Differentially Private and Communication Efficient Collaborative Learning

Jiahao Ding, [1] **Guannan Liang,** [2] **Jinbo Bi,** [2] **Miao Pan**[1]
[1]University of Houston
[2]University of Connecticut
{jding7, mpan2}@uh.edu, {guannan.liang, jinbo.bi}@uconn.edu

Abstract

Collaborative learning has received huge interests due to its capability of exploiting the collective computing power of the wireless 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 bottleneck. In this paper, to address these problems, we propose two differentially private (DP) and communication efficient algorithms, called `Q-DPSGD-1` and `Q-DPSGD-2`. 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 track the privacy loss of both approaches under the Rényi DP and provide convergence analysis for both convex and non-convex loss functions. The proposed methods are evaluated in extensive experiments on real-world datasets and the empirical results validate our theoretical findings.

Introduction

Machine learning is increasingly deployed into large-scale distributive systems that can improve the quality of our life, such as smart home security (Komninos, Philippou, and Pitsillides 2014), and AI-aided medical diagnosis (Jiang, Li, and Lv 2017). 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 (Abadi et al. 2016a; Dean et al. 2012; McDonald, Hall, and Mann 2010). 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 (Fredrikson, Jha, and Ristenpart 2015; Agarwal et al. 2018). Fortunately, differential privacy (Dwork et al. 2006) 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 (Wang, Ye, and Xu 2017; Abadi et al. 2016b) and output perturbation approaches (Chaudhuri, Monteleoni, and Sarwate 2011; Wu et al. 2017; Ding et al. 2019b). 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 (Wang et al. 2018; Bernstein et al. 2018; Rothchild et al. 2020) or data privacy (Shokri and Shmatikov 2015; Ding et al. 2019b; Jayaraman et al. 2018). However, only a limited amount of works consider both (Agarwal et al. 2018; Zhang et al. 2020; Wang, Jia, and Song 2021). Agarwal et al. (Agarwal et al. 2018) 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 (Zhang et al. 2020), 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 paper, 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 re-

verse 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 (Canonne, Kamath, and Steinke 2020). The reason is that CDP does not support privacy amplification from subsampling and analytical moments accountant (Zhu and Wang 2019), 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 (Ferdinand et al. 2018) 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 (Agarwal et al. 2018). 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.

Related Works

Decentralized consensus optimization has been studied extensively. The most popular first-order choices for the convex setting are distributed gradient descent-type methods (Jakovetić, Xavier, and Moura 2014; Qu and Li 2019), distributed variants of the alternating direction method of multipliers (ADMM) (Shi et al. 2014), and dual averaging (Duchi, Agarwal, and Wainwright 2011). Recently, there have been some works which study non-convex decentralized consensus optimization and establish convergence to a stationary point (Zeng and Yin 2018; Lian et al. 2017). 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 (Reisizadeh et al. 2019a,b; Li et al. 2021) and sparsification (Tang et al. 2018; Stich, Cordonnier, and Jaggi 2018). Another line is to reduce the number of communication rounds by techniques such as periodic averaging that pay more local computation for less communication (Zhou and Cong 2018). 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 (Ding et al. 2019a,b, 2020; Shokri and Shmatikov 2015) have studied how to effectively integrate distributed learning algorithms (ADMM, gradient descent) with DP. However, most of them ignore the communication efficiency aspect.

Problem Setting and Preliminaries

In this paper, we aim to solve the following population risk problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} F(\boldsymbol{x}) = \mathbb{E}_{\theta\sim\mathcal{P}}\, l(\boldsymbol{x},\theta) \tag{1}$$

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 (1), we consider minimizing the following Empirical Risk Minimization (ERM) problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} F_N(\boldsymbol{x}, D) = \frac{1}{mn}\sum_{\theta\in D} l(\boldsymbol{x},\theta) \tag{2}$$

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

In collaborative training, our goal is to collaboratively solve problem (2) 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 $\mathcal{N} = \{1, \cdots, n\}$, 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 $G = \{\mathcal{N}, \mathscr{E}\}$, where $\mathscr{E} \subseteq \mathcal{N} \times \mathcal{N}$ 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

$$\min_{\boldsymbol{x}\in\mathbb{R}^p} f(\boldsymbol{x}, D) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}, D_i) \tag{3}$$

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 (3) in a decentralized manner, we then rewrite it as a consensus

optimization problem as follows,

$$\min_{\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n} \hat{F}(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}_i, D_i), \quad (4)$$
$$\text{s.t.} \quad \boldsymbol{x}_i = \boldsymbol{x}_j,\ \forall i,j \in \mathcal{N}_i$$

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, (3) and (4) are equivalent, i.e., the optimal solution $\{\boldsymbol{x}_i^*\}_{i=1}^n$ of problem (3) holds that $\boldsymbol{x}^* = \boldsymbol{x}_1^* = \boldsymbol{x}_2^* = \cdots = \boldsymbol{x}_n^*$.

To solve Problem (4) 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 (Fredrikson, Jha, and Ristenpart 2015; Wu et al. 2020) and membership inference attack (Shokri et al. 2017) 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 paper, 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 (Reisizadeh et al. 2019a,b). 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 $\langle\nabla\ell(\boldsymbol{x},\theta) - \nabla\ell(\hat{\boldsymbol{x}},\theta), \boldsymbol{x} - \hat{\boldsymbol{x}}\rangle \geq \mu\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|^2$.*

Differential Privacy

Definition 1 ((ϵ,δ)-DP (Dwork et al. 2006))**.** *Given any two neighboring datasets D and $\hat{D}$ differing in at most one single data sample, we say that a randomized mechanism $\mathcal{M}$ satisfies (ϵ,δ)-DP if for any possible output $o \in Range(\mathcal{M})$, we have $\Pr[\mathcal{M}(D) = o] \leq e^\epsilon \Pr[\mathcal{M}(\hat{D}) = o] + \delta$.*

Next, we introduce a generalization of DP, called Rényi differential privacy (RDP) (Mironov 2017), which is widely used in stochastic iterative learning algorithms due to the tighter composition and subsample amplification results.

Definition 2 (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 (Mironov 2017) show that Gaussian mechanism can achieve RDP.

Lemma 1. *Given any function $\mathcal{M}_q : \mathcal{D} \to \mathbb{R}^d$, the Gaussian Mechanism is defined as: $\mathcal{M}_G(D,q) = \mathcal{M}_q(D) + \mathbf{u}$, where $\mathbf{u}$ is drawn from a Gaussian distribution $\mathcal{N}(0,\sigma^2 I_d)$, and Δ_2 is the L_2-sensitivity of function $\mathcal{M}_q$, i.e., $\Delta_2 = \sup_{D\sim\hat{D}}\|\mathcal{M}_q(D) - \mathcal{M}_q(\hat{D})\|$. Gaussian Mechanism satisfies $(\rho, \rho\Delta_2^2/(2\sigma^2))$-RDP.*

Main Methods

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 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 1. 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

$$Q(x) = \begin{cases} k\eta & \text{w.p. } 1 - (x - k\eta)/\eta, \\ (k+1)\eta & \text{w.p. } (x - k\eta)/\eta, \end{cases} \quad (5)$$

where $x \in [k\eta, (k+1)\eta]$. Note that the above quantizer satisfied Assumption 2 (Reisizadeh et al. 2019a).

Note that Q-DPSGD-1 is different from the fixed (mini-batch) gradient computation in previous works (Reisizadeh

Algorithm 1 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

et al. 2019a; Zhang et al. 2020; Agarwal et al. 2018), where each agent i selects a subset of local data samples to estimate the stochastic gradient. Motivated by (Ferdinand et al. 2018; Reisizadeh et al. 2019b), 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 $\mathcal{N}(0, \sigma^2 K^2 I_p))$ 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

$$\begin{aligned}\boldsymbol{x}_{i,t+1} =& (1 - \varepsilon + \varepsilon w_{ii})\boldsymbol{x}_{i,t} + \varepsilon \sum_{j \in \mathcal{N}_i} w_{ij} \boldsymbol{z}_{j,t} \\ & - \alpha\varepsilon(\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t}) + \zeta_{i,t}) \end{aligned} \tag{6}$$

where $\zeta_{i,t} \sim \mathcal{N}(0, \sigma^2 K^2 I_p))$ 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 1. *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 1. *Since we adopt a deadline based scheme in Q-DPSGD-1 algorithm instead of the fixed mini-batch scheme used in (Zhang et al. 2020; Abadi et al. 2016b), 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 (Ferdinand et al. 2018; Reisizadeh et al. 2019b), 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 1 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 2 (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*

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

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 (4)*.*

Remark 2. *Theorem 2 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 $1/T^{\tilde{\delta}}$ 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 3. *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 (Bottou and Bousquet 2008), 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 3 (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:*

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

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

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

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 (Agarwal et al. 2018) 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 (Kairouz et al. 2019), 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, (Canonne, Kamath, and Steinke 2020) has shown that discrete Gaussian provides the same CDP (Bun and Steinke 2016) 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 paper provide the RDP analysis for discrete Gaussian, which can use tight composition theory like analytical moments accountant (Zhu and Wang 2019).

Definition 3 (Discrete Gaussian (Canonne, Kamath, and Steinke 2020))**.** *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 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 4 (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 $\mathcal{N}_{\eta\mathbb{Z}}(0, \sigma^2)$.*

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

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

The details of Q-DPSGD-2 is given in Algorithm 2. 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

$$\begin{aligned}\boldsymbol{x}_{i,t} =& (1-\varepsilon+\varepsilon w_{ii})\boldsymbol{x}_{i,t-1} + \varepsilon\sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{z}_{j,t-1} \\ &- \alpha\varepsilon\widetilde{\nabla} f_i(\boldsymbol{x}_{i,t-1}).\end{aligned} \tag{7}$$

Algorithm 2 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

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 5. *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 4. *From Theorem 5, 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 6 (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 $T \geq T_{\mathsf{min}}^{\mathsf{sc}}$ the iterates generated by the Q-DPSGD-2 algorithm satisfy*

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

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 (4).

Remark 5. *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 $F_N(\boldsymbol{x}_{i,T}) - F_N(\boldsymbol{x}^*) = f(\boldsymbol{x}_{i,T}) - f(\boldsymbol{x}^*) \leq \mathcal{O}((\frac{p(2\log(1/\delta)/\epsilon+1)(\alpha\varepsilon K/\eta+\sqrt{p}B)^2}{\mu\epsilon m^2})^{2/3})$. 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 7 (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:*

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

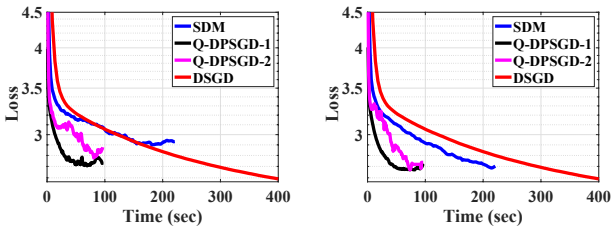
$$\frac{1}{T} \sum_{t=0}^{T-1} \frac{1}{n} \sum_{i=1}^n \mathbb{E} \|\overline{\boldsymbol{x}}_t - \boldsymbol{x}_{i,t}\|^2 \leq \mathcal{O}\left(\frac{\gamma^2}{m(1-\beta)^2} \right) \frac{1}{T^{1/3}} + \mathcal{O}\left(\frac{\hat{K}^2}{(1-\beta)^4} \frac{\gamma^2}{m} + \frac{\hat{K}}{(1-\beta)^2} \frac{(\widetilde{\sigma}^2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

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.

Experimental Results

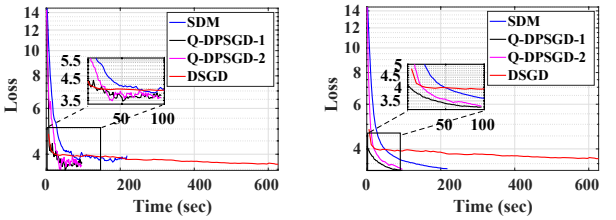
In this section, we present the performance evaluation of the proposed two algorithms for solving a non-convex decentralized optimization problem. In particular, we compare the privacy-accuracy trade-off and the total run-time of our proposed algorithms against the ones for two baselines:

- Decentralized SGD (DSGD) (Yuan, Ling, and Yin 2016): 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) (Zhang et al. 2020): 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 1: Compare loss on MNIST ($T_c = 3$, batch size $B = 20$, $s = 3$, $c = 0.3$).



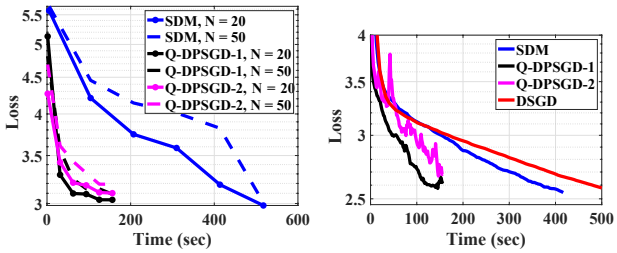
(a) $\epsilon = 1$ (b) $\epsilon = 1.5$

Figure 2: 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 (Zhang et al. 2020) 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 (5) 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 compressed vector are proportioned according the quantization level and the compressed rate c, respectively.



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

Figure 3: 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.

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 1. 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 runtime, 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 5.

Moreover, we also consider the impact of number of agents on the algorithm convergence, as shown in Fig. 3(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 1(a) and Fig. 3(b).

Conclusion

In this paper, we have developed two differentially private and communication efficient collaborate learning algorithms, `Q-DPSGD-1` and `Q-DPSGD-2`. In `Q-DPSGD-1`, the Gaussian mechanism is applied before random quantization. In `Q-DPSGD-2`, we adopt the Gaussian mechanism after random quantization. From theoretical analysis and experimental results, `Q-DPSGD-1` outperforms `Q-DPSGD-2` in terms of the expected population risk and convergence. Our algorithms give practical guidelines for differentially private and communication efficient collaborate learning, and are superior to the state-of-the-art works.

Acknowledgments

We thank the reviewers for their insightful comments. The work of J. Ding and M. Pan was supported in part by the

U.S. National Science Foundation under grants US CNS-1646607, CNS-1801925, and CNS-2029569. The work of J. Bi was partially supported by NSF grants: CCF-1514357 and IIS-1718738, and NIH grant 5K02DA043063-03 and R01-DA051922-01.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016a. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv:1603.04467* .

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016b. Deep learning with differential privacy. In *ACM CCS*.

Agarwal, N.; Suresh, A. T.; Yu, F. X. X.; Kumar, S.; and McMahan, B. 2018. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*.

Bernstein, J.; Zhao, J.; Azizzadenesheli, K.; and Anandkumar, A. 2018. signSGD with majority vote is communication efficient and fault tolerant. *arXiv:1810.05291* .

Bottou, L.; and Bousquet, O. 2008. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, 161–168.

Bun, M.; and Steinke, T. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography*, 635–658. Springer.

Canonne, C.; Kamath, G.; and Steinke, T. 2020. The discrete gaussian for differential privacy. *arXiv preprint arXiv:2004.00010* .

Chaudhuri, K.; Monteleoni, C.; and Sarwate, A. D. 2011. Differentially private empirical risk minimization. *JMLR* 12(Mar): 1069–1109.

Dean, J.; Corrado, G.; Monga, R.; Chen, K.; Devin, M.; Mao, M.; Ranzato, M.; Senior, A.; Tucker, P.; Yang, K.; et al. 2012. Large scale distributed deep networks. In *Advances in neural information processing systems*, 1223–1231.

Ding, J.; Errapotu, S. M.; Zhang, H.; Gong, Y.; Pan, M.; and Han, Z. 2019a. Stochastic admm based distributed machine learning with differential privacy. In *International conference on security and privacy in communication systems*.

Ding, J.; Wang, J.; Liang, G.; Bi, J.; and Pan, M. 2020. Towards Plausible Differentially Private ADMM Based Distributed Machine Learning. In *ACM International Conference on Information and Knowledge Management*.

Ding, J.; Zhang, X.; Chen, M.; Xue, K.; Zhang, C.; and Pan, M. 2019b. Differentially Private Robust ADMM for Distributed Machine Learning. In *IEEE International Conference on Big Data*. Los Angeles, CA.

Duchi, J. C.; Agarwal, A.; and Wainwright, M. J. 2011. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control* 57(3): 592–606.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*. Springer Berlin Heidelberg.

Ferdinand, N.; Al-Lawati, H.; Draper, S.; and Nokleby, M. 2018. Anytime minibatch: Exploiting stragglers in online distributed optimization. In *International Conference on Learning Representations*.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 1322–1333.

Jakovetić, D.; Xavier, J.; and Moura, J. M. 2014. Fast distributed gradient methods. *IEEE Transactions on Automatic Control* 59(5): 1131–1146.

Jayaraman, B.; Wang, L.; Evans, D.; and Gu, Q. 2018. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, 6343–6354.

Jiang, D.; Li, W.; and Lv, H. 2017. An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. *Neurocomputing* 220: 160–169.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2019. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977* .

Komninos, N.; Philippou, E.; and Pitsillides, A. 2014. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials* 16(4): 1933–1954.

Li, L.; Shi, D.; Hou, R.; Li, H.; Pan, M.; and Han, Z. 2021. To Talk or to Work: Flexible Communication Compression for Energy Efficient Federated Learning over Heterogeneous Mobile Edge Devices. In *Proc. IEEE Conference on Computer Communications (INFOCOM)*.

Lian, X.; Zhang, C.; Zhang, H.; Hsieh, C.-J.; Zhang, W.; and Liu, J. 2017. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, 5330–5340.

McDonald, R.; Hall, K.; and Mann, G. 2010. 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*, 456–464.

Mironov, I. 2017. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*.

Qu, G.; and Li, N. 2019. Accelerated distributed Nesterov gradient descent. *IEEE Transactions on Automatic Control* .

Reisizadeh, A.; Mokhtari, A.; Hassani, H.; and Pedarsani, R. 2019a. An exact quantized decentralized gradient descent algorithm. *IEEE Transactions on Signal Processing* 67(19): 4934–4947.

Reisizadeh, A.; Taheri, H.; Mokhtari, A.; Hassani, H.; and Pedarsani, R. 2019b. Robust and communication-efficient collaborative learning. In *Advances in Neural Information Processing Systems*, 8386–8397.

Rothchild, D.; Panda, A.; Ullah, E.; Ivkin, N.; Stoica, I.; Braverman, V.; Gonzalez, J.; and Arora, R. 2020. Fetchsgd: Communication-efficient federated learning with sketching. *arXiv preprint arXiv:2007.07682* .

Shi, W.; Ling, Q.; Yuan, K.; Wu, G.; and Yin, W. 2014. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Transactions on Signal Processing* 62(7): 1750–1761.

Shokri, R.; and Shmatikov, V. 2015. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE S&P*, 3–18. San Jose, CA: IEEE.

Stich, S. U.; Cordonnier, J.-B.; and Jaggi, M. 2018. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*, 4447–4458.

Tang, H.; Gan, S.; Zhang, C.; Zhang, T.; and Liu, J. 2018. Communication compression for decentralized training. In *Advances in Neural Information Processing Systems*.

Wang, D.; Ye, M.; and Xu, J. 2017. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, 2722–2731.

Wang, H.; Sievert, S.; Liu, S.; Charles, Z.; Papailiopoulos, D.; and Wright, S. 2018. Atomo: Communication-efficient learning via atomic sparsification. In *Advances in Neural Information Processing Systems*, 9850–9861.

Wang, L.; Jia, R.; and Song, D. 2021. D2P-Fed: Differentially Private Federated Learning With Efficient Communication. *arXiv preprint arXiv:2006.13039* .

Wu, M.; Zhang, X.; Ding, J.; Nguyen, H.; Yu, R.; Pan, M.; and Wong, S. T. 2020. Evaluation of Inference Attack Models for Deep Learning on Medical Data. *arXiv preprint arXiv:2011.00177* .

Wu, X.; Li, F.; Kumar, A.; Chaudhuri, K.; Jha, S.; and Naughton, J. 2017. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *Proceedings of the 2017 ACM International Conference on Management of Data*, 1307–1322.

Yuan, K.; Ling, Q.; and Yin, W. 2016. On the convergence of decentralized gradient descent. *SIAM Journal on Optimization* 26(3): 1835–1854.

Zeng, J.; and Yin, W. 2018. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing* 66(11): 2834–2848.

Zhang, X.; Fang, M.; Liu, J.; and Zhu, Z. 2020. 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)*.

Zhou, F.; and Cong, G. 2018. On the convergence properties of a K-step averaging stochastic gradient descent algorithm for nonconvex optimization. In *International Joint Conference on Artificial Intelligence*.

Zhu, Y.; and Wang, Y.-X. 2019. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, 7634–7642.