i)\|_2\\&\leq\frac{1}{|D_i|}\|y_i^1\mathcal{L}'(y_i^1o^Tz_i^1)z_i^1-\hat{y}_i^1\mathcal{L}'(\hat{y}_i^1o^T\hat{x}_i^1)\hat{x}_i^1\|_2\\&\leq\frac{2}{|D_i|},\end{aligned}$$

where the last inequality follows as $|\mathcal{L}'(\cdot)|\leq 1$.

Setting $\sigma_{i1}=\frac{2\sqrt{2\ln(1.25/\delta_{i1})}}{|D_i|\epsilon_{i3}}$, we can get

$$\frac{\mathrm{pdf}(B_{i1}^{t+1}(o,D_i);\epsilon_{i1},\delta_{i1})}{\mathrm{pdf}(B_{i1}^{t+1}(o,\hat{D}_i);\epsilon_{i1},\delta_{i1})}\leq e^{\epsilon_{i3}}\ w.p.\ \geq 1-\delta_{i1}$$

from the guarantees of the Gaussian mechanism.

Consider the second part, the Jacobian matrix $\mathbf{J}(o \to B_{i1}^{t+1}(o, D_i))$ is given as

$$\mathbf{J}(o \to B_{i1}^{t+1}(o, D_i)) = -\frac{1}{|D_i|}\sum_{n=1}^{|D_i|}(y_i^n)^2\mathcal{L}''(y_i^n o^T z_i^n)z_i^n(z_i^n)^T - \frac{\hat{\lambda}}{N}\nabla^2\mathcal{R}(o) - 2\eta_i^{t+1}|\mathcal{V}_i|I_d.$$

Define $A(t+1) = \frac{1}{|D_i|}((\hat{y}_i^n)^2\mathcal{L}''(\hat{y}_i^1 o^T \hat{x}_i^1)\hat{x}_i^1(\hat{x}_i^1)^T - (y_i^n)^2\mathcal{L}''(y_i^1 o^T z_i^1)z_i^1(z_i^1)^T)$ and $E(t+1) = -\mathbf{J}(o \to B_{i1}^{t+1}(o, D_i))$. Then, we have

$$\begin{aligned}\frac{|\det(\mathbf{J}(o \to B_{i1}^{t+1}(o, D_i)))|}{|\det(\mathbf{J}(o \to B_{i1}^{t+1}(o, \hat{D}_i)))|} &= \frac{|\det(E(t+1))|}{|\det(E(t+1) + A(t+1))|} \\ &= \frac{1}{|\det(I + E(t+1)^{-1}A(t+1))|} \\ &= \frac{1}{|\prod_{j=1}^{r}(1 + \lambda_j(E(t+1)^{-1}A(t+1))|},\end{aligned}$$

where $\lambda_j(E(t+1)^{-1}A(t+1)$ represents the j-th largest eigenvalue of $E(t+1)^{-1}A(t+1)$ and $E(t+1)^{-1}A(t+1)$ has rank at most 2.

Also, since $0 < \mathcal{L}'' \leq c_1$ and the regularizer $\mathcal{R}$ is 1-strongly convex and twice differentiable, the eigenvalues of $E(t+1)$ and $A(t+1)$ satisfy

$$\lambda_j(E(t+1)) \geq \frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i| > 0$$

$$\text{and} \quad -\frac{c_1}{|D_i|} \leq \lambda_j(A(t+1)) \leq \frac{c_1}{|D_i|},$$

which implies

$$-\frac{c_1}{|D_i|(\frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i|)} \leq \lambda_j(E(t+1)^{-1}A(t+1))) \leq \frac{c_1}{|D_i|(\frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i|)}.$$

If we choose η such that $2c_1 < |D_i|(\frac{\hat{\lambda}}{N} + 2\eta|\mathcal{V}_i|)$, we have $-\frac{1}{2} \leq \lambda_j(E(t+1)^{-1}A(t+1))) \leq \frac{1}{2}$.

Since $\lambda_{min}(E(t+1)^{-1}A(t+1))) > -1$, there is

$$\frac{1}{|1+\lambda_{max}(E(t+1)^{-1}A(t+1)))|^2} \leq \frac{1}{|\det(I+E(t+1)^{-1}A(t+1))|} \leq \frac{1}{|1+\lambda_{min}(E(t+1)^{-1}A(t+1)))|^2}.$$

Therefore,

$$\frac{|\det(\mathbf{J}(o \to B_{i1}^{t+1}(o, D_i)))|}{|\det(\mathbf{J}(o \to B_{i1}^{t+1}(o, \hat{D}_i)))|} \leq \frac{1}{|1-\frac{c_1}{|D_i|(\frac{\hat{\lambda}}{N}+2\eta|\mathcal{V}_i|)}|^2} = e^{-2\ln(1-\frac{c_1}{|D_i|(\frac{\hat{\lambda}}{N}+2\eta|\mathcal{V}_i|)})} \leq e^{\frac{2.8c_1}{|D_i|(\frac{\hat{\lambda}}{N}+2\eta|\mathcal{V}_i|)}} \leq e^{\epsilon_{i1}-\epsilon_{i3}},$$

where in the second inequality, we use the fact that for any real number $x \in [0, 0.5]$, $-\ln(1-x) < 1.4x$. If choosing $\hat{\lambda} \geq \max_i(\frac{2.8Nc_1}{(\epsilon_{i1}-\epsilon_{i3})|D_i|} - 2\eta|\mathcal{V}_i|) \geq \max_i \frac{2.8Nc_1}{(\epsilon_{i1}-\epsilon_{i3})|D_i|}$, the last inequality follows.

Combine the first and second part, we can get that $\frac{\mathrm{pdf}_{D_i}(\tilde{x}_i^{t+1}=o)}{\mathrm{pdf}_{\hat{D}_i}(\tilde{x}_i^{t+1}=o)} \leq e^{\epsilon_{i1}}$ $w.p.$ $\geq 1-\delta_{i1}$. In other words, obtaining the exact minimizer $\tilde{x}_i^{t+1}$ of equation (21) provides $(\epsilon_{i1}, \delta_{i1})$-DP, and ρ_{i1}-zCDP with $\rho_{i1} = \epsilon_{i1}^2/(4\ln(1/\delta_{i1}))$.

Now, since we have $x_i^{t+1} = \tilde{x}_i^{t+1} + (\hat{x}_i^{t+1} - \tilde{x}_i^{t+1} + b_{i2})$, we will prove that releasing $(\hat{x}_i^{t+1} - \tilde{x}_i^{t+1} + b_{i2})$ is ρ_{i2}-zCDP. The L_2-sensitivity of $(\hat{x}_i^{t+1} - \tilde{x}_i^{t+1})$ is bounded by

$$\|\hat{x}_i^{t+1} - \tilde{x}_i^{t+1}\|_2 \leq \frac{1}{\frac{\hat{\lambda}}{N}+2\eta|\mathcal{V}_i|}\|\nabla\mathcal{L}_{per}(\hat{x}_i^{t+1}, D_i) - \nabla\mathcal{L}_{per}(\tilde{x}_i^{t+1}, D_i)\|_2 \leq \frac{\beta}{\frac{\hat{\lambda}}{N}+2\eta|\mathcal{V}_i|}.$$

According to Lemma 1 with $b_{i2} \sim \mathcal{N}(0, \sigma_{i2}^2 I_d)$ and $\sigma_{i2} = \frac{\beta}{\sqrt{2\rho_{i2}}(\frac{\hat{\lambda}}{N}+2\eta|\mathcal{V}_i|)}$, releasing $(\hat{x}_i^{t+1} - \tilde{x}_i^{t+1} + b_{i2})$ provides ρ_{i2}-zCDP.

Finally, by the composition of zCDP in Lemma 2, it follows that the privacy guarantee of outputting the perturbed approximated solution at $t+1$ iteration provides $(\rho_{i1}+\rho_{i2})$-zCDP. Considering T iterations, the total privacy loss for each agent i is bounded by $\rho_i = T(\rho_{i1}+\rho_{i2})$. □

Proof of Theorem 8

Proof. Let $\tilde{f}(x) = \mathbb{L}(x) + \frac{\hat{\lambda}}{2N}\|x\|_2^2$ and $\tilde{\mathbf{x}}_i = \underset{x}{\operatorname{argmin}} \quad \tilde{f}(x)$. Let $x_i^{opt} = \underset{x}{\operatorname{argmin}} \quad f_i(x, D_i)$ be agent i's classifier learned with its own dataset. Let $x_i^{appopt} = \mathcal{O}(f_i(x, D_i), \beta)$, which is agent i's approximate classifier trained with its own dataset. Then, using the analysis method in [60], we have

$$\mathbb{L}(x^*) = \mathbb{L}(x_{ref}) + \left(\tilde{f}(x^*) - \tilde{f}(x_i^{appopt})\right) + \left(\tilde{f}(x_i^{appopt}) - \tilde{f}(\tilde{x}_i)\right) + \left(\frac{\hat{\lambda}}{2N}\|x_{ref}\|_2^2 - \frac{\hat{\lambda}}{2N}\|x^*\|_2^2\right) + \left(\tilde{f}(\tilde{x}_i) - \tilde{f}(\tilde{x}_{ref})\right).$$

By [61], we have $\tilde{f}(x_i^{appopt}) - \tilde{f}(\tilde{x}_i) \leq (1+a)(f_i(x_i^{appopt}) - f_i(x_i^{opt})) + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|})$ holds $\forall a > 0$ with probability $1-\xi$, where $\mathcal{O}$ is the big-$\mathcal{O}$ notation. Moreover, we have

$$\begin{aligned} f_i(x_i^{appopt}) - f_i(x_i^{opt}) &\leq |f_i(x_i^{appopt}) - f_i(x_i^{opt})| \\ &\leq \|x_i^{appopt} - x_i^{opt}\|_2 \\ &\leq \frac{N}{\hat{\lambda}}\|\nabla f_i(x_i^{appopt}) - \nabla f_i(x_i^{opt})\|_2 \\ &\leq \frac{N\beta}{\hat{\lambda}}. \end{aligned}$$

Since $\tilde{\mathbf{x}}_i = \underset{x}{\operatorname{argmin}} \quad \tilde{f}(x)$, then $\tilde{f}(\tilde{x}_i) \leq \tilde{f}(\tilde{x}_{ref})$.

Moreover, we also assume the difference of expected loss under x_{ref} and x_i^{appopt} is bounded by $v > 0$, i.e., $\tilde{f}(x^*) - \tilde{f}(x_i^{appopt}) \leq \frac{\hat{\lambda}}{2N}(\|x^*\|_2^2 - \|x_i^{appopt}\|_2^2) + v$. Then, the following holds $\forall a > 0$ with probability $1-\xi$

$$\mathbb{L}(x^*) \leq \mathbb{L}(x_{ref}) + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|}) + v + \frac{(1+a)N\beta}{\hat{\lambda}} + \left(\frac{\hat{\lambda}}{2N}\|x_{ref}\|_2^2 - \frac{\hat{\lambda}}{2N}\|x_i^{appopt}\|_2^2\right).$$

We assume that v is quite small in comparison to other terms. If choosing $\hat{\lambda} \leq \frac{N(a_{acc} - \Delta_{i,non}^{t+1})}{\|x_{ref}\|_2^2}$ then, $\frac{\hat{\lambda}}{2N}\|x_{ref}\|_2^2 \leq \frac{a_{acc} - \Delta_{i,non}^{t+1}}{2}$.

Thus, $\mathbb{L}(x^*) \leq \mathbb{L}(x_{ref}) + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|}) + \frac{(1+a)N\beta}{\hat{\lambda}} + \frac{a_{acc} - \Delta_{i,non}^{t+1}}{2}$ holds with probability $1-\xi$.

If $|D_i|$ satisfies $\mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|}) + \frac{(1+a)N\beta}{\hat{\lambda}} \leq \frac{a_{acc}-\Delta_{i,non}^{t+1}}{2}$, we have

$$|D_i| \geq V \max_t \{ \frac{\log(1/\xi)}{\frac{(a_{acc}-\Delta_{i,non}^{t+1})^2}{2\|x_{ref}\|_2^2} - (1+a)\beta} \}$$

for some constant V. Therefore, the following holds with probability $1-\xi$

$$\mathbb{L}(x^*) \leq \mathbb{L}(x_{ref}) + a_{acc} - \Delta_{i,non}^{t+1}.$$

Since $\mathbb{L}(\hat{x}_{i,non}^{t+1}) \leq \mathbb{L}(x^*) + \Delta_{i,non}^{t+1}$, then $\mathbb{L}(\hat{x}_{i,non}^{t+1}) \leq \mathbb{L}(x_{ref}) + a_{acc}$ holds with probability $1-\xi$. □

Proof of Theorem 9

Proof. Let $\tilde{f}(x) = \mathbb{L}(x) + \frac{\hat{\lambda}}{2N}\|x\|_2^2$ and $\tilde{\mathbf{x}}_i = \underset{x}{\operatorname{argmin}} \quad \tilde{f}(x)$. Let $x_i^{opt} = \underset{x}{\operatorname{argmin}} \quad f_i(x, D_i)$ be agent i's classifier learned with its own dataset. Let $x_i^{appopt} = \mathcal{O}(f_i(x, D_i), \beta)$ which is agent i's approximate classifier trained with its own dataset. Let $x_{i,new}^{appopt} = \mathcal{O}(f_i^{new}(x, D_i), \beta)$ and $x_{i,new}^{privopt} = \underset{x}{\operatorname{argmin}} \quad f_i^{new}(x, D_i)$ and $x_{i,new} = x_{i,new}^{appopt} + b_{i2}$.

Using the analysis method in [60], we have

$$\begin{aligned}\mathbb{L}(x_{new}^*) = \mathbb{L}(x_{ref}) &+ \left(\tilde{f}(x_{new}^*) - \tilde{f}(x_{i,new})\right) + \left(\tilde{f}(x_{i,new}) - \tilde{f}(\tilde{x}_i)\right) \\ &+ \left(\frac{\hat{\lambda}}{2N}\|x_{ref}\|_2^2 - \frac{\hat{\lambda}}{2N}\|x_{new}^*\|_2^2\right) + \left(\tilde{f}(\tilde{x}_i) - \tilde{f}(\tilde{x}_{ref})\right).\end{aligned}$$

Here, x_{new}^* is the centralized classifier trained with all data samples, and $x_{i,new}$ is the private classifier trained with data from agent i. We also assume that the difference of expected loss between x_{new}^* and $x_{i,new}$ is bounded, i.e., $\tilde{f}(x_{new}^*) - \tilde{f}(x_{i,new}) \leq \frac{\hat{\lambda}}{2N}(\|x_{new}^*\|_2^2 - \|x_{i,new}\|_2^2) + v$.

By [61], $\tilde{f}(x_{i,new}) - \tilde{f}(\tilde{x}_i) \leq (1+a)\left(f_i(x_{i,new}) - f_i(x_i^{opt})\right) + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|})$ holds $\forall a > 0$ with probability $1-\xi$, where $\mathcal{O}$ is the big-$\mathcal{O}$ notation. Then, we have

$$f_i(x_{i,new}) - f_i(x_i^{opt}) = \left(f_i(x_{i,new}) - f_i(x_{i,new}^{privopt})\right) + \left(f_i(x_{i,new}^{privopt}) - f_i(x_i^{opt})\right).$$

We first bound $f_i(x_{i,new}) - f_i(x_{i,new}^{privopt})$. We have w.p. $\geq 1-\xi$

$$\begin{aligned} f_i(x_{i,new}) - f_i(x_{i,new}^{privopt}) &\leq |f_i(x_{i,new}) - f_i(x_{i,new}^{privopt})| \\ &\leq \|x_{i,new} - x_{i,new}^{privopt}\|_2 \\ &= \|x_{i,new}^{appopt} - x_{i,new}^{privopt} + b_{i2}\|_2 \\ &\leq \|x_{i,new}^{appopt} - x_{i,new}^{privopt}\|_2 + \|b_{i2}\|_2 \\ &\leq \frac{N\beta}{\hat{\lambda}} + \sigma_{i2}\sqrt{2d\log\frac{1}{\xi}}. \end{aligned}$$

Next, we bound $f_i(x_{i,new}^{privopt}) - f_i(x_i^{opt})$. We have w.p. $1-\xi$

$$\begin{aligned} f_i(x_{i,new}^{privopt}) - f_i(x_i^{opt}) &\leq {b_{i1}}^T(x_i^{opt} - x_{i,new}^{privopt}) \\ &\leq \frac{N\|b_{i1}\|_2^2}{\hat{\lambda}} \leq \frac{2N\sigma_{i1}^2 d\log\frac{1}{\xi}}{\hat{\lambda}}, \end{aligned}$$

where the second inequality follows the $\frac{N}{\hat{\lambda}}$-strongly convex properties of f_i^{new} and f_i, and the last inequality follows Lemma 14. Thus, $f_i(x_{i,new}) - f_i(x_i^{opt}) \leq \frac{N\beta}{\hat{\lambda}} + \sigma_{i2}\sqrt{2d\log\frac{1}{\xi}} + \frac{2N\sigma_{i1}^2 d\log\frac{1}{\xi}}{\hat{\lambda}}$ holds w.p. $1-2\xi$. Therefore, $\tilde{f}(x_{i,new}) - \tilde{f}(\tilde{x}_i) \leq (1+a)(\frac{N\beta}{\hat{\lambda}} + \sigma_{i2}\sqrt{2d\log\frac{1}{\xi}} + \frac{2N\sigma_{i1}^2 d\log\frac{1}{\xi}}{\hat{\lambda}}) + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|})$ holds $\forall a > 0$ with probability $1-3\xi$.

Since $\tilde{\mathbf{x}}_i = \underset{x}{\operatorname{argmin}} \quad \tilde{f}(x)$, then $\tilde{f}(\tilde{x}_i) \leq \tilde{f}(\tilde{x}_{ref})$. We then have $\forall a > 0$ w.p. $1-3\xi$

$$\begin{aligned} \mathbb{L}(x_{new}^*) \leq \mathbb{L}(x_{ref}) + v + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|}) + (1+a)(\frac{N\beta}{\hat{\lambda}} + \sigma_{i2}\sqrt{2d\log\frac{1}{\xi}} + \frac{2N\sigma_{i1}^2 d\log\frac{1}{\xi}}{\hat{\lambda}}) \\ + \left(\frac{\hat{\lambda}}{2N}\|x_{ref}\|_2^2 - \frac{\hat{\lambda}}{2N}\|x_{i,new}\|_2^2\right). \end{aligned}$$

We assume that v is quite small in comparison to other terms. If choosing $\hat{\lambda} \leq \frac{N(a_{acc} - \Delta_{i,new}^{t+1})}{\|x_{ref}\|_2^2}$ then, $\frac{\hat{\lambda}}{2N}\|x_{ref}\|_2^2 \leq \frac{a_{acc} - \Delta_{i,new}^{t+1}}{2}$.

Thus, we have

$$\mathbb{L}(x^*_{new}) \leq \mathbb{L}(x_{ref}) + \mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|}) + (1+a)(\frac{N\beta}{\hat{\lambda}} + \sigma_{i2}\sqrt{2d\log\frac{1}{\xi}} + \frac{2N\sigma_{i1}^2 d\log\frac{1}{\xi}}{\hat{\lambda}}) + \frac{a_{acc} - \Delta^{t+1}_{i,new}}{2}$$

holds w.p. $1-3\xi$.

If $|D_i|$ satisfies $\mathcal{O}(\frac{N\log(1/\xi)}{\hat{\lambda}|D_i|}) + (1+a)(\frac{N\beta}{\hat{\lambda}} + \sigma_{i2}\sqrt{2d\log\frac{1}{\xi}} + \frac{2N\sigma_{i1}^2 d\log\frac{1}{\xi}}{\hat{\lambda}}) \leq \frac{a_{acc}-\Delta^{t+1}_{i,new}}{2}$, i.e., for some constant V, we have

$$|D_i| \geq V \max_t \{\frac{\log(1/\xi)}{\frac{(a_{acc}-\Delta^{t+1}_{i,new})^2}{2\|x_{ref}\|_2^2} - (1+a)(\beta+\mathcal{H})}\},$$

where $\mathcal{H} = \frac{\sigma_{i2}(a_{acc}-\Delta^{t+1}_{i,new})\sqrt{2d\log\frac{1}{\xi}}}{\|x_{ref}\|_2^2} + 2\sigma_{i1}^2 d\log\frac{1}{\xi}$.

Therefore, the followsing holds with probability $1-3\xi$

$$\mathbb{L}(x^*_{new}) \leq \mathbb{L}(x_{ref}) + a_{acc} - \Delta^{t+1}_{i,new}.$$

Since $\mathbb{L}(x_i^{t+1}) \leq \mathbb{L}(x^*_{new}) + \Delta^{t+1}_{i,new}$, then $\mathbb{L}(x_i^{t+1}) \leq \mathbb{L}(x_{ref}) + a_{acc}$ holds with probability $1-3\xi$. □

Lemma 14. *[62] Let X be a random variable drawn from distribution $\mathcal{N}(0, I_d)$. Then we have w.p. $\geq 1-\xi$, $\|X\|_2 \leq \sqrt{2d\log\frac{1}{\xi}}$.*

4 Differentially Private and Communication Efficient Decentralized Gradient Descent

4.1 Introduction

Machine learning is increasingly deployed into large-scale distributive systems that can improve the quality of our life, such as smart home security [1], and AI-aided medical diagnosis [2]. With the proliferation of mobile phone devices, a vast amount of data has been generated at an ever-increasing rate, which leads to significant computational complexity for data collection and processing via a centralized machine learning approach. Therefore, collaborative training of a machine learning model among edge computing devices is beneficial and essential in dealing with large scale decentralized learning tasks [3, 4, 5]. However, since the dimension of learning model increases (which is the current trend in large-scale distributed machine learning), model exchanges among agents become the significant communication bottleneck. Moreover, the computation speed and computational load of local agents vary greatly, which can substantially slow down the overall system efficiency.

While communication is a key concern in collaborative machine learning, an equally important consideration is the critical privacy leakage of sensitive training data during the training process [11, 63]. Fortunately, differential privacy [25] has been exploited as a well-defined framework for providing privacy protection in machine learning, which guarantees that the adversary with arbitrary background knowledge cannot extract any sensitive information about the training data. Many existing mechanisms have been proposed to ensure DP, like gradient perturbation [64, 53] and output perturbation approaches [56, 65, 17]. However, directly hammering those centralized mechanisms into distributed settings will potentially introduce a heavy communication burden.

A majority of the existing research focuses on either communication efficiency [66, 67, 68] or data privacy [69, 17, 70]. However, only a limited amount of works consider both [63, 71, 72]. Agarwal et al. [63] proposed the cpSGD algorithm based on the randomized quantization and Binomial mechanism. However, the method was specialized for the

distributed mean estimation problem under the server/worker architecture. The performance of collaborative learning is not clear in general network topologies. Furthermore, in [71], Zhang et al. adopted a sparsification operator to compress the differentially private local differentials before transmitting to neighboring agents to reduce the communication cost while guaranteeing privacy. However, the above works ignore the critical impact of the straggling agents, which may significantly slow down the wall-clock time of the convergence.

In this work, we propose two differentially private and communication efficient algorithms, named `Q-DPSGD-1` and `Q-DPSGD-2`, by considering different orders between the random quantization and DP mechanism. Particularly, in `Q-DPSGD-1`, *a Gaussian mechanism is applied before random quantization*, and the privacy guarantee of quantized model roots from the post-processing property of DP. In `Q-DPSGD-2`, we consider an alternative design in reverse order, i.e., *by applying a Gaussian mechanism after random quantization.* Due to the discretization of the quantizatized local model parameters, we propose to sample Gaussian noises from a discretization of Gaussian distribution and add the discrete Gaussian noise to the quantization values without sacrificing the communication efficiency. We provide the privacy analysis of discrete Gaussian mechanism under the Rényi DP (RDP) instead of Concentrated DP, i.e., CDP [73]. The reason is that CDP does not support privacy amplification from subsampling and analytical moments accountant [35], both of which may broaden the practical applications of discrete Gaussian mechanisms. Moreover, a deadline based scheme for local computations is leveraged in both algorithms to address the straggler problems and reduce the elapsed time of convergence. We also provide convergence results of both algorithms for convex and non-convex loss functions. Our salient contributions are summarized as follows.

- We propose a `Q-DPSGD-1` method which will update the local models by integrating DP noise and random quantization operator to simultaneously enforce DP and communication efficiency. Especially, different from the fixed (mini-batch) gradient computation approaches, we utilize a deadline based approach [74] to effectively integrate DP and random quantization for collaborative learning, where no privacy budget

will be consumed if there is no gradient computation before the deadline. We prove the convergence results under convex and non-convex cases, and analyze the trade-off between privacy and accuracy in terms of expected population risk.

- To exploit the capability of perturbing quantized local model by DP noise in collaborative learning, we propose a `Q-DPSGD-2` method that employs discrete Gaussian mechanism after random quantization, instead of using Binomial mechanism [63]. We analyze privacy guarantee of discrete Gaussian mechanism under the RDP that breaks its limited application under CDP. The convergence results of `Q-DPSGD-2` are also provided for both convex and non-convex objectives.

- Through extensive experiments on the CIFAR-10 and MNIST datasets, we show the superior performance of the proposed algorithms over the baseline algorithms, and the experimental results validate our theoretical analysis.

4.2 Related Work

Decentralized consensus optimization has been studied extensively. The most popular first-order choices for the convex setting are distributed gradient descent-type methods [75, 76], distributed variants of the alternating direction method of multipliers (ADMM) [49], and dual averaging [77]. Recently, there have been some works which study non-convex decentralized consensus optimization and establish convergence to a stationary point [78, 79]. There are two categories of communication-efficiency of distributed optimization. One way to improve communication-efficiency of distributed optimization procedures is by communicating compressed local gradients or models to parameter server via quantization [80, 81, 82] and sparsification [83, 84]. Another line is to reduce the number of communication rounds by techniques such as periodic averaging that pay more local computation for less communication [85]. However, most of the above communication-efficient schemes ignore the privacy aspect.

To prevent privacy leakage in distributed machine learning, many related works focus on secure multi-party computation or homomorphic encryption, which involve both high

computation and communication overhead, and cannot prevent the information leakage from the final learned model. Thus, many works [21, 17, 18, 69] have studied how to effectively integrate distributed learning algorithms (ADMM, gradient descent) with DP. However, most of them ignore the communication efficiency aspect.

4.3 Problem Setting and Preliminaries

In this work, we aim to solve the population risk problem as

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} F(\boldsymbol{x}) = \mathbb{E}_{\theta\sim\mathcal{P}}\, l(\boldsymbol{x},\theta), \tag{22}$$

where $\ell : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ is a stochastic loss function, $\theta \in \mathbb{R}^q$ is a data sample drawn from an unknown probability distribution $\mathcal{P}$. Instead of directly solving (22), we consider minimizing the following Empirical Risk Minimization (ERM) problem as

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} F_N(\boldsymbol{x}, D) = \frac{1}{mn}\sum_{\theta\in D} l(\boldsymbol{x},\theta), \tag{23}$$

where $D = \{\theta_1, \cdots, \theta_{mn}\}$ is the overall data samples.

In collaborative training, our goal is to collaboratively solve problem (23) to train a common classifier $\boldsymbol{x} \in \mathbb{R}^p$ in a decentralized manner (i.e., no centralized controller) while keeping the privacy for each data sample. Thus, we consider a wireless edge network containing n agents with a node set $\mathscr{N} = \{1, \cdots, n\}$, and each agent i has a dataset $D_i = \{\theta_i^1, \cdots, \theta_i^m\}$. The communication among agents can be represented by an undirected connected graph $G = \{\mathscr{N}, \mathscr{E}\}$, where $\mathscr{E} \subseteq \mathscr{N} \times \mathscr{N}$ denotes the set of communication links between agents. Note that two agents i and j can communicate with each other only when they are neighbors, i.e., $(i, j) \in \mathscr{E}$. We denote the set of neighbors of agent i as $\mathcal{N}_i$. Thus, the collaborative ERM problem can be formulated as

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} f(\boldsymbol{x}, D) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}, D_i), \tag{24}$$

where $D = D_1 \cup \cdots \cup D_n$ is the union of all local datasets, and $f_i(\boldsymbol{x}, D_i) = \frac{1}{m} \sum_{\theta \in D_i} \ell(\boldsymbol{x}; \theta)$, which is only observable to agent i. In order to collaboratively solve problem (24) in a decentralized manner, we then rewrite it as a consensus optimization problem as follows,

$$\begin{aligned} \min_{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n} \hat{F}(x) &= \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{x}_i, D_i) \\ \text{s.t.} \quad & \boldsymbol{x}_i = \boldsymbol{x}_j, \ \forall i, j \in \mathcal{N}_i, \end{aligned} \tag{25}$$

where the vector $x = [\boldsymbol{x}_1; \cdots; \boldsymbol{x}_n] \in \mathbb{R}^{np}$ denotes the concatenation of all the local models $\boldsymbol{x}_i$ at agent i, and the constraint here is to enforce all the local classifiers reach consensus, i.e., $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \cdots = \boldsymbol{x}_n$. Thus, (24) and (25) are equivalent, i.e., the optimal solution $\{\boldsymbol{x}_i^*\}_{i=1}^n$ of problem (24) holds that $\boldsymbol{x}^* = \boldsymbol{x}_1^* = \boldsymbol{x}_2^* = \cdots = \boldsymbol{x}_n^*$.

To solve Problem (25) in a decentralized manner, each agent i can minimize the local objective function $f_i(\boldsymbol{x}, D_i)$ over its own private dataset D_i, and exchange local model $\boldsymbol{x}_i$ among its neighboring agents $j \in \mathcal{N}_i$ to enforce $\boldsymbol{x}_i$ close enough to the local model $\boldsymbol{x}_j$ of its neighbors j. Although there is no need to share the local private dataset during this iterative process, local model exchange between the distributed agents imposes the risk of information leakage. For example, the adversary may perform model inversion attack [11, 86] and membership inference attack [10] together with some background knowledge to infer sensitive information in the dataset. Furthermore, model exchange also brings a potentially heavy communication burden, and this problem becomes worse when performing on edge devices due to the receiver sensitivity and transmitter power constraints, etc. Therefore, in this work, our objective is to achieve communication efficient collaborative learning while preserving DP guarantee at the same time.

We then assume that the weight matrix, the quantizer, and local objective functions satisfy the assumptions, which are commonly used in related works [80, 81]. We use $\lceil x \rceil$ to denote the least integer greater than or equal to x, and $\|\cdot\|$ to denote the l_2-norm of a vector.

Assumption 1. *The weight matrix $W \in \mathbb{R}^{n \times n}$ with entries $w_{ij} \geq 0$ satisfies the following*

conditions: $W = W^\top$, $W\mathbf{1} = \mathbf{1}$ *and* $null(I - W) = span(\mathbf{1})$.

Assumption 2. *The random quantizer* $Q(\cdot)$ *is unbiased and variance-bounded, i.e.,* $\mathbb{E}[Q(\boldsymbol{x})|\boldsymbol{x}] = \boldsymbol{x}$ *and* $\mathbb{E}[\|Q(\boldsymbol{x}) - \boldsymbol{x}\|^2|\boldsymbol{x}] \leq \widetilde{\sigma}^2$, *for any* $\boldsymbol{x} \in \mathbb{R}^p$*; and quantizations are carried out independently.*

Assumption 3. *The local loss function* ℓ *is* $\hat{K}$*-smooth and* K*-Lipschitz continuous with respect to* $\boldsymbol{x}$*, i.e., for any* $\boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathbb{R}^p$ *and any* $\theta \in \mathcal{D}$, $\|\nabla\ell(\boldsymbol{x}, \theta) - \nabla\ell(\hat{\boldsymbol{x}}, \theta)\| \leq \hat{K}\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$, *and* $\|\ell(\boldsymbol{x}, \theta) - \ell(\hat{\boldsymbol{x}}, \theta)\| \leq K\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|$.

Assumption 4. *Stochastic gradients* $\nabla\ell(\boldsymbol{x}, \theta)$ *are unbiased and variance bounded, i.e.,* $\mathbb{E}_{\theta\sim\mathcal{P}}[\nabla\ell(\boldsymbol{x}, \theta)] = \nabla F(\boldsymbol{x})$ *and* $\mathbb{E}_{\theta\sim\mathcal{P}}\left[\|\nabla\ell(\boldsymbol{x}, \theta) - \nabla F(\boldsymbol{x})\|^2\right] \leq \gamma^2$.

Assumption 5. *The function* ℓ *is* μ*-strongly convex, i.e., for any* $\boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathbb{R}^p$ *and* $\theta \in \mathcal{D}$ *we have that* $\langle\nabla\ell(\boldsymbol{x}, \theta) - \nabla\ell(\hat{\boldsymbol{x}}, \theta), \boldsymbol{x} - \hat{\boldsymbol{x}}\rangle \geq \mu\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$.

The condition in Assumptions 3 and 5 imply that the objective function f is strongly convex and

the local gradients of each node $\nabla f_i(\boldsymbol{x})$ are also unbiased estimators of the expected risk gradient $\nabla L(\boldsymbol{x})$ and their variance is bounded above by γ^2/m as it is defined as an average over m realizations.

4.4 Main Methods

4.4.1 `Q-DPSGD-1`

In this section, we introduce `Q-DPSGD-1` algorithm that takes into account privacy-preservation and communication efficiency in collaborative learning. To ensure DP guarantee, each agent utilizes Gaussian mechanism to perturb the gradients of model update and then performs noisy SGD to update the local model before sharing to neighboring agents. To reduce the communication overhead, we consider that each agent only exchanges a randomly quantized version of its local model to its neighbors. Exchanging quantized local model instead of the original model indeed improves the communication efficiency at the

Algorithm 4 `Q-DPSGD-1` run by agent i

Input: Weights $\{w_{ij}\}_{j=1}^{n}$; Deadline T_d.
Set initial variables $\boldsymbol{x}_{i,0} = 0$ and $\boldsymbol{z}_{i,0} = Q(\boldsymbol{x}_{i,0})$.
for $t = 0, \cdots, T-1$ **do**
    Broadcast $\boldsymbol{z}_{i,t} = Q(\boldsymbol{x}_{i,t})$ to all neighbors $j \in \mathcal{N}_i$.
    Receive $\boldsymbol{z}_{j,t}$ from its neighbor $j \in \mathcal{N}_i$.
    Take and evaluate stochastic gradients $\{\nabla \ell(\boldsymbol{x}_{i,t}; \theta) : \theta \in \mathcal{S}_{i,t}\}$ till reaching the deadline T_d, with $\mathcal{S}_{i,t} \subseteq \{1, \cdots, m\}$.
    Generate gradient:
    $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) = \frac{1}{|\mathcal{S}_{i,t}|} \sum_{\theta \in \mathcal{S}_{i,t}} \nabla \ell(\boldsymbol{x}_{i,t}; \theta)$
    Update $\boldsymbol{x}_{i,t+1} = (1 - \varepsilon + \varepsilon w_{ii}) \boldsymbol{x}_{i,t} + \varepsilon \sum_{j \in \mathcal{N}_i} w_{ij} \boldsymbol{z}_{j,t} - \alpha\varepsilon(\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) + \zeta_{i,t})$, where $\zeta_{i,t} \sim \mathcal{N}(0, \sigma^2 K^2 I_p)$).
end for

cost of injecting quantization noise to the information received by the agents in the network. However, using quantized model and Gaussian mechanism induces extra noise in the training process which makes the analysis of our algorithm more challenging.

The details of `Q-DPSGD-1` algorithm are given in Algorithm 4. At each iteration t, consider $\boldsymbol{x}_{i,t}$ as the local classifier, each agent i sends $\boldsymbol{z}_{i,t} = Q(\boldsymbol{x}_{i,t})$, the quantized version of the vector $\boldsymbol{x}_{i,t}$, to all neighbors $j \in \mathcal{N}_i$ to reduce the communication burden on the shared bus. For instance, we consider the precision quantizer decribed by quantization resolution η and s bits with the representation range $\{-\eta \cdot 2^{s-1}, \cdots, \eta \cdot (2^s - 1)\}$. Then the quantization function $Q(x)$ can be expressed as

$$Q(x) = \begin{cases} k\eta & \text{w.p. } 1 - (x - k\eta)/\eta, \\ (k+1)\eta & \text{w.p. } (x - k\eta)/\eta, \end{cases} \tag{26}$$

where $x \in [k\eta, (k+1)\eta]$. Note that the above quantizer satisfied Assumption 2 [80].

Note that `Q-DPSGD-1` is different from the fixed (mini-batch) gradient computation in previous works [80, 71, 63], where each agent i selects a subset of local data samples to estimate the stochastic gradient. Motivated by [74, 81], `Q-DPSGD-1` considers a deadline based approach by setting a deadline T_d to limit the time that each agent can perform stochastic gradient estimation. Further, this deadline based approach can also avoid waiting for the slowest agent to finish its local model update, i.e., straggler's delay problem. Thus,

at iteration t, each agent is given a deadline time T_d to compute its per sample gradient $\nabla\ell(\boldsymbol{x}_{i,t};\theta)$. At the end of the deadline, each agent computes its local mini-batch gradient $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) = \frac{1}{|\mathcal{S}_{i,t}|}\sum_{\theta\in\mathcal{S}_{i,t}} \nabla\ell(\boldsymbol{x}_{i,t};\theta)$, where we treat the set of collected samples as $\mathcal{S}_{i,t}$. Note that we set $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) = 0$ when there are not any gradient estimates by deadline T_d, i.e., $|\mathcal{S}_{i,t}| = 0$.

In order to enforce DP guarantee, each agent i adds a noise $\zeta_{i,t}$ drawn from a Gaussian distribution $\mathcal{N}(0, \sigma^2K^2I_p))$ to perturb the local stochastic gradient $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})$. After that, the perturbed local stochastic gradient, its local variables $\boldsymbol{x}_{i,t}$ and the local variables received from its neighbors $\{\boldsymbol{z}_{j,t} = Q(\boldsymbol{x}_{j,t}); j \in \mathcal{N}_i\}$ are used to update its local model $\boldsymbol{x}_{i,t+1}$. Note that we denote the communication matrix w_{ij} as the weight that agent i assigns to the information that it receives from agent j. If agents i and j are not neighbors, $w_{ij} = 0$. In particular, at iteration t, agent i updates $\boldsymbol{x}_{i,t+1}$ according to the update

$$\boldsymbol{x}_{i,t+1} = (1 - \varepsilon + \varepsilon w_{ii})\boldsymbol{x}_{i,t} + \varepsilon \sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{z}_{j,t} - \alpha\varepsilon(\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) + \zeta_{i,t}), \tag{27}$$

where $\zeta_{i,t} \sim \mathcal{N}(0, \sigma^2K^2I_p))$ and α and ε are positive constants. The parameter α behaves as the step size of the gradient descent step regarding to the local objective function f_i and ε acts as an averaging parameter between performing the distributed gradient update $\varepsilon(w_{ii}\boldsymbol{x}_{i,t} + \sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{z}_{j,t} - \alpha(\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) + \zeta_{i,t})$ versus using the previous decision variable $(1-\varepsilon)\boldsymbol{x}_{i,t}$.

Privacy guarantee The following theorem provides the privacy guarantee of `Q-DPSGD-1` algorithm.

Theorem 11. *The `Q-DPSGD-1` algorithm satisfies (ϵ, δ)-DP with $\epsilon = \epsilon(\rho) + \frac{\log(1/\delta)}{\rho-1}$ and $\epsilon(\rho) = \max_i \sum_{t=0}^{T-1} \epsilon'_{i,t}(\rho)$ with $\epsilon'_{i,t}(\rho) = \frac{8\rho}{m^2\sigma^2}$ if $|\mathcal{S}_{i,t}| \neq 0$, and $\rho = 2\log(1/\delta)/\epsilon + 1$.*

Remark 4. *Since we adopt a deadline based scheme in `Q-DPSGD-1` algorithm instead of the fixed mini-batch scheme used in [71, 53], the size of mini-batch $\mathcal{S}_{i,t}$, i.e., $|\mathcal{S}_{i,t}|$ is not deterministic but a random variable. We then need to carefully state our computation model used for the processing time of agents in the communication network. Following the similar*

approach in [74, 81], we denote the processing speed of each machine as the number of per-example gradient $\nabla\ell(\boldsymbol{x}_{i,t};\theta)$ that it computes per second. We also assume that the processing speed of each machine i at iteration t is a random variable $V_{i,t}$, and $V_{i,t}$'s are i.i.d with probability distribution $F_V(v)$. We further assume that the domain of the random variable V is bounded and its realizations are in $[\underline{v},\overline{v}]$. If $V_{i,t}$ is the number of stochastic gradient which can be computed per second, the size of mini-batch $\mathcal{S}_{i,t}$ is given by $|\mathcal{S}_{i,t}| = V_{i,t}T_d$. Therefore, the privacy budget ϵ in Theorem 11 is also a random variable and provides a good manner to characterize the privacy consumption of decentralized learning under the straggler's delay problem. For instance, when $\mathcal{S}_{i,t} \subseteq \emptyset$, i.e., there is no gradient computation by deadline T_d, agent i then updates $\boldsymbol{x}_{i,t+1}$ by $\boldsymbol{x}_{i,t+1} = (1-\varepsilon+\varepsilon w_{ii})\boldsymbol{x}_{i,t} + \varepsilon\sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{z}_{j,t}$ and broadcasts $\boldsymbol{z}_{i,t+1} = Q(\boldsymbol{x}_{i,t+1})$ without spending any privacy budget while preventing stragglers holding up the entire network.

Convergence analysis We characterize the convergence of `Q-DPSGD-1` algorithm for strongly convex and non-convex objectives, respectively.

Theorem 12 (Strongly Convex)**.** *If the conditions in Assumptions 1–5 are satisfied and step-sizes are picked as $\varepsilon = T^{-3\widetilde{\boldsymbol{\delta}}/2}$, $\alpha = 2T^{-\widetilde{\boldsymbol{\delta}}/2}$, for any $\widetilde{\boldsymbol{\delta}} \in (0, 1/2)$, then for large enough number of iterations $T \geq T^{\mathsf{c}}_{\mathsf{min}}$, the iterates generated by the `Q-DPSGD-1` algorithm satisfy*

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_{i,T}-\boldsymbol{x}^*\right\|^2 \leq \mathcal{O}\left(\frac{E^2(\hat{K}/\mu)^2}{(1-\beta)^2}+\frac{\widetilde{\sigma}^2}{\mu}\right)\frac{1}{T^{\widetilde{\boldsymbol{\delta}}}} + \mathcal{O}\left(\frac{\gamma^2}{\mu}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}+\frac{pK^2\sigma^2}{\mu}\right)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}},$$

where $E^2 = 2K\sum_{i=1}^{n}(f_i(0)-f_i^*)$, and $f_i^* = \min_{\boldsymbol{x}\in\mathbb{R}^p} f_i(\boldsymbol{x})$ and $\boldsymbol{x}^*$ is the solution of Problem (25).

Remark 5. *Theorem 12 shows that the exact convergence of each local model to the global optimal can be achieved with the sublinear convergence rate which is $\mathcal{O}(1/\sqrt{T})$ by setting $\widetilde{\boldsymbol{\delta}}$ close to $1/2$. Furthermore, the above results also show the effect of stochastic gradients variance γ^2, the Gaussian noise σ^2 used to provide privacy guarantee, as well as the deadline*

based scheme parameters $\mathbb{E}[1/V]/T_d$ *that describes the inverse of the batch size computed before the deadline* T_d*. Moreover, the coefficient of* $1/T^{\widetilde{\delta}}$ *describes the effects of objective function condition number* K/μ*, variance* $\widetilde{\sigma}^2$ *introduced by random quantization, and the graph connectivity parameter* $1/(1-\beta)$*. Notice that the error term introduced by DP decays faster than the one introduced by random quantization.*

Remark 6. *Utilizing the strong convexity of objective function, if we choose* $|\mathcal{S}_{i,t}| = B$ *and* $\sigma^2 = \frac{16T(2\log(1/\delta)/\epsilon+1)}{m^2\epsilon}$ *and* $T = \mathcal{O}(\frac{m^4\epsilon^2\mu^2}{(\log(1/\delta)/\epsilon+1)^2p^2K^4})$*,* `Q-DPSGD-1` *is* (ϵ,δ)*-DP and the empirical risk* $F_N(\boldsymbol{x}_{i,T}) - F_N(\boldsymbol{x}^*) = f(\boldsymbol{x}_{i,T}) - f(\boldsymbol{x}^*) \leq \mathcal{O}(\frac{(2\log(1/\delta)/\epsilon+1)pK^2}{m^2\epsilon\mu})$*. Then according to [87], the difference between population risk* F *and empirical risk* F_N *over* mn *data samples is bounded by* $\sup_{\boldsymbol{x}} |F(\boldsymbol{x}) - F_N(\boldsymbol{x})| \leq \mathcal{O}(1/mn)$*. Thus, the overall error of* `Q-DPSGD-1` *with respect to population risk* F *is* $\mathcal{O}(\frac{(2\log(1/\delta)/\epsilon+1)pK^2}{m^2\epsilon\mu} + \frac{1}{mn})$*.*

We next present the convergence result of `Q-DPSGD-1` for non-convex objectives regarding to first-order optimality and consensus convergence rate.

Theorem 13 (Non-convex)**.** *Under Assumptions 1–4, and for step-sizes* $\alpha = T^{-1/6}$ *and* $\varepsilon = T^{-1/2}$*,* `Q-DPSGD-1` *guarantees the following convergence and consensus rates:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{x}}_t\right)\right\|^2 \leq \mathcal{O}\left(\frac{\hat{K}\widetilde{\sigma}^2}{n} + \frac{\hat{K}^2\gamma^2}{(1-\beta)^2 m} + \frac{\sigma^2K^2\hat{K}^2p}{(1-\beta)^2}\right)\frac{1}{T^{1/3}} + \mathcal{O}\left(\hat{K}\frac{\gamma^2}{n}\max\left\{\frac{\mathbb{E}[1/V]}{T_d}, \frac{1}{m}\right\} + \frac{\sigma^2\hat{K}K^2p}{n}\right)\frac{1}{T^{2/3}}$$

$$\text{and}\quad \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\overline{\boldsymbol{x}}_t - \boldsymbol{x}_{i,t}\right\|^2 \leq \mathcal{O}\left(\frac{\gamma^2}{m(1-\beta)^2}\right)\frac{1}{T^{1/3}} + \mathcal{O}\left(\frac{\hat{K}^2}{(1-\beta)^4}\frac{\gamma^2}{m} + \frac{\hat{K}}{(1-\beta)^2}\frac{\widetilde{\sigma}^2}{n} + \frac{\sigma^2K^2\hat{K}^2p}{(1-\beta)^4}\right)\frac{1}{T^{2/3}}$$

for large enough number of iterations $T \geq T_{\mathsf{min}}^{\mathsf{nc}}$*. Here* $\overline{\boldsymbol{x}}_t = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_{i,t}$ *denotes the average models at iteration* t*.*

Remark 7. *Theorem 13 shows that* `Q-DPSGD-1` *finds first-order stationary points and the approximation error decays with a rate of* $\mathcal{O}(1/T^{1/3})$*. Moreover, the local models reach*

consensus as fast as $\mathcal{O}(1/T^{1/3})$. *Notice that the consensus rate result shows a balance between the variance of Gaussian noise and the graph connectivity.*

4.4.2 `Q-DPSGD-2`

Note that in `Q-DPSGD-1`, the DP noise is applied before random quantization, and the privacy guarantee of quantization operator roots from the post-processing property of DP. Is it possible that we can implement communication efficient and private collaborative learning in a reverse order, i.e., adopting DP noise after random quantization? Agarwal et al. in [63] indeed implemented such a design on the distributed mean estimation problem by applying the Binomial mechanism after random quantization. However, compared with the Gaussian mechanism, Binomial mechanism has very complicated privacy analysis and incurs large noise errors under the same privacy budget. Besides, as pointed out by [88], the Binomial mechanism cannot inherently benefit from the powerful privacy accountant like the moments accountant method. Thus, we consider to add Gaussian noises after quantization instead of Binomial noises to implement the collaborative learning.

The main challenge is that the transmitted values now are real numbers and the benefits of model quantization are lost, if we directly adding Gaussian noise after quantization. Our solution is to sample Gaussian noise from a discretization of Gaussian distribution and add the discrete Gaussian noise to the quantization values without sacrificing the communication efficiency. However, the problem here is whether the discrete Gaussian distribution still guarantees the same DP as the continuous Gaussian distribution. Fortunately, [73] has shown that discrete Gaussian provides the same CDP [31] as the continuous one. In general, the RDP view of privacy is broader than the CDP view as it captures finer information. Unlike RDP, CDP cannot enjoy the benefit from the privacy amplification of subsampling. Therefore, we in this work provide the RDP analysis for discrete Gaussian, which can use tight composition theory like analytical moments accountant [35].

Definition 8 (Discrete Gaussian [73])**.** *The discrete Gaussian distribution with location* $\mu \in \mathbb{R}$ *and scale* $\sigma \in \mathbb{R}$ *is denoted as* $\mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)$. *The corresponding probability distribution*

Algorithm 5 `Q-DPSGD-2` run by agent i

Input: Weights $\{w_{ij}\}_{j=1}^n$; Deadline T_d.
Set initial variables $\boldsymbol{x}_{i,0} = 0$ and $\boldsymbol{z}_{i,0} = Q(\boldsymbol{x}_{i,0})$.
for $t = 0, \cdots, T-1$ **do**
    Broadcast $\boldsymbol{z}_{i,t} = Q(\boldsymbol{x}_{i,t}) + \zeta_{i,t}$ to all neighbors $j \in \mathcal{N}_i$, where $\zeta_{i,t} \sim \mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2 K^2 I_p)$.
    Receive $\boldsymbol{z}_{j,t}$ from its neighbor $j \in \mathcal{N}_i$.
    Take and evaluate stochastic gradients $\{\nabla \ell(\boldsymbol{x}_{i,t}; \theta) : \theta \in \mathcal{S}_{i,t}\}$ till reaching the deadline T_d, with $\mathcal{S}_{i,t} \subseteq \{1, \cdots, m\}$.
    Generate gradient:
    $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) = \frac{1}{|\mathcal{S}_{i,t}|} \sum_{\theta \in \mathcal{S}_{i,t}} \nabla \ell(\boldsymbol{x}_{i,t}; \theta)$.
    Update $\boldsymbol{x}_{i,t+1} = (1 - \varepsilon + \varepsilon w_{ii}) \boldsymbol{x}_{i,t} + \varepsilon \sum_{j \in \mathcal{N}_i} w_{ij} \boldsymbol{z}_{j,t} - \alpha\varepsilon \widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})$.
end for

supported on the integers and defined by

$$\forall x \in \mathbb{Z}, \quad \mathop{\mathbb{P}}_{X \sim \mathcal{N}_{\mathbb{Z}}(\mu, \sigma^2)}[X = x] = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sum_{y \in \mathbb{Z}} e^{-(y-\mu)^2/2\sigma^2}}.$$

Theorem 14 (Discrete Gaussian Satisfies RDP)**.** *Let $\Delta, \sigma > 0$, $\rho > 1$. Let $\mathcal{M}_q : \mathcal{D} \to \mathbb{Z}$ satisfy $|\mathcal{M}_q(D) - \mathcal{M}_q(\hat{D})| \leq \Delta$ for all $D, \hat{D} \in \mathcal{D}$ differing on a single sample. Define a randomized algorithm $\mathcal{M}(D) = \mathcal{M}_q(D) + X$, where X is drawn from a discrete Gaussian distribution $\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$. Then $\mathcal{M}$ satisfies $(\rho, \rho\Delta^2/(2\sigma^2))$-RDP.*

Corollary 1 (Discrete Gaussian with Arbitrary Precision)**.** *Let $\Delta, \sigma, \eta > 0$, $\rho > 1$. Let $\mathcal{M}_q : \mathcal{D} \to \eta\mathbb{Z}$ with $\eta\mathbb{Z} = \{\eta z : z \in \mathbb{Z}\}$ satisfy $|\mathcal{M}_q(D) - \mathcal{M}_q(\hat{D})| \leq \Delta$ for all $D, \hat{D} \in \mathcal{D}$ differing on a single sample. Define a randomized algorithm $\mathcal{M}(D) = \mathcal{M}_q(D) + Y$, where Y is drawn from a discrete Gaussian distribution $\mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2)$, i.e.,*

$$\forall x \in \eta\mathbb{Z}, \quad \mathop{\mathbb{P}}_{X \sim \mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2)}[X = x] = \frac{e^{-x^2/2\sigma^2}}{\sum_{y \in \eta\mathbb{Z}} e^{-y^2/2\sigma^2}}.$$

Then $\mathcal{M}$ satisfies $(\rho, \rho\Delta^2/(2\sigma^2))$-RDP.

The details of `Q-DPSGD-2` is given in Algorithm 5. At iteration $t-1$, each agent is given a deadline time T_d to compute its per sample gradient $\nabla \ell(\boldsymbol{x}_{i,t-1}; \theta)$. At the end of the deadline, each agent computes its local mini-batch gradient $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t-1}) = \frac{1}{|\mathcal{S}_{i,t-1}|} \sum_{\theta \in \mathcal{S}_{i,t-1}} \nabla \ell(\boldsymbol{x}_{i,t-1}; \theta)$, where $\mathcal{S}_{i,t-1}$ is the batch size in such time period. Formally,

agent i updates $x_{i,t}$ according to

$$\boldsymbol{x}_{i,t} =(1-\varepsilon+\varepsilon w_{ii})\boldsymbol{x}_{i,t-1}+\varepsilon\sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{z}_{j,t-1}-\alpha\varepsilon\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t-1}). \tag{28}$$

Local variables x_i are then exchanged between neighboring agents. To reduce the communication cost of exchanging such variables, the quantization operator $Q(\cdot)$ is enforced to reduce the required number of bits. Thus, each agent i sends $\boldsymbol{z}_{j,t} = Q(\boldsymbol{x}_{i,t}) + \zeta_{i,t}$ to all neighbors $j \in \mathcal{N}_i$, where $\zeta_{i,t} \sim \mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2 K^2 I_p)$ is used to enforce DP guarantee of the quantization model variables. If the range of private local model $\boldsymbol{z}_{j,t}$ surpasses the representation range, post-processing (i.e., truncating) can be used to limit it.

Privacy guarantee We then provide the privacy guarantee of `Q-DPSGD-2` algorithm in the following theorem.

Theorem 15. *The `Q-DPSGD-2` algorithm satisfies (ϵ, δ)-DP with $\epsilon = \epsilon(\rho) + \frac{\log(1/\delta)}{\rho-1}$ and $\epsilon(\rho) = \max_i \ \sum_{t=0}^{T-1} \frac{8\rho}{\sigma^2 m^2}(\alpha\varepsilon + \frac{\eta\sqrt{p}}{K}|\mathcal{S}_{i,t}|)^2$ with $\rho = 2\log(1/\delta)/\epsilon + 1$.*

Remark 8. *From Theorem 15, we can see that the privacy budget is related to the step sizes α and ε, and the quantization resolution η and model dimension p. Diminishing step sizes α and ε can not only help balance the randomness introduced by exchanging quantized and private local models, but also improve the privacy guarantee (i.e., reduce the privacy budget).*

Convergence analysis The following is the convergence rate of `Q-DPSGD-2` algorithm for strongly convex and non-convex objectives, respectively.

Theorem 16 (Strongly Convex)**.** *If the conditions in Assumptions 1–5 are satisfied and step-sizes are picked as $\varepsilon = T^{-3\widetilde{\boldsymbol{\delta}}/2}$, $\alpha = T^{-\widetilde{\boldsymbol{\delta}}/2}$, for any $\widetilde{\boldsymbol{\delta}} \in (0, 1/2)$, then for large enough number of iterations $T \geq T^{\mathsf{c}}_{\mathsf{min}}$ the iterates generated by the `Q-DPSGD-2` algorithm satisfy*

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\boldsymbol{x}_{i,T}-\boldsymbol{x}^*\right\|^2 \leq \mathcal{O}\left(\frac{E^2(\hat{K}/\mu)^2}{(1-\beta)^2}+\frac{\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2}}{\mu}\right)\frac{1}{T^{\widetilde{\boldsymbol{\delta}}}}+\mathcal{O}\left(\frac{\gamma^2}{\mu}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}},$$

where $E^2 = 2K\sum_{i=1}^n (f_i(0) - f_i^*)$, *and* $f_i^* = \min_{\boldsymbol{x}\in\mathbb{R}^p} f_i(\boldsymbol{x})$ *and* $\boldsymbol{x}^*$ *is the solution of Problem* (25).

Remark 9. *From Theorem 16, the coefficient of* $1/T^{\widetilde{\boldsymbol{\delta}}}$ *is dominated by the error term introduced by random quantization and DP, i.e., the error term decays slower than that in Theorem 12. Nevertheless,* `Q-DPSGD-2` *still finds the global optimal for each agent with a* $\mathcal{O}(1/\sqrt{T})$ *convergence rate by choosing* $\widetilde{\boldsymbol{\delta}}$ *close to* $1/2$.

Remark 10. *Utilizing the strong convexity of objective function, if we choose* $|\mathcal{S}_{i,t}| = B$ *and* $\sigma^2 = \frac{16T(2\log(1/\delta)/\epsilon+1)(\alpha\varepsilon+\eta B\sqrt{p}/K)^2}{m^2\epsilon}$ *and* $T = (\frac{\mu\epsilon m^2}{p(2\log(1/\delta)/\epsilon+1)(\alpha\varepsilon K/\eta+\sqrt{p}B)^2})^{2/3}$, *then* `Q-DPSGD-2` *is* (ϵ,δ)*-DP and the empirical risk* $F_N(\boldsymbol{x}_{i,T}) - F_N(\boldsymbol{x}^*) = f(\boldsymbol{x}_{i,T}) - f(\boldsymbol{x}^*) \leq \mathcal{O}((\frac{p(2\log(1/\delta)/\epsilon+1)(\alpha\varepsilon K/\eta+\sqrt{p}B)^2}{\mu\epsilon m^2})^{2/3})$, *where* $\hat{\boldsymbol{x}}^*$ *is the minimizer of the empirical risk* F_N. *The overall error of* `Q-DPSGD-2` *regarding to population risk* F *is* $\mathcal{O}((\frac{p(2\log(1/\delta)/\epsilon+1)(\alpha\varepsilon K/\eta+\sqrt{p}B)^2}{\mu\epsilon m^2})^{2/3} + \frac{1}{mn})$. *Notice that the overall risk of* `Q-DPSGD-2`, *i.e.,* $\widetilde{\mathcal{O}}(\frac{p^{4/3}}{m^{4/3}\epsilon^{4/3}})$, *is higher than that of* `Q-DPSGD-1`, *i.e.,* $\widetilde{\mathcal{O}}(\frac{p}{m^2\epsilon^2})$, *where* $\widetilde{\mathcal{O}}$ *term omits logarithmic and other factors.*

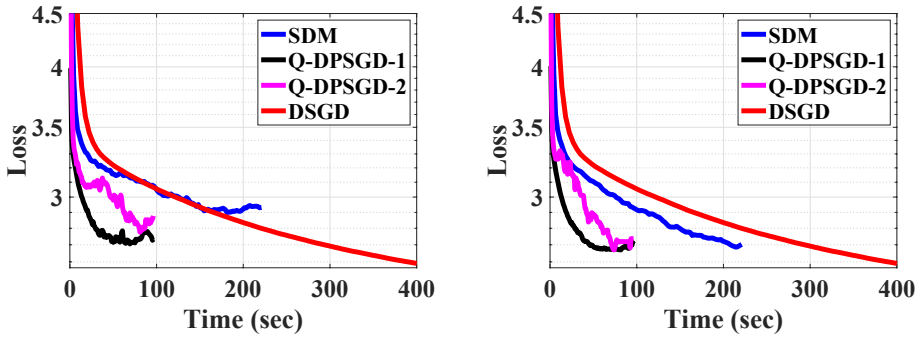
Theorem 17 (Non-convex)**.** *Under Assumptions 1–4, and for step-sizes* $\alpha = T^{-1/6}$ *and* $\varepsilon = T^{-1/2}$, `Q-DPSGD-2` *guarantees the following convergence and consensus rates as*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{x}}_t\right)\right\|^2 \leq \mathcal{O}\left(\hat{K}\frac{\gamma^2}{n}\max\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\}\right)\frac{1}{T^{2/3}} + \mathcal{O}\left(\frac{\hat{K}}{n}(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2}) + \frac{\hat{K}^2\gamma^2}{(1-\beta)^2 m}\right)\frac{1}{T^{1/3}}$$

$$\text{and}\quad \frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\overline{\boldsymbol{x}}_t - \boldsymbol{x}_{i,t}\right\|^2 \leq \mathcal{O}\left(\frac{\gamma^2}{m(1-\beta)^2}\right)\frac{1}{T^{1/3}} + \mathcal{O}\left(\frac{\hat{K}^2}{(1-\beta)^4}\frac{\gamma^2}{m} + \frac{\hat{K}}{(1-\beta)^2}\frac{(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{n}\right)\frac{1}{T^{2/3}}$$

for large enough number of iterations $T \geq T_{\mathsf{min}}^{\mathsf{nc}}$. *Here* $\overline{\boldsymbol{x}}_t = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_{i,t}$ *denotes the average models at iteration* t.

Remark 11. *From first upper bound in Theorem 17,* `Q-DPSGD-2` *indeed finds the first-order stationary points with a rate of* $\mathcal{O}(1/T^{1/3})$, *while in the second upper bound of Theorem 13,*



(a) $\epsilon = 1$ (b) $\epsilon = 1.5$

Figure 14: Compare loss on MNIST ($T_c = 3$, batch size $B = 20$, $s = 3$, $c = 0.3$).

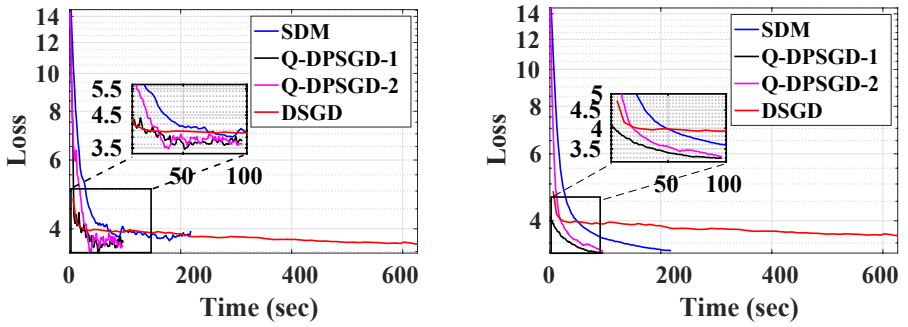
the error term due to DP appears in the both of coefficients of $1/T^{1/3}$ *and* $1/T^{2/3}$. *Moreover, the consensus error decays the same rate as* `Q-DPSGD-1`.

4.5 Experimental Results

In this section, we present the performance evaluation of the proposed two algorithms for solving a non-convex decentralized optimization problem. In particular, we compare the privacy-accuracy trade-off and the total run-time of our proposed algorithms against the ones for two baselines:

- Decentralized SGD (DSGD) [89]: Each agent updates its local model parameter as $\boldsymbol{x}_{i,t+1} = \sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{x}_{j,t} - \alpha\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})$. Note that the exchanged local parameter $\boldsymbol{x}_i$ with its neighbors is not quantized or compressed and the local gradients $\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})$ are computed for a fixed batch size.

- Sparse differential Gaussian-masked stochastic gradients (SDM) [71]: This algorithm communicates compressed local differentials $\mathbf{d}_{i,t-1} = \mathbf{y}_{i,t-1} - \boldsymbol{x}_{i,t-1}$ with its neighbors and then estimating neighbor's copies $\boldsymbol{x}_{i,t} = \boldsymbol{x}_{i,t-1} + S(\mathbf{d}_{i,t})$, where $S(\cdot)$ is a sparsifier operator. The output of $S(\cdot)$ follows the $Bernoulli(c)$ distribution, i.e., $\Pr[S(x) = x/c] = c$ and $\Pr[S(x) = 0] = 1 - c$. Thus, the update rule of SDM is $\mathbf{y}_{i,t} = (1-\theta)\boldsymbol{x}_{i,t} + \theta(\sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{x}_{j,t} - \alpha(\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) + \zeta_{i,t}))$, where $\zeta_{i,t}$ is a Gaussian random noise.



(a) $\epsilon = 1$ (b) $\epsilon = 1.5$

Figure 15: Compare loss on CIFAR-10 ($T_c = 3$, batch size $B = 20$, $s = 3$, $c = 0.3$).

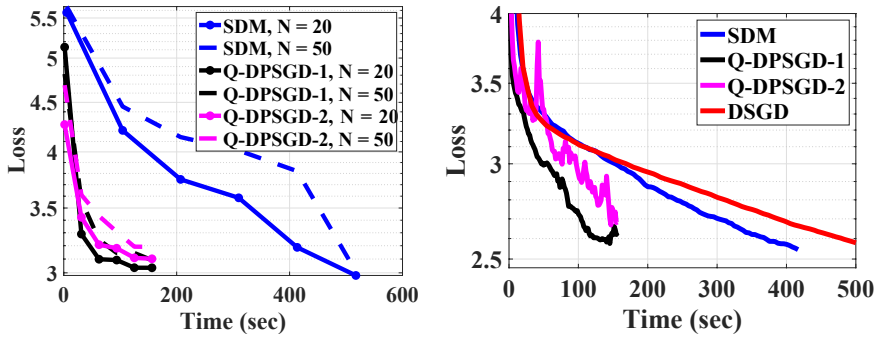
Dataset and Experiment Settings We conduct the experiments over two benchmark datasets: MNIST and CIFAR-10. For MNIST, we consider a fully connected network with a hidden layer of size 50. The image is transformed to a vector of length 784. For CIFAR-10, we use a fully connected neural network with one hidden layer with 40 neurons to classify the input image into 10 classes, where the input image is converted to a vector with 3072 dimensions. We use sigmoid function as the activation in both network.

In the experiments, we set the step sizes $(\alpha, \varepsilon) = (0.3/T^{1/6}, 11/T^{1/2})$ for `Q-DPSGD-1` and `Q-DPSGD-2`, and $\alpha = 0.2$ for DSGD and SDM. Moreover, we also set $\theta = 0.6$ as stated in [71] for SDM. To control the sensitivity of the gradient, we adopt gradient clipping threshold technique,

$\nabla \ell(\boldsymbol{x}_{i,t}; \theta) = \nabla \ell(\boldsymbol{x}_{i,t}; \theta) / \max\left(1, \|\nabla \ell(\boldsymbol{x}_{i,t}; \theta)\| / K\right)$. Here, we set $K = 0.5$ for `Q-DPSGD-1` and `Q-DPSGD-2` and SDM. In each simulation, we randomly sample $10,000$ records for training and divide them into n parties, and thus each party consists of $10000/n$ data samples (i.e., $m = 10000/n$). In all experiments, we set $\delta = 10^{-5}$.

We also set the processing speed of each machine follows a uniform distribution given as

$V \sim \text{Uniform}(10, 90)$, and then choose the deadline $T_d = B/\mathbb{E}[V]$, where B is the expected batch size used in each machine. We consider a low precision quantizer in (26) with various quantization levels s, and we denote T_c as the communication time of a p-dimension vector without quantization (16 bits). Thus, the communication time for a quantized vector and



(a) $\epsilon = 1$ (b) $\epsilon = 1$

Figure 16: Left: loss comparisons for different number of agents on MNIST ($B = 20$, $T_c = 3$, $c = 0.3$); Right: loss comparisons for large batch size $B = 50$ on MNIST.

compressed vector are proportioned according the quantization level and the compressed rate c, respectively.

Network Model We adopt a network with 10 agents, where the communication graph G is generated by the ERdös-Rényi graph with edge connectivity $p_c = 0.4$. The weight matrix is designed as $W = I - L/\kappa$ with Laplacian matrix L of G and $\kappa > \lambda_{\max}(L)/2$, where $\lambda_{\max}$ is the largest eigenvalue of L.

We present the convergence performance (i.e., loss) of different algorithms on MNIST and CIFAR-10 under the same budgets and same communication time, as shown in Figure 14. We can observe that when privacy budget decreases from 1.5 to 1, the loss values of private algorithms increase. Moreover, our proposed algorithms significantly outperform the baseline algorithms in terms of total run-time, since the utilization of quantization and deadline based scheme can reduce the communication cost while mitigating the straggler problem. Notice that `Q-DPSGD-2` exhibits a lower convergence rate compared to `Q-DPSGD-1`, which is consistent with our theoretical analysis in Remark 10.

Moreover, we also consider the impact of number of agents on the algorithm convergence, as shown in Fig. 16(a), The results shows that the proposed algorithms continue to have the highest accuracy for large networks. To evaluate the effect of batch sizes, we observe that large batch size can further reduce the loss while consuming more training time from Fig 14(a) and Fig. 16(b).

4.6 Omitted Proofs

Proof of Theorem 11

Proof. At iteration t, agent i updates $\boldsymbol{x}_{i,t+1}$ according to the update

$$\boldsymbol{x}_{i,t+1} =(1-\varepsilon+\varepsilon w_{ii})\boldsymbol{x}_{i,t}+\varepsilon\sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{z}_{j,t}-\alpha\varepsilon(\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})+\zeta_{i,t}),$$

where $\zeta_{i,t}\sim\mathcal{N}(0,\sigma^2K^2I_p))$.

We first show that the L_2-sensitivity of gradient $\hat{\mathcal{M}}_{i,t}=\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})=\frac{1}{|\mathcal{S}_{i,t}|}\sum_{\theta\in\mathcal{S}_{i,t}}\nabla\ell(\boldsymbol{x}_{i,t};\theta)$. Assume neighboring datasets $\mathcal{S}_{i,t}$ and $\mathcal{S}'_{i,t}$ differ by one samples θ_s and θ'_s, by the definition of sensitivity, we have

$$\begin{aligned}\Delta(\hat{\mathcal{M}}_{i,t}) &= \sup_{\mathcal{S}_{i,t}\sim\mathcal{S}'_{i,t}}\|\frac{1}{|\mathcal{S}_{i,t}|}\sum_{\theta\in|\mathcal{S}_{i,t}|}\nabla\ell(\boldsymbol{x}_{i,t};\theta)-\frac{1}{|\mathcal{S}'_{i,t}|}\sum_{\theta\in\mathcal{S}'_{i,t}}\nabla\ell(\boldsymbol{x}_{i,t};\theta)\|\\ &=\frac{1}{|\mathcal{S}_{i,t}|}\sup\|\nabla\ell(\boldsymbol{x}_{i,t};\theta_s)-\nabla\ell(\boldsymbol{x}_{i,t};\theta'_s)\|\\ &\leq\frac{2K}{|\mathcal{S}_{i,t}|}.\end{aligned}$$

By Lemma 5, $\hat{\mathcal{M}}_{i,t}$ preserves $(\rho,\epsilon_{i,t}(\rho))$-RDP with respect to $\mathcal{S}_{i,t}$, i.e., $\epsilon_{i,t}(\rho)=\frac{2\rho}{|\mathcal{S}_{i,t}|^2\sigma^2}$. Since $\mathcal{S}_{i,t}$ is a randomized subsample of D_i, by Lemma 8 and its approximate version [90] with sampling probability $\mathbf{p}=\frac{|\mathcal{S}_{i,t}|}{m}$, we then can compute $\epsilon'_{i,t}(\rho)$ so that $\hat{\mathcal{M}}_{i,t}$ preserves $(\rho,\epsilon'_{i,t}(\rho))$-RDP with respect to D_i, i.e., $\epsilon'_{i,t}(\rho)\approx\frac{8\rho}{m\sigma^2}$. Note that if $|\mathcal{S}_{i,t}|=0$, i.e., there are not any gradient computation by deadline T_d, we set $\epsilon'_{i,t}(\rho)=0$. Since the algorithms has run T iterations, according to Lemma 6 and parallel composition [25], `Q-DPSGD-1` is $(\rho,\epsilon(\rho))$-RDP with $\epsilon(\rho)=\max_i\ \sum_{t=0}^{T-1}\epsilon'_{i,t}(\rho)$. Moreover, `Q-DPSGD-1` is also (ϵ,δ)-DP with $\epsilon=\frac{\epsilon(\rho)+\log(1/\delta)}{\rho-1}$.

□

Proof of Theorem 12

In our analysis for both convex and non-convex scenarios, we need to have the noise of various stochastic gradient functions evaluated. Hence, let us start this section by the following

lemma which bounds the variance of stochastic gradient functions under our customary Assumption 4.

Lemma 15. *Assumption 4 results in the followings for any* $\boldsymbol{x} \in \mathbb{R}^p$ *and* $i \in [n]$*:*

i. $\mathbb{E}_{D_i}[\nabla f_i(\boldsymbol{x}, D_i)] = \mathbb{E}_\theta[\nabla l(\boldsymbol{x}, \theta)] = \nabla F(\boldsymbol{x})$

ii. $\mathbb{E}_{D_i}\left[\|\nabla f_i(\boldsymbol{x}, D_i) - \nabla F(\boldsymbol{x})\|^2\right] \leq \frac{\gamma^2}{m}$

iii. $\mathbb{E}_D\left[\|\nabla f(\boldsymbol{x}, D) - \nabla F(\boldsymbol{x})\|^2\right] \leq \frac{\gamma^2}{nm}$

iv. $\mathbb{E}_\theta\left[\|\nabla f_i(\boldsymbol{x}, D_i) - \nabla f(\boldsymbol{x}, D)\|^2\right] \leq \gamma_1^2 \coloneqq \gamma^2\left(\frac{1}{m} + \frac{1}{nm}\right)$

v. $\mathbb{E}\left[\widetilde{\nabla} f_i(\boldsymbol{x})\right] = \nabla F(\boldsymbol{x})$

vi. $\mathbb{E}\left[\left\|\widetilde{\nabla} f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\right\|^2\right] \leq \gamma_2^2 \coloneqq 2\gamma^2 \max\left\{\frac{\mathbb{E}[1/V]}{T_d}, \frac{1}{m}\right\}$

Proof. The first five expressions (i)-(v) in the lemma are immediate results of Assumption 4 together with the fact that the noise of the stochastic gradient scales down with the sample size. To prove (vi), let $\mathcal{S}_i$ denote the sample set for which node i has computed the gradients, i.e., $\widetilde{\nabla} f_i(\boldsymbol{x}) = \frac{1}{|\mathcal{S}_i|}\sum_{\theta \in \mathcal{S}_i} \nabla \ell(\boldsymbol{x}; \theta)$. We have

$$\begin{aligned}
\mathbb{E}\left[\left\|\widetilde{\nabla} f_i(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\right\|^2\right] &= \sum_b \Pr\left[|\mathcal{S}_i| = b\right] \mathbb{E}\left\|\frac{1}{b}\sum_{\theta \in \mathcal{S}_i} \nabla \ell(\boldsymbol{x}; \theta) - \nabla F(\boldsymbol{x})\right\|^2 \\
&\leq \gamma^2 \sum_b \Pr\left[|\mathcal{S}_i| = b\right] \frac{1}{b} \\
&= \gamma^2 \mathbb{E}[1/|\mathcal{S}_i|] \\
&= \gamma^2 \frac{\mathbb{E}[1/V]}{T_d},
\end{aligned}$$

and therefore

$$\begin{aligned}
\mathbb{E}\left[\left\|\widetilde{\nabla} f_i(\boldsymbol{x}) - \nabla f_i(\boldsymbol{x})\right\|^2\right] &= \mathbb{E}\left[\left\|\widetilde{\nabla} f_i(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\right\|^2\right] + \mathbb{E}\left[\|\nabla f_i(\boldsymbol{x}) - \nabla F(\boldsymbol{x})\|^2\right] \\
&\leq \gamma^2\left(\frac{\mathbb{E}[1/V]}{T_d} + \frac{1}{m}\right) \\
&\leq 2\gamma^2 \max\left\{\frac{\mathbb{E}[1/V]}{T_d}, \frac{1}{m}\right\}.
\end{aligned}$$

□

We first establish two Lemmas 16 and 17 and then easily conclude the theorem from the two results.

The main problem is to minimize the global objective defined in (25). We denote the vector $x = [\boldsymbol{x}_1; \cdots; \boldsymbol{x}_n] \in \mathbb{R}^{np}$ denotes the concatenation of all the local models. Clearly, $\widetilde{\boldsymbol{x}}^* \coloneqq [\boldsymbol{x}^*; \cdots; \boldsymbol{x}^*]$ is the solution to (25). We also define the matrix $\boldsymbol{W} = W \otimes I \in \mathbb{R}^{np \times np}$ as the Kronecker product of the weight matrix W $in \mathbb{R}^{n \times n}$ and the identity matrix $I \in \mathbb{R}^{n \times n}$ given in Assumption 1. Similarly, we further define $\boldsymbol{W}_d = W_d \otimes I \in \mathbb{R}^{np \times np}$, where $W_d = [w_{ii}] \in \mathbb{R}^{n \times n}$ is the diagonal matrix of the entries on the main diagonal of W. We also denote the boldface $\boldsymbol{I}$ as the identity matrix of size np. Then the constraint in the alternative problem (25) can be stated as $(\boldsymbol{I} - \boldsymbol{W})^{1/2} x = 0$. Inspired by this fact, we define the following penalty function for every α

$$h_\alpha(x) = \frac{1}{2} x^\top (\boldsymbol{I} - \boldsymbol{W}) x + \alpha n \hat{F}(x), \tag{29}$$

and denote by $\boldsymbol{x}_\alpha^*$ the (unique) minimizer of $h_\alpha(x)$. That is

$$\boldsymbol{x}_\alpha^* = \operatorname*{arg\,min}_{x \in \mathbb{R}^{np}} h_\alpha(x) = \operatorname*{arg\,min}_{x \in \mathbb{R}^{np}} \frac{1}{2} x^\top \big(\boldsymbol{I} - \boldsymbol{W}\big) x + \alpha n \hat{F}(x). \tag{30}$$

Next lemma characterizes the deviation of the models generated by the `Q-DPSGD-1` method at iteration T, that is $\boldsymbol{x}_T = [\boldsymbol{x}_{1,T}; \cdots; \boldsymbol{x}_{n,T}]$ from the optimizer of the penalty function, i.e. $\boldsymbol{x}_\alpha^*$.

Lemma 16. *Suppose Assumptions 1–5 hold. Then, the expected deviation of the output of* `Q-DPSGD-1` *from the solution to Problem* (29) *is upper bounded by*

$$\begin{aligned} \mathbb{E}\left[\|\boldsymbol{x}_T - \boldsymbol{x}_\alpha^*\|^2\right] \leq \mathcal{O}\left(\frac{n\sigma^2}{\mu} \|W - W_D\|^2\right) \frac{1}{T^{\widetilde{\boldsymbol{\delta}}}} + \mathcal{O}\left(\frac{n\gamma^2}{\mu}\left(\frac{\mathbb{E}[1/V]}{T_d} + \frac{1}{m}\right)\right) \frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}} \\ + \mathcal{O}\left(\frac{npK^2\sigma^2}{\mu} \frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}\right) \end{aligned} \tag{31}$$

for $\varepsilon = T^{-3\widetilde{\boldsymbol{\delta}}/2}$, $\alpha = 2T^{-\widetilde{\boldsymbol{\delta}}/2}$, *any* $\widetilde{\boldsymbol{\delta}} \in (0, 1/2)$ *and* $T \geq T^{\mathsf{c}}_{\mathsf{min1}}$, *where*

$$T^{\mathsf{c}}_{\mathsf{min1}} \coloneqq max\left\{ \left\lceil \left(\frac{(2+K)^2}{\mu} \right)^{\frac{1}{\widetilde{\boldsymbol{\delta}}}} \right\rceil, \left\lceil e^{e^{\frac{1}{1-2\widetilde{\boldsymbol{\delta}}}}} \right\rceil, \left\lceil \mu^{\frac{1}{2\widetilde{\boldsymbol{\delta}}}} \right\rceil \right\}. \tag{32}$$

Proof of Lemma 16. First note that the gradient of the penalty function h_α defined in (29) is as

$$\nabla h_\alpha(\boldsymbol{x}_t) = (\boldsymbol{I} - \boldsymbol{W})\, \boldsymbol{x}_t + \alpha n \nabla \hat{F}(\boldsymbol{x}_t), \tag{33}$$

where $\boldsymbol{x}_t = [\boldsymbol{x}_{1,t}; \cdots; \boldsymbol{x}_{n,t}]$ denotes the concatenation of models at iteration t. Now consider the following stochastic gradient function for h_α

$$\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) = (\boldsymbol{W}_D - \boldsymbol{W})\, \boldsymbol{z}_t + (\boldsymbol{I} - \boldsymbol{W}_D)\, \boldsymbol{x}_t + \alpha n \widetilde{\nabla} \hat{F}(\boldsymbol{x}_t), \tag{34}$$

where $\widetilde{\nabla} F(\boldsymbol{x}_t) = \left[\frac{1}{n} \widetilde{\nabla} f_1(\boldsymbol{x}_{1,t}); \cdots; \frac{1}{n} \widetilde{\nabla} f_n(\boldsymbol{x}_{n,t}) \right]$, and $\boldsymbol{z}_t = [\boldsymbol{z}_{1,t}; \cdots; \boldsymbol{z}_{n,t}]$ as the concatenation of the quantized variant of the local updates $\boldsymbol{x}_t$.

We let $\mathcal{F}^t$ denote a sigma algebra that measures the history of the system up until time t. According to Assumptions 2 and 4, the stochastic gradient defined above is unbiased, that is

$$\begin{aligned}
\mathbb{E}\left[\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) | \mathcal{F}^t \right] &= (\boldsymbol{W}_D - \boldsymbol{W})\, \mathbb{E}\left[\boldsymbol{z}_t | \mathcal{F}^t \right] + (\boldsymbol{I} - \boldsymbol{W}_D)\, \boldsymbol{x}_t + \alpha n \mathbb{E}\left[\widetilde{\nabla} \hat{F}(\boldsymbol{x}_t) | \mathcal{F}^t \right] \\
&= (\boldsymbol{I} - \boldsymbol{W})\, \boldsymbol{x}_t + \alpha n \nabla \hat{F}(\boldsymbol{x}_t) \\
&= \nabla h_\alpha(\boldsymbol{x}_t).
\end{aligned}$$

By denoting $\zeta_t = [\zeta_{1,t}; \cdots; \zeta_{n,t}]$ with $\zeta_{i,t} \sim \mathcal{N}(0, \sigma^2 K^2 I_p))$ as the concatenation of noise vectors at iteration t, we can also write the update rule of `Q-DPSGD-1` method as

$$\begin{aligned}
\boldsymbol{x}_{t+1} &= \boldsymbol{x}_t - \varepsilon \left((\boldsymbol{W}_D - \boldsymbol{W})\, \boldsymbol{z}_t + (\boldsymbol{I} - \boldsymbol{W}_D)\, \boldsymbol{x}_t + \alpha n \widetilde{\nabla} \hat{F}(\boldsymbol{x}_t) + \alpha \zeta_t \right) \\
&= \boldsymbol{x}_t - \varepsilon \widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) - \varepsilon \alpha \zeta_t,
\end{aligned} \tag{35}$$

which also represents an iteration of the Stochastic Gradient Descent (SGD) algorithm with

step-size ε in order to minimize the penalty function $h_\alpha(\boldsymbol{x})$ over $\boldsymbol{x} \in \mathbb{R}^{np}$. We can bound the deviation of the iteration generated by `Q-DPSGD-1` from the optimizer $\boldsymbol{x}_\alpha^*$ as

$$
\begin{aligned}
&\mathbb{E}\left[\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_\alpha^*\|^2 \,|\mathcal{F}^t\right] \qquad (36)\\
&= \mathbb{E}\left[\left\|\boldsymbol{x}_t - \varepsilon\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) - \varepsilon\alpha\zeta_t - \boldsymbol{x}_\alpha^*\right\|^2 |\mathcal{F}^t\right]\\
&= \|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 - 2\varepsilon\left\langle \boldsymbol{x}_t - \boldsymbol{x}_\alpha^*, \mathbb{E}\left[\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) - \varepsilon\alpha\zeta_t|\mathcal{F}^t\right]\right\rangle + \varepsilon^2\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) - \alpha\zeta_t\right\|^2 |\mathcal{F}^t\right]\\
&= \|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 - 2\varepsilon\left\langle \boldsymbol{x}_t - \boldsymbol{x}_\alpha^*, \nabla h_\alpha(\boldsymbol{x}_t)\right\rangle + \varepsilon^2\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) - \alpha\zeta_t\right\|^2 |\mathcal{F}^t\right]\\
&\le (1 - 2\mu_\alpha\varepsilon)\|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 + 2\varepsilon^2\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)\right\|^2 |\mathcal{F}^t\right] + 2\varepsilon^2\alpha^2K^2\sigma^2np, \qquad (37)
\end{aligned}
$$

where we used the fact that the penalty function h_α is strongly convex with parameter $\mu_\alpha \coloneqq \alpha\mu$, and $\mathbb{E}[\|a+b\|^2] \le 2\mathbb{E}\|a\|^2 + 2\mathbb{E}\|b\|^2$. Moreover, we can bound the second term in RHS of (37) as

$$
\begin{aligned}
&\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)\right\|^2 |\mathcal{F}^t\right]\\
&= \mathbb{E}\left[\left\|(\boldsymbol{W}_D - \boldsymbol{W})\,\boldsymbol{z}_t + (\boldsymbol{I} - \boldsymbol{W}_D)\,\boldsymbol{x}_t + \alpha n\widetilde{\nabla}\hat{F}(\boldsymbol{x}_t)\right\|^2 |\mathcal{F}^t\right]\\
&= \mathbb{E}\left[\left\|(\boldsymbol{I} - \boldsymbol{W})\,\boldsymbol{x}_t + \alpha n\nabla\hat{F}(\boldsymbol{x}_t) + (\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t - \boldsymbol{x}_t) + \alpha n\widetilde{\nabla}F(\boldsymbol{x}_t) - \alpha n\nabla\hat{F}(\boldsymbol{x}_t)\right\|^2 |\mathcal{F}^t\right]\\
&= \|\nabla h_\alpha(\boldsymbol{x}_t)\|^2 + \mathbb{E}\left[\|(\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t - \boldsymbol{x}_t)\|^2 |\mathcal{F}^t\right] + \alpha^2n^2\mathbb{E}\left[\left\|\widetilde{\nabla}\hat{F}(\boldsymbol{x}_t) - \nabla\hat{F}(\boldsymbol{x}_t)\right\|^2 |\mathcal{F}^t\right]\\
&\le K_\alpha^2\|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 + n\widetilde{\sigma}^2\|W - W_D\|^2 + \alpha^2n\gamma_2^2. \qquad (38)
\end{aligned}
$$

To derive (38), we used the facts that h_α is smooth with parameter $K_\alpha \coloneqq 1 - \lambda_n(W) + \alpha K$; the quantizer is unbiased with variance $\le \widetilde{\sigma}^2$ (Assumption 2); stochastic gradients of the loss function are unbiased and variance-bounded (Assumption 4 and Lemma 15). Plugging

(38) in (37) yields

$$\begin{aligned}
&\mathbb{E}\left[\left\|\boldsymbol{x}_{t+1}-\boldsymbol{x}_{\alpha}^{*}\right\|^{2}|\mathcal{F}^{t}\right] \\
&\leq\left(1-2\mu_{\alpha}\varepsilon+2\varepsilon^{2}K_{\alpha}^{2}\right)\left\|\boldsymbol{x}_{t}-\boldsymbol{x}_{\alpha}^{*}\right\|^{2}+2\varepsilon^{2}n\widetilde{\sigma}^{2}\left\|W-W_{D}\right\|^{2}+2\alpha^{2}\varepsilon^{2}n\gamma_{2}^{2}+2\varepsilon^{2}\alpha^{2}K^{2}\sigma^{2}np.
\end{aligned} \tag{39}$$

To ease the notation, let $e_t := \mathbb{E}[\,\|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2]$ denote the expected deviation of the models at iteration t i.e. $\boldsymbol{x}_t$ from the optimizer $\boldsymbol{x}_\alpha^*$ with respect to all the randomnesses from iteration $t = 0$. Therefore, we have

$$\begin{aligned}
e_{t+1} &\leq \left(1-2\mu_{\alpha}\varepsilon+2\varepsilon^{2}K_{\alpha}^{2}\right)e_{t}+2\varepsilon^{2}n\widetilde{\sigma}^{2}\left\|W-W_{D}\right\|^{2}+2\alpha^{2}\varepsilon^{2}\gamma_{2}^{2}+2\varepsilon^{2}\alpha^{2}K^{2}\sigma^{2}np \\
&=\left(1-\varepsilon(2\mu_{\alpha}-2\varepsilon K_{\alpha}^{2})\right)e_{t}+2\varepsilon^{2}n\widetilde{\sigma}^{2}\left\|W-W_{D}\right\|^{2}+2\alpha^{2}\varepsilon^{2}n\gamma_{2}^{2}+2\varepsilon^{2}\alpha^{2}K^{2}\sigma^{2}np.
\end{aligned} \tag{40}$$

For any $T \geq T_{\mathsf{min1}}^{\mathsf{c}}$ and the proposed pick $\varepsilon = T^{-3\widetilde{\boldsymbol{\delta}}/2}$, we have $T^{\widetilde{\boldsymbol{\delta}}} \geq (T_{\mathsf{min1}}^{\mathsf{c}})^{\widetilde{\boldsymbol{\delta}}} \geq \frac{(2+K)^2}{\mu}$ and therefore

$$\begin{aligned}
\varepsilon &= \frac{1}{T^{3\widetilde{\boldsymbol{\delta}}/2}} \\
&\leq \frac{\mu}{(2+K)^2}\cdot\frac{1}{T^{\widetilde{\boldsymbol{\delta}}/2}} \\
&\leq \frac{\mu_\alpha}{2(2+\alpha K)^2} \\
&\leq \frac{\mu_\alpha}{2K_\alpha^2}.
\end{aligned}$$

Hence, we can further bound (40) as

$$\begin{aligned}
e_{t+1} &\leq \left(1-\varepsilon\left(2\mu_{\alpha}-2\varepsilon K_{\alpha}^{2}\right)\right)e_{t}+2\varepsilon^{2}n\widetilde{\sigma}^{2}\left\|W-W_{D}\right\|^{2}+2\alpha^{2}\varepsilon^{2}n\gamma_{2}^{2}+2\varepsilon^{2}\alpha^{2}K^{2}\sigma^{2}np \\
&\leq \left(1-\mu_{\alpha}\varepsilon\right)e_{t}+2\varepsilon^{2}n\widetilde{\sigma}^{2}\left\|W-W_{D}\right\|^{2}+\alpha^{2}\varepsilon^{2}n(2\gamma_{2}^{2}+2K^{2}\sigma^{2}p) \\
&=\left(1-\frac{2\mu}{T^{2\widetilde{\boldsymbol{\delta}}}}\right)e_{t}+\frac{2n\widetilde{\sigma}^{2}\left\|W-W_{D}\right\|^{2}}{T^{3\widetilde{\boldsymbol{\delta}}}}+\frac{2n\gamma_{2}^{2}+2npK^{2}\sigma^{2}}{T^{4\widetilde{\boldsymbol{\delta}}}}.
\end{aligned}$$

Now, we let $(a,b,c) = (2\mu, 2n\widetilde{\sigma}^2\left\|W-W_D\right\|^2, 2n\gamma_2^2 + 2npK^2\sigma^2)$ and employ Lemma 18

which yields

$$
\begin{aligned}
e_T &= \mathbb{E}\left[\|\boldsymbol{x}_T - \boldsymbol{x}_\alpha^*\|^2\right] \\
&\leq \mathcal{O}\left(\frac{b/a}{T^{\widetilde{\boldsymbol{\delta}}}}\right) + \mathcal{O}\left(\frac{c/a}{T^{2\widetilde{\boldsymbol{\delta}}}}\right) \\
&= \mathcal{O}\left(\frac{n\widetilde{\sigma}^2}{\mu}\|W - W_D\|^2 \frac{1}{T^{\widetilde{\boldsymbol{\delta}}}}\right) + \mathcal{O}\left(\frac{n\gamma^2}{\mu}\left(\frac{\mathbb{E}[1/V]}{T_d} + \frac{1}{m}\right)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}\right) + \mathcal{O}\left(\frac{npK^2\sigma^2}{\mu}\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}\right),
\end{aligned}
$$

and the proof of Lemma 16 is concluded. □

Now we also bound the deviation of the optimizers of the penalty function and the main loss function, that is $\boldsymbol{x}_\alpha^*$ and $\widetilde{\boldsymbol{x}}^*$.

Lemma 17. *[81] Suppose Assumptions 1, 3–5 hold. Then the difference between the optimal solutions to* (25) *and its penalized version* (30) *is bounded above by*

$$
\|\boldsymbol{x}_\alpha^* - \widetilde{\boldsymbol{x}}^*\| \leq \mathcal{O}\left(\frac{\sqrt{2n}c_2 D\,(3 + 2K/\mu)}{1-\beta}\frac{1}{T^{\widetilde{\boldsymbol{\delta}}/2}}\right),
$$

for $\alpha = T^{-\widetilde{\boldsymbol{\delta}}/2}$, *any* $\widetilde{\boldsymbol{\delta}} \in (0, 1/2)$ *and* $T \geq T_{\mathsf{min2}}^{\mathsf{c}}$ *with*

$$
T_{\mathsf{min2}}^{\mathsf{c}} := max\left\{\left\lceil\left(\frac{K}{1+\lambda_n(W)}\right)^{\frac{2}{\widetilde{\boldsymbol{\delta}}}}\right\rceil, \left\lceil(\mu+K)^{\frac{2}{\widetilde{\boldsymbol{\delta}}}}\right\rceil\right\},
$$

where $E^2 = 2K\sum_{i=1}^n (f_i(0) - f_i^*)$, *and* $f_i^* = \min_{\boldsymbol{x}\in\mathbb{R}^p} f_i(\boldsymbol{x})$.

Having proved Lemmas 16 and 17, we can now plug them in Theorem 12 and write for

$T \geq T^{\mathsf{c}}_{\mathsf{min}} \coloneqq \max\{T^{\mathsf{c}}_{\mathsf{min1}}, T^{\mathsf{c}}_{\mathsf{min2}}\}$. We then have

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\|\boldsymbol{x}_{i,T}-\boldsymbol{x}^*\|^2\right] &= \frac{1}{n}\mathbb{E}\left[\|\boldsymbol{x}_T-\widetilde{\boldsymbol{x}}^*\|^2\right] \\
&= \frac{1}{n}\mathbb{E}\left[\|\boldsymbol{x}_T-\boldsymbol{x}^*_\alpha+\boldsymbol{x}^*_\alpha-\widetilde{\boldsymbol{x}}^*\|^2\right] \\
&\leq \frac{2}{n}\mathbb{E}\left[\|\boldsymbol{x}_T-\boldsymbol{x}^*_\alpha\|^2\right]+\frac{2}{n}\|\boldsymbol{x}^*_\alpha-\widetilde{\boldsymbol{x}}^*\|^2 \\
&\leq \mathcal{O}\bigg(\frac{E^2(K/\mu)^2}{(1-\beta)^2}+\frac{\widetilde{\sigma}^2}{\mu}\bigg)\frac{1}{T^{\widetilde{\boldsymbol{\delta}}}}+\mathcal{O}\bigg(\frac{\gamma^2}{\mu}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\bigg)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}} \\
&\quad+\mathcal{O}\left(\frac{pK^2\sigma^2}{\mu}\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}\right).
\end{aligned}$$

In the end, we state and proof Lemma 18 which we used its result earlier in the proof of Lemma 16.

Lemma 18. *[81] Let the non-negative sequence e_t satisfy the inequality*

$$e_{t+1} \leq \left(1-\frac{a}{T^{2\widetilde{\boldsymbol{\delta}}}}\right)e_t+\frac{b}{T^{3\widetilde{\boldsymbol{\delta}}}}+\frac{c}{T^{3\widetilde{\boldsymbol{\delta}}}}, \tag{41}$$

for $t=0,1,2,\cdots$, positive constants a,b,c and $\widetilde{\boldsymbol{\delta}} \in (0,1/2)$. Then, after

$$T \geq \max\left\{\left\lceil e^{e^{\frac{1}{1-2\widetilde{\boldsymbol{\delta}}}}}\right\rceil, \left\lceil a^{\frac{1}{2\widetilde{\boldsymbol{\delta}}}}\right\rceil\right\}$$

iterations, the iterate e_T satisfies

$$e_T \leq \mathcal{O}\left(\frac{b/a}{T^{\widetilde{\boldsymbol{\delta}}}}\right)+\mathcal{O}\left(\frac{c/a}{T^{2\widetilde{\boldsymbol{\delta}}}}\right). \tag{42}$$

Proof of Theorem 13

We the characterize the convergence rate of `Q-DPSGD-1` for non-convex and smooth objectives. We are interested in finding a set of local models which satisfy first-order optimality condition approximately, while the models are close to each other and satisfy the consensus condition up to a small error. To be more precise, we are interested in finding a set of local models $\{\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*\}$ where their average $\overline{\boldsymbol{x}}^* \coloneqq \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^*$ (approximately) satisfy first-order optimality condition, i.e., $\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{x}}^*\right)\right\|^2 \leq \nu$, while the iterates are close to their average, i.e., $\mathbb{E}\|\overline{\boldsymbol{x}}^* - \boldsymbol{x}_i^*\|^2 \leq \rho$.

To ease the notation, we agree in this section on the following shorthand notations for $t = 0, 1, 2, \cdots$,

$$
\begin{aligned}
X_t &= [\boldsymbol{x}_{1,t} \ \cdots \ \boldsymbol{x}_{n,t}] \in \mathbb{R}^{p\times n},\\
Z_t &= [\boldsymbol{z}_{1,t} \ \cdots \ \boldsymbol{z}_{n,t}] \in \mathbb{R}^{p\times n},\\
\overline{\boldsymbol{x}}_t &= \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_{i,t} \in \mathbb{R}^p,\\
\overline{X}_t &= [\overline{\boldsymbol{x}}_t \ \cdots \ \overline{\boldsymbol{x}}_t] \in \mathbb{R}^{p\times n},\\
\widetilde{\partial} f(X_t) &= \left[\widetilde{\nabla} f_1(\boldsymbol{x}_{1,t}) \ \cdots \ \widetilde{\nabla} f_n(\boldsymbol{x}_{n,t})\right] \in \mathbb{R}^{p\times n},\\
\partial f(X_t) &= [\nabla f_1(\boldsymbol{x}_{1,t}) \ \cdots \ \nabla f_n(\boldsymbol{x}_{n,t})] \in \mathbb{R}^{p\times n},\\
\text{and} \ \ \boldsymbol{\zeta}_t &= [\zeta_{1,t} \ \cdots \ \zeta_{n,t}] \in \mathbb{R}^{p\times n} \quad \text{with} \quad \zeta_{i,t} \sim \mathcal{N}(0, \sigma^2 K^2 I_p)).
\end{aligned}
$$

As stated before, we can write the update rule of the proposed `Q-DPSGD-1` in the matrix form as

$$
X_{t+1} = X_t\left((1-\varepsilon)I + \varepsilon W\right) + \varepsilon(Z_t - X_t)(W - W_D) - \alpha\varepsilon\widetilde{\partial} f(X_t) - \alpha\varepsilon\boldsymbol{\zeta}_t. \tag{43}
$$

Let us denote $W_\varepsilon = (1-\varepsilon)I + \varepsilon W$ and write (43) as

$$
X_{t+1} = X_t W_\varepsilon + \varepsilon(Z_t - X_t)(W - W_D) - \alpha\varepsilon\widetilde{\partial} f(X_t) - \alpha\varepsilon\boldsymbol{\zeta}_t. \tag{44}
$$

Assumption 1 implies that W is symmetric and doubly stochastic. Moreover, all the eigenvalues of W are in $(-1, 1]$, i.e., $1 = \lambda_1(W) \geq \lambda_2(W) \geq \cdots \geq \lambda_n(W) > -1$. We also denote by $1 - \beta$ the spectral gap associated with the stochastic matrix W, where $\beta = \max\{|\lambda_2(W)|, |\lambda_n(W)|\}$. Clearly for any $\varepsilon \in (0, 1]$, W_ε is also doubly stochastic with eigenvalues $\lambda_i(W_\varepsilon) = 1-\varepsilon+\varepsilon\lambda_i(W)$ and spectral gap $1-\beta_\varepsilon = 1-\max\{|\lambda_2(W_\varepsilon)|, |\lambda_n(W_\varepsilon)|\}$.

We start the convergence analysis by using the smoothness property of the objectives and write

$$\begin{aligned}
\mathbb{E} f\left(\frac{X_{t+1}\mathbf{1}_n}{n}\right) &= \mathbb{E} f\left(\frac{X_t W_\varepsilon \mathbf{1}_n}{n} + \frac{\varepsilon(Z_t - X_t)(W - W_D)\mathbf{1}_n}{n} - \frac{\alpha\varepsilon\widetilde{\partial} f(X_t)\mathbf{1}_n}{n} - \frac{\alpha\varepsilon\boldsymbol{\zeta}_t\mathbf{1}_n}{n}\right) \\
&\overset{\text{Assumption 3}}{\leq} \mathbb{E} f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \alpha\varepsilon\mathbb{E}\left\langle \nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right), \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\rangle \\
&\quad + \frac{\varepsilon^2 K}{2}\mathbb{E}\left\|\frac{(Z_t - X_t)(W - W_D)\mathbf{1}_n}{n} - \alpha\frac{\widetilde{\partial} f(X_t)\mathbf{1}_n}{n} - \alpha\frac{\boldsymbol{\zeta}_t\mathbf{1}_n}{n}\right\|^2. \qquad (45)
\end{aligned}$$

We specifically used the following equivalent form of the smoothness (Assumption 3) for every local and hence the global objective

$$f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}) + \langle \nabla f_i(\boldsymbol{x}), \boldsymbol{x}_1 - \boldsymbol{x}\rangle + \frac{K}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}\|^2, \quad \text{for all } i \in [n], \boldsymbol{x}, \boldsymbol{x}_1 \in \mathbb{R}^p.$$

Also, we used the simple fact in Assumption 1 as

$$W_\varepsilon\mathbf{1}_n = ((1-\varepsilon)I + \varepsilon W)\mathbf{1}_n = (1-\varepsilon)\mathbf{1}_n + \varepsilon W\mathbf{1}_n = \mathbf{1}_n.$$

Now let us bound the term in (45) as

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{(Z_t - X_t)(W - W_D)\mathbf{1}_n}{n} - \alpha\frac{\widetilde{\partial} f(X_t)\mathbf{1}_n}{n} - \alpha\frac{\boldsymbol{\zeta}_t\mathbf{1}_n}{n}\right\|^2 \\
&\leq 3\mathbb{E}\left\|\frac{(Z_t - X_t)(W - W_D)\mathbf{1}_n}{n}\right\|^2 + 3\mathbb{E}\left\|\alpha\frac{\widetilde{\partial} f(X_t)\mathbf{1}_n}{n}\right\|^2 + 3\mathbb{E}\left\|\alpha\frac{\boldsymbol{\zeta}_t\mathbf{1}_n}{n}\right\|^2 \\
&= \frac{3}{n^2}\sum_{i=1}^{n}(1 - w_{ii})^2\mathbb{E}\left\|\boldsymbol{z}_{i,t} - \boldsymbol{x}_{i,t}\right\|^2 + 3\alpha^2\mathbb{E}\left\|\frac{\widetilde{\partial} f(X_t)\mathbf{1}_n}{n}\right\|^2 + \frac{3\alpha^2K^2\sigma^2p}{n} \\
&\leq \frac{3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p}{n} + 3\alpha^2\mathbb{E}\left\|\frac{\widetilde{\partial} f(X_t)\mathbf{1}_n}{n}\right\|^2, \qquad (46)
\end{aligned}
$$

where we used Assumption 2 to derive the first term in (46). To bound the second term in (46), we have

$$
\begin{aligned}
\mathbb{E}\left\|\frac{\widetilde{\partial} f(X_t)\mathbf{1}_n}{n}\right\|^2 &= \mathbb{E}\left\|\frac{\sum_{i=1}^{n}\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 \\
&= \mathbb{E}\left\|\frac{\sum_{i=1}^{n}\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) - \nabla f_i(\boldsymbol{x}_{i,t}) + \nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 \\
&\leq \mathbb{E}\left\|\frac{\sum_{i=1}^{n}\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) - \nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 + \mathbb{E}\left\|\frac{\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 \\
&\leq \frac{\gamma^2}{n}\left(\frac{\mathbb{E}[1/V]}{T_d} + \frac{1}{m}\right) + \mathbb{E}\left\|\frac{\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 \\
&= \frac{\gamma_2^2}{n} + \mathbb{E}\left\|\frac{\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2, \qquad (47)
\end{aligned}
$$

where the last inequality follows from Lemma 15.

Plugging (47) in (45) yields

$$
\begin{aligned}
&\mathbb{E} f\left(\frac{X_{t+1}\mathbf{1}_n}{n}\right) \\
&\leq \mathbb{E} f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \alpha\varepsilon\mathbb{E}\left\langle \nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right), \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\rangle \\
&+ \frac{\varepsilon^2 K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2 p) + \frac{3\alpha^2\varepsilon^2 K}{2n}\gamma_2^2 + \frac{3\alpha^2\varepsilon^2 K}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^n \nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 \\
&= \mathbb{E} f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\alpha\varepsilon - 3\alpha^2\varepsilon^2 K}{2}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 - \frac{\alpha\varepsilon}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&+ \frac{\varepsilon^2 K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2 p) + \frac{3\alpha^2\varepsilon^2 K}{2n}\gamma_2^2 + \frac{\alpha\varepsilon}{2}\underbrace{\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2}_{T_1},
\end{aligned} \tag{48}
$$

where we used the identity $2\langle \mathbf{a}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a}-\mathbf{b}\|^2$. The term T_1 defined in (48) can be bounded as

$$
\begin{aligned}
T_1 &= \mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 \\
&\leq \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left\|\nabla f_i\left(\frac{X_t\mathbf{1}_n}{n}\right) - \nabla f_i(\boldsymbol{x}_{i,t})\right\|^2 \\
&\leq \frac{K^2}{n}\sum_{i=1}^n \underbrace{\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n} - \boldsymbol{x}_{i,t}\right\|^2}_{Q_{i,t}}.
\end{aligned}
$$

Let us define $Q_{i,t} \coloneqq \mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n} - \boldsymbol{x}_{i,t}\right\|^2$ and $M_t \coloneqq \frac{1}{n}\sum_{i=1}^n Q_{i,t} = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n} - \boldsymbol{x}_{i,t}\right\|^2$. Here, $Q_{i,t}$ captures the deviation of the model at node i from the average model at iteration t and M_t aggregates them to measure the average total consensus error. To bound M_t, we need to evaluate the recursive expressions as

$$
\begin{aligned}
X_t &= X_{t-1}W_\varepsilon + \varepsilon(Z_{t-1} - X_{t-1})(W - W_D) - \alpha\varepsilon\widetilde{\partial} f(X_{t-1}) - \alpha\varepsilon\boldsymbol{\zeta}_{t-1} \\
&= X_0 W_\varepsilon^t + \varepsilon\sum_{s=0}^{t-1}(Z_s - X_s)(W - W_D)W_\varepsilon^{t-s-1} - \alpha\varepsilon\sum_{s=0}^{t-1}\widetilde{\partial} f(X_s)W_\varepsilon^{t-s-1} - \alpha\varepsilon\sum_{s=0}^{t-1}\boldsymbol{\zeta}_s W_\varepsilon^{t-s-1}.
\end{aligned} \tag{49}
$$

Now, using (49) we can write

$$
\begin{aligned}
M_t &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n}-\boldsymbol{x}_{i,t}\right\|^2\\
&= \frac{1}{n}\mathbb{E}\left\|\overline{X}_t - X_t\right\|_F^2\\
&= \frac{1}{n}\mathbb{E}\left\|X_t\frac{\mathbf{1}\mathbf{1}^\top}{n}-X_t\right\|_F^2\\
&= \frac{1}{n}\mathbb{E}\Bigg\|X_0\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^t\right)+\varepsilon\sum_{s=0}^{t-1}(Z_s-X_s)(W-W_D)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\\
&\qquad -\alpha\varepsilon\sum_{s=0}^{t-1}\widetilde{\partial}f(X_s)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)-\alpha\varepsilon\sum_{s=0}^{t-1}\boldsymbol{\zeta}_s\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\Bigg\|_F^2\\
&= \frac{(\alpha\varepsilon)^2}{n}\underbrace{\mathbb{E}\left\|\sum_{s=0}^{t-1}\widetilde{\partial}f(X_s)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2}_{T_2}+\frac{(\alpha\varepsilon)^2}{n}\underbrace{\mathbb{E}\left\|\sum_{s=0}^{t-1}\boldsymbol{\zeta}_s\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2}_{T_4}\\
&\quad +\frac{\varepsilon^2}{n}\underbrace{\mathbb{E}\left\|\sum_{s=0}^{t-1}(Z_s-X_s)(W-W_D)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2}_{T_3},
\end{aligned}
$$

where we used the fact that quantiziations, stochastic gradients, and DP noise are statistically independent and $X_0=0$. According the proof of Theorem 2 in [81], we can bound T_2 as

$$
\begin{aligned}
T_2 &= \mathbb{E}\left\|\sum_{s=0}^{t-1}\widetilde{\partial}f(X_s)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2\\
&= \mathbb{E}\left\|\sum_{s=0}^{t-1}\left(\widetilde{\partial}f(X_s)-\partial f(X_s)+\partial f(X_s)\right)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2\\
&\le 2\mathbb{E}\left\|\sum_{s=0}^{t-1}\left(\widetilde{\partial}f(X_s)-\partial f(X_s)\right)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2+2\mathbb{E}\left\|\sum_{s=0}^{t-1}\partial f(X_s)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2\\
&\le \frac{2n\gamma_2^2}{1-\beta_\varepsilon^2}+6K^2\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right\|^2+6\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right\|^2\\
&\quad +12\sum_{s=0}^{t-1}\left(3K^2\sum_{i=1}^{n}Q_{i,s}+3\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\right)\frac{\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}+18n\gamma_1^2\frac{1}{(1-\beta_\varepsilon)^2}.
\end{aligned}
$$

Moreover, the term T_3 can be bounded as

$$\begin{aligned} T_3 &= \mathbb{E}\left\|\sum_{s=0}^{t-1}(Z_s - X_s)(W - W_D)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right)\right\|_F^2 \\ &\leq \mathbb{E}\sum_{s=0}^{t-1}\|Z_s - X_s\|_F^2\,\|W - W_D\|^2\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right\|^2 \\ &\leq \frac{4n\widetilde{\sigma}^2}{1-\beta_\varepsilon^2}, \end{aligned}$$

where we used the fact that $\|W - W_D\| \leq 2$.

We can also bound term T_4 as

$$\begin{aligned} T_4 &= \mathbb{E}\left\|\sum_{s=0}^{t-1}\boldsymbol{\zeta}_s\left(\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right)\right\|_F^2 \\ &\leq \mathbb{E}\sum_{s=0}^{t-1}\|\boldsymbol{\zeta}_s\|_F^2\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right\|^2 \\ &\leq \frac{n\sigma^2K^2p}{1-\beta_\varepsilon^2}. \end{aligned}$$

Now we use the bounds derived for T_2 and T_3 and T_4 to bound the consensus error M_t as

$$
\begin{aligned}
M_t &\leq \frac{\alpha^2\varepsilon^2}{n}T_2 + \frac{\varepsilon^2}{n}T_3 + \frac{\alpha^2\varepsilon^2}{n}T_4 \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{6\alpha^2\varepsilon^2K^2}{n}\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right\|^2 \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right\|^2 \\
&\quad + \frac{12\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\left(3K^2\sum_{i=1}^{n}Q_{i,s} + 3\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\right)\frac{\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} \\
&\quad + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2} + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2} \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2} + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2} + \frac{6\alpha^2\varepsilon^2K^2}{n}\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\beta_\varepsilon^{2(t-s-1)} \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\beta_\varepsilon^{2(t-s-1)} \\
&\quad + \frac{12\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\left(3K^2\sum_{i=1}^{n}Q_{i,s} + 3\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\right)\frac{\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2} + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2} \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\beta_\varepsilon^{2(t-s-1)} + \frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}\right) \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}K^2\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\left(\frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} + \beta_\varepsilon^{2(t-s-1)}\right).
\end{aligned} \tag{50}
$$

As we defined earlier, we have $M_s = \frac{1}{n}\sum_{i=1}^{n}Q_{i,s}$ which simplifies (50) as

$$
\begin{aligned}
M_t &\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2} + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2} \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\beta_\varepsilon^{2(t-s-1)} + \frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}\right) \\
&\quad + 6\alpha^2\varepsilon^2K^2\sum_{s=0}^{t-1}M_s\left(\frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} + \beta_\varepsilon^{2(t-s-1)}\right).
\end{aligned} \tag{51}
$$

Now we can sum (51) over $t = 0, 1, \cdots, T-1$, which yields

$$\begin{aligned}
\sum_{t=0}^{T-1} M_t \leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2}T \\
& + \frac{6\alpha^2\varepsilon^2}{n}\sum_{t=0}^{T-1}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\beta_\varepsilon^{2(t-s-1)} + \frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}\right) \\
& + 6\alpha^2\varepsilon^2K^2\sum_{t=0}^{T-1}\sum_{s=0}^{t-1}M_s\left(\frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} + \beta_\varepsilon^{2(t-s-1)}\right) \\
\leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2} \\
& + \frac{6\alpha^2\varepsilon^2}{n}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\sum_{k=0}^{\infty}\beta_\varepsilon^{2k} + \frac{2\sum_{k=0}^{\infty}\beta_\varepsilon^k}{1-\beta_\varepsilon}\right) \\
& + 6\alpha^2\varepsilon^2K^2\sum_{t=0}^{T-1}M_t\left(\frac{2\sum_{k=0}^{\infty}\beta_\varepsilon^k}{1-\beta_\varepsilon} + \sum_{k=0}^{\infty}\beta_\varepsilon^{2k}\right) \\
\leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2} \\
& + \frac{18\alpha^2\varepsilon^2}{n(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2 + \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}M_t. \qquad (52)
\end{aligned}$$

Note that $\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2 = n\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2$, which simplifies (52) as

$$\begin{aligned}
\sum_{t=0}^{T-1} M_t \leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2}T \\
& + \frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2 + \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}M_t. \qquad (53)
\end{aligned}$$

Rearranging the terms implies that

$$\begin{aligned}
\left(1 - \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}\right)\sum_{t=0}^{T-1} M_t \leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{1-\beta_\varepsilon^2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{1-\beta_\varepsilon^2}T \\
& + \frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2. \qquad (54)
\end{aligned}$$

Now define $D_2 \coloneqq 1 - \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}$ and rewrite (54) as

$$\sum_{t=0}^{T-1} M_t \leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2D_2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2. \tag{55}$$

Note that from definition of T_1 we have $T_1 \leq \frac{K^2}{n}\sum_{i=1}^n Q_{i,t} = K^2M_t$. Now use the above fact in the recursive equation (48) which we started with, that is

$$\mathbb{E}f\left(\frac{X_{t+1}\mathbf{1}_n}{n}\right) \leq \mathbb{E}f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\alpha\varepsilon - 3\alpha^2\varepsilon^2K}{2}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 - \frac{\alpha\varepsilon}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 + \frac{\varepsilon^2K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p) + \frac{3\alpha^2\varepsilon^2K}{2n}\gamma_2^2 + \frac{\alpha\varepsilon K^2}{2}M_t. \tag{56}$$

If we sum (56) over $t = 0, 1, \cdots, T-1$, we get

$$\begin{aligned}
&\frac{\alpha\varepsilon - 3\alpha^2\varepsilon^2K}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 + \frac{\alpha\varepsilon}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq f(0) - f^* + \frac{\varepsilon^2K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p)T + \frac{3\alpha^2\varepsilon^2K}{2n}\gamma_2^2T + \frac{\alpha\varepsilon K^2}{2}\sum_{t=0}^{T-1}M_t \\
&\overset{\text{from (55)}}{\leq} f(0) - f^* + \frac{\varepsilon^2K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p)T + \frac{3\alpha^2\varepsilon^2K}{2n}\gamma_2^2T \\
&\quad + \frac{\alpha\varepsilon K^2}{2}\left\{\frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{(1-\beta_\varepsilon^2)D_2}T\right\} \\
&\quad + \frac{9\alpha^3\varepsilon^3K^2}{(1-\beta_\varepsilon)^2D_2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2.
\end{aligned} \tag{57}$$

We can rearrange the terms in (57) and rewrite it as

$$\begin{aligned}
&\frac{\alpha\varepsilon - 3\alpha^2\varepsilon^2K}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 + \alpha\varepsilon\left(\frac{1}{2} - \frac{9\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2D_2}\right)\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq f(0) - f^* + \frac{\varepsilon^2K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p)T + \frac{3\alpha^2\varepsilon^2K}{2n}\gamma_2^2T \\
&\quad + \frac{\alpha\varepsilon K^2}{2}\left\{\frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{\alpha^2\varepsilon^2\sigma^2K^2p}{(1-\beta_\varepsilon^2)D_2}T\right\}.
\end{aligned} \tag{58}$$

Now, we define D_1 as $D_1 \coloneqq \frac{1}{2} - \frac{9\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2D_2}$ and replace in (58) which yields

$$\begin{aligned}
&\frac{1}{\alpha\varepsilon T}\left\{\frac{\alpha\varepsilon - 3\alpha^2\varepsilon^2K}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 + \alpha\varepsilon D_1\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2\right\}\\
&\leq \frac{1}{\alpha\varepsilon T}(f(0)-f^*) + \frac{\varepsilon}{\alpha}\frac{K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p) + \alpha\varepsilon\frac{3K\gamma_2^2}{2n}\\
&\quad + \frac{\alpha^2\varepsilon^2}{1-\beta_\varepsilon^2}\frac{K^2\gamma_2^2}{D_2} + \frac{\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\frac{9K^2\gamma_1^2}{D_2} + \frac{\varepsilon^2}{1-\beta_\varepsilon^2}\frac{2K^2\widetilde{\sigma}^2}{D_2} + \frac{\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\frac{\sigma^2K^4p}{2D_2}. \qquad (59)
\end{aligned}$$

To balance the terms in RHS of (59), we need to know how β_ε behaves with ε. As we defined before, $W_\varepsilon = (1-\varepsilon)I + \varepsilon W$. Hence, $\lambda_i(W_\varepsilon) = 1 - \varepsilon + \varepsilon\lambda_i(W)$. Therefore, for $\varepsilon \leq \frac{1}{1-\lambda_n(W)}$, we have

$$\begin{aligned}
\beta_\varepsilon &= \max\left\{|\lambda_2(W_\varepsilon)|, |\lambda_n(W_\varepsilon)|\right\}\\
&= \max\left\{|1-\varepsilon+\varepsilon\lambda_2(W)|, |1-\varepsilon+\varepsilon\lambda_n(W)|\right\}\\
&= \max\left\{1-\varepsilon+\varepsilon\lambda_2(W), 1-\varepsilon+\varepsilon\lambda_n(W)\right\}\\
&= 1 - \varepsilon\left(1-\lambda_2(W)\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
1-\beta_\varepsilon &= \varepsilon\left(1-\lambda_2(W)\right) \geq \varepsilon(1-\beta)\\
\text{and}\quad 1-\beta_\varepsilon^2 &= 2\varepsilon\left(1-\lambda_2(W)\right) - \varepsilon^2\left(1-\lambda_2(W)\right)^2 \geq \varepsilon(1-\beta^2).
\end{aligned}$$

Moreover, if $\alpha\varepsilon \leq \frac{6}{K}$, we have

$$\begin{aligned}
\frac{D_1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 &\leq \frac{1}{\alpha\varepsilon T}(f(0)-f^*) + \frac{\varepsilon}{\alpha}\frac{K}{2n}(3\widetilde{\sigma}^2 + 3\alpha^2K^2\sigma^2p) + \alpha\varepsilon\frac{3K\gamma_2^2}{2n}\\
&\quad + \frac{\alpha^2\varepsilon}{1-\beta^2}\frac{K^2\gamma_2^2}{D_2} + \frac{\alpha^2}{(1-\beta)^2}\frac{9K^2\gamma_1^2}{D_2} + \frac{\varepsilon}{1-\beta^2}\frac{2K^2\widetilde{\sigma}^2}{D_2}\\
&\quad + \frac{\alpha^2}{(1-\beta)^2}\frac{\sigma^2K^4p}{2D_2}. \qquad (60)
\end{aligned}$$

For $\alpha \leq \frac{1-\beta}{6K}$, we have

$$\begin{aligned} D_2 &= 1 - \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2} \\ &= 1 - \frac{18\alpha^2\varepsilon^2K^2}{\varepsilon^2(1-\beta)^2} \\ &= 1 - \frac{18\alpha^2K^2}{(1-\beta)^2} \\ &\geq \frac{1}{2}, \end{aligned} \tag{61}$$

and for $\alpha \leq \frac{1-\beta}{6\sqrt{2K}}$ we have

$$\begin{aligned} D_1 &= \frac{1}{2} - \frac{9\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2 D_2} \\ &\geq \frac{1}{2} - \frac{18\alpha^2\varepsilon^2K^2}{\varepsilon^2(1-\beta)^2} \\ &= \frac{1}{2} - \frac{18\alpha^2K^2}{(1-\beta)^2} \\ &\geq \frac{1}{4}. \end{aligned}$$

Now, we pick the step-sizes as

$$\alpha = \frac{1}{T^{1/6}}, \tag{62}$$

$$\text{and} \quad \varepsilon = \frac{1}{T^{1/2}}. \tag{63}$$

It is clear that in order to satisfy the conditions mentioned before, that are $\varepsilon \leq \frac{1}{1-\lambda_n(W)}$, $\alpha\varepsilon \leq \frac{6}{K}$ and $\alpha \leq \frac{1-\beta}{6\sqrt{2K}}$, it suffices to pick T as large as

$$T \geq T_{\mathsf{min}}^{\mathsf{nc}} \coloneqq \max\left\{ \left(1-\lambda_n(W)\right)^2, (K/6)^{3/2}, \left(\frac{6\sqrt{2}K}{1-\beta}\right)^6 \right\}. \tag{64}$$

For such T, we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq \frac{1}{T^{1/3}}4(f(0)-f^*)+\frac{1}{T^{1/3}}\frac{K}{n}(6\widetilde{\sigma}^2+\frac{1}{T^{1/3}}6K^2\sigma^2 p)+\frac{1}{T^{2/3}}\frac{6K\gamma_2^2}{n} \\
&\quad+\frac{1}{T^{5/6}}\frac{8K^2\gamma_2^2}{1-\beta^2}+\frac{1}{T^{1/3}}\frac{72K^2\gamma_1^2}{(1-\beta)^2}+\frac{1}{T^{1/2}}\frac{16K^2\widetilde{\sigma}^2}{1-\beta^2}+\frac{1}{T^{1/3}}\frac{4\sigma^2K^4p}{(1-\beta)^2} \\
&=\frac{B_1}{T^{1/3}}+\frac{B_2}{T^{1/2}}+\frac{B_3}{T^{2/3}}+\frac{B_4}{T^{5/6}} \qquad (65)\\
&=\mathcal{O}\left(\frac{K\widetilde{\sigma}^2}{n}+\frac{K^2\gamma^2}{(1-\beta)^2m}+\frac{\sigma^2K^4p}{(1-\beta)^2}\right)\frac{1}{T^{1/3}}+\mathcal{O}\left(\frac{K^2}{1-\beta^2}\sigma^2\right)\frac{1}{T^{1/2}} \\
&\quad+\mathcal{O}\left(K\frac{\gamma^2}{n}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}+\frac{\sigma^2K^3p}{n}\right)\frac{1}{T^{2/3}} \\
&\quad+\mathcal{O}\left(\frac{K^2}{1-\beta^2}\gamma^2\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T^{5/6}},
\end{aligned}
$$

where

$$
\begin{aligned}
B_1 &\coloneqq 4(f(0)-f^*)+\frac{K}{n}(6\widetilde{\sigma}^2)+\frac{72K^2\gamma_1^2}{(1-\beta)^2}+\frac{4\sigma^2K^4p}{(1-\beta)^2},\\
B_2 &\coloneqq \frac{16K^2\widetilde{\sigma}^2}{1-\beta^2},\\
B_3 &\coloneqq \frac{6K\gamma_2^2}{n}+\frac{6\sigma^2K^3p}{n},\\
\text{and}\quad B_4 &\coloneqq \frac{8K^2\gamma_2^2}{1-\beta^2}.
\end{aligned}
$$

Now we bound the consensus error. From (55) we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n}-\boldsymbol{x}_{i,t}\right\|^2\\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}+\frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}+\frac{4\varepsilon^2\widetilde{\sigma}^2}{(1-\beta_\varepsilon^2)D_2}+\frac{\alpha^2\varepsilon^2\sigma^2K^2p}{(1-\beta_\varepsilon^2)D_2}\\
&\quad+\frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2D_2}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2\\
&\leq \alpha^2\varepsilon\frac{2\gamma_2^2}{(1-\beta^2)D_2}+\alpha^2\frac{18\gamma_1^2}{(1-\beta)^2D_2}+\varepsilon\frac{4\widetilde{\sigma}^2}{(1-\beta^2)D_2}+\alpha^2\varepsilon\frac{\sigma^2K^2p}{(1-\beta^2)D_2}\\
&\quad+\alpha^2\frac{18}{(1-\beta)^2D_2}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2.
\end{aligned}
$$

For the same step-sizes α and ε defined in (62) and large enough T as in (64), we can use the convergence result in (65) which yields

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n}-\boldsymbol{x}_{i,t}\right\|^2\\
&\leq \frac{1}{T^{5/6}}\frac{4\gamma_2^2}{1-\beta^2}+\frac{1}{T^{1/3}}\frac{36\gamma_1^2}{(1-\beta)^2}+\frac{1}{T^{1/2}}\frac{8\widetilde{\sigma}^2}{(1-\beta^2)}+\frac{1}{T^{5/6}}\frac{2\sigma^2K^2p}{(1-\beta^2)}\\
&\quad+\frac{1}{T^{1/3}}\frac{36}{(1-\beta)^2}\left(\frac{B_1}{T^{1/3}}+\frac{B_2}{T^{1/2}}+\frac{B_3}{T^{2/3}}+\frac{B_4}{T^{5/6}}\right)\\
&=\frac{C_1}{T^{1/3}}+\frac{C_2}{T^{1/2}}+\frac{C_3}{T^{2/3}}+\frac{C_4}{T^{5/6}}+\frac{C_5}{T}+\frac{C_6}{T^{7/6}}\\
&=\mathcal{O}\left(\frac{\gamma^2}{m(1-\beta)^2}\right)\frac{1}{T^{1/3}}+\mathcal{O}\left(\frac{\widetilde{\sigma}^2}{1-\beta^2}\right)\frac{1}{T^{1/2}}\\
&\quad+\mathcal{O}\left(\frac{K^2}{(1-\beta)^4}\frac{\gamma^2}{m}+\frac{K}{(1-\beta)^2}\frac{\widetilde{\sigma}^2}{n}+\frac{\sigma^2K^4p}{(1-\beta)^4}\right)\frac{1}{T^{2/3}}\\
&\quad+\mathcal{O}\left(\frac{\gamma^2}{1-\beta^2}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}+\frac{K^2\widetilde{\sigma}^2}{(1-\beta)^4}+\frac{2\sigma^2K^2p}{(1-\beta^2)}\right)\frac{1}{T^{5/6}}\\
&\quad+\mathcal{O}\left(\frac{K}{(1-\beta)^2}\frac{\gamma^2}{n}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}+\frac{\sigma^2K^3p}{(1-\beta)^2n}\right)\frac{1}{T}\\
&\quad+\mathcal{O}\left(\frac{K^2}{(1-\beta)^4}\gamma^2\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T^{7/6}},
\end{aligned}
$$

where

$$
\begin{aligned}
C_1 &\coloneqq \frac{36\gamma_1^2}{(1-\beta)^2}, \\
C_2 &\coloneqq \frac{8\widetilde{\sigma}^2}{1-\beta^2}, \\
C_3 &\coloneqq \frac{36}{(1-\beta)^2}(4(f(0)-f^*)) + \frac{216K\widetilde{\sigma}^2}{(1-\beta)^2 n} + \frac{2592K^2\gamma_1^2}{(1-\beta)^4} + \frac{144\sigma^2K^4p}{(1-\beta)^4}, \\
C_4 &\coloneqq \frac{4\gamma_2^2}{1-\beta^2} + \frac{2\sigma^2K^2p}{(1-\beta^2)} + \frac{576K^2\widetilde{\sigma}^2}{(1-\beta)^2(1-\beta^2)}, \\
C_5 &\coloneqq \frac{216K\gamma_2^2}{(1-\beta)^2 n} + \frac{216\sigma^2K^3p}{(1-\beta)^2 n}, \\
\text{and } \quad C_6 &\coloneqq \frac{288K^2\gamma_2^2}{(1-\beta)^2(1-\beta^2)}.
\end{aligned}
$$

Proof of Theorem 14

Proof. For $\rho > 1, \epsilon > 0$, a randomized mechanism $\mathcal{M}$ satisfies (ρ, ϵ)-Rényi differential privacy, i.e., (ρ, ϵ)-RDP, if for all adjacent datasets $D, \hat{D}$ differing by one element, we have

$$\mathscr{D}_\rho\big(\mathcal{M}(D)||\mathcal{M}(\hat{D})\big) := \log \mathbb{E}\big(\mathcal{M}(D)/\mathcal{M}(\hat{D})\big)^\rho/(\rho-1) \leq \epsilon,$$

where the expectation is taken over $\mathcal{M}(\hat{D})$. Then we get

$$\begin{aligned}
&e^{(\rho-1)\mathscr{D}_\rho\left(\mathcal{M}(D)||\mathcal{M}(\hat{D})\right)} \\
&= e^{(\rho-1)\mathscr{D}_\rho\left(\mathcal{N}_{\mathbb{Z}}(\mathcal{M}_q(D),\sigma^2)||\mathcal{N}_{\mathbb{Z}}(\mathcal{M}_q(\hat{D}),\sigma^2)\right)} \\
&= \sum_{x\in\mathbb{Z}} \mathop{\mathbb{P}}_{X\sim\mathcal{N}_{\mathbb{Z}}(\mathcal{M}_q(D),\sigma^2)}[X=x]^\rho \mathop{\mathbb{P}}_{X\sim\mathcal{N}_{\mathbb{Z}}(\mathcal{M}_q(\hat{D}),\sigma^2)}[X=x]^{1-\rho} \\
&= \sum_{x\in\mathbb{Z}} \left(\frac{e^{-(x-\mathcal{M}_q(D))^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{\frac{-(y-\mathcal{M}(D))^2}{2\sigma^2}}}\right)^\rho \left(\frac{e^{-(x-\mathcal{M}_q(\hat{D}))^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{\frac{-(y-\mathcal{M}_q(\hat{D}))^2}{2\sigma^2}}}\right)^{1-\rho} \\
&= \frac{\sum_{x\in\mathbb{Z}} e^{(-x^2+2\rho(\mathcal{M}_q(D)-\mathcal{M}_q(\hat{D}))x-\rho(\mathcal{M}_q(D)-\mathcal{M}_q(\hat{D}))^2)/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-y^2/2\sigma^2}} \\
&= e^{\rho(\rho-1)\Delta^2/2\sigma^2} \frac{\sum_{x\in\mathbb{Z}} e^{-(x-\rho\Delta)^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-y^2/2\sigma^2}} \\
&\leq e^{\rho(\rho-1)\Delta^2/2\sigma^2}.
\end{aligned}$$

Thus, we have $\mathscr{D}_\rho\big(\mathcal{M}(D)||\mathcal{M}(\hat{D}))\big) \leq \rho\Delta^2/(2\sigma^2)$. According to Definition 6, $\mathcal{M}$ satisfies $(\rho, \rho\Delta^2/(2\sigma^2))$-RDP. □

Proof of Corollary 1

Proof. We have

$$
\begin{aligned}
&e^{(\rho-1)\mathscr{D}_\rho\left(\mathcal{M}(D)||\mathcal{M}(\hat{D})\right)} \\
&= e^{(\rho-1)\mathscr{D}_\rho\left(\mathcal{N}_{\eta\mathbb{Z}}(\mathcal{M}_q(D),\sigma^2)||\mathcal{N}_{\eta\mathbb{Z}}(\mathcal{M}_q(\hat{D}),\sigma^2)\right)} \\
&= \sum_{x\in\eta\mathbb{Z}} \mathop{\mathbb{P}}_{X\sim\mathcal{N}_{\eta\mathbb{Z}}(\mathcal{M}_q(D),\sigma^2)}[X=x]^\rho \mathop{\mathbb{P}}_{X\sim\mathcal{N}_{\eta\mathbb{Z}}(\mathcal{M}_q(\hat{D}),\sigma^2)}[X=x]^{1-\rho} \\
&= \sum_{x\in\eta\mathbb{Z}} \mathop{\mathbb{P}}_{X\sim\mathcal{N}_{\mathbb{Z}}(\mathcal{M}_q(D)/\eta,\sigma^2/\eta)}[X=\frac{x}{\eta}]^\rho \mathop{\mathbb{P}}_{X\sim\mathcal{N}_{\mathbb{Z}}(\mathcal{M}_q(\hat{D})/\eta,\sigma^2/\eta)}[X=\frac{x}{\eta}]^{1-\rho} \\
&= \sum_{x\in\eta\mathbb{Z}} \left(\frac{e^{-(x-\mathcal{M}(D))^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{\frac{-(y-\mathcal{M}_q(D)/\eta)^2}{2\sigma^2/\eta^2}}}\right)^\rho \left(\frac{e^{-(x-\mathcal{M}_q(\hat{D}))^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{\frac{-(y-\mathcal{M}_q(\hat{D})/\eta)^2}{2\sigma^2/\eta^2}}}\right)^{1-\rho} \\
&= \sum_{x\in\eta\mathbb{Z}} \left(\frac{e^{-(x-\mathcal{M}_q(D))^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{\frac{-(\eta y-\mathcal{M}_q(D))^2}{2\sigma^2}}}\right)^\rho \left(\frac{e^{-(x-\mathcal{M}_q(\hat{D}))^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{\frac{-(\eta y-\mathcal{M}_q(\hat{D}))^2}{2\sigma^2}}}\right)^{1-\rho} \\
&= \frac{\sum_{x\in\eta\mathbb{Z}} e^{(-x^2+2\rho(\mathcal{M}_q(D)-\mathcal{M}_q(\hat{D}))x-\rho(\mathcal{M}_q(D)-\mathcal{M}_q(\hat{D}))^2)/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-y^2/2\sigma^2}} \\
&= e^{\rho(\rho-1)\Delta^2/2\sigma^2} \frac{\sum_{x\in\eta\mathbb{Z}} e^{-(x-\rho\Delta)^2 2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-\eta^2 y^2/2\sigma^2}} \\
&= e^{\rho(\rho-1)\Delta^2/2\sigma^2} \frac{\sum_{y\in\mathbb{Z}} e^{-(\eta y-\rho\Delta)^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-\eta^2 y^2/2\sigma^2}} \\
&\leq e^{\rho(\rho-1)\Delta^2/2\sigma^2},
\end{aligned}
$$

where the last inequality is from Lemma 6 in [73]. Then, we have $\mathscr{D}_\rho\big(\mathcal{M}(D)||\mathcal{M}(\hat{D}))\big) \leq \rho\Delta^2/(2\sigma^2)$. According to Definition 6, $\mathcal{M}$ satisfies $(\rho, \rho\Delta^2/(2\sigma^2))$-RDP.

□

Proof of Theorem 15

Lemma 19. *The sensitivity of quantized local model $Q(\boldsymbol{x}_{i,t})$, denoted by $\Delta_{i,t}$, is $\frac{2K\alpha\varepsilon}{|\mathcal{S}_{i,t-1}|} + 2\eta\sqrt{p}$.*

Proof. Assume neighboring mini-batches $\mathcal{S}_{i,t-1}$ and $\mathcal{S}'_{i,t-1}$ differ by one samples θ_s and θ'_s.

Then, by the definition of sensitivity, we have

$$\begin{aligned}
\Delta_{i,t} &= \sup_{\mathcal{S}_{i,t-1}\sim\mathcal{S}'_{i,t-1}} \|Q(\boldsymbol{x}_{i,t}(\mathcal{S}_{i,t-1})) - Q(\boldsymbol{x}_{i,t}(\mathcal{S}'_{i,t-1}))\| \\
&\leq \sup_{\mathcal{S}_{i,t-1}\sim\mathcal{S}'_{i,t-1}} \|\boldsymbol{x}_{i,t}(\mathcal{S}_{i,t-1}) + \eta I_p - \boldsymbol{x}_{i,t}(\mathcal{S}'_{i,t-1}) - \eta I_p)\| \\
&\leq \sup_{\mathcal{S}_{i,t-1}\sim\mathcal{S}'_{i,t-1}} \|\boldsymbol{x}_{i,t}(\mathcal{S}_{i,t-1}) - \boldsymbol{x}_{i,t}(\mathcal{S}'_{i,t-1})\| + 2\eta\sqrt{p} \\
&\leq \sup_{\mathcal{S}_{i,t-1}\sim\mathcal{S}'_{i,t-1}} \alpha\varepsilon\|\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}, \mathcal{S}_{i,t-1}) - \widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}, \mathcal{S}'_{i,t-1})\| + 2\eta\sqrt{p} \\
&\leq \frac{\alpha\varepsilon}{|\mathcal{S}_{i,t-1}|}\|\nabla\ell(\boldsymbol{x}_{i,t}; \theta_s) - \nabla\ell(\boldsymbol{x}_{i,t}; \theta'_s)\| + 2\eta\sqrt{p} \\
&\leq \frac{2K\alpha\varepsilon}{|\mathcal{S}_{i,t-1}|} + 2\eta\sqrt{p}.
\end{aligned}$$

□

Proof. According to Lemma 19, we have the sensitivity $\Delta_{i,t}$ of $Q(\boldsymbol{x}_{i,t}(\mathcal{S}_{i,t-1}))$ with respect to the subsample dataset $\mathcal{S}_{i,t-1}$, is $\frac{2K\alpha\varepsilon}{|\mathcal{S}_{i,t-1}|} + 2\eta\sqrt{p}$. By Corollary 1, $Q(\boldsymbol{x}_{i,t})$ satisfies $(\rho, \epsilon_{i,t}(\rho))$-RDP with $\epsilon_{i,t}(\rho) = \frac{2\rho}{\sigma^2}(\frac{\alpha\varepsilon}{|\mathcal{S}_{i,t-1}|} + \frac{\eta\sqrt{p}}{K})^2$. Since $\mathcal{S}_{i,t}$ is a randomized subsample of D_i, by Lemma 8 and its approximate version in [90], for each agent i, each iteration of Algorithm 5 preserves $(\rho, \epsilon'_{i,t}(\rho))$-RDP with respect to D_i, i.e., $\epsilon'_{i,t}(\rho) \approx \frac{8\rho^2}{\sigma^2 m^2}(\alpha\varepsilon + \frac{\eta\sqrt{p}}{K}|\mathcal{S}_{i,t-1}|)^2$. Since the algorithms has run T iterations, according to Lemma 6 and parallel composition, `Q-DPSGD-2` is $(\rho, \epsilon(\rho))$-RDP with $\epsilon(\rho) = \max_i \sum_{t=0}^{T-1} \epsilon'_{i,t}(\rho)$. Moreover, `Q-DPSGD-1` is also (ϵ, δ)-DP with $\epsilon = \frac{\epsilon(\rho)+\log(1/\delta)}{\rho-1}$.

□

Proposition 2 (Variance of Discrete Gaussian [73])**.** *For all $\sigma \in \mathbb{R}$ with $\sigma > 0$, we have $Var[\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)] \leq \sigma^2\left(1 - \frac{4\pi^2\sigma^2}{e^{4\pi^2\sigma^2}-1}\right)$ and $Var[\mathcal{N}_{\mathbb{Z}}(0, \sigma^2)] \geq \frac{1}{e^{1/\sigma^2}-1}$.*

Corollary 2 (Variance of Discrete Gaussian with Arbitrary Precision)**.** *For all $\sigma \in \mathbb{R}$ with $\sigma > 0$, and $\eta > 0$, $Var[\mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2)] \leq \sigma^2/\eta^2\left(1 - \frac{4\pi^2\sigma^2/\eta^2}{e^{4\pi^2\sigma^2/\eta^2}-1}\right) \leq \sigma^2/\eta^2$ and $Var[\mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2)] \geq \frac{1}{e^{\eta^2/\sigma^2}-1}$.*

Proof of Theorem 16

Next lemma characterizes the deviation of the models generated by the Q-DPSGD-2 method at iteration T, that is $\boldsymbol{x}_T = [\boldsymbol{x}_{1,T}; \cdots; \boldsymbol{x}_{n,T}]$ from the optimizer of the penalty function, i.e. $\boldsymbol{x}_\alpha^*$.

Lemma 20. *Suppose Assumptions 1–5 hold. Then, the expected deviation of the output of Q-DPSGD-2 from the solution to Problem* (29) *is upper bounded by*

$$\mathbb{E}\left[\|\boldsymbol{x}_T - \boldsymbol{x}_\alpha^*\|^2\right] \leq \mathcal{O}\left(\frac{n(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{\mu}\|W - W_D\|^2 \frac{1}{T^{\widetilde{\boldsymbol{\delta}}}}\right) + \mathcal{O}\left(\frac{n\gamma^2}{\mu}\left(\frac{\mathbb{E}[1/V]}{T_d} + \frac{1}{m}\right)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}\right) \tag{66}$$

for $\varepsilon = T^{-3\widetilde{\boldsymbol{\delta}}/2}$, $\alpha = T^{-\widetilde{\boldsymbol{\delta}}/2}$, *any* $\widetilde{\boldsymbol{\delta}} \in (0, 1/2)$ *and* $T \geq T^{\mathsf{c}}_{\mathsf{min1}}$, *where*

$$T^{\mathsf{c}}_{\mathsf{min1}} \coloneqq max\left\{\left\lceil\left(\frac{(2+K)^2}{\mu}\right)^{\frac{1}{\widetilde{\boldsymbol{\delta}}}}\right\rceil, \left\lceil e^{\frac{1}{1-2\widetilde{\boldsymbol{\delta}}}}\right\rceil, \left\lceil \mu^{\frac{1}{2\widetilde{\boldsymbol{\delta}}}}\right\rceil\right\}. \tag{67}$$

Proof of Lemma 20. First note that the gradient of the penalty function h_α defined in (29) is

$$\nabla h_\alpha(\boldsymbol{x}_t) = (\boldsymbol{I} - \boldsymbol{W})\,\boldsymbol{x}_t + \alpha n \nabla \hat{F}(\boldsymbol{x}_t), \tag{68}$$

where $\boldsymbol{x}_t = [\boldsymbol{x}_{1,t}; \cdots; \boldsymbol{x}_{n,t}]$ denotes the concatenation of models at iteration t. Now consider the stochastic gradient function for h_α as

$$\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) = (\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t + \zeta_t) + (\boldsymbol{I} - \boldsymbol{W}_D)\,\boldsymbol{x}_t + \alpha n \widetilde{\nabla} \hat{F}(\boldsymbol{x}_t), \tag{69}$$

where $\widetilde{\nabla} F(\boldsymbol{x}_t) = \left[\frac{1}{n}\widetilde{\nabla} f_1(\boldsymbol{x}_{1,t}); \cdots; \frac{1}{n}\widetilde{\nabla} f_n(\boldsymbol{x}_{n,t})\right]$, and $\zeta_t = [\zeta_{1,t}; \cdots; \zeta_{n,t}]$ with $\zeta_{i,t} \sim \mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2 K^2 I_p)$ is the concatenation of noise vectors at iteration t. Moreover, $\boldsymbol{z}_t = [\boldsymbol{z}_{1,t}; \cdots; \boldsymbol{z}_{n,t}]$ as the concatenation of the quantized variant of the local updates $\boldsymbol{x}_t$.

We let $\mathcal{F}^t$ denote a sigma algebra that measures the history of the system up until time t. According to Assumptions 2 and 4, the stochastic gradient defined above is unbiased,

that is,

$$\begin{aligned}\mathbb{E}\left[\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)|\mathcal{F}^t\right] &= (\boldsymbol{W}_D - \boldsymbol{W})\,\mathbb{E}\left[\boldsymbol{z}_t + \zeta_t|\mathcal{F}^t\right] + (\boldsymbol{I} - \boldsymbol{W}_D)\,\boldsymbol{x}_t + \alpha n \mathbb{E}\left[\widetilde{\nabla}\hat{F}(\boldsymbol{x}_t)|\mathcal{F}^t\right] \\ &= (\boldsymbol{I} - \boldsymbol{W})\,\boldsymbol{x}_t + \alpha n \nabla F(\boldsymbol{x}_t) \\ &= \nabla h_\alpha(\boldsymbol{x}_t).\end{aligned}$$

We then can also write the update rule of `Q-DPSGD-2` method as

$$\begin{aligned}\boldsymbol{x}_{t+1} &= \boldsymbol{x}_t - \varepsilon\left((\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t + \zeta_t) + (\boldsymbol{I} - \boldsymbol{W}_D)\,\boldsymbol{x}_t + \alpha n \widetilde{\nabla}(F(\boldsymbol{x}_t)\right) \\ &= \boldsymbol{x}_t - \varepsilon\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t),\end{aligned} \tag{70}$$

which also represents an iteration of the Stochastic Gradient Descent (SGD) algorithm with step-size ε in order to minimize the penalty function $h_\alpha(\boldsymbol{x})$ over $\boldsymbol{x} \in \mathbb{R}^{np}$. We can bound the deviation of the iteration generated by `Q-DPSGD-2` from the optimizer $\boldsymbol{x}_\alpha^*$ as

$$\begin{aligned}\mathbb{E}\left[\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_\alpha^*\|^2\,|\mathcal{F}^t\right] &= \mathbb{E}\left[\left\|\boldsymbol{x}_t - \varepsilon\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t) - \boldsymbol{x}_\alpha^*\right\|^2|\mathcal{F}^t\right] \\ &= \|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 - 2\varepsilon\left\langle \boldsymbol{x}_t - \boldsymbol{x}_\alpha^*, \mathbb{E}\left[\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)|\mathcal{F}^t\right]\right\rangle + \varepsilon^2\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right] \\ &= \|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 - 2\varepsilon\left\langle \boldsymbol{x}_t - \boldsymbol{x}_\alpha^*, \nabla h_\alpha(\boldsymbol{x}_t)\right\rangle + \varepsilon^2\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right] \\ &\leq (1 - 2\mu_\alpha\varepsilon)\,\|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 + \varepsilon^2\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right],\end{aligned} \tag{71}$$

where we used the fact that the penalty function h_α is strongly convex with parameter $\mu_\alpha \coloneqq \alpha\mu$. Moreover, we can bound the second term in RHS of (71) as

$$\begin{aligned}&\mathbb{E}\left[\left\|\widetilde{\nabla} h_\alpha(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right] \\ &= \mathbb{E}\left[\left\|(\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t + \zeta_t) + (\boldsymbol{I} - \boldsymbol{W}_D)\,\boldsymbol{x}_t + \alpha n\widetilde{\nabla} F(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right] \\ &= \mathbb{E}\left[\left\|(\boldsymbol{I} - \boldsymbol{W})\,\boldsymbol{x}_t + \alpha n\nabla F(\boldsymbol{x}_t) + (\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t - \boldsymbol{x}_t + \zeta_t) + \alpha n\widetilde{\nabla}\hat{F}(\boldsymbol{x}_t) - \alpha n\nabla\hat{F}(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right] \\ &= \|\nabla h_\alpha(\boldsymbol{x}_t)\|^2 + \mathbb{E}\left[\|(\boldsymbol{W}_D - \boldsymbol{W})\,(\boldsymbol{z}_t - \boldsymbol{x}_t + \zeta_t)\|^2|\mathcal{F}^t\right] + \alpha^2 n^2\mathbb{E}\left[\left\|\widetilde{\nabla}\hat{F}(\boldsymbol{x}_t) - \nabla\hat{F}(\boldsymbol{x}_t)\right\|^2|\mathcal{F}^t\right] \\ &\leq K_\alpha^2\,\|\boldsymbol{x}_t - \boldsymbol{x}_\alpha^*\|^2 + (n\widetilde{\sigma}^2 + \frac{npK^2\sigma^2}{\eta^2})\,\|W - W_D\|^2 + \alpha^2 n\gamma_2^2.\end{aligned} \tag{72}$$

To derive (72), we used the facts that h_α is smooth with parameter $K_\alpha \coloneqq 1-\lambda_n(W)+\alpha K$; the quantizer is unbiased with variance $\leq \widetilde{\sigma}^2$ (Assumption 2); stochastic gradients of the loss function are unbiased and variance-bounded (Assumption 4 and Lemma 15). Plugging (72) in (71) yields

$$\mathbb{E}\left[\|\boldsymbol{x}_{t+1}-\boldsymbol{x}_\alpha^*\|^2\,|\mathcal{F}^t\right] \leq \left(1-2\mu_\alpha\varepsilon+\varepsilon^2K_\alpha^2\right)\|\boldsymbol{x}_t-\boldsymbol{x}_\alpha^*\|^2+\varepsilon^2 n(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})\|W-W_D\|^2 + \alpha^2\varepsilon^2 n\gamma_2^2. \tag{73}$$

To ease the notation, let $e_t := \mathbb{E}[\,\|\boldsymbol{x}_t-\boldsymbol{x}_\alpha^*\|^2]$ denote the expected deviation of the models at iteration t, i.e. $\boldsymbol{x}_t$ from the optimizer $\boldsymbol{x}_\alpha^*$ with respect to all the randomnesses from iteration $t=0$. Therefore, we have

$$\begin{aligned} e_{t+1} &\leq \left(1-2\mu_\alpha\varepsilon+\varepsilon^2K_\alpha^2\right)e_t+\varepsilon^2 n(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})\|W-W_D\|^2+\alpha^2\varepsilon^2 n\gamma_2^2 \\ &= \left(1-\varepsilon(2\mu_\alpha-\varepsilon K_\alpha^2)\right)e_t+\varepsilon^2 n(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})\|W-W_D\|^2+\alpha^2\varepsilon^2 n\gamma_2^2. \end{aligned} \tag{74}$$

For any $T \geq T_{\mathsf{min1}}^{\mathsf{c}}$ and the proposed pick $\varepsilon = T^{-3\widetilde{\boldsymbol{\delta}}/2}$, we have $T^{\widetilde{\boldsymbol{\delta}}} \geq (T_{\mathsf{min1}}^{\mathsf{c}})^{\widetilde{\boldsymbol{\delta}}} \geq \frac{(2+K)^2}{\mu}$ and therefore

$$\begin{aligned} \varepsilon &= \frac{1}{T^{3\widetilde{\boldsymbol{\delta}}/2}} \\ &\leq \frac{\mu}{(2+K)^2}\cdot\frac{1}{T^{\widetilde{\boldsymbol{\delta}}/2}} \\ &\leq \frac{\mu_\alpha}{(2+\alpha K)^2} \\ &\leq \frac{\mu_\alpha}{K_\alpha^2}. \end{aligned}$$

Hence, we can further bound (74) as

$$
\begin{aligned}
e_{t+1} &\leq \left(1-\varepsilon(2\mu_\alpha - \varepsilon K_\alpha^2)\right) e_t + \varepsilon^2 n(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) \left\|W - W_D\right\|^2 + \alpha^2\varepsilon^2 n\gamma_2^2 \\
&\leq (1-\mu_\alpha\varepsilon)\, e_t + \varepsilon^2 n(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) \left\|W - W_D\right\|^2 + \alpha^2\varepsilon^2 n\gamma_2^2 \\
&= \left(1 - \frac{\mu}{T^{2\widetilde{\boldsymbol{\delta}}}}\right) e_t + \frac{n(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) \left\|W - W_D\right\|^2}{T^{3\widetilde{\boldsymbol{\delta}}}} + \frac{n\gamma_2^2}{T^{4\widetilde{\boldsymbol{\delta}}}}.
\end{aligned}
$$

Now, we let $(a,b,c) = (\mu, n(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})\widetilde{\sigma}^2 \left\|W - W_D\right\|^2, n\gamma_2^2)$ and employ Lemma 18, which yields

$$
\begin{aligned}
e_T &= \mathbb{E}\left[\left\|\boldsymbol{x}_T - \boldsymbol{x}_\alpha^*\right\|^2\right] \\
&\leq \mathcal{O}\left(\frac{b/a}{T^{\widetilde{\boldsymbol{\delta}}}}\right) + \mathcal{O}\left(\frac{c/a}{T^{2\widetilde{\boldsymbol{\delta}}}}\right) \\
&= \mathcal{O}\left(\frac{n(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{\mu} \left\|W - W_D\right\|^2 \frac{1}{T^{\widetilde{\boldsymbol{\delta}}}}\right) + \mathcal{O}\left(\frac{n\gamma^2}{\mu}\left(\frac{\mathbb{E}[1/V]}{T_d} + \frac{1}{m}\right)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}\right).
\end{aligned}
$$

□

Combining Lemmas 20 and 17, we can now plug them in Theorem 12 and write for $T \geq T_{\mathsf{min}}^{\mathsf{c}} \coloneqq \max\{T_{\mathsf{min1}}^{\mathsf{c}}, T_{\mathsf{min2}}^{\mathsf{c}}\}$

$$
\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left\|\boldsymbol{x}_{i,T} - \boldsymbol{x}^*\right\|^2\right] &= \frac{1}{n}\mathbb{E}\left[\left\|\boldsymbol{x}_T - \widetilde{\boldsymbol{x}}^*\right\|^2\right] \\
&= \frac{1}{n}\mathbb{E}\left[\left\|\boldsymbol{x}_T - \boldsymbol{x}_\alpha^* + \boldsymbol{x}_\alpha^* - \widetilde{\boldsymbol{x}}^*\right\|^2\right] \\
&\leq \frac{2}{n}\mathbb{E}\left[\left\|\boldsymbol{x}_T - \boldsymbol{x}_\alpha^*\right\|^2\right] + \frac{2}{n}\left\|\boldsymbol{x}_\alpha^* - \widetilde{\boldsymbol{x}}^*\right\|^2 \\
&\leq \mathcal{O}\left(\frac{E^2(K/\mu)^2}{(1-\beta)^2} + \frac{\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}}{\mu}\right)\frac{1}{T^{\widetilde{\boldsymbol{\delta}}}} \\
&\quad + \mathcal{O}\left(\frac{\gamma^2}{\mu}\max\left\{\frac{\mathbb{E}[1/V]}{T_d}, \frac{1}{m}\right\}\right)\frac{1}{T^{2\widetilde{\boldsymbol{\delta}}}}.
\end{aligned}
$$

Proof of Theorem 17

We then characterize the convergence rate of `Q-DPSGD-2` for non-convex and smooth objectives. We are interested in finding a set of local models which satisfy first-order optimality condition approximately, while the models are close to each other and satisfy the consensus condition up to a small error. To be more precise, we are interested in finding a set of local models $\{\boldsymbol{x}_1^*, \dots, \boldsymbol{x}_n^*\}$ where their average $\overline{\boldsymbol{x}}^* \coloneqq \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i^*$ (approximately) satisfy first-order optimality condition, i.e., $\mathbb{E}\left\|\nabla f\left(\overline{\boldsymbol{x}}^*\right)\right\|^2 \leq \nu$, while the iterates are close to their average, i.e., $\mathbb{E}\|\overline{\boldsymbol{x}}^* - \boldsymbol{x}_i^*\|^2 \leq \rho$.

To ease the notation, we agree in this section on the following shorthand notations for $t = 0, 1, 2, \cdots$,

$$
\begin{aligned}
X_t &= [\boldsymbol{x}_{1,t} \ \cdots \ \boldsymbol{x}_{n,t}] \in \mathbb{R}^{p\times n},\\
Z_t &= [\boldsymbol{z}_{1,t} \ \cdots \ \boldsymbol{z}_{n,t}] \in \mathbb{R}^{p\times n},\\
\overline{\boldsymbol{x}}_t &= \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_{i,t} \in \mathbb{R}^p,\\
\overline{X}_t &= [\overline{\boldsymbol{x}}_t \ \cdots \ \overline{\boldsymbol{x}}_t] \in \mathbb{R}^{p\times n},\\
\widetilde{\partial} f(X_t) &= \left[\widetilde{\nabla} f_1(\boldsymbol{x}_{1,t}) \ \cdots \ \widetilde{\nabla} f_n(\boldsymbol{x}_{n,t})\right] \in \mathbb{R}^{p\times n},\\
\partial f(X_t) &= [\nabla f_1(\boldsymbol{x}_{1,t}) \ \cdots \ \nabla f_n(\boldsymbol{x}_{n,t})] \in \mathbb{R}^{p\times n},\\
\text{and} \ \ \boldsymbol{\zeta}_t &= [\zeta_{1,t} \ \cdots \ \zeta_{n,t}] \in \mathbb{R}^{p\times n} \quad \text{with} \quad \zeta_{i,t} \sim \mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2 K^2 I_p)).
\end{aligned}
$$

As stated before, we can write the update rule of the proposed `Q-DPSGD-2` in the matrix form as

$$
X_{t+1} = X_t\left((1-\varepsilon)I + \varepsilon W\right) + \varepsilon(Z_t - X_t + \boldsymbol{\zeta}_t)(W - W_D) - \alpha\varepsilon\widetilde{\partial} f(X_t). \tag{75}
$$

Let us denote $W_\varepsilon = (1-\varepsilon)I + \varepsilon W$ and write (75) as

$$
X_{t+1} = X_t W_\varepsilon + \varepsilon(Z_t - X_t + \boldsymbol{\zeta}_t)(W - W_D) - \alpha\varepsilon\widetilde{\partial} f(X_t). \tag{76}
$$

We start the convergence analysis by using the smoothness property of the objectives and write

$$\begin{aligned}
\mathbb{E}f\left(\frac{X_{t+1}\mathbf{1}_n}{n}\right) &= \mathbb{E}f\left(\frac{X_tW_\varepsilon\mathbf{1}_n}{n} + \frac{\varepsilon(Z_t - X_t + \boldsymbol{\zeta}_t)(W - W_D)\mathbf{1}_n}{n} - \frac{\alpha\varepsilon\widetilde{\partial}f(X_t)\mathbf{1}_n}{n}\right) \\
&\overset{\text{Assumption 3}}{\leq} \mathbb{E}f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \alpha\varepsilon\mathbb{E}\left\langle\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right), \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\rangle \\
&\quad + \frac{\varepsilon^2K}{2}\mathbb{E}\left\|\frac{(Z_t - X_t + \boldsymbol{\zeta}_t)(W - W_D)\mathbf{1}_n}{n} - \alpha\frac{\widetilde{\partial}f(X_t)\mathbf{1}_n}{n}\right\|^2. \qquad (77)
\end{aligned}$$

We specifically used the following equivalent form of the smoothness (Assumption 3) for every local and hence the global objective is bounded by

$$f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}) + \langle\nabla f_i(\boldsymbol{x}), \boldsymbol{x}_1 - \boldsymbol{x}\rangle + \frac{K}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}\|^2, \quad \text{for all } i \in [n], \boldsymbol{x}, \boldsymbol{x}_1 \in \mathbb{R}^p.$$

Also, we used the simple fact in Assumption 1 as

$$W_\varepsilon\mathbf{1}_n = ((1-\varepsilon)I + \varepsilon W)\mathbf{1}_n = (1-\varepsilon)\mathbf{1}_n + \varepsilon W\mathbf{1}_n = \mathbf{1}_n.$$

Now let us bound the term in (77) as

$$\begin{aligned}
&\mathbb{E}\left\|\frac{(Z_t - X_t + \boldsymbol{\zeta}_t)(W - W_D)\mathbf{1}_n}{n} - \alpha\frac{\widetilde{\partial}f(X_t)\mathbf{1}_n}{n}\right\|^2 \\
&\leq \mathbb{E}\left\|\frac{(Z_t - X_t + \boldsymbol{\zeta}_t)(W - W_D)\mathbf{1}_n}{n}\right\|^2 + \mathbb{E}\left\|\alpha\frac{\widetilde{\partial}f(X_t)\mathbf{1}_n}{n}\right\|^2 \\
&= \frac{1}{n^2}\sum_{i=1}^n(1-w_{ii})^2\mathbb{E}\|\boldsymbol{z}_{i,t} - \boldsymbol{x}_{i,t} + \zeta_{i,t}\|^2 + \alpha^2\mathbb{E}\left\|\frac{\widetilde{\partial}f(X_t)\mathbf{1}_n}{n}\right\|^2 \\
&\leq \frac{\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}}{n} + \alpha^2\mathbb{E}\left\|\frac{\widetilde{\partial}f(X_t)\mathbf{1}_n}{n}\right\|^2 \\
&\leq \frac{\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}}{n} + \frac{\gamma_2^2}{n} + \mathbb{E}\left\|\frac{\sum_{i=1}^n\nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2, \qquad (78)
\end{aligned}$$

where we used Assumption 2 to derive the first term in (46) and the second term is from (47).

Plugging (78) in (77) yields

$$\begin{aligned}
\mathbb{E}f\left(\frac{X_{t+1}\mathbf{1}_n}{n}\right) &\leq \mathbb{E}f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \alpha\varepsilon\mathbb{E}\left\langle \nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right), \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\rangle \\
&\quad + \frac{\varepsilon^2 K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) + \frac{\alpha^2\varepsilon^2 K}{2n}\gamma_2^2 + \frac{\alpha^2\varepsilon^2 K}{2}\mathbb{E}\left\|\frac{\sum_{i=1}^n \nabla f_i(\boldsymbol{x}_{i,t})}{n}\right\|^2 \\
&= \mathbb{E}f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\alpha\varepsilon - \alpha^2\varepsilon^2 K}{2}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 - \frac{\alpha\varepsilon}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\quad + \frac{\varepsilon^2 K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) + \frac{\alpha^2\varepsilon^2 K}{2n}\gamma_2^2 + \frac{\alpha\varepsilon}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 \\
&\leq \mathbb{E}f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\alpha\varepsilon - \alpha^2\varepsilon^2 K}{2}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 - \frac{\alpha\varepsilon}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\quad + \frac{\varepsilon^2 K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) + \frac{\alpha^2\varepsilon^2 K}{2n}\gamma_2^2 + \frac{K^2}{n}\sum_{i=1}^n \underbrace{\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n} - \boldsymbol{x}_{i,t}\right\|^2}_{Q_{i,t}}, \qquad (79)
\end{aligned}$$

where we used the identity $2\langle \mathbf{a}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a}-\mathbf{b}\|^2$.

Let us define $Q_{i,t} \coloneqq \mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n} - \boldsymbol{x}_{i,t}\right\|^2$ and $M_t \coloneqq \frac{1}{n}\sum_{i=1}^n Q_{i,t} = \frac{1}{n}\sum_{i=1}^n \mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n} - \boldsymbol{x}_{i,t}\right\|^2$.

Here, $Q_{i,t}$ captures the deviation of the model at node i from the average model at iteration t and M_t aggregates them to measure the average total consensus error. To bound M_t, we need to evaluate the following recursive expressions, i.e.,

$$\begin{aligned}
X_t &= X_{t-1}W_\varepsilon + \varepsilon(Z_{t-1} - X_{t-1} + \boldsymbol{\zeta}_{t-1})(W - W_D) - \alpha\varepsilon\widetilde{\partial} f(X_{t-1}) \\
&= X_0 W_\varepsilon^t + \varepsilon\sum_{s=0}^{t-1}(Z_s - X_s + \boldsymbol{\zeta}_s)(W - W_D)W_\varepsilon^{t-s-1} - \alpha\varepsilon\sum_{s=0}^{t-1}\widetilde{\partial} f(X_s)W_\varepsilon^{t-s-1}. \qquad (80)
\end{aligned}$$

Now, using (80) we can write

$$
\begin{aligned}
M_t &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n}-\boldsymbol{x}_{i,t}\right\|^2 \\
&= \frac{1}{n}\mathbb{E}\left\|\overline{X}_t - X_t\right\|_F^2 \\
&= \frac{1}{n}\mathbb{E}\left\|X_t\frac{\mathbf{1}\mathbf{1}^\top}{n}-X_t\right\|_F^2 \\
&= \frac{1}{n}\mathbb{E}\Bigg\|X_0\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^t\right)+\varepsilon\sum_{s=0}^{t-1}(Z_s-X_s+\boldsymbol{\zeta}_s)(W-W_D)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right) \\
&\quad -\alpha\varepsilon\sum_{s=0}^{t-1}\widetilde{\partial}f(X_s)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\Bigg\|_F^2 \\
&= \frac{(\alpha\varepsilon)^2}{n}\underbrace{\mathbb{E}\left\|\sum_{s=0}^{t-1}\widetilde{\partial}f(X_s)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2}_{T_2} \\
&\quad +\frac{\varepsilon^2}{n}\underbrace{\mathbb{E}\left\|\sum_{s=0}^{t-1}(Z_s-X_s+\boldsymbol{\zeta}_s)(W-W_D)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2}_{T_3},
\end{aligned}
$$

where we used the fact that quantiziations, stochastic gradients, and differential privacy noise are statistically independent and $X_0 = 0$.

Moreover, the term T_3 can be bounded as

$$
\begin{aligned}
T_3 &= \mathbb{E}\left\|\sum_{s=0}^{t-1}(Z_s-X_s+\boldsymbol{\zeta}_s)(W-W_D)\left(\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right)\right\|_F^2 \\
&\le \mathbb{E}\sum_{s=0}^{t-1}\left\|Z_s-X_s+\boldsymbol{\zeta}_s\right\|_F^2\left\|W-W_D\right\|^2\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n}-W_\varepsilon^{t-s-1}\right\|^2 \\
&\le \frac{4n(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2},
\end{aligned}
$$

where we used the fact that $\|W - W_D\| \le 2$.

Now we use the bounds derived for T_2 and T_3 and T_4 to bound the consensus error M_t as

$$
\begin{aligned}
M_t &\leq \frac{\alpha^2\varepsilon^2}{n}T_2 + \frac{\varepsilon^2}{n}T_3 \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{6\alpha^2\varepsilon^2K^2}{n}\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right\|^2 \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left\|\frac{\mathbf{1}\mathbf{1}^\top}{n} - W_\varepsilon^{t-s-1}\right\|^2 \\
&\quad + \frac{12\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\left(3K^2\sum_{i=1}^{n}Q_{i,s} + 3\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\right)\frac{\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} \\
&\quad + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2} \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2} + \frac{6\alpha^2\varepsilon^2K^2}{n}\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\beta_\varepsilon^{2(t-s-1)} \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\beta_\varepsilon^{2(t-s-1)} \\
&\quad + \frac{12\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\left(3K^2\sum_{i=1}^{n}Q_{i,s} + 3\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\right)\frac{\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2} \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\beta_\varepsilon^{2(t-s-1)} + \frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}\right) \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}K^2\sum_{s=0}^{t-1}\sum_{i=1}^{n}Q_{i,s}\left(\frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} + \beta_\varepsilon^{2(t-s-1)}\right). \qquad (81)
\end{aligned}
$$

As we defined earlier, we have $M_s = \frac{1}{n}\sum_{i=1}^{n}Q_{i,s}$ which simplifies (81) to

$$
\begin{aligned}
M_t &\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2} + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2} + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2} \\
&\quad + \frac{6\alpha^2\varepsilon^2}{n}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\beta_\varepsilon^{2(t-s-1)} + \frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}\right) \\
&\quad + 6\alpha^2\varepsilon^2K^2\sum_{s=0}^{t-1}M_s\left(\frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} + \beta_\varepsilon^{2(t-s-1)}\right). \qquad (82)
\end{aligned}
$$

Now we can sum (82) over $t = 0, 1, \cdots, T-1$, which yields

$$\begin{aligned}
\sum_{t=0}^{T-1} M_t \leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2}T \\
& + \frac{6\alpha^2\varepsilon^2}{n}\sum_{t=0}^{T-1}\sum_{s=0}^{t-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\beta_\varepsilon^{2(t-s-1)} + \frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon}\right) \\
& + 6\alpha^2\varepsilon^2K^2\sum_{t=0}^{T-1}\sum_{s=0}^{t-1}M_s\left(\frac{2\beta_\varepsilon^{t-s-1}}{1-\beta_\varepsilon} + \beta_\varepsilon^{2(t-s-1)}\right) \\
\leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2}T \\
& + \frac{6\alpha^2\varepsilon^2}{n}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2\left(\sum_{k=0}^{\infty}\beta_\varepsilon^{2k} + \frac{2\sum_{k=0}^{\infty}\beta_\varepsilon^k}{1-\beta_\varepsilon}\right) \\
& + 6\alpha^2\varepsilon^2K^2\sum_{t=0}^{T-1}M_t\left(\frac{2\sum_{k=0}^{\infty}\beta_\varepsilon^k}{1-\beta_\varepsilon} + \sum_{k=0}^{\infty}\beta_\varepsilon^{2k}\right) \\
\leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2}T \\
& + \frac{18\alpha^2\varepsilon^2}{n(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2 + \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}M_t. \qquad (83)
\end{aligned}$$

Note that $\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\mathbf{1}_n^\top\right\|_F^2 = n\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2$, which simplifies (83) as

$$\begin{aligned}
\sum_{t=0}^{T-1} M_t \leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2 \\
& + \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}M_t. \qquad (84)
\end{aligned}$$

Rearranging the terms implies that

$$\begin{aligned}
\left(1 - \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}\right)\sum_{t=0}^{T-1} M_t \leq & \frac{2\alpha^2\varepsilon^2\gamma_2^2}{1-\beta_\varepsilon^2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2}T + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{1-\beta_\varepsilon^2}T \\
& + \frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2. \qquad (85)
\end{aligned}$$

Now define $D_2 \coloneqq 1 - \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2}$ and rewrite (85) as

$$\sum_{t=0}^{T-1} M_t \leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}T + \frac{4\varepsilon^2(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2D_2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2. \tag{86}$$

Note that from definition of T_1 we have $T_1 \leq \frac{K^2}{n}\sum_{i=1}^n Q_{i,t} = K^2M_t$. Now use the above fact in the recursive equation (79) which we started with, that is

$$\mathbb{E}f\left(\frac{X_{t+1}\mathbf{1}_n}{n}\right) \leq \mathbb{E}f\left(\frac{X_t\mathbf{1}_n}{n}\right) - \frac{\alpha\varepsilon - \alpha^2\varepsilon^2K}{2}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 - \frac{\alpha\varepsilon}{2}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 + \frac{\varepsilon^2K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) + \frac{\alpha^2\varepsilon^2K}{2n}\gamma_2^2 + \frac{\alpha\varepsilon K^2}{2}M_t. \tag{87}$$

If we sum (87) over $t = 0, 1, \cdots, T-1$, we get

$$\begin{aligned}
&\frac{\alpha\varepsilon - \alpha^2\varepsilon^2K}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 + \frac{\alpha\varepsilon}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq f(0) - f^* + \frac{\varepsilon^2K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})T + \frac{\alpha^2\varepsilon^2K}{2n}\gamma_2^2T + \frac{\alpha\varepsilon K^2}{2}\sum_{t=0}^{T-1}M_t \\
&\overset{\text{from (86)}}{\leq} f(0) - f^* + \frac{\varepsilon^2K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})T + \frac{\alpha^2\varepsilon^2K}{2n}\gamma_2^2T \\
&\quad + \frac{\alpha\varepsilon K^2}{2}\left\{\frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{(1-\beta_\varepsilon^2)D_2}T\right\} \\
&\quad + \frac{9\alpha^3\varepsilon^3K^2}{(1-\beta_\varepsilon)^2D_2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2.
\end{aligned} \tag{88}$$

We can rearrange the terms in (88) and rewrite it as

$$\begin{aligned}
&\frac{\alpha\varepsilon - \alpha^2\varepsilon^2K}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 + \alpha\varepsilon\left(\frac{1}{2} - \frac{9\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2D_2}\right)\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq f(0) - f^* + \frac{\varepsilon^2K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2})T + \frac{\alpha^2\varepsilon^2K}{2n}\gamma_2^2T \\
&\quad + \frac{\alpha\varepsilon K^2}{2}\left\{\frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}T + \frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}T + \frac{4\varepsilon^2\widetilde{\sigma}^2}{(1-\beta_\varepsilon^2)D_2}T\right\}.
\end{aligned} \tag{89}$$

Now, we define D_1 as $D_1 \coloneqq \frac{1}{2} - \frac{9\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2D_2}$ and replace in (89) which yields

$$\begin{aligned}
&\frac{1}{\alpha\varepsilon T}\left\{\frac{\alpha\varepsilon - \alpha^2\varepsilon^2K}{2}\sum_{t=0}^{T-1}\mathbb{E}\left\|\frac{\partial f(X_t)\mathbf{1}_n}{n}\right\|^2 + \alpha\varepsilon D_1\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2\right\} \\
&\leq \frac{1}{\alpha\varepsilon T}(f(0)-f^*) + \frac{\varepsilon}{\alpha}\frac{K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) + \alpha\varepsilon\frac{K\gamma_2^2}{2n} + \frac{\alpha^2\varepsilon^2}{1-\beta_\varepsilon^2}\frac{K^2\gamma_2^2}{D_2} + \frac{\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2}\frac{9K^2\gamma_1^2}{D_2}.
\end{aligned} \tag{90}$$

To balance the terms in RHS of (90), we need to know how β_ε behaves with ε. As we defined before, $W_\varepsilon = (1-\varepsilon)I + \varepsilon W$. Hence, $\lambda_i(W_\varepsilon) = 1 - \varepsilon + \varepsilon\lambda_i(W)$. Therefore, for $\varepsilon \leq \frac{1}{1-\lambda_n(W)}$, we have

$$\begin{aligned}
\beta_\varepsilon &= \max\left\{|\lambda_2(W_\varepsilon)|, |\lambda_n(W_\varepsilon)|\right\} \\
&= \max\left\{|1-\varepsilon+\varepsilon\lambda_2(W)|, |1-\varepsilon+\varepsilon\lambda_n(W)|\right\} \\
&= \max\left\{1-\varepsilon+\varepsilon\lambda_2(W), 1-\varepsilon+\varepsilon\lambda_n(W)\right\} \\
&= 1 - \varepsilon\left(1-\lambda_2(W)\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
1-\beta_\varepsilon &= \varepsilon\left(1-\lambda_2(W)\right) \geq \varepsilon(1-\beta) \\
\text{and}\quad 1-\beta_\varepsilon^2 &= 2\varepsilon\left(1-\lambda_2(W)\right) - \varepsilon^2\left(1-\lambda_2(W)\right)^2 \geq \varepsilon(1-\beta^2).
\end{aligned}$$

Moreover, if $\alpha\varepsilon \leq \frac{1}{K}$ we have from (90) that

$$\begin{aligned}
\frac{D_1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 &\leq \frac{1}{\alpha\varepsilon T}(f(0)-f^*) + \frac{\varepsilon}{\alpha}\frac{K}{2n}(\widetilde{\sigma}^2 + \frac{pK^2\sigma^2}{\eta^2}) + \alpha\varepsilon\frac{K\gamma_2^2}{2n} \\
&\quad + \frac{\alpha^2\varepsilon}{1-\beta^2}\frac{K^2\gamma_2^2}{D_2} + \frac{\alpha^2}{(1-\beta)^2}\frac{9K^2\gamma_1^2}{D_2} + \frac{\varepsilon}{1-\beta^2}\frac{2K^2\widetilde{\sigma}^2}{D_2}.
\end{aligned} \tag{91}$$

For $\alpha \leq \frac{1-\beta}{6K}$, we have

$$\begin{aligned} D_2 &= 1 - \frac{18\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2} \\ &= 1 - \frac{18\alpha^2\varepsilon^2K^2}{\varepsilon^2(1-\beta)^2} \\ &= 1 - \frac{18\alpha^2K^2}{(1-\beta)^2} \\ &\geq \frac{1}{2}, \end{aligned} \tag{92}$$

and for $\alpha \leq \frac{1-\beta}{6\sqrt{2K}}$, we have

$$\begin{aligned} D_1 &= \frac{1}{2} - \frac{9\alpha^2\varepsilon^2K^2}{(1-\beta_\varepsilon)^2 D_2} \\ &\geq \frac{1}{2} - \frac{18\alpha^2\varepsilon^2K^2}{\varepsilon^2(1-\beta)^2} \\ &= \frac{1}{2} - \frac{18\alpha^2K^2}{(1-\beta)^2} \\ &\geq \frac{1}{4}. \end{aligned}$$

Now, we pick the step-sizes as

$$\alpha = \frac{1}{T^{1/6}}, \tag{93}$$

$$\text{and} \quad \varepsilon = \frac{1}{T^{1/2}}. \tag{94}$$

It is clear that in order to satisfy the conditions mentioned before, that are $\varepsilon \leq \frac{1}{1-\lambda_n(W)}$, $\alpha\varepsilon \leq \frac{1}{K}$ and $\alpha \leq \frac{1-\beta}{6\sqrt{2K}}$, it suffices to pick T as large as

$$T \geq T_{\mathsf{min}}^{\mathsf{nc}} \coloneqq \max\left\{ \left(1-\lambda_n(W)\right)^2, K^{3/2}, \left(\frac{6\sqrt{2}K}{1-\beta}\right)^6 \right\}. \tag{95}$$

For such T we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_t\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq \frac{1}{T^{1/3}}4(f(0)-f^*)+\frac{1}{T^{1/3}}\frac{K}{n}(2\widetilde{\sigma}^2+2\frac{pK^2\sigma^2}{\eta^2})+\frac{1}{T^{2/3}}\frac{2K\gamma_2^2}{n} \\
&\quad+\frac{1}{T^{5/6}}\frac{8K^2\gamma_2^2}{1-\beta^2}+\frac{1}{T^{1/3}}\frac{72K^2\gamma_1^2}{(1-\beta)^2}+\frac{1}{T^{1/2}}\frac{16K^2\widetilde{\sigma}^2}{1-\beta^2} \\
&=\frac{B_1}{T^{1/3}}+\frac{B_2}{T^{1/2}}+\frac{B_3}{T^{2/3}}+\frac{B_4}{T^{5/6}} \\
&=\mathcal{O}\left(\frac{K}{n}(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})+\frac{K^2\gamma^2}{(1-\beta)^2m}\right)\frac{1}{T^{1/3}}+\mathcal{O}\left(\frac{K^2}{1-\beta^2}\sigma^2\right)\frac{1}{T^{1/2}} \\
&\quad+\mathcal{O}\left(K\frac{\gamma^2}{n}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T^{2/3}}+\mathcal{O}\left(\frac{K^2}{1-\beta^2}\gamma^2\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T^{5/6}},
\end{aligned}
\tag{96}
$$

where

$$
\begin{aligned}
B_1 &\coloneqq 4(f(0)-f^*)+\frac{K}{n}(2\widetilde{\sigma}^2+2\frac{pK^2\sigma^2}{\eta^2})+\frac{72K^2\gamma_1^2}{(1-\beta)^2}, \\
B_2 &\coloneqq \frac{16K^2\widetilde{\sigma}^2}{1-\beta^2}, \\
B_3 &\coloneqq \frac{2K\gamma_2^2}{n}, \\
\text{and}\quad B_4 &\coloneqq \frac{8K^2\gamma_2^2}{1-\beta^2}.
\end{aligned}
$$

Now we bound the consensus error. From (86) we have

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n}-\boldsymbol{x}_{i,t}\right\|^2 \\
&\leq \frac{2\alpha^2\varepsilon^2\gamma_2^2}{(1-\beta_\varepsilon^2)D_2}+\frac{18\alpha^2\varepsilon^2\gamma_1^2}{(1-\beta_\varepsilon)^2D_2}+\frac{4\varepsilon^2(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{(1-\beta_\varepsilon^2)D_2}+\frac{18\alpha^2\varepsilon^2}{(1-\beta_\varepsilon)^2D_2}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2 \\
&\leq \alpha^2\varepsilon\frac{2\gamma_2^2}{(1-\beta^2)D_2}+\alpha^2\frac{18\gamma_1^2}{(1-\beta)^2D_2}+\varepsilon\frac{4(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{(1-\beta^2)D_2} \\
&\quad+\alpha^2\frac{18}{(1-\beta)^2D_2}\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\left\|\nabla f\left(\frac{X_s\mathbf{1}_n}{n}\right)\right\|^2.
\end{aligned}
$$

For the same step-sizes α and ε defined in (93) and large enough T as in (95), we can use

the convergence result in (96) which yields

$$\begin{aligned}
\frac{1}{T}\sum_{t=0}^{T-1}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left\|\frac{X_t\mathbf{1}_n}{n}-\boldsymbol{x}_{i,t}\right\|^2 &\leq \frac{1}{T^{5/6}}\frac{4\gamma_2^2}{1-\beta^2}+\frac{1}{T^{1/3}}\frac{36\gamma_1^2}{(1-\beta)^2}+\frac{1}{T^{1/2}}\frac{8(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{(1-\beta^2)} \\
&\quad+\frac{1}{T^{1/3}}\frac{36}{(1-\beta)^2}\left(\frac{B_1}{T^{1/3}}+\frac{B_2}{T^{1/2}}+\frac{B_3}{T^{2/3}}+\frac{B_4}{T^{5/6}}\right) \\
&=\frac{C_1}{T^{1/3}}+\frac{C_2}{T^{1/2}}+\frac{C_3}{T^{2/3}}+\frac{C_4}{T^{5/6}}+\frac{C_5}{T}+\frac{C_6}{T^{7/6}} \\
&=\mathcal{O}\left(\frac{\gamma^2}{m(1-\beta)^2}\right)\frac{1}{T^{1/3}}+\mathcal{O}\left(\frac{(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{1-\beta^2}\right)\frac{1}{T^{1/2}} \\
&\quad+\mathcal{O}\left(\frac{K^2}{(1-\beta)^4}\frac{\gamma^2}{m}+\frac{K}{(1-\beta)^2}\frac{(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{n}\right)\frac{1}{T^{2/3}} \\
&\quad+\mathcal{O}\left(\frac{\gamma^2}{1-\beta^2}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}+\frac{K^2\widetilde{\sigma}^2}{(1-\beta)^4}\right)\frac{1}{T^{5/6}} \\
&\quad+\mathcal{O}\left(\frac{K}{(1-\beta)^2}\frac{\gamma^2}{n}\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T} \\
&\quad+\mathcal{O}\left(\frac{K^2}{(1-\beta)^4}\gamma^2\max\left\{\frac{\mathbb{E}[1/V]}{T_d},\frac{1}{m}\right\}\right)\frac{1}{T^{7/6}},
\end{aligned}$$

where

$$\begin{aligned}
C_1 &\coloneqq \frac{36\gamma_1^2}{(1-\beta)^2}, \\
C_2 &\coloneqq \frac{8(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{1-\beta^2}, \\
C_3 &\coloneqq \frac{36}{(1-\beta)^2}(4(f(0)-f^*))+\frac{72K(\widetilde{\sigma}^2+\frac{pK^2\sigma^2}{\eta^2})}{(1-\beta)^2 n}+\frac{2592K^2\gamma_1^2}{(1-\beta)^4}, \\
C_4 &\coloneqq \frac{4\gamma_2^2}{1-\beta^2}+\frac{576K^2\widetilde{\sigma}^2}{(1-\beta)^2(1-\beta^2)}, \\
C_5 &\coloneqq \frac{432K\gamma_2^2}{(1-\beta)^2 n}, \\
\text{and}\quad C_6 &\coloneqq \frac{288K^2\gamma_2^2}{(1-\beta)^2(1-\beta^2)}.
\end{aligned}$$

5 Differentially Private and Fair Classification via Calibrated Functional Mechanism

5.1 Introduction

In this big data era, machine learning has been becoming a powerful technique for automated and data-driven decision making processes in various domains, such as spam filtering, credit ratings, housing allocation, and so on. However, as the success of machine learning mainly rely on a vast amount of individual data (e.g., financial transactions, tax payments), there are growing concerns about the potential for privacy leakage and unfairness in training and deploying machine learning algorithms [11, 91]. Thus, the problem of fairness and privacy in machine learning has attracted considerable attention.

Fairness-aware learning has received growing attentions in the machine learning field due to the social inequities and unfair behaviors observed in classification models. For example, a classification model of automated job hiring system is more likely to hire candidates from certain racial or gender groups [15, 16]. Hence, substantial effort has centered on developing algorithmic methods for designing fair classification models and balancing the trade-off between accuracy and fairness, mainly including two groups: pre/post-processing methods [92, 93, 94] and in-processing methods [95, 96]. Pre/post-processing methods achieve fairness by directly changing values of the sensitive attributes or class labels in the training data. As pointed out in [96], pre/post-processing methods treat the learning algorithm as a black box, which can result in unpredictable loss of the classification utility. Thus, in-processing methods, which introduce fairness constraints or regularization terms to the objective function to remove the discriminatory effect of classifiers, have been shown a great success.

At the same time, differential privacy [26] has emerged as the de facto standard for measuring the privacy leakage associated with algorithms on sensitive databases, which has recently received considerable attentions by large-scale corporations such as Google [97] and Microsoft [98], etc. Generally speaking, differential privacy ensures that there is no

statistical difference to the output of a randomized algorithm whether a single individual opts in to, or out of its input. A large class of mechanisms has been proposed to ensure differential privacy. For instance, the Laplace mechanism is employed by introducing random noise drawn from the Laplace distribution to the output of queries such that the adversary will not be able to confirm a single individual is in the input with high confidence [25]. To design private machine learning models, more complicated perturbation mechanisms have been proposed like objective perturbation [56] and functional mechanism [99], which inject random noise into the objective function rather than model parameters.

Thus, in this work, we mainly focus on achieving classification models that simultaneously provide differential privacy and fairness. As pointed out in recent study [100], achieving both requirements efficiently is quite challenging, due to the different aims of differential privacy and fairness. Differential privacy in a classification model focuses on the individual level, i.e., differential privacy guarantees that the model output is independent of whether any individual record presents or absents in the dataset, while fairness in a classification model focuses on the group level, i.e., fairness guarantees that the model predictions of the protected group (such as female group) are same to those of the unprotected group (such as male group). Lots of researches have emerged in achieving both privacy protection and fairness. Specifically, in [92], Dwork et al. gave a new definition of fairness that is an extended definition of differential privacy. In [101], Hajian et al. imposed fairness and k-anonymity via a pattern sanitization method. Moreover, Ekstrand et al. in [102] put forward a set of questions about whether fairness are compatible with privacy. However, only Xu et al. in [100] studied how to meet the requirements of both differential privacy and fairness in classification models by combining functional mechanism and decision boundary fairness together. Therefore, how to simultaneously meet the requirements of differential privacy and fairness in machine learning algorithms is under exploited.

In this work, we propose **P**urely and **A**pproximately **D**ifferential private and **F**air **C**lassification algorithms, called PDFC and ADFC, respectively, by incorporating functional mechanism and decision boundary covariance, a novel measure of decision boundary

fairness. As shown in [103], due to the correlation between input features (attributes), the discrimination of classification still exists even if removing the protected attribute from the dataset before training. Hence, different from [100], which adds same scale of noise in each attribute, in PDFC, we consider a calibrated functional mechanism, i.e., injecting different amounts of Laplace noise regarding different attributes to the polynomial coefficients of the constrained objective function to ensure ϵ-differential privacy and reduce effects of discrimination. To further improve the model accuracy, in ADFC, we propose a relaxed functional mechanism by inserting Gaussian noise instead of Laplace noise and leverage it to perturb coefficients of the polynomial representation of the constrained objective function to enforce (ϵ, δ)-differential privacy and fairness. Our salient contributions are listed as follows.

- We propose two approaches PDFC and ADFC to learn a logistic regression model with differential privacy and fairness guarantees by applying functional mechanism to a constrained objective function of logistic regression that decision boundary fairness constraint is treated as a penalty term and added to the original objective function.
- For PDFC, different magnitudes of Laplace noise regarding different attributes are added to the polynomial coefficients of the constrained objective function to enforce ϵ-differential privacy and fairness.
- For ADFC, we further improve the model accuracy by proposing the relaxed functional mechanism based on Extended Gaussian mechanism, and leverage it to introduce Gaussian noise with different scales to perturb objective function.
- Using real-world datasets, we show that the performance of PDFC and ADFC significantly outperforms the baseline algorithms while jointly providing differential privacy and fairness.

6 Problem Statement

This work considers a training dataset D that includes n tuples $t_1, t_2, \cdots, t_n$. We also denote each tuple $t_i = (\boldsymbol{x}_i, y_i)$ where the feature vector $\boldsymbol{x}_i$ contains d attributes, i.e.,

$\boldsymbol{x}_i = (x_{i1}, x_{i2}, \cdots, x_{id})$, and y_i is the corresponding label. Without loss of generality, we assume $\sqrt{\sum_{j=1}^{d} x_{ij}^2} \leq 1$ where $x_{ij} \geq 0$, and $y_i \in \{0, 1\}$ for binary classification tasks. The objective is to construct a binary classification model $\rho(\boldsymbol{x}, w)$ with model parameters $w = (w_1, w_2, \cdots, w_d)$ that taken $\boldsymbol{x}$ as input, can output the prediction $\hat{y}$, by minimizing the empirical loss on the training dataset D over the parameter space w of ρ.

In general, we have the optimization problem as

$$w^* = \arg\min_w f(D, w) = \arg\min_w \sum_{i=1}^{n} f(t_i, w), \tag{97}$$

where f is the loss function. In this work, we consider logistic regression as the loss function, i.e., $f(D, w) = \sum_{i=1}^{n} [\log(1 + exp(\boldsymbol{x}_i^T w)) - y_i \boldsymbol{x}_i^T w]$. Thus, the classification model has the form $\rho(\boldsymbol{x}, w^*) = \frac{exp(\boldsymbol{x}^T w^*)}{1+exp(\boldsymbol{x}^T w^*)}$.

Although there is no need to share the dataset during the training procedure, the risk of information leakage still exists when we release the classification model parameter w^*. For example, the adversary may perform model inversion attack [11] over the release model w^* together with some background knowledge about the training dataset to infer sensitive information in the dataset.

Furthermore, if labels in the training dataset are associated with a protected attribute z_i (note that we denote $\boldsymbol{x}_i$ as unprotected attributes), like gender, the classifier may be biased, i.e., $P(\hat{y}_i = 1 | z_i = 0) \neq P(\hat{y}_i = 1 | z_i = 1)$, where we assume the protected attribute $z_i \in \{0, 1\}$. According to [104], even if the protected attribute is not used to build the classification model, this unfair behavior may happen when the protected attribute is correlated with other unprotected attributes.

Therefore, in this work, our objective is to learn a binary classification model, which is able to guarantee differential privacy and fairness while preserving good model utility.

6.1 Background

In this section, we first introduce some background knowledge of Functional Mechanism in differential privacy, which helps us to build private classification models. Then we present

fairness definition, which helps us to enforce classification fairness.

6.1.1 Functional Mechanism

Functional mechanism, introduced by [99], as an extension of the Laplace mechanism is designed for regression analysis. To preserve ϵ-differential privacy, functional mechanism injects differentially private noise into the objective function $f(D, w)$ and then publishs a noisy model parameter $\hat{w}$ derived from minimizing the perturbed objective function $\hat{f}(D, w)$ rather than the original one. As a result of the objective function being a complex function of w, in functional mechanism, $f(D, w)$ is represented in polynomial forms trough Taylor Expansion. The model parameter w is a vector consisting of several values $w_1, w_2, \cdots, w_d$. We denote $\phi(w)$ as a product of $w_1, w_2, \cdots, w_d$, namely, $\phi(w) = w_1^{c_1} w_2^{c_2} \cdots w_d^{c_d}$ for some $c_1, c_2, \cdots, c_d \in \mathbb{N}$. We also denote $\Phi_j (j \in \mathbb{N})$ as the set of all products of $w_1, w_2, \cdots, w_d$ with degree j, i.e., $\Phi_j = \{w_1^{c_1} w_2^{c_2} \cdots w_d^{c_d} | \sum_{l=1}^{d} c_l = j\}$.

According to the Stone-Weierstrass Theorem [105], any continuous and differentiable function can always be expressed as a polynomial form. Therefore, the objective function $f(D, w)$ can be written as

$$f(D, w) = \sum_{i=1}^{n} \sum_{j=0}^{J} \sum_{\phi \in \Phi_j} \lambda_{\phi t_i} \phi(w), \tag{98}$$

where $\lambda_{\phi t_i}$ represents the coefficient of $\phi(w)$ in polynomial.

To preserve ϵ-differential privacy, the objective function $f(D, w)$ is perturbed by adding Laplace noise into the polynomial coefficients, i.e., $\lambda_\phi = \sum_{i=1}^{n} \lambda_{\phi t_i} + Lap(\Delta_1/\epsilon)$, where $\Delta_1 = 2 \max_t \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \|\lambda_{\phi t}\|_1$. And then the model parameter $\hat{w}$ is obtained by minimizing the noisy objective function $\hat{f}(D, w)$. The sensitivity of logistic regression is given in the following lemma

Lemma 21 (l_1**-Sensitivity of Logistic Regression).** *Let $f(D, w)$ and $f(D', w)$ be the logistic regression on two neighboring datasets D and D', respectively, and denote their polynomial representations as $f(D, w) = \sum_{i=1}^{n} \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \lambda_{\phi t_i} \phi(w)$ and*

$f(D', w) = \sum_{i=1}^{n} \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \lambda_{\phi t'_i} \phi(w)$. *Then, we have the following inequality*

$$\Delta_1 = \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \| \sum_{t_i \in D} \lambda_{\phi t_i} - \sum_{t'_i \in D'} \lambda_{\phi t'_i} \|_1 \leq 2 \max_t \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \|\lambda_{\phi t}\|_1 \leq \frac{d^2}{4} + d,$$

where t_i, t'_i or t is an arbitrary tuple.

6.1.2 Classification Fairness

The goal of classification fairness is to find a classifier that minimizes the empirical loss while guaranteeing certain fairness requirements. Many fairness definitions have been proposed for in the literature including mistreatment parity [106], demographic parity [104], etc.

Demographic parity, the most widely-used fairness definition in the classification fairness domain, requires the decision made by the classifier is not dependent on the protected attribute z, for instance, sex or race.

Definition 9. *(***Demographic Parity in a Classifier)** *Given a classification model $\hat{y} = \rho(\boldsymbol{x}, w)$ and a labeled dataset D, the property of demographic parity in a classifier is defined by $\Pr(\hat{y} = 1|z = 1) = \Pr(\hat{y} = 1|z = 0)$ where $z \in \{0, 1\}$ is the protected attribute.*

Moreover, demographic parity is quantified in terms of the risk difference (RD) [107], i.e., the difference of the positive decision made in between the protected group and unprotected group. Thus, the risk difference produced by a classifier is defined as $RD = |\Pr(\hat{y} = 1|z = 1) - \Pr(\hat{y} = 1|z = 0)|$.

One of the in-processing methods, called decision boundary fairness [96], to ensure classification fairness is to find a model parameter w that minimizes the loss function $f(D, w)$ under a fairness constraint. Thus, the fair classification problem is formulated as

$$\begin{aligned} &\textit{minimize} \quad f(D, w) \\ &\textit{subject to} \quad g(D, w) \leq \tau, g(D, w) \geq -\tau, \end{aligned} \tag{99}$$

where $g(D, w)$ is a constraint term, and τ is the threshold. For instance, Zafar et al. [96] have proposed to adopt the decision boundary covariance to define the fairness constraint, i.e.,

$$g(D, w) = \mathbb{E}[(z - \bar{z})d(\boldsymbol{x}, w)] - \mathbb{E}[z - \bar{z}]d(\boldsymbol{x}, w) \propto \sum_{i=1}^{n}(z_i - \bar{z})d(\boldsymbol{x}_i, w), \tag{100}$$

where $\{d(\boldsymbol{x}_i, w)\}_{i=1}^n$ is decision boundary, $\bar{z}$ is the average of the protected attribute and $\mathbb{E}[z-\bar{z}] = 0$. For logistic regression classification models, the decision boundary is defined by $\boldsymbol{x}^T w$. The decision boundary covariance (100) then reduces to $g(D, w) = \sum_{i=1}^n (z_i - \bar{z})\boldsymbol{x}_i^T w$.

6.2 Differentially Private and Fair Classification

In this section, we first present our approach PDFC to achieve fair logistic regression with ϵ-differentially private guarantee. Then we propose a relaxed functional mechanism by injecting Gaussian noise instead of Laplace noise to provide (ϵ, δ)-differential privacy. By leveraging the relaxed functional mechanism, we will show that our second approach ADFC can jointly provide (ϵ, δ)-differential privacy and fairness.

6.2.1 Purely DP and Fair Classification

In order to meet the requirements of ϵ-differential privacy and fairness, motivated by [100], we consider to combine the functional mechanism and decision boundary fairness. We first consider to transform the constrained optimization problem (99) into unconstrained problem by treating the fairness constraint as a penalty term, where the fairness constraints are shifted to the original objective function $f(D, w)$. Then, we have the new objective function $\tilde{f}_D(w)$ defined as $\tilde{f}(D, w) = f(D, w) + \alpha_1|g(D, w) - \tau|$, where we consider α_1 as a hyperparameter to optimize the trade-off between model utility and fairness. For convenience of discussion, we set $\tau = 0$ and choose suitable values to make $\alpha_1 = 1$. Note that our theoretical results still hold if we choose other values of α_1 and τ. By equation

(100), we have

$$\tilde{f}(D,w) = \sum_{i=1}^{n}[\log(1+exp(\boldsymbol{x}_i^T w)) - y_i \boldsymbol{x}_i^T w] + \left|\sum_{i=1}^{n}(z_i - \bar{z})\boldsymbol{x}_i^T w\right|. \tag{101}$$

To apply functional mechanism, we first write the approximate objective function $\bar{f}(D,w)$ based on (98) as

$$\begin{aligned}\bar{f}(D,w) &= \sum_{i=1}^{n}\sum_{j=0}^{2}\frac{f_1^{(j)}(0)}{j!}(\boldsymbol{x}_i^T w)^j - \left(\sum_{i=1}^{n} y_i \boldsymbol{x}_i^T\right) w + \left|\sum_{i=1}^{n}(z_i - \bar{z})\boldsymbol{x}_i^T w\right| \\ &= \sum_{i=1}^{n}\sum_{j=0}^{2}\sum_{\phi\in\Phi_j}\bar{\lambda}_{\phi t_i}\phi(w),\end{aligned} \tag{102}$$

where $\bar{\lambda}_{\phi t_i}$ denotes the coefficient of $\phi(w)$ in the polynomial of $\bar{f}(t_i,w)$ and $f_1(\cdot) = \log(1+\exp(\cdot))$.

The attributes involving in the dataset may not be independent from each other, which means some unprotected attributes in $\boldsymbol{x}$ are quite correlated with the protected attribute z. For instance, the protected attribute, like gender, may be correlated with the attribute, marital status. Thus, to reduce the discrimination between the protected attribute z and the labels y, it is important to weaken the correlation between these most correlated attributes and protected attribute z. However, it is often impossible to determine the degree of relation between an unprotected attribute and the protected attribute. Therefore, we randomly select an unprotected attribute x_s and leverage functional mechanism to add noise with large scale to the corresponding polynomial coefficients of the monomials involving w_s. Interestingly, this approach not only helps to reduce the correlation between attributes x_s and z, but also improve the privacy on attribute x_s to prevent model inversion attacks, as shown in [108].

The key steps of PDFC are outlined in Algorithm 6. We first set two different privacy budgets, ϵ_s and ϵ_n, for attribute x_s and the rest of attributes $\{\boldsymbol{x} \setminus x_s\}$. Before injecting noise to the coefficients, all coefficients ϕ should be separated into two groups Φ_s and Φ_n by considering whether w_s involves in the corresponding monomials (i.e., whether their the

Algorithm 6 Purely DP and Fair Classification (PDFC)

Input: Dataset D; The objective function $f(D, w)$; The fairness constraint $g(D, w)$; The privacy budget ϵ_s for unprotected attribute x_s; The privacy budget ϵ_n for other unprotected attributes $\{\boldsymbol{x} \setminus x_s\}$; l_1-sensitivity Δ_1.
Output: $\hat{w}$, ϵ.
Set the approximate function $\bar{f}(D, w)$ by equation (102).
Set two sets $\Phi_s = \{\}$, $\Phi_n = \{\}$.
for $1 \leq j \leq 2$ **do**
    for each $\phi \in \Phi_j$ **do**
        if ϕ includes w_s for a particular attribute x_s **then**
            Put ϕ into Φ_s.
        else
            Put ϕ into Φ_n.
        end if
    end for
end for
for $1 \leq j \leq 2$ **do**
    for each $\phi \in \Phi_j$ **do**
        if $\phi \in \Phi_s$ **then**
            Set $\hat{\lambda}_\phi = \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} + Lap(\Delta_1/(\epsilon_s))$.
        else
            Set $\hat{\lambda}_\phi = \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} + Lap(\Delta_1/(\epsilon_n))$.
        end if
    end for
end for
Let $\hat{f}(D, w) = \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \hat{\lambda}_\phi \phi(w)$.
Compute $\hat{w} = \arg\min_w \hat{f}(D, w)$.
Compute $\epsilon = \epsilon_s/d + \epsilon_n(d-1)/d$.
return: $\hat{w}$, ϵ.

coefficients contain attribute x_s). We then add Laplace noises drawn from $Lap(\Delta_1/\epsilon_s)$ and $Lap(\Delta_1/\epsilon_n)$ to the coefficients of $\phi \in \Phi_s$ and $\phi \in \Phi_n$ respectively to reconstruct the differentially private objective function $\hat{f}(D, w)$, where Δ_1 can be found in Lemma 22. Finally, the differentially private model parameter $\hat{w}$ is obtained by minimizing $\hat{f}(D, w)$. Note that $\hat{w}$ also ensures classification fairness due to the objective function involving fairness constraint.

Lemma 22. *Let D and D' be any two neighboring datasets differing in at most one tuple. Let $\bar{f}(D, w)$ and $\bar{f}(D', w)$ be the approximate objective function on D and D', then we have*

the inequality as,

$$\Delta_1 = \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \| \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i'} \|_1 \leq \frac{d^2}{4} + 3d.$$

Proof. Assume that D and D' differ in the last tuple t_n and t_n'. We have that

$$\begin{aligned}
\Delta_1 &= \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \| \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i'} \|_1 \\
&= \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \| \bar{\lambda}_{\phi t_n} - \bar{\lambda}_{\phi t_n'} \|_1 \\
&\leq 2 \max_{t=(\boldsymbol{x},y)} \sum_{j=1}^{2} \sum_{\phi \in \Phi_j} \| \bar{\lambda}_{\phi t} \|_1 \\
&\leq 2 \max_{t=(\boldsymbol{x},y)} (\frac{1}{2} - y_i + |z_i - \bar{z}|) \sum_{e=1}^{d} x_{(e)} + \frac{1}{8} \sum_{1 \leq e,l \leq d} x_{(e)} x_{(l)} \\
&\leq \frac{d^2}{4} + 3d,
\end{aligned}$$

where $x_{(e)}$ represents e-th element in feature vector $\boldsymbol{x}$. □

The following theorem shows the privacy guarantee of PDFC.

Theorem 18. *The output model parameter $\hat{w}$ in PDFC (Algorithm 6) preserves ϵ-differential privacy, where $\epsilon = \frac{1}{d}\epsilon_s + \frac{d-1}{d}\epsilon_n$.*

Proof. We assume there are two neighboring datasets D and D' that differ in the last tuple t_n and t_n'. As shown in the Algorithm 6, all polynomial coefficients ϕ are divided into two subsets Φ_s and ϕ_n in view of whether they include sensitive attribute x_s or not. After injecting Laplace noise, we have

$$\Pr\left(\hat{f}(D,w)\right) = \prod_{\phi \in \Phi_s} \exp\left(\frac{\epsilon_s \| \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_{\phi} \|_1}{\Delta_1}\right) \prod_{\phi \in \Phi_n} \exp\left(\frac{\epsilon_n \| \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_{\phi} \|_1}{\Delta_1}\right).$$

Then, the following inequality holds

$$
\begin{aligned}
&\frac{\Pr\left(\hat{f}(D,w)\right)}{\Pr\left(\hat{f}(D',w)\right)} \\
&\leq \prod_{\phi\in\Phi_s} \exp\left(\frac{\epsilon_s\|\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}\|_1}{\Delta_1}\right)\prod_{\phi\in\Phi_n}\exp\left(\frac{\epsilon_n\|\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}\|_1}{\Delta_1}\right) \\
&\leq \prod_{\phi\in\Phi_s} \exp\left(\frac{\epsilon_s\|\bar{\lambda}_{\phi t_n}-\bar{\lambda}_{\phi t'_n}\|_1}{\Delta_1}\right)\prod_{\phi\in\Phi_n}\exp\left(\frac{\epsilon_n\|\bar{\lambda}_{\phi t_n}-\bar{\lambda}_{\phi t'_n}\|_1}{\Delta_1}\right) \\
&\leq \prod_{\phi\in\Phi_s} \exp\left(\frac{\epsilon_s}{\Delta_1}2\max_t\|\lambda_{\phi t}\|_1\right)\prod_{\phi\in\Phi_n}\exp\left(\frac{\epsilon_n}{\Delta_1}2\max_t\|\lambda_{\phi t}\|_1\right) \\
&= \exp\left(\epsilon_s/d+\epsilon_n(d-1)/d\right) \\
&= \exp\left(\epsilon\right).
\end{aligned}
$$

In the last second equality, we directly adopt the result in [108].

□

6.2.2 Approximately DP and Fair Classification

We now focus on using the relaxed version of ϵ-differential privacy, i.e., (ϵ,δ)-differential privacy to further improve the utility of differentially private and fair logistic regression. Hence, in order to satisfy (ϵ,δ)-differential privacy, we propose the relaxed functional mechanism by making use of Extended Gaussian mechanism. As shown in Definition 4, before applying Extended Gaussian mechanism, we first calculate the sensitivity of a query function, i.e., the objective function of logistic regression $f(D,w)=\sum_{i=1}^{n}[\log(1+exp(\boldsymbol{x}_i^T w))-y_i\boldsymbol{x}_i^T w]$, given in the following lemma.

Lemma 23 (l_2**-Sensitivity of Logistic Regression).** *For polynomial representations of logistic regression, two $f(D,w)$ and $f(D',w)$ given in Lemma 21, we have the inequality as*

$$\Delta_2=\|\mathscr{A}_1-\mathscr{A}_2\|_2\leq\sqrt{\frac{d^2}{16}+d},$$

where we denote $\mathscr{A}_1=\{\sum_{i=1}^{n}\lambda_{\phi t_i}\}_{\phi\in\cup_{j=1}^{J}\Phi_j}$ and $\mathscr{A}_2=\left\{\sum_{i=1}^{n}\lambda_{\phi t'_i}\right\}_{\phi\in\cup_{j=1}^{J}\Phi_j}$ as the set of

polynomial coefficients of $f(D, w)$ and $f(D', w)$. And we denote t_i or t'_i as an arbitrary tuple.

Proof. Assume that D and D' differ in the last tuple t_n and t'_n. For logistic regression, we have

$$\begin{aligned} f(D, w) &= \sum_{i=1}^{n} \sum_{j=0}^{2} \frac{f_1^{(j)}(0)}{j!} (\boldsymbol{x}_i^T w)^j - \left(\sum_{i=1}^{n} y_i \boldsymbol{x}_i^T \right) w \\ &= \sum_{i=1}^{n} \sum_{j=0}^{2} \sum_{\phi \in \Phi_j} \lambda_{\phi t_i} \phi(w), \end{aligned}$$

where we have

$$\begin{aligned} \{\lambda_{\phi t_i}\}_{\phi \in \Phi_1} =: \lambda_{1 t_i} &= \frac{f_1^{(1)}(0)}{1!} \boldsymbol{x}_i - y_i \boldsymbol{x}_i = (\frac{1}{2} - y_i) \boldsymbol{x}_i \\ \text{and} \quad \{\lambda_{\phi t_i}\}_{\phi \in \Phi_2} =: \lambda_{2 t_i} &= \frac{f_1^{(2)}(0)}{2!} \boldsymbol{x}_i^2 = \frac{1}{8} \boldsymbol{x}_i^2. \end{aligned}$$

Denote $\mathscr{A}_1 = \{\sum_{i=1}^{n} \lambda_{\phi t_i}\}_{\phi \in \cup_{j=1}^{2} \Phi_j}$ and $\mathscr{A}_2 = \left\{\sum_{i=1}^{n} \lambda_{\phi t'_i}\right\}_{\phi \in \cup_{j=1}^{2} \Phi_j}$ as the set of polynomial coefficients of $f(D, w)$ and $f(D', w)$, and $\mathscr{E} = \begin{pmatrix} (\frac{1}{2} - y) x_{(1)} \\ \cdots \\ (\frac{1}{2} - y) x_{(d)} \\ \frac{1}{8} x_{(1)} x_{(1)} \\ \cdots \\ \frac{1}{8} x_{(d)} x_{(d)} \end{pmatrix}_{(d + d^2) \times 1}$, where $x_{(e)}$ represents e-th element in feature vector $\boldsymbol{x}$.

Then, we have

$$\begin{aligned}
\Delta_2 &= \|\mathscr{A}_1 - \mathscr{A}_2\|_2 \\
&= \|\{\sum_{i=1}^{n} \lambda_{\phi t_i} - \sum_{i=1}^{n} \lambda_{\phi t_i'}\}_{\phi \in \cup_{j=1}^{2} \Phi_j}\|_2 \\
&= \|\{\lambda_{\phi t_n} - \lambda_{\phi t_n'}\}_{\phi \in \cup_{j=1}^{2} \Phi_j}\|_2 \\
&\leq 2 \max_{t=(\boldsymbol{x},y)} \|\mathscr{C}\|_2 \\
&= 2 \max_{t=(\boldsymbol{x},y)} \sqrt{\sum_{j=1}^{d}((\frac{1}{2} - y)x_j)^2 + \sum_{1 \leq e,l \leq d} (\frac{1}{8} x_{(e)} x_{(l)})^2} \\
&= \sqrt{\frac{d^2}{16} + d},
\end{aligned}$$

where t is an arbitrary tuple. □

We then perturb $f(D, w)$ by injecting Gaussian noise drawn from $\mathcal{N}(0, \sigma^2)$ with $\sigma = \frac{\sqrt{2}\Delta_2}{2\epsilon}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}) + \epsilon})$ into its polynomial coefficients, and obtain the differentially private model parameter $\hat{w}$ by minimizing the noisy function $\hat{f}(D, w)$, as shown in Algorithm 7. Finally, we provide a privacy guarantee of proposed relaxed functional mechanism by the following theorem.

Theorem 19. *The relaxed functional mechanism in Algorithm 7 guarantees (ϵ, δ)-differential privacy.*

Proof. Assume that the neighboring datasets D and D' differ in the last tuple t_n and t_n'.

$$\begin{aligned}
\left|\log \frac{\Pr\left(\hat{f}(D, w)\right)}{\Pr\left(\hat{f}(D', w)\right)}\right| &= \left|\log \frac{\prod_{j=1}^{J} \prod_{\phi \in \Phi_j} \exp\left(-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_\phi\right)^2\right)}{\prod_{j=1}^{J} \prod_{\phi \in \Phi_j} \exp\left(-\frac{1}{2\sigma^2}\left(\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i'} - \hat{\lambda}_\phi\right)^2\right)}\right| \\
&= \frac{1}{2\sigma^2}\left|\sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \left(\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_\phi\right)^2 - \left(\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i'} - \hat{\lambda}_\phi\right)^2\right| \\
&= \frac{1}{2\sigma^2}\left|\|\mathscr{A}\|_2^2 - \|\mathscr{A} + \mathscr{B}\|_2^2\right|,
\end{aligned}$$

where $\mathscr{A} = \left\{\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_{\phi}\right\}_{\phi \in \cup_{j=1}^{J} \Phi_j}$ and $\mathscr{B} = \left\{\sum_{i=1}^{n} \bar{\lambda}_{\phi t'_i} - \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i}\right\}_{\phi \in \cup_{j=1}^{J} \Phi_j}$.

We know the fact that the distribution of a spherically symmetric normal is not dependent of the orthogonal basis where its constituent normals are drawn. Thus, we work in a basis aligned with $\mathscr{B}$. Fix such a basis $\mathscr{C}_1, \cdots, \mathscr{C}_{\left|\cup_{j=1}^{J} \Phi_j\right|}$ and draw $\mathscr{A}$ by first drawing signed lengths $\mathscr{V}_\phi \sim \mathcal{N}(0, \sigma^2)$ for $\phi \in \cup_{j=1}^{J} \Phi_j$, then let $\mathscr{A}_\phi = \mathscr{V}_\phi \mathscr{C}_\phi$ and $\mathscr{A} = \sum_{\phi \in \cup_{j=1}^{J} \Phi_j} \mathscr{A}_\phi$. Without loss of generality, we assume that $\mathscr{C}_1$ is parallel to $\mathscr{B}$. Based on the triangle with the base $\mathscr{B} + \mathscr{A}_1$ and the edge $\sum_{\phi=2}^{\left|\cup_{j=1}^{J} \Phi_j\right|} \mathscr{A}_\phi$ orthogonal to $\mathscr{B}$, apparently, we have $\|\mathscr{A} + \mathscr{B}\|_2^2 - \|\mathscr{A}\|_2^2 = \|\mathscr{B}\|_2^2 + 2\mathscr{C}_1\|\mathscr{B}\|_2$. Since $\|\mathscr{B}\|_2 \leq \Delta_2$, we have $\left|\Pr\left(\hat{f}(D, w)\right) / \Pr\left(\hat{f}(D', w)\right)\right| \leq \frac{1}{2\sigma^2}\left|\Delta_2^2 + 2|\mathscr{V}_1|\Delta_2\right|$. When $|\mathscr{V}_1| \leq \frac{1}{2}(2\sigma^2\epsilon - 1)$, the privacy loss is bounded by $\epsilon (\epsilon > 0)$, i.e., $\left|\Pr\left(\hat{f}(D, w)\right) / \Pr\left(\hat{f}(D', w)\right)\right| \leq \epsilon$. Next, we need to prove that the privacy loss is bounded by ϵ with probability at least $1 - \delta$, which requires $\Pr\left(\mathscr{V}_1 > \frac{1}{2}(2\sigma^2\epsilon - 1)\right) \leq \delta/2$. Now we use the tail bound of $\mathscr{V}_1 \sim \mathcal{N}(0, \sigma^2)$, we have

$$\Pr\left(\mathscr{V}_1 > r\right) \leq \frac{\sigma}{\sqrt{2}r} \exp(-\frac{r^2}{2\sigma^2}).$$

By letting $r = \frac{1}{2}(2\sigma^2\epsilon - 1)$ in the above inequality, we have

$$\Pr\left(\mathscr{V}_1 > \frac{1}{2}(2\sigma^2\epsilon - 1)\right) \leq \frac{\sqrt{2}\sigma}{2\sigma^2\epsilon - 1} \exp\left(-\frac{1}{2}\left(\frac{2\sigma^2\epsilon - 1}{2\sigma}\right)^2\right).$$

When $\sigma \geq \frac{\sqrt{2}\Delta_2}{2\epsilon}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}) + \epsilon})$, $\epsilon > 0$ and δ is very small, we have

$$\Pr\left(\mathscr{V}_1 > \frac{1}{2}(2\sigma^2\epsilon - 1)\right) \leq \delta/2.$$

We then can easily prove

$$\Pr\left(|\mathscr{V}_1| \leq \frac{1}{2}(2\sigma^2\epsilon - 1)\right) \geq 1 - \delta.$$

Based on the proof above, we know that the privacy loss $\left|\Pr\left(\hat{f}(D, w)\right) / \Pr\left(\hat{f}(D', w)\right)\right|$ is bounded by ϵ with probability at least $1 - \delta$, which represents the the computation of

Algorithm 7 Relaxed Functional Mechanism

Input: Dataset D; The objective function $f(D, w) = \sum_{i=1}^{n} \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \lambda_{\phi t_i} \phi(w)$; The privacy parameters ϵ, δ.
Output: $\hat{w}$
Set Δ_2 according Lemma 23.
for $1 \leq j \leq J$ **do**
    for each $\phi \in \Phi_j$ **do**
        Set $\lambda_\phi = \sum_{i=1}^{n} \lambda_{\phi t_i} + \mathcal{N}(0, \sigma^2)$, where $\sigma = \frac{\sqrt{2}\Delta_2}{2\epsilon}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta}) + \epsilon})$.
    end for
end for
Let $\hat{f}(D, w) = \sum_{j=1}^{J} \sum_{\phi \in \Phi_j} \lambda_\phi \phi(w)$.
Compute $\hat{w} = \arg\min_w \hat{f}(D, w)$.
return: $\hat{w}$.

$\hat{f}(D, w)$ satisfies (ϵ, δ)-differential privacy. Therefore, Algorithm 7 satisfies (ϵ, δ)-differential privacy. □

Our second approach called, ADFC, applies the relaxed functional mechanism into the objective function with decision boundary fairness constraint to enforce (ϵ, δ)-differential privacy and fairness. As shown in Algorithm 8, we first derive the polynomial representation $\bar{f}(D, w)$ according to (102), and employ random Gaussian noise to perturb the objective function $\bar{f}(D, w)$, i.e., injecting Gaussian noise into its polynomial coefficients. Furthermore, we also allocate differential privacy parameters, (ϵ_s, δ_s) and (ϵ_n, δ_n) for a particular unprotected attribute x_s and the rest of unprotected attributes $\{\boldsymbol{x} \setminus x_s\}$ to improve the privacy on attribute x_s and reduce the correlation between attributes x_s and z. Hence, we add random noise drawn from $\mathcal{N}(0, \sigma_s^2)$ to polynomial coefficients of $\phi \in \Phi_s$. For polynomial coefficients in Φ_n, we inject noise drawn from $\mathcal{N}(0, \sigma_n^2)$.

Lemma 24. *Let D and D' be any two neighboring datasets differing in at most one tuple. Let $\bar{f}(D, w)$ and $\bar{f}(D', w)$ be the approximate objective function on D and D', then we have the inequality as*

$$\Delta_2' = \|\mathscr{A}_1' - \mathscr{A}_2'\|_2 \leq \sqrt{\frac{d^2}{16} + 9d}.$$

Algorithm 8 Approximately DP and Fair Classification (ADFC)

Input: Dataset D; The objective function $f(D, w)$; The fairness constraint $g(D, w)$; The privacy parameters ϵ_s, δ_s for unprotected attribute x_s; The privacy parameters ϵ_n, δ_n for other unprotected attributes $\{\boldsymbol{x} \setminus x_s\}$.
Output: $\hat{w}$, ϵ and δ.
Set the approximate function $\bar{f}(D, w)$ by equation (102).
Set two sets $\Phi_s = \{\}$, $\Phi_n = \{\}$.
for $1 \leq j \leq 2$ **do**
    for each $\phi \in \Phi_j$ **do**
        if ϕ includes w_s for a particular attribute x_s **then**
            Put ϕ into Φ_s.
        else
            Put ϕ into Φ_n.
        end if
    end for
end for
Set l_2-sensitivity Δ_2' by Lemma 24.
for $1 \leq j \leq 2$ **do**
    for each $\phi \in \Phi_j$ **do**
        if $\phi \in \Phi_s$ **then**
            Set $\hat{\lambda}_\phi = \sum_{i=1}^n \bar{\lambda}_{\phi t_i} + \mathcal{N}(0, \sigma_s^2)$, where $\sigma_s = \frac{\sqrt{2}\Delta_2'}{2\epsilon_s}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta_s})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta_s}) + \epsilon_s})$.
        else
            Set $\hat{\lambda}_\phi = \sum_{i=1}^n \bar{\lambda}_{\phi t_i} + \mathcal{N}(0, \sigma_n^2)$, where $\sigma_n = \frac{\sqrt{2}\Delta_2'}{2\epsilon_n}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta_n})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta_n}) + \epsilon_n})$.
        end if
    end for
end for
Let $\hat{f}(D, w) = \sum_{j=1}^2 \sum_{\phi \in \Phi_j} \hat{\lambda}_\phi \phi(w)$.
Compute $\hat{w} = \arg\min_w \hat{f}(D, w)$.
Compute $\epsilon = \frac{1}{d}\epsilon_s + \frac{d-1}{d}\epsilon_n$ and $\delta = 1 - (1 - \delta_s)(1 - \delta_n)$.
return: $\hat{w}$, ϵ and δ.

where we denote $\mathscr{A}_1' = \left\{\sum_{i=1}^n \bar{\lambda}_{\phi t_i}\right\}_{\phi \in \cup_{j=1}^2 \Phi_j}$ *and* $\mathscr{A}_2' = \left\{\sum_{i=1}^n \bar{\lambda}_{\phi t_i'}\right\}_{\phi \in \cup_{j=1}^2 \Phi_j}$ *as the set of polynomial coefficients of* $\bar{f}(D, w)$ *and* $\bar{f}(D', w)$. *And we denote* t_i *or* t_i' *as an arbitrary tuple.*

Proof. Assume that D and D' differ in the last tuple t_n and t_n'. For objective $\bar{f}(D, w)$, we

have

$$\begin{aligned}\bar{f}(D,w) &= \sum_{i=1}^{n}\sum_{j=0}^{2}\frac{f_1^{(j)}(0)}{j!}(\boldsymbol{x}_i^T w)^j - \left(\sum_{i=1}^{n} y_i \boldsymbol{x}_i^T\right) w + \left|\sum_{i=1}^{n}(z_i - \bar{z})\boldsymbol{x}_i^T w\right| \\ &= \sum_{i=1}^{n}\sum_{j=0}^{2}\sum_{\phi\in\Phi_j}\bar{\lambda}_{\phi t_i}\phi(w),\end{aligned}$$

where we have

$$\begin{aligned}\{\bar{\lambda}_{\phi t_i}\}_{\phi\in\Phi_1} &=: \bar{\lambda}_{1t_i} = (\frac{1}{2} - y_i + |z_i - \bar{z}|)\boldsymbol{x}_i \\ \text{and}\quad \{\bar{\lambda}_{\phi t_i}\}_{\phi\in\Phi_2} &=: \bar{\lambda}_{2t_i} = \frac{1}{8}\boldsymbol{x}_i^2.\end{aligned}$$

Denote $\mathscr{A}_1' = \left\{\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}\right\}_{\phi\in\cup_{j=1}^{2}\Phi_j}$ and $\mathscr{A}_2' = \left\{\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i'}\right\}_{\phi\in\cup_{j=1}^{2}\Phi_j}$ as the set of polynomial coefficients of $\bar{f}(D,w)$ and $\bar{f}(D',w)$, and $\mathscr{E} = \begin{pmatrix} (\frac{1}{2} - y + |z - \bar{z}|)x_{(1)} \\ \cdots \\ (\frac{1}{2} - y + |z - \bar{z}|)x_{(d)} \\ \frac{1}{8}x_{(1)}x_{(1)} \\ \cdots \\ \frac{1}{8}x_{(d)}x_{(d)} \end{pmatrix}_{(d+d^2)\times 1}$, where $x_{(e)}$ represents e-th element in feature vector $\boldsymbol{x}$.

Then, we have

$$
\begin{aligned}
\Delta_2 &= \|\mathscr{A}_1' - \mathscr{A}_2'\|_2 \\
&= \|\{\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \sum_{i=1}^{n} \bar{\lambda}_{\phi t_i'}\}_{\phi \in \cup_{j=1}^{2} \Phi_j}\|_2 \\
&= \|\{\bar{\lambda}_{\phi t_n} - \bar{\lambda}_{\phi t_n'}\}_{\phi \in \cup_{j=1}^{2} \Phi_j}\|_2 \\
&\leq 2 \max_{t=(\boldsymbol{x},z,y)} \|\mathscr{E}\|_2 \\
&= 2 \max_{t=(\boldsymbol{x},z,y)} \sqrt{\sum_{j=1}^{d}((\frac{1}{2} - y + |z - \bar{z}|)x_j)^2 + \sum_{1 \leq e,l \leq d} (\frac{1}{8} x_{(e)} x_{(l)})^2} \\
&= \sqrt{\frac{d^2}{16} + 9d},
\end{aligned}
$$

where t is an arbitrary tuple. □

Finally, by minimizing the differentially private objective function $\hat{f}(D, w)$, we derive the model parameter $\hat{w}$, which achieves differential privacy and fairness at the same time. We now show that ADFC satisfies (ϵ, δ)-differential privacy in the following theroem.

Theorem 20. *The output model parameter $\hat{w}$ in ADFC (Algorithm 8) guarantees (ϵ, δ)-differential privacy, where $\epsilon = \frac{1}{d}\epsilon_s + \frac{d-1}{d}\epsilon_n$ and $\delta = 1 - (1 - \delta_s)(1 - \delta_n)$.*

Proof. Assume that the neighboring datasets D and D' differ in the last tuple t_n and t_n'. We have

$$
\Pr\left(\hat{f}(D, w)\right) = \prod_{\phi \in \Phi_s} \exp\left(-\frac{1}{2\sigma_s^2}\left(\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_{\phi}\right)^2\right) \prod_{\phi \in \Phi_n} \exp\left(-\frac{1}{2\sigma_n^2}\left(\sum_{i=1}^{n} \bar{\lambda}_{\phi t_i} - \hat{\lambda}_{\phi}\right)^2\right).
$$

Considering the absolute values, we have

$$\begin{aligned}
\left|\log\frac{\Pr\left(\hat{f}(D,w)\right)}{\Pr\left(\hat{f}(D',w)\right)}\right| &= \left|\frac{1}{2\sigma_s^2}\sum_{\phi\in\Phi_s}\left(\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\hat{\lambda}_\phi\right)^2-\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}-\hat{\lambda}_\phi\right)^2\right)\right. \\
&\quad \left.+\frac{1}{2\sigma_n^2}\sum_{\phi\in\Phi_n}\left(\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\hat{\lambda}_\phi\right)^2-\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}-\hat{\lambda}_\phi\right)^2\right)\right| \\
&\leq \left|\frac{1}{2\sigma_s^2}\sum_{\phi\in\Phi_s}\left(\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\hat{\lambda}_\phi\right)^2-\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}-\hat{\lambda}_\phi\right)^2\right)\right| \\
&\quad +\left|\frac{1}{2\sigma_n^2}\sum_{\phi\in\Phi_n}\left(\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\hat{\lambda}_\phi\right)^2-\left(\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}-\hat{\lambda}_\phi\right)^2\right)\right| \\
&= \frac{1}{2\sigma_s^2}\left|\|\mathscr{A}'\|_2^2-\|\mathscr{A}'+\mathscr{B}'\|_2^2\right|+\frac{1}{2\sigma_n^2}\left|\|\mathscr{A}''\|_2^2-\|\mathscr{A}''+\mathscr{B}''\|_2^2\right|,
\end{aligned}$$

where we let $\mathscr{A}' = \left\{\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\hat{\lambda}_\phi\right\}_{\phi\in\Phi_s}$, $\mathscr{B}' = \left\{\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}-\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}\right\}_{\phi\in\Phi_s}$, $\mathscr{A}'' = \left\{\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}-\hat{\lambda}_\phi\right\}_{\phi\in\Phi_n}$ and $\mathscr{B}'' = \left\{\sum_{i=1}^{n}\bar{\lambda}_{\phi t'_i}-\sum_{i=1}^{n}\bar{\lambda}_{\phi t_i}\right\}_{\phi\in\Phi_n}$.

Now we will use the fact that the distribution of a spherically symmetric normal is not dependent of the orthogonal basis where its constituent normals are drawn. Thus, we work in two basis aligned with $\mathscr{B}'$ and $\mathscr{B}''$ separately. Fix the basis $\mathscr{C}'_1,\cdots,\mathscr{C}'_{|\Phi_s|}$ of $\mathscr{B}'$ and draw $\mathscr{A}'$ by first drawing signed lengths $\mathscr{V}'_\phi\sim\mathcal{N}(0,\sigma_s^2)$ for $\phi\in\Phi_s$, then let $\mathscr{A}'_\phi=\mathscr{V}'_\phi\mathscr{C}'_\phi$ and $\mathscr{A}'=\sum_{\phi\in\Phi_s}\mathscr{A}'_\phi$. Fix the basis $\mathscr{C}''_1,\cdots,\mathscr{C}''_{|\Phi_n|}$ of $\mathscr{B}''$ and draw $\mathscr{A}''$ by first drawing signed lengths $\mathscr{V}''_\phi\sim\mathcal{N}(0,\sigma_n^2)$ for $\phi\in\Phi_n$, then let $\mathscr{A}''_\phi=\mathscr{V}''_\phi\mathscr{C}''_\phi$ and $\mathscr{A}''=\sum_{\phi\in\Phi_n}\mathscr{A}''_\phi$.

Without loss of generality, we assume that $\mathscr{C}'_1$ is parallel to $\mathscr{B}'$ and $\mathscr{C}''_1$ is parallel to $\mathscr{B}''$. Based on the triangle with the base $\mathscr{B}'+\mathscr{A}'_1$ and the edge $\sum_{\phi=2}^{|\Phi_s|}\mathscr{A}'_\phi$ orthogonal to $\mathscr{B}'$, we have $\|\mathscr{A}'+\mathscr{B}'\|_2^2-\|\mathscr{A}'\|_2^2=\|\mathscr{B}'\|_2^2+2\mathscr{C}'_1\|\mathscr{B}'\|_2$. Since $\|\mathscr{B}'\|_2\leq\frac{1}{d}\Delta_2$, we have $\frac{1}{2\sigma_s^2}\left|\|\mathscr{A}'\|_2^2-\|\mathscr{A}'+\mathscr{B}'\|_2^2\right|\leq\frac{1}{2d\sigma_s^2}\left|\Delta_2^2+2|\mathscr{V}'_1|\Delta_2\right|$. Similarly, consider that the triangle with the base $\mathscr{B}''+\mathscr{A}''_1$ and the edge $\sum_{\phi=2}^{|\Phi_n|}\mathscr{A}''_\phi$ orthogonal to $\mathscr{B}''$, we have $\|\mathscr{A}''+\mathscr{B}''\|_2^2-\|\mathscr{A}''\|_2^2=\|\mathscr{B}''\|_2^2+2\mathscr{C}''_1\|\mathscr{B}''\|_2$. Since $\|\mathscr{B}''\|_2\leq\frac{d-1}{d}\Delta_2$, we have $\frac{1}{2\sigma_n^2}\left|\|\mathscr{A}''\|_2^2-\|\mathscr{A}''+\mathscr{B}''\|_2^2\right|\leq\frac{d-1}{2d\sigma_n^2}\left|\Delta_2^2+2|\mathscr{V}''_1|\Delta_2\right|$. When set $|\mathscr{V}'_1|\leq\frac{1}{2}(2\sigma_s^2\epsilon_s-1)$ and $|\mathscr{V}''_1|\leq\frac{1}{2}(2\sigma_n^2\epsilon_n-1)$, we have $\frac{1}{2d\sigma_s^2}\left|\Delta_2^2+2|\mathscr{V}'_1|\Delta_2\right|\leq\frac{1}{d}\epsilon_s$ and $\frac{d-1}{2d\sigma_n^2}\left|\Delta_2^2+2|\mathscr{V}''_1|\Delta_2\right|\leq\frac{d-1}{d}\epsilon_n$. Thus, we have the privacy loss $\left|\Pr\left(\hat{f}(D,w)\right)/\Pr\left(\hat{f}(D',w)\right)\right|\leq\frac{1}{d}\epsilon_s+\frac{d-1}{d}\epsilon_n=\epsilon$.

To ensure the privacy loss is bounded by ϵ with probability at least $1-\delta$, we require

$$\begin{aligned}
&\Pr\left(|\mathscr{V}_1'| \leq \frac{1}{2}(2\sigma_s^2\epsilon_s - 1), |\mathscr{V}_1''| \leq \frac{1}{2}(2\sigma_n^2\epsilon_n - 1)\right) \\
&= \Pr\left(|\mathscr{V}_1'| \leq \frac{1}{2}(2\sigma_s^2\epsilon_s - 1)\right) \Pr\left(|\mathscr{V}_1''| \leq \frac{1}{2}(2\sigma_n^2\epsilon_n - 1)\right) \\
&\geq 1-\delta.
\end{aligned}$$

Now we will give the upper bound of $\Pr\left(|\mathscr{V}_1'| \leq \frac{1}{2}(2\sigma_s^2\epsilon_s - 1)\right)$ by using the tail bound of $\mathscr{V}_1' \sim \mathcal{N}(0, \sigma_s^2)$. Hence, we have $\Pr\left(\mathscr{V}_1' > r'\right) \leq \frac{\sigma_s}{\sqrt{2}r'} \exp(-\frac{r'^2}{2\sigma_s^2})$. By letting $r' = \frac{1}{2}(2\sigma_s^2\epsilon_s - 1)$ in the above inequality, we have $\Pr\left(\mathscr{V}_1' > \frac{1}{2}(2\sigma_s^2\epsilon_s - 1)\right) \leq \frac{\sqrt{2}\sigma_s}{2\sigma_s^2\epsilon_s - 1} \exp\left(-\frac{1}{2}\left(\frac{2\sigma_s^2\epsilon_s - 1}{2\sigma_s}\right)^2\right)$. When $\sigma_s \geq \frac{\sqrt{2}\Delta_2}{2\epsilon_s}(\sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta})} + \sqrt{\log(\sqrt{\frac{2}{\pi}}\frac{1}{\delta_s}) + \epsilon_s})$, $\epsilon_s > 0$ and δ_s is very small, we have $\Pr\left(\mathscr{V}_1' > \frac{1}{2}(2\sigma_s^2\epsilon_s - 1)\right) \leq \delta_s/2$. Thus, we can prove that $\Pr\left(|\mathscr{V}_1'| \leq \frac{1}{2}(2\sigma_s^2\epsilon_s - 1)\right) \geq 1 - \delta_s$. In the same way, we can prove $\Pr\left(|\mathscr{V}_1''| \leq \frac{1}{2}(2\sigma_n^2\epsilon_n - 1)\right) \geq 1 - \delta_n$. Therefore, if we let $\delta = 1 - (1-\delta_s)(1-\delta_n)$, we have

$$\Pr\left(|\mathscr{V}_1'| \leq \frac{1}{2}(2\sigma_s^2\epsilon_s - 1), |\mathscr{V}_1''| \leq \frac{1}{2}(2\sigma_n^2\epsilon_n - 1)\right) \geq 1 - \delta,$$

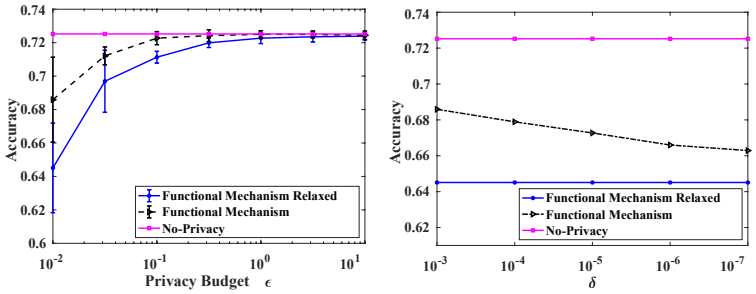
which proves that the computation of $\hat{f}(D, w)$ satisfies (ϵ, δ)-differential privacy. Apparently, the final result $\hat{w}$ also satisfies (ϵ, δ)-differential privacy. □

6.3 Performance Evaluation

6.3.1 Simulation Setup

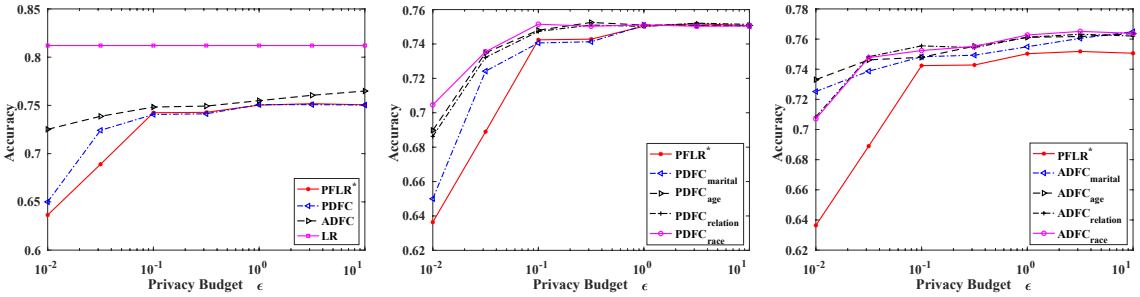
Data preprocessing We evaluate the performance on two datasets, *Adult* dataset and *US* dataset. The *Adult* dataset from UCI Machine Learning Repository [4] contains information about 13 different features (e.g., work-class, education, race, age, sex, and so on) of 48,842 individuals. The label is to predict whether the annual income of those individuals is above 50K or not. The *US* dataset is from Integrated Public Use Microdata

[4] `http://archive.ics.uci.edu/ml/datasets/Adult`



(a) (b)

Figure 17: Compare accuracy under different values privacy budgets ϵ and δ on *US*.



(a) (b) (c)

Figure 18: Compare accuracy under different privacy budgets on *Adult* ($\delta = 10^{-3}$).

Series [5] and consists of 370,000 records of census microdata, which includes features like age, sex, education, family size, etc. The goal is to predict whether the income is over 25K a year. In both datasets, we consider sex as a binary protected attribute.

Baseline algorithms In our experiments, we compare our approaches, PDFC, and ADFC against several baseline algorithms, namely, LR and PFLR*. LR is a logistic regression model. PFLR* [100] is a differentially private and fair logistic regression model that injects Laplace noise with shifted mean to the objective function of logistic regression with fairness constraint. Moreover, we compare our relaxed functional mechanism against the original functional mechanism proposed in [99] and No-Privacy, which is the original functional mechanism without injecting any noise to the polynomial coefficients.

Evaluation The utility of algorithms is measured by *Accuracy*, defined as $Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made}$, which demonstrates the quality of a classifier. The fairness

[5] `http://international.ipums.org`

of classification models is qualified by *risk difference (RD)*,i.e.,

$$RD = |\Pr(\hat{y} = 1|z = 1) - \Pr(\hat{y} = 1|z = 0)|,$$

where z is the protected attribute. We consider a random 80-20 training-testing split and conduct 10 independent runs of algorithms. We then record the mean values and standard deviation values of *Accuracy* and *RD* on the testing dataset. For the parameters of differential privacy, we consider $\epsilon = \{10^{-2}, 10^{-1.5}, 10^{-1}, 10^{0}, 10^{0.5}, 10^{1}\}$, and $\delta = \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}\}$.

6.3.2 Results and Analysis

In Figure 17(a), we show the accuracy of each algorithm, functional mechanism, relaxed functional mechanism and No-Privacy, as a function of the privacy budget with fixed $\delta = 10^{-3}$. We can see that the accuracy of No-Privacy remains unchanged for all values of ϵ, as it does not provide any differential privacy guarantee. Our relaxed functional mechanism exhibits quite higher accuracy than functional mechanism in high privacy regime, and the accuracy of relaxed functional mechanism is the same as No-Privacy baseline when $\epsilon > 10^{-1}$. Figure 17(b) studies the accuracy of each algorithm under different values of δ with fixed $\epsilon = 10^{-2}$. Relaxed functional mechanism incurs lower accuracy when δ decreases, as a smaller δ requires a larger scale of noise to be injected in the objective function. But the accuracy of functional mechanism remains considerably lower than relaxed functional mechanism in all cases.

Figure 18(a) studies the accuracy comparison among PFLR*, LR, PDFC and ADFC on *Adult* dataset with the particular unprotected attribute x_s denoted by marital status. We can observe that ADFC continuously achieves better accuracy than PFLR* in all privacy regime, and PDFC only outperforms PFLR* when ϵ is small. We also evaluate the effect of choosing different attributes as x_s by performing experiments on *Adult* dataset. As shown in Figure 18(b) and Figure 18(c), choosing different attributes, marital status, age, relation and race, has different effects on the accuracy of PDFC and ADFC. However, PDFC and

ADFC still outperform PFLR* under varying values of ϵ. As expected, as the value of ϵ increases, the accuracy of each algorithm becomes higher in above three figures.

Table 1 shows how different privacy budgets affect the risk difference of LR, PFLR*, PDFC and ADFC on two datasets. Note that we consider the attribute x_s as race on *Adult* dataset, and work on *US* dataset. It is clear that PDFC and ADFC produce less risk difference compared to PFLR* in most cases of ϵ. The key reason is that adding different amounts of noise regarding different attributes indeed reduces the correlation between unprotected attributes and protected attributes.

Table 1: Risk difference with different privacy budgets ϵ on two datasets ($\delta = 10^{-3}$).

Data	ϵ	LR	PFLR*	PDFC	ADFC
Adult	0.01	0.187 ± 0.049	0.045 ± 0.095	0.048 ± 0.108	0.146 ± 0.131
	0.1	0.187 ± 0.049	0.004 ± 0.009	0.005 ± 0.022	0.068 ± 0.028
	1	0.187 ± 0.049	0.022 ± 0.088	0.002 ± 0.011	0.045 ± 0.027
	10	0.187 ± 0.049	0.003 ± 0.001	0.035 ± 0.041	0.019 ± 0.003
US	0.01	0.191 ± 0.014	0.037 ± 0.038	0.003 ± 0.034	0.004 ± 0.007
	0.1	0.191 ± 0.014	0.078 ± 0.021	0.001 ± 0.006	0.008 ± 0.003
	1	0.191 ± 0.014	0.069 ± 0.007	0.022 ± 0.047	0.031 ± 0.004
	10	0.191 ± 0.014	0.067 ± 0.003	0.022 ± 0.031	0.045 ± 0.002

7 Future Work

In my future research, I will continue my investigation on the privacy, efficiency, and fairness of (collaborative) machine learning.

For the studies on private collaborative learning, we have proposed a number of differentially private Alternating Direction Method of Multipliers (ADMM) algorithms to balance the privacy-accuracy tradeoff. However, the privacy analyses and convergence rate of our current approach crucially rely on the convexity and smoothness of the objective function. More complicated non-convex problems arise in the context of neural networks, it is worthwhile to study the privacy guarantee and analyze the convergence rate in this practical case. Furthermore, since only the total privacy guarantee after T iterations would be of interest in practice, an adaptive privacy budget allocation (i.e., different ϵ for different iterations) may be preferable to a fixed allocation (as long as the total privacy cost is the same). Thus, we will also develop several adaptive privacy allocation schemes where each iteration has a different share of the overall privacy budget. We also plan to verify our trade-off analysis with sensitive medical data.

Our current works on collaborative learning needs to assume that each agent has access to data generated IID (identically and independently distributed) from a single distribution. However, in practice, the local data is generated and stored across the clients. The totality of data is typically highly imbalanced (clients have different quantities of training data) and statistically heterogeneous (the training samples on clients may come from different distributions). When the goal is to train a single global model for all agents, non-IID data partitioning can be challenging, especially with limited communication budgets. Moreover, number of clients in collaborative learning can be extremely large and some of clients may be temporarily unavailable, dropping out or joining during the training. One potential future direction is to design robust and efficiency collaborative learning algorithms that can tolerate the limited availability of the clients, limited reliability of the network, and the heterogeneous of data distribution.

To address the issues of privacy and discrimination in machine learning, we have designed

classification models with fairness and differential privacy guarantees by jointly combining functional mechanism and decision boundary fairness. However, the privacy guarantee of using functional mechanism needs strict assumption of objective function. We will thus attempt to mitigate the disparate impacts of DP in learning SGD framework. DP-SGD does not restrict focus on convex loss functions rendering it an appealing framework for DP learning tasks. Furthermore, we will also investigate how to achieve privacy and fairness in distributed learning setting.

Bibliography

[1] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: Issues, challenges and countermeasures," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1933–1954, April 2014.

[2] D. Jiang, W. Li, and H. Lv, "An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications," *Neurocomputing*, vol. 220, no. 1, pp. 160–169, November 2017.

[3] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, and M. Devin, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[4] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, and K. Yang, "Large scale distributed deep networks," in *Advances in neural information processing systems*, Lake Tahoe, NV, December 2012.

[5] R. McDonald, K. Hall, and G. Mann, "Distributed training strategies for the structured perceptron," in *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*, Los Angeles, CA, June 2010.

[6] K. Rebello and P. Kuhne, "Sharing and utilizing health data for ai — a new report and recommendations for hhs," FedScoop, July 2019. [Online]. Available: https://www.fedscoop.com/hhs-report-artificial-intelligence-health-care/

[7] H. Claver, "Data sharing key for AI in agriculture," Future Farming, February 2019. [Online]. Available: https://www.futurefarming.com/Tools-data/Articles/2019/2/Data-sharing-key-for-AI-in-agriculture-389844E

[8] J. Conway, "Artificial intelligence and machine learning: Current applications in real estate," Ph.D. dissertation, Massachusetts Institute of Technology, 2018. [Online]. Available: https://dspace.mit.edu/bitstream/handle/1721.1/120609/1088413444-MIT.pdf

[9] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *IEEE 31st Computer Security Foundations Symposium (CSF)*, Oxford, UK, July 2018.

[10] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE symposium on security and privacy (SP)*, San Jose, CA, May 2017.

[11] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM Conference on Computer and Communications Security*, Denver, CO, October 2015.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, June 2016.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, San Diego, CA, May 2015.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, June 2019.

[15] V. Giang, "The potential hidden bias in automated hiring systems," Fast Company, September 2018. [Online]. Available: https://www.fastcompany.com/40566971/the-potential-hidden-bias-in-automated-hiring-systems

[16] S. Wachter-Boettcher, "Ai recruiting tools do not eliminate bias," TIME, September 2018. [Online]. Available: http://time.com/4993431/ai-recruiting-tools-do-not-eliminate-bias

[17] J. Ding, X. Zhang, M. Chen, K. Xue, C. Zhang, and M. Pan, "Differentially private robust admm for distributed machine learning," in *IEEE International Conference on Big Data*, Los Angeles, CA, December 2019.

[18] J. Ding, J. Wang, G. Liang, J. Bi, and M. Pan, "Towards plausible differentially private admm based distributed machine learning," in *ACM International Conference on Information and Knowledge Management*, Virtual Event, October 2020.

[19] J. Ding, G. Liang, J. Bi, and M. Pan, "Differentially private and communication efficient collaborative learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Virtual Event, February 2021.

[20] J. Ding, X. Zhang, X. Li, J. Wang, R. Yu, and M. Pan, "Differentially private and fair classification via calibrated functional mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, New York, NY, February 2020.

[21] J. Ding, S. M. Errapotu, H. Zhang, Y. Gong, M. Pan, and Z. Han, "Stochastic admm based distributed machine learning with differential privacy," in *International conference on security and privacy in communication systems*, Orlando, FL, December 2019.

[22] J. Ding, S. M. Errapotu, Y. Guo, H. Zhang, D. Yuan, and M. Pan, "Private empirical risk minimization with analytic gaussian mechanism for healthcare system," *IEEE Transactions on Big Data*, vol. 1, no. 1, pp. 1–1, May 2020.

[23] J. Ding, X. Qin, W. Xu, Y. Gong, C. Zhang, and M. Pan, "Differentially private admm for distributed medical machine learning," *arXiv preprint arXiv:1901.02094*, 2019.

[24] J. Ding, G. Liang, D. Wang, and M. Pan, "Universal analysis of adaptive gradient methods with better generalization," *In submission to a conference*, 2022.

[25] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*, New York, NY, March 2006.

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

[27] X. Zhang, J. Ding, S. M. Errapotu, X. Huang, P. Li, and M. Pan, "Differentially private functional mechanism for generative adversarial networks," in *IEEE Global Communications Conference (GLOBECOM)*, Waikoloa, HI, December 2019.

[28] D. Shi, J. Ding, S. M. Errapotu, H. Yue, W. Xu, X. Zhou, and M. Pan, "Deep q-network-based route scheduling for tnc vehicles with passengers' location differential privacy," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7681–7692, March 2019.

[29] J. Wang, X. Zhang, H. Zhang, H. Lin, H. Tode, M. Pan, and Z. Han, "Data-driven optimization for utility providers with differential privacy of users' energy profile," in *IEEE Global Communications Conference (GLOBECOM)*, Singapore, December 2018.

[30] N. Phan, M. N. Vu, Y. Liu, R. Jin, D. Dou, X. Wu, and M. T. Thai, "Heterogeneous gaussian mechanism: Preserving differential privacy in deep learning with provable robustness," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, China, August 2019.

[31] M. Bun and T. Steinke, "Concentrated differential privacy: Simplifications, extensions, and lower bounds," in *Theory of Cryptography*, China, October 2016.

[32] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum, and S. Vadhan, "On the complexity of differentially private data release: efficient algorithms and hardness results," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*, Bethesda, MD, May 2009.

[33] M. Lyu, D. Su, and N. Li, "Understanding the sparse vector technique for differential privacy," *Proceedings of the VLDB Endowment*, vol. 10, no. 6, pp. 637–648, February 2017.

[34] I. Mironov, "Rényi differential privacy," in *IEEE 30th Computer Security Foundations Symposium (CSF)*, Santa Barbara, CA, August 2017.

[35] Y. Zhu and Y.-X. Wang, "Poission subsampled rényi differential privacy," in *International Conference on Machine Learning*, Long Beach, CA, June 2019.

[36] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update 2015-2020," 2016. [Online]. Available: https://www.cisco.com/c/dam/m/en_in/innovation/enterprise/assets/mobile-white-paper-c11-520862.pdf

[37] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, "A survey of machine learning for big data processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, p. 67, May 2016.

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

[39] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, December 2010.

[40] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc wsns with noisy links – Part I: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, January 2008.

[41] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *IEEE Conference on Decision and Control (CDC)*, Maui, HI, December 2012.

[42] T. Zhang and Q. Zhu, "A dual perturbation approach for differential private admm-based distributed empirical risk minimization," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security*, Austria, October 2016.

[43] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proceedings of the 35th International Conference on Machine Learning*, Sweden, July 2018.

[44] X. Zhang, M. M. Khalili, and M. Liu, "Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms," in *56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Urbana, IL, October 2018.

[45] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Russia, May 2006.

[46] Z. Huang, R. Hu, Y. Guo, E. Chan-Tin, and Y. Gong, "Dp-admm: Admm-based distributed learning with differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 1002–1012, July 2019.

[47] J. Ding, S. M. Errapotu, H. Zhang, M. Pan, and Z. Han, "Stochastic admm based distributed machine learning with differential privacy," in *International conference on security and privacy in communication systems*, Orlando, FL, October 2019.

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

[49] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the

ADMM in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, April 2014.

[50] A. Makhdoumi and A. Ozdaglar, "Convergence rate of distributed ADMM over networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5082–5095, October 2017.

[51] Q. Li, B. Kailkhura, R. Goldhahn, P. Ray, and P. K. Varshney, "Robust decentralized learning using ADMM with unreliable agents," *arXiv preprint arXiv:1710.05241*, 2018.

[52] J. Ding, Y. Gong, C. Zhang, M. Pan, and Z. Han, "Optimal differentially private ADMM for distributed machine learning," *arXiv preprint arXiv:1901.02094*, 2019.

[53] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Austria, October 2016.

[54] K. Chaudhuri and S. A. Vinterbo, "A stability-based validation procedure for differentially private machine learning," in *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, December 2013.

[55] T. Zhang and Q. Zhu, "Dynamic differential privacy for admm-based distributed classification learning," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 172–187, September 2016.

[56] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. 29, pp. 1069–1109, March 2011.

[57] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *IEEE Symposium on Security and Privacy (SP)*, Los Alamitos, CA, May 2019.

[58] D. Kifer, A. Smith, and A. Thakurta, "Private convex empirical risk minimization and high-dimensional regression," in *Proceedings of the 25th Annual Conference on Learning Theory*, Scotland, June 2012.

[59] P. Billingsley, *Probability and measure*, 3rd ed. Wiley, 1995.

[60] S. Shalev-Shwartz and N. Srebro, "Svm optimization: inverse dependence on training set size," in *Proceedings of the 25th international conference on Machine learning*, Finland, July 2008.

[61] K. Sridharan, S. Shalev-Shwartz, and N. Srebro, "Fast rates for regularized objectives," in *Advances in neural information processing systems*, Canada, December 2009.

[62] S. Dasgupta and L. Schulman, "A probabilistic analysis of em for mixtures of separated, spherical gaussians," *Journal of Machine Learning Research*, vol. 8, no. 7, pp. 203–226, February 2007.

[63] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpsgd: Communication-efficient and differentially-private distributed sgd," in *Advances in Neural Information Processing Systems*, Canada, December 2018.

[64] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Advances in Neural Information Processing Systems*, Long Beach, CA, December 2017.

[65] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proceedings of the 2017 ACM International Conference on Management of Data*, Chicago, IL, May 2017.

[66] H. Wang, S. Sievert, S. Liu, Z. Charles, D. Papailiopoulos, and S. Wright, "Atomo: Communication-efficient learning via atomic sparsification," in *Advances in Neural Information Processing Systems*, Canada, December 2018.

[67] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signSGD with majority vote is communication efficient and fault tolerant," in *International Conference on Learning Representations*, New Orleans, LA, May 2018.

[68] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora, "Fetchsgd: Communication-efficient federated learning with sketching," in *Proceedings of the 37th International Conference on Machine Learning*, Virtual Event, July 2020.

[69] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, Denver, Co, October 2015.

[70] B. Jayaraman, L. Wang, D. Evans, and Q. Gu, "Distributed learning without distress: Privacy-preserving empirical risk minimization," in *Advances in Neural Information Processing Systems*, Canada, December 2018.

[71] X. Zhang, M. Fang, J. Liu, and Z. Zhu, "Private and communication-efficient edge learning: A sparse differential gaussian-masking distributed sgd approach," in *ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc)*, Virtual Event, October 2020.

[72] L. Wang, R. Jia, and D. Song, "D2p-fed: Differentially private federated learning with efficient communication," *arXiv preprint arXiv:2006.13039*, 2021.

[73] C. Canonne, G. Kamath, and T. Steinke, "The discrete gaussian for differential privacy," in *Advances in Neural Information Processing Systems*, Virtual Event, December 2020.

[74] N. Ferdinand, H. Al-Lawati, S. Draper, and M. Nokleby, "Anytime minibatch: Exploiting stragglers in online distributed optimization," in *International Conference on Learning Representations*, Canada, April 2018.

[75] D. Jakovetić, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1131–1146, May 2014.

[76] G. Qu and N. Li, "Accelerated distributed nesterov gradient descent," *IEEE Transactions on Automatic Control*, vol. 65, no. 6, pp. 2566–2581, June 2020.

[77] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, March 2011.

[78] J. Zeng and W. Yin, "On nonconvex decentralized gradient descent," *IEEE Transactions on signal processing*, vol. 66, no. 11, pp. 2834–2848, June 2018.

[79] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems*, Long Beach, CA, December 2017.

[80] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Transactions on Signal Processing*, vol. 67, no. 19, pp. 4934–4947, October 2019.

[81] A. Reisizadeh, H. Taheri, A. Mokhtari, H. Hassani, and R. Pedarsani, "Robust and communication-efficient collaborative learning," in *Advances in Neural Information Processing Systems*, Canada, December 2019.

[82] L. Li, D. Shi, R. Hou, H. Li, M. Pan, and Z. Han, "To talk or to work: Flexible communication compression for energy efficient federated learning over heterogeneous mobile edge devices," in *IEEE Conference on Computer Communications (INFOCOM)*, Virtual Conference, May 2021.

[83] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Advances in Neural Information Processing Systems*, Canada, December 2018.

[84] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," in *Advances in Neural Information Processing Systems*, Canada, December 2018.

[85] F. Zhou and G. Cong, "On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, Sweden, July 2018.

[86] M. Wu, X. Zhang, J. Ding, H. Nguyen, R. Yu, M. Pan, and S. T. Wong, "Evaluation of inference attack models for deep learning on medical data," *arXiv preprint arXiv:2011.00177*, 2020.

[87] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in neural information processing systems*, Canada, December 2008.

[88] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, R. G. L. D'Oliveira, H. Eichner, S. E. Rouayheb, D. Evans, J. Gardner, Z. Garrett, A. Gascón, B. Ghazi, P. B. Gibbons, M. Gruteser, Z. Harchaoui, C. He, L. He, Z. Huo, B. Hutchinson, J. Hsu, M. Jaggi, T. Javidi, G. Joshi, M. Khodak, J. Konečný, A. Korolova, F. Koushanfar, S. Koyejo, T. Lepoint, Y. Liu, P. Mittal, M. Mohri, R. Nock, A. Özgür, R. Pagh, M. Raykova, H. Qi, D. Ramage, R. Raskar, D. Song, W. Song, S. U. Stich, Z. Sun, A. T. Suresh, F. Tramèr, P. Vepakomma, J. Wang, L. Xiong, Z. Xu, Q. Yang, F. X. Yu, H. Yu, and S. Zhao, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, June 2021.

[89] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, September 2016.

[90] Z. Liang, B. Wang, Q. Gu, S. Osher, and Y. Yao, "Exploring private federated learning with laplacian smoothing," *arXiv preprint arXiv:2005.00218*, 2020.

[91] A. Datta, M. C. Tschantz, and A. Datta, "Automated experiments on ad privacy settings," in *Proceedings on privacy enhancing technologies*, Philadelphia, PA, June 2015.

[92] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, Cambridge, MA, January 2012.

[93] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Australia, August 2015.

[94] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, Spain, December 2016.

[95] T. Kamishima, S. Akaho, and J. Sakuma, "Fairness-aware learning through regularization approach," in *2011 IEEE 11th International Conference on Data Mining Workshops*, Canada, December 2011.

[96] M. B. Zafar, I. Valera, M. G. Rodriguez, and K. P. Gummadi, "Fairness constraints: Mechanisms for fair classification," in *The 20th International Conference on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, April 2017.

[97] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *21st ACM Conference on Computer and Communications Security*, Scottsdale, AZ, November 2014.

[98] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Advances in Neural Information Processing Systems*, Long Beach, CA, December 2017.

[99] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: regression analysis under differential privacy," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, July 2012.

[100] D. Xu, S. Yuan, and X. Wu, "Achieving differential privacy and fairness in logistic regression," in *Companion Proceedings of The 2019 World Wide Web Conference*, San Francisco, CA, May 2019.

[101] S. Hajian, J. Domingo-Ferrer, A. Monreale, D. Pedreschi, and F. Giannotti, "Discrimination-and privacy-aware patterns," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1733–1782, November 2015.

[102] M. D. Ekstrand, R. Joshaghani, and H. Mehrpouyan, "Privacy for all: Ensuring fair and equitable privacy protections," in *Conference on Fairness, Accountability and Transparency*, New York, NY, February 2018.

[103] F. Kamiran and T. Calders, "Data preprocessing techniques for classification without discrimination," *Knowledge and Information Systems*, vol. 33, no. 1, pp. 1–33, October 2012.

[104] D. Pedreshi, S. Ruggieri, and F. Turini, "Discrimination-aware data mining," in *14th ACM SIGKDD international conference on Knowledge discovery and data mining*, Las Vegas, NV, August 2008.

[105] W. Rudin, *Principles of mathematical analysis*. McGraw-hill New York, 1964.

[106] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate

mistreatment," in *The 26th International Conference on World Wide Web*, Australia, April 2017.

[107] D. Pedreschi, S. Ruggieri, and F. Turini, "A study of top-k measures for discrimination discovery," in *The 27th Annual ACM Symposium on Applied Computing*, Italy, March 2012.

[108] Y. Wang, C. Si, and X. Wu, "Regression model fitting under differential privacy and model inversion attack," in *24th International Joint Conference on Artificial Intelligence*, Argentina, July 2015.