Layered Randomized Quantization for Communication-Efficient and Privacy-Preserving Distributed Learning

Guangfeng Yan*, Tan Li*, Tian Lan, Kui Wu and Linqi Song

***Abstract*—In distributed learning systems, ensuring efficient communication and privacy protection are two significant challenges. Although several existing works have attempted to address these challenges simultaneously, they often overlook essential learning-oriented features such as dynamic gradient and communication characteristics. In this paper, we propose a communication-efficient and privacy-preserving distributed SGD algorithm. Our proposed algorithm employs a layered randomized quantizer (LRQ) to reduce communication overhead, which also ensures that quantization errors follow an exact Gaussian distribution, thus achieving client-level differential privacy. We analyze the trade-off between convergence error, communication, and privacy under non-IID data distributions. Besides, we modify the algorithm to be training-adaptive by adjusting the per-round privacy budget allocation in response to i) dynamic gradient features and ii) real-time changing communication rounds. Both closed-form solutions are derived by solving the minimization problem of convergence error subject to the privacy budget constraint. Finally, we evaluate the effectiveness of our approach through extensive experiments on various datasets, including MNIST, CIFAR-10, and CIFAR-100, demonstrating its superiority in terms of communication cost privacy protection, and model performance compared to state-of-the-art methods.**

***Index Terms*—Distributed Learning, Communication Efficiency, Quantization, Privacy**

I. Introduction

Distributed learning is a crucial framework for realizing edge intelligence, facilitating the deployment of advanced AI capabilities directly at the network edge. By leveraging computational resources across various AI nodes, this approach enables the training of large models and processing of extensive datasets within a short time [1]. Among the algorithms employed in this paradigm, Distributed Stochastic Gradient Descent (Distributed SGD) is particularly prevalent, noted for its effectiveness in handling large-scale learning tasks [2], [3]. Despite the apparent advantages, implementing Distributed SGD in wireless environments is faced with several critical concerns: i) *Communication efficiency*: Due to the sheer volume of devices involved, nodes within this learning setup require data sharing or parameter updates, and these processes may contribute to substantial communication overhead, potentially imposing inhibitive bottlenecks on the system [4], [5] ii) *Privacy*: Exchanging model parameters or even just gradients may lead to the privacy leakage of local data [6]–[8]. iii) *Convergence performance*: Unlike centralized training, distributed learning often requires extended time to achieve model convergence. Furthermore, methods aimed at improving communication efficiency or privacy, such as lossy compression and disrupting gradient, may result in additional training noise/errors that negatively impact convergence. There is thus a strong desire for methods that *systematically provide privacy guarantees while maintaining high communication efficiency and good convergence performance*.

Extensive research has been conducted to address the problem. Several previous studies [9]–[11] have focused on injecting additional discrete noise into post-compressed gradients, effectively avoiding the disruption of communication efficiency that continuous noise might cause. Common types of discrete noise added include binomial noise [9] and discrete Gaussian noise [10], which asymptotically converges to a Gaussian distribution. However, these methods often suffer from “noise explosion” and bias resulting from modulo operation. In response, recent works [12]–[14] have shifted focus towards exploiting the inherent privacy properties of the compression itself. These studies are based on the understanding that the errors introduced by compression can serve as intentional noise perturbations. Such perturbations are supposed to mask the true parameters of the model updates, thus providing privacy protection against attacks. Our work follows this line of research, specifically, we investigate the privacy gain brought by a special quantization technique, called Layered Random Quantization [15], [16].

We develop an algorithm known as Layered Randomized Quantization-aided distributed SGD (LRQ-SGD), which is designed to improve both privacy protection and communication efficiency within the traditional distributed SGD framework. In this algorithm, each client employs the LRQ technique to quantize their model updates locally before sending them to the central server for aggregation. *We shape the quantization noise to mimic an exact Gaussian distribution, thereby offering the same level of differential privacy (DP) as the Gaussian Mechanism [17], [18], which uses continuous Gaussian noise.* However, our method has the added advantage of reducing communication overhead. The LRQ-SGD algorithm is privacy-

*G. Yan and T. Li contributed to the work equally and should be regarded as co-first authors. (Corresponding author: Linqi Song)

G. Yan, L. Song are with the Department of Computer Science, City University of Hong Kong, and City University of Hong Kong Shenzhen Research Institute, Shenzhen, China. (e-mail: gfyan2-c@my.cityu.edu.hk, linqi.song@cityu.edu.hk)

T. Li is with the Department of Computer Science, The Hang Seng University of Hong Kong. (e-mail: tanli6-c@my.cityu.edu.hk)

T. Lan is with the Department of Electrical and Computer Engineering, George Washington University. (e-mail: tlan@gwu.edu)

K. Wu is with the Department of Computer Science, University of Victoria. (e-mail: wkui@uvic.ca)

oriented, taking a (ϵ, δ) privacy budget as input. This budget is distributed evenly across all clients per round, further determining the quantization bit for each client. One of our primary theoretical contributions is the analysis of the trade-offs among communication, privacy, and convergence within the LRQ-SGD framework, particularly under non-IID data conditions. Our findings suggest that stricter privacy settings (lower ϵ) result in smaller quantization bits and consequently, a larger reduction in total communication cost.

We next transfer the proposed LRQ-SGD to a training-adaptive algorithm, specific to distributed AI model training tasks. The first learning-oriented feature we consider is the dynamic behavior of model gradients among training stages. Recent studies [19], [20] have observed that at the early stages of training, the models are far from convergence, and the gradients tend to display large variances, making them easily distinguishable. This observation suggests allocating a smaller privacy budget (i.e., more noise) during these initial stages. However, there has been a lack of theoretical frameworks that guide the optimal distribution of this privacy budget throughout the training process. Our work seeks to fill this gap by developing a dynamic version of the algorithm, named Dynamic LRQ-SGD (DLRQ-SGD), which optimizes the balance between privacy protection and model performance across different phases of the training process. We have formulated and solved the convergence error minimization problem, deriving a closed-form solution that guides our privacy budgeting strategy. Our theoretical analysis indicates that a larger portion of the privacy budget should be reserved for the later stages of training, which is in line with the empirical evidence presented in [19], [20].

The second learning-oriented feature we consider is the chaining communication frequency during the training process. Initially, our approach to determining the (dynamic) per-round privacy budget relies on the predefined number of communication rounds K. However, during training, K may change due to various factors such as the application of early stopping strategies [21] and constraints imposed by battery level. To adapt to these changes, we have implemented a feedback mechanism that dynamically adjusts the privacy budget allocation based on the actual remaining number of communication rounds and the total privacy budget. Our theoretical solution aligns with an intuitive notion: more communication rounds may increase privacy risks. When the number of communication rounds K increases, the per-round privacy budget allocation decreases, leading to an increase in noise and enhanced privacy protection. Conversely, when K decreases, the usable quantization bits per round increase, resulting in reduced quantization noise, which corresponds to an increase in the privacy budget allocation.

Finally, we validate our theoretical analysis through extensive experiments on staple learning tasks with non-IID data, including MNIST, CIFAR-10, and CIFAR-100. The results demonstrate significant improvement over sole privacy-preserving schemes, communication-efficiency schemes, and their naive combination. Besides, DLRQ-SGD outperforms LRQ-SGD, demonstrating the dynamic privacy allocation can yield extra accuracy improvements for the models.

II. Related Work

The development of distributed learning systems has been increasingly influenced by the dual pressures of communication overhead and privacy concerns. In response, compression techniques such as sparsification [22], sketching [23], and quantization [4] have been widely applied to reduce communication overhead. Meanwhile, Differential Privacy [24] provides a framework for injecting controlled noise into raw model parameters, thereby ensuring privacy protections.

In recent years, there has been a growing trend toward systematically addressing both communication efficiency and privacy issues within a unified framework. A pioneering effort in this domain is the integration of compression techniques with the addition of discrete noise. For instance, the introduction of cpSGD [9] is the first to achieve communication-efficient distributed learning with DP by merging quantization with the binomial mechanism. To further enhance DP guarantees, subsequent works [10] incorporate discrete Gaussian noise. These mechanisms are used in conjunction with quantization, aiming to enhance privacy. However, these approaches often suffer from several common drawbacks: 1) since the noise is asymptotically normal, in the high-privacy regimes, the variance of the noise (hence the communication cost) explodes. 2) these methods have not fully explored the inherent privacy properties of compression itself.

Recognizing these issues, some researchers have opted to in favor of more intrinsic methods of achieving privacy. For instance, [25] developed PBM where local devices' gradients are first encoded into a binomial distribution, and subsequently, a sample is drawn from this distribution. Randomized Quantization Mechanism (RQM) is adopted by another study [14], which leverages a two-stage quantization process to achieve privacy. Both approaches have successfully obtained guarantees under Renyi DP. However, these methods encounter limitations because they lack control over the shape of the error introduced by encoding or compression, which means that traditional DP analysis cannot be readily applied. It is noteworthy that our previous work [12] also delved into analyzing the inherent privacy properties of compression. Our findings aligned with the conclusions drawn from the RQM, which indicated that quantization alone is insufficient for ensuring privacy. Consequently, additional noise or randomization is necessary to enhance privacy protections.

Our work closely aligns with the study presented in [13], where similar layered randomized quantization is employed. However, our focus diverges significantly from that of [13]. Our paper emphasizes how the existing distributed SGD framework can benefit from the LRQ in terms of both privacy and communication efficiency. We provide a detailed analysis of how the utilization of LRQ impacts the overall trade-offs among communication, privacy, and convergence performance within the distributed SGD. Moreover, our work considers three learning-oriented features that have previously been overlooked: non-iid data distribution, dynamic model feature and communication requirement during the training process. This makes our approach more adaptable to real-world learning tasks within a distributed learning system.

III. Preliminaries

A. Distributed local SGD

We consider a distributed learning problem, where N clients collaboratively participate in training a shared model via a central server. The local dataset located at client i is denoted as $\mathcal{D}^{(i)}$ and could have different distribution from other clients (which means the IID assumption is relaxed), and the union of all local datasets $\mathcal{D} = \{\mathcal{D}^{(1)}, ..., \mathcal{D}^{(N)}\}$. Our goal is to find a set of global optimal parameters $\boldsymbol{\theta}$ by minimizing the objective function $F : \mathbb{R}^d \to \mathbb{R}$,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) = \frac{1}{N}\sum\nolimits_{i=1}^{N} F_i(\boldsymbol{\theta}), \tag{1}$$

where $F_i(\boldsymbol{\theta}) \triangleq \sum_{\xi^{(i)}\in\mathcal{D}^{(i)}}[l(\boldsymbol{\theta};\xi^{(i)})]$ is the local expected loss of client i, and $l(\boldsymbol{\theta};\xi^{(i)})$ is the local loss function of the model $\boldsymbol{\theta}$) towards one data sample $\xi^{(i)}$.

A standard approach to solve this problem is local SGD [26] with K communication rounds. Specifically, at the k-th round ($0 \leq k \leq K-1$), the server selects a subset $\mathcal{B}_k$ of B clients ($B \leq N$). Each selected client $i \in \mathcal{B}_k$ downloads the global model $\boldsymbol{\theta}_k$ from server and initialize $\boldsymbol{\theta}_{k,0}^{(i)} = \boldsymbol{\theta}_k$. It then performs Q local SGD to update the local model:

$$\boldsymbol{\theta}_{k,q+1}^{(i)} = \boldsymbol{\theta}_{k,q}^{(i)} - \eta_l \boldsymbol{g}_{k,q}^{(i)} \tag{2}$$

for step $q = 0, 1, ..., Q-1$, where η_l is the local learning rate, $\boldsymbol{g}_{k,q}^{(i)} \triangleq \frac{1}{A}\sum_{\xi^{(i)}\in\mathcal{A}_{k,q}^{(i)}} \nabla l(\boldsymbol{\theta}_{k,q}^{(i)};\xi^{(i)})$ is local mini-batch stochastic gradient based on local model parameter $\boldsymbol{\theta}_{k,q}^{(i)}$. Here, $\mathcal{A}_{k,q}^{(i)}$ is a batch of samples with size A randomly selects from $\mathcal{D}^{(i)}$. Let $\boldsymbol{\Delta}_k^{(i)} = \boldsymbol{\theta}_{k,Q}^{(i)} - \boldsymbol{\theta}_{k,0}^{(i)}$ denotes the model update in the k-th communication round. Then the server aggregates model updates from $\mathcal{B}_k$ and sends the updated global model $\boldsymbol{\theta}_{k+1}$ back to all clients for the next round's local training:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \frac{\eta_g}{B}\sum\nolimits_{i\in\mathcal{B}_k} \boldsymbol{\Delta}_k^{(i)} \tag{3}$$

where η_g is the global learning rate. We make the following two common assumptions on such the stochastic gradient $\nabla l(\boldsymbol{\theta};\xi^{(i)})$ and the objective function $F(\boldsymbol{\theta})$ [27], [28]:

Assumption 1 (Bounded Variance)**.** *For parameter $\boldsymbol{\theta}$, the stochastic gradient evaluated on a sample point $\nabla l(\boldsymbol{\theta};\xi^{(i)})$ of each client has a bounded variance uniformly, satisfying:*

$$\mathbb{E}[\|\nabla l(\boldsymbol{\theta};\xi^{(i)}) - \nabla F_i(\boldsymbol{\theta})\|^2] \leq \alpha_l^2. \tag{4}$$

and the deviation between local and global gradient satisfies:

$$\mathbb{E}[\|\nabla F_i(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta})\|^2] \leq \alpha_g^2. \tag{5}$$

Assumption 2 (Smoothness)**.** *The objective function $F(\boldsymbol{\theta})$ is ν-smooth: $\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, $\|\nabla F(\boldsymbol{\theta}) - \nabla F(\boldsymbol{\theta}')\| \leq \nu\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|$.*

Assumption 2 further implies that $\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in \mathbb{R}^d$, we have

$$F(\boldsymbol{\theta}') \leq F(\boldsymbol{\theta}) + \nabla F(\boldsymbol{\theta})^\top(\boldsymbol{\theta}' - \boldsymbol{\theta}) + \frac{\nu}{2}\|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2. \tag{6}$$

Assumption 3 (Bounded Norm)**.** *The raw gradient $\boldsymbol{g}_{k,q}^{(i)}$ has a bounded norm uniformly: $\|\boldsymbol{g}_{k,q}^{(i)}\|_2 \leq G_T$ and $\|\boldsymbol{g}_{k,q}^{(i)}\|_\infty \leq G_I$.*

In this paper, we follow the setting in [29] and utilize a weighted mean of the gradient norm over K communication rounds to evaluate the learning performance, denoted as

$$\mathcal{E} = \frac{\sum_{k=0}^{K-1} \tau^{-k}\|\nabla F(\boldsymbol{\theta}_k)\|^2}{\sum_{k=0}^{K-1} \tau^{-k}} \tag{7}$$

where $\tau \leq 1$, we give more significant weight to the gradient norm in the later stage of training, which can better capture the convergence characteristics of non-convex problems.

B. Client-Level DP and Gaussian Mechanism

In this work, we use client-level differential privacy (CLDP) to quantify the privacy guarantees of the proposed algorithm. Unlike data sample-level differential privacy [30], which protect a single data point's contribution in learning a model, CLDP prevents the eavesdroppers from identifying the participation of a client by observing the aggregated model update. The formal definition of (ϵ,δ)-CLDP is as follows.

Definition 1 (((ϵ,δ) - CLDP [17], [24])**.** Given a set of data sets $\mathcal{D} = \{\mathcal{D}^{(1)}, ..., \mathcal{D}^{(N)}\}$ and a query function $q : \mathcal{D} \to \mathcal{X}$, a mechanism $\mathcal{M} : \mathcal{X} \to \mathcal{O}$ to release the answer of the query, is defined to be (ϵ,δ) - CLDP if for any adjacent datasets $(\mathcal{D},\mathcal{D}')$ constructed by adding or removing all records of any one client and any measurable subset outputs $O \in \mathcal{O}$,

$$\Pr\{\mathcal{M}[q(\mathcal{D})] \in O\} \leq \Pr\{\mathcal{M}[q(\mathcal{D}')] \in O\}e^\epsilon + \delta, \tag{8}$$

where $\epsilon > 0$ is the distinguishable bound of all outputs on adjacent datasets $\mathcal{D},\mathcal{D}'$ that differ in at most one client's local dataset. δ represents the event that the ratio of the probabilities for two adjacent datasets $\mathcal{D},\mathcal{D}'$ cannot be bounded by e^ϵ after privacy-preserving mechanism $\mathcal{M}$.

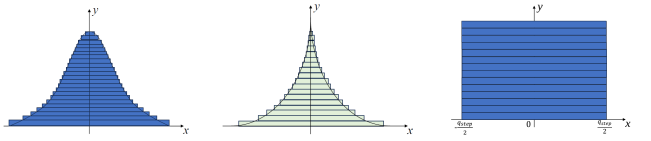
The Gaussian mechanism [17], [24] is commonly used for achieving CLDP in distributed learning by distorting the update aggregation with additive continuous Gaussian noise. When applying Gaussian mechanism in Distributed SGD, we first perform **norm clipping** on each gradient update $\boldsymbol{\Delta}_k^{(i)}$ to ensure that the l_2 norm is no larger than S, i.e., the update $\boldsymbol{\Delta}_k^{(i)}$ is scaled by $\frac{\boldsymbol{\Delta}_k^{(i)}}{\max\{1,\|\boldsymbol{\Delta}_k^{(i)}\|_2/S\}}$. Then the Gaussian noise with parameter σ is added to the sum of all scaled updates in $\mathcal{B}_k$. Dividing the Gaussian mechanism's output by B yields an averaged private update:

$$\bar{\boldsymbol{\Delta}}_k = \frac{1}{B}\sum_{i\in\mathcal{B}_k}\Big[\frac{\boldsymbol{\Delta}_k^{(i)}}{\max\{1,\|\boldsymbol{\Delta}_k^{(i)}\|_2/S\}} + \mathcal{N}(\boldsymbol{0},\sigma^2\boldsymbol{I})\Big] \tag{9}$$

We use the following Lemma to describe the privacy property for the averaged private update.

Lemma 1 (Gaussian mechanism for CLDP [17], [18])**.** *Using Gaussian mechanism in Eq.* (9)*, the averaged private update $\bar{\boldsymbol{\Delta}}_k$ after K communication rounds satisfied $(\frac{2S\sqrt{KB\ln[1/\delta]}}{N\sigma},\delta)$-CLDP.*

From the lemma above, we can see that privacy protection can be achieved by injecting continuous Gaussian noise into the raw model update. However, this contradicts the goal of communication efficiency, as adding continuous noise to the compressed update disrupts the compression results.



(a) Gaussian distribution (b) Laplace distribution (c) Uniform distribution

Fig. 1. Expressing the Gaussian and Laplace distribution as a convex combination of uniform distributions.

C. Layered Randomized Quantization

To avoid the detrimental effects of adding continuous noise to the post-compressed update, recent works start to explore the methods for achieving privacy protection by leveraging the noise inherently generated by the compression process itself. Among them, we follow the research conducted by [15], [16], and proceed to design a Layered Randomized Quantization (LRQ) scheme by combing the dithered quantization [31], [32] and layered multishift coupling [15], to generate exact quantization noise distribution. This approach ensures that the integrity of the quantization is maintained while still fulfilling privacy requirements.

Definition 2 (Dithered Quantizer)**.** The dithered quantizer is defined using i) a *deterministic* quantization step size q_{step} and a dither signal $x \sim U(-\frac{q_{step}}{2}, \frac{q_{step}}{2})$; ii) a *encoder* to encode the input u to $m = \left\lfloor \frac{u+x}{q_{step}} + \frac{1}{2} \right\rfloor$; and iii) a *decoder* to decode m to $\hat{u} = mq_{step} - x$.

For an input $u \in [a_1, a_2]$, the quantization noise $u - \hat{u}$ follows a uniform distribution $U(-\frac{q_{step}}{2}, \frac{q_{step}}{2})$ [31]. By using layered multishift coupling [15], we can create a quantizer where the quantization step q_{step} is not fixed but random. This results in a quantization noise that can be thought of as a mixture of uniform distributions. As shown in Fig. 1, by choosing a suitable distribution of q_{step}, we can obtain any *symmetric unimodal distribution* as the quantization error distribution, for example, Gaussian distribution (Fig. 1(a)) and Laplace distribution (Fig. 1(b)). A Dithered Quantizer can be regarded as a special type of layered multishift coupling (Fig. 1(c)), generating a uniform error distribution.

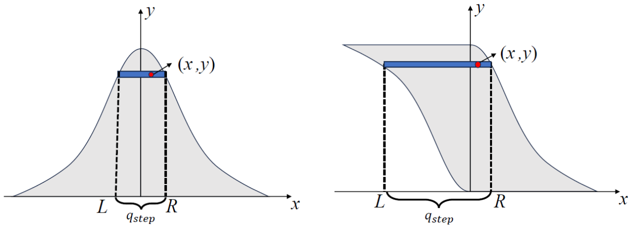
Next, we will explain how to design a specific version of LRQ that can generate exact Gaussian noise, and integrate it into a distributed SGD framework to achieve both communication efficiency and privacy protection.

IV. Layered Randomized Quantization aided Distributed SGD

In this section, we first describe how to utilize LRQ to generate quantization noise that follows the exact Gaussian distribution. Subsequently, we integrate the developed LRQ approach into a distributed SGD framework and provide a theoretical analysis of its performance, focusing on privacy, communication costs, and the bounds of convergence error, particularly in the context of non-IID data settings.

A. Layered Randomized Quantizer with Gaussian Error

To meet the requirement that the quantization error precisely follows a Gaussian distribution, we introduce the following instance of the LRQ:



(a) Original Gaussian distribution (b) Flipped Gaussian distribution

Fig. 2. Compute the quantization step of LRQ by sampling point (x, y).

Definition 3 (Layered Randomized Quantizer with Gaussian error)**.** Given a parameter σ_ϵ, let x is randomly sampled from $\mathcal{N}(0, \sigma_\epsilon^2)$ and y from $\exp\{-\frac{(x/\sigma_\epsilon)^2}{2}\} \cdot U(0,1)$. If $x < 0$, then $y = 1 - y$. Then the LRQ is defined using i) a *random* quantization step $q_{step} = R - L$ determined by $L = -\sigma_\epsilon\sqrt{-2\ln{(1-y)}}$ and $R = \sigma_\epsilon\sqrt{-2\ln y}$; ii) a *encoder* to encoder the input u to $m = \lfloor \frac{u+R+x}{R-L} \rfloor$ and iii) a *decoder* to decode m to $\hat{u} = m(R-L) - x$.

As shown in Fig. 2(a), we can draw x (w.r.t σ_ϵ) from the Gaussian distribution, determine the range of possible values for y, and then pick y uniformly at random from within this range. To compute L and R, we need to be able to invert the probability density function, which we can easily do for the Gaussian distribution. However, in practice, we choose to draw the (x, y) from the flipped Gaussian distribution (Fig. 2(b)) rather than the original Gaussian distribution (Fig. 2(a))

There are two reasons why we consider the flipped Gaussian distribution. Firstly, in Fig. 2(b), the left region is obtained by vertically reflecting the left region of the original Gaussian distribution. Therefore, the flip operation will not affect the fact that x follows the Gaussian distribution. Secondly, for Fig. 2(a), there is no lower bound on how small the q_{step} can be. Quantization becomes meaningless as the value of q_{step} tends to infinity. In contrast, for Fig. 2(b), there is a positive minimum length for the q_{step}. This minimum ensures that there is a practical lower bound to the quantization step, which aids in maintaining the utility and applicability of the quantization process across different scales of data values.

To characterize the properties of LRQ, we derive the following two lemmas.

Lemma 2 (Quantization Noise Shape of LRQ [15], [16])**.** *For an input u and the quantized output $\hat{u}$, the quantization noise of LRQ $u - \hat{u}$ follows Gaussian distribution $\mathcal{N}(0, \sigma_\epsilon^2)$.*

Lemma 3 (Quantization Bit of LRQ)**.** *For an input $u \in [a_1, a_2]$, the LRQ needs at most $\log_2 [\frac{a_2 - a_1}{2\sigma_\epsilon\sqrt{2\ln 2}} + 1]$ bits to express the quantized output $\hat{u}$.*

Proof. Since $q_{step} = R - L = \sigma_\epsilon\sqrt{-2\ln y} + \sigma_\epsilon\sqrt{-2\ln{(1-y)}}$, the minimum value of q_{step} is $2\sigma_\epsilon\sqrt{2\ln 2}$. That means that the maximum quantization level is: $l = \frac{a_2 - a_1}{q_{step}} \leq \frac{a_2 - a_1}{2\sigma_\epsilon\sqrt{2\ln 2}}$. Hence, the needed quantization bits are upper bounded by $\log_2 \left[\frac{a_2 - a_1}{2\sigma_\epsilon\sqrt{2\ln 2}} + 1\right]$. □

The two lemmas above demonstrate that fewer quantization bits result in greater quantization noise, which leads to more significant disruption to the model updtates.

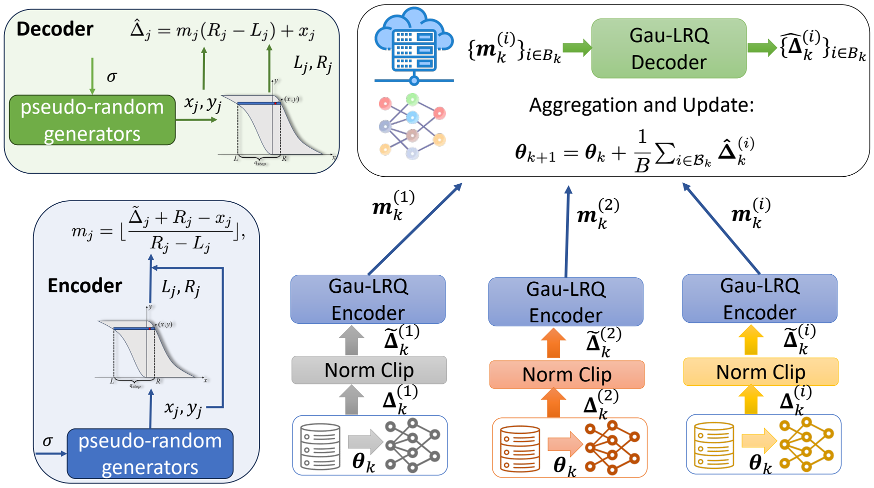

Decoder
$\hat{\Delta}_j = m_j(R_j - L_j) + x_j$
σ
L_j, R_j
pseudo-random generators
x_j, y_j
L
R
q_{step}
(x,y)
x
y
Encoder
$m_j = \lfloor \frac{\tilde{\Delta}_j + R_j - x_j}{R_j - L_j} \rceil,$
L_j, R_j
x_j, y_j
σ
pseudo-random generators
$\{\boldsymbol{m}_k^{(i)}\}_{i \in B_k}$
Gau-LRQ Decoder
$\{\widehat{\boldsymbol{\Delta}}_k^{(i)}\}_{i \in B_k}$
Aggregation and Update:
$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \frac{1}{B} \sum_{i \in \mathcal{B}_k} \hat{\boldsymbol{\Delta}}_k^{(i)}$
$\boldsymbol{m}_k^{(1)}$
$\boldsymbol{m}_k^{(2)}$
$\boldsymbol{m}_k^{(i)}$
Gau-LRQ Encoder
Gau-LRQ Encoder
Gau-LRQ Encoder
$\widetilde{\boldsymbol{\Delta}}_k^{(1)}$
$\widetilde{\boldsymbol{\Delta}}_k^{(2)}$
$\widetilde{\boldsymbol{\Delta}}_k^{(i)}$
Norm Clip
Norm Clip
Norm Clip
$\boldsymbol{\Delta}_k^{(1)}$
$\boldsymbol{\Delta}_k^{(2)}$
$\boldsymbol{\Delta}_k^{(i)}$
$\boldsymbol{\theta}_k$
$\boldsymbol{\theta}_k$
$\boldsymbol{\theta}_k$

Fig. 3. LRQ-SGD framework.

B. LRQ-aided Distributed SGD

After developing the specific LRQ based on Definition 3, a natural idea is to use the Gaussian quantization noise as the "perturbation" to achieve CLDP. We next incorporate the developed LRQ into the distributed SGD framework, leading to the LRQ-aided distributed SGD algorithm (Alg. 1), to concurrently attain communication efficiency and privacy preservation. The algorithm pipeline is summarized in Fig. 3. Each communication round k consists of several steps:

Norm Clip at Client i. To limit the impact of one client's samples on the whole model, we first clip the raw model update $\boldsymbol{\Delta}_k^{(i)}$ into l_2 norm with threshold S:

$$\tilde{\boldsymbol{\Delta}}_k^{(i)} = \frac{\boldsymbol{\Delta}_k^{(i)}}{\max\{1, \|\boldsymbol{\Delta}_k^{(i)}\|_2 / S\}}. \tag{10}$$

Encode and Upload at Client i. Then, each client i uses the LRQ to quantize the clipped model updates in an element-wise way. In particular, the j-th element of $\tilde{\boldsymbol{\Delta}}_k^{(i)}$, denoted as $\tilde{\Delta}_j$ for simplicity, is encoded as:

$$m_j = \lfloor \frac{\tilde{\Delta}_j + R_j - x_j}{R_j - L_j} \rceil, \tag{11}$$

where x_j, L_j and R_j are determined by Definition 3 with the settled σ_ϵ in Line 2. After encoding, client i sends the quantized model update $\mathbf{m}_k^{(i)}$ instead of raw $\boldsymbol{\Delta}_k^{(i)}$ to server.

Decode and Aggregate at Server. The server decodes the j-th element of the quantized model update $\mathbf{m}_k^{(i)}$ as:

$$\hat{\Delta}_j = m_j(R_j - L_j) + x_j \tag{12}$$

The decoding of $\hat{\Delta}_j$ requires the server to know the numbers x_j and y_j, which determine L_j and R_j. However, directly transmitting these values could hurt communication efficiency. To address this, we use pseudo-random generators on both server and client to generate x_j and y_j synchronously. This setup requires a single initial seed distribution to the generators, using methods like on-demand key distribution [33] before training. Finally, the server aggregates the decoded model by $\bar{\boldsymbol{\Delta}}_k = \frac{1}{B} \sum_{i \in \mathcal{B}_k} \hat{\boldsymbol{\Delta}}_k^{(i)}$ and updates the global model.

We use the following Theorem to demonstrate the performance of Alg. 1 in terms of privacy, communication cost, and model convergence.

Theorem 1 (Performance of Privacy-oriented LRQ-SGD)**.** *For an N-client distributed learning problem, Alg. 1 satisfies the following performance:*
Privacy: *The LRQ-SGD satisfied (ϵ, δ)-CLDP.*
Communication: *The LRQ-SGD incurs communication cost*

$$KBd \log_2 \left[\frac{\eta_l Q G_I N \epsilon}{2S\sqrt{2 \ln 2KB \ln[1/\delta]}} + 1 \right] \tag{13}$$

Convergence: *By letting the learning rate satisfied $\frac{\eta_l^2 \nu^2 Q(Q-1)}{2} + 2\nu\eta_g\eta_l Q \leq 1$, the convergence error for smooth objective is upper bounded by*

$$\frac{\sum_{k=0}^{K-1} \check{\beta}_k \tau^{-k} \|\nabla F(\boldsymbol{\theta}_k)\|^2}{\sum_{k=0}^{K-1} \tau^{-k}} \leq \mathcal{E}_{LSGD} + \underbrace{\alpha_3 \frac{S^2 dK \ln[1/\delta]}{N^2 \epsilon^2}}_{\mathcal{E}_{Privacy}}$$
$$\underbrace{+ \alpha_1 \Big[\mathbb{E} \frac{1}{N} \sum_{i=1}^{N} (\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}) \Big] + \alpha_1 \alpha_2 \mathbb{E} \| \frac{1}{B} \sum_{i \in \mathcal{B}_k} [\tilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}] \|^2}_{\mathcal{E}_{clip}} \tag{14}$$

where $\alpha_1 = \frac{2G_T^2}{Q}, \alpha_2 = \nu\eta_g\eta_l, \alpha_3 = \frac{4\eta_g}{\eta_l Q}$,

$$\widetilde{\beta}_k^{(i)} := \frac{1}{\max\{1, \eta_l \| \sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)} \|_2 / S\}},$$

$$\check{\beta}_k := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\max\{1, \eta_l \| \mathbb{E}[\sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)}] \|_2 / S\}}$$

Remark 1. *Note that the first term $\mathcal{E}_{LSGD}$ in the convergence error originates from vanilla local SGD [26], which is not subjected to any compression or privacy-protection techniques. Since this term is not the main focus of this paper, its complete form is provided in the Appendix. VIII-A.*

Next, we will conduct a detailed analysis of the conclusions from the theorem above and the trade-offs derived from it.

Algorithm 1 Layered Randomized Quantized Local SGD (LRQ-SGD)

```
Input: Learning rate $\eta_l$, $\eta_g$, initial point $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, local updates $Q$, communication round $K$, clipping threshold $S$, privacy budget $(\epsilon,\ \delta)$;
Set $\sigma_\epsilon = \frac{2S\sqrt{KB\ln[1/\delta]}}{N\epsilon}$ as the parameter for LRQ;
for each communication rounds $k = 0, 1, ..., K$: do
    On each client $i \in \mathcal{B}_k$:
    Download $\boldsymbol{\theta}_{k,0}^{(i)} = \boldsymbol{\theta}_k$ from server;
    for each local updates $q = 0, 1, ..., Q-1$ do
        Performe local SGD using Eq. (2);
    end for
    Compute the model update $\boldsymbol{\Delta}_k^{(i)} = \boldsymbol{\theta}_{k,Q}^{(i)} - \boldsymbol{\theta}_{k,0}^{(i)}$;
    Clipping $\boldsymbol{\Delta}_k^{(i)}$ to $\tilde{\boldsymbol{\Delta}}_k^{(i)}$ using Eq. (10);
    Encode $\tilde{\boldsymbol{\Delta}}_k^{(i)}$ as $\mathbf{m}_k^{(i)}$ using LRQ's encoder (Eq. (11));
    Send $\mathbf{m}_k^{(i)}$ to the server;
    On the server:
    Decode $\hat{\boldsymbol{\Delta}}_k^{(i)}$ using LRQ's decoder (Eq. (12));
    Aggregate decoded updates $\bar{\boldsymbol{\Delta}}_k = \frac{1}{B}\sum_{i\in\mathcal{B}_k}\hat{\boldsymbol{\Delta}}_k^{(i)}$;
    Update global model parameter: $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_g\bar{\boldsymbol{\Delta}}_k$;
end for
```

• **Privacy**.Our Alg. 1 is privacy-oriented, taking a privacy budget (ϵ, δ) as input. The privacy protection is achieved through the LRQ without the addition of any external noise. According to Lemma 2, the quantization error, following the Gaussian distribution with σ_ϵ as the standard deviation, can be viewed as Gaussian noise added to the clipped model update. By controlling the value of $\sigma_\epsilon = \frac{2S\sqrt{KB\ln[1/\delta]}}{N\epsilon}$ (Lemma 2), we can achieve the same (ϵ, δ)-CLDP of privacy protection as the Gaussian Mechanism we introduced in Section III.

• **Communication**. According to Lemma 3, σ_ϵ determines the size of the quantified updates. Thus, our Alg. 1 also yields additional benefits of communication reduction. By considering $\tilde{\Delta}_j \in [-\eta_l Q G_I, \eta_l Q G_I]$ and $\sigma_\epsilon = \frac{2S\sqrt{KB\ln[1/\delta]}}{N\epsilon}$, we can determine the upper bound of quantization bits for each element of the encoded update:

$$b = \log_2\left[\frac{\eta_l Q G_I N\epsilon}{2S\sqrt{2\ln 2KB\ln[1/\delta]}} + 1\right] \tag{15}$$

In Alg. 1, we use fixed-length coding to transmit $\mathbf{m}_k^{(i)}$, thus, each element is represented by b bits. At communication round k, client i quantizes each element of the d-dimensional model update from b_{init} to b bits, where b_{init} is the number of bits of full-precision floating point, e.g., $b_{init} = 32$ or $b_{init} = 64$. Thus, we can achieve the total communication cost in Eq. (13) by summing over B selected clients, and K communication rounds. Compared with vanilla local SGD, we can reduce $BKd(b_{init} - b)$ bits communication overhead.

• **Privacy-Communication Trade-off**. From Eq. (13), it is evident that the communication cost is directly proportional to ϵ. This implies that stringent privacy measures (a lower privacy budget ϵ) lead to smaller quantization bits and consequently, a reduction in total communication cost. Additionally, the variance of the added noise, σ_ϵ, is proportional to $\sqrt{K}$. This suggests that frequent communication may pose higher risks of privacy breaches. Consequently, the algorithm compensates by increasing the noise added to each model update during encoding and reducing the number of bits used. This adjustment ensures the fulfillment of the privacy requirements.

• **Convergence**. We finally show how the LRQ affects learning convergence. The first item in Eq. (14), denoted as $\mathcal{E}_{LSGD}$, refers to the convergence error bound of vanilla local SGD with *non-compressed* and *privacy-free* model updates. The second item $\mathcal{E}_{\text{privacy}}$ is the privacy error. The impact of our Alg. 1 on convergence error is primarily reflected in this term. Intuitively, $\mathcal{E}_{\text{privacy}}$ is of the order (dK/ϵ^2), suggesting it is directly proportional to the number of communication rounds K and the update dimension d, and inversely proportional to ϵ^2. This implies that frequent communication, a large model (information-rich model), and a strict privacy budget will all lead to a larger convergence error.

The third item $\mathcal{E}_{\text{clip}}$ is caused by Norm Clipping (Eq. (10). We focus our analysis on the term $\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}$, as it is directly related to the value of $\mathcal{E}_{\text{clip}}$. To facilitate this, we introduce an intermediate variable $\acute{\beta}_k^{(i)} := \frac{1}{\max\{1, \eta_l \|\mathbb{E}\sum_{q=0}^{Q-1}\boldsymbol{g}_{k,q}^{(i)}\|_2/S\}}$ to decompose $\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}$ into the following two components:

$$\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)} = \underbrace{\widetilde{\beta}_k^{(i)} - \acute{\beta}_k^{(i)}}_{(1)} + \underbrace{\acute{\beta}_k^{(i)} - \check{\beta}_k^{(i)}}_{(2)} \tag{16}$$

The component (1) is associated with the *local update variance*, which is bounded by (α_l^2/A) as specified in Assumption 1. This component arises due to the stochastic nature of SGD, which depends on the batch size use. This bias can be minimized or even eliminated by using larger batch sizes or full dataset updates. The component (2) is determined by the variance in gradients across different clients, denoted by α_g^2. This bias is introduced due to the non-IID nature of the data among clients. When the data distribution is homogeneous across all clients, this variance component also disappears.

Our Alg. 1 and Theorem 1 are both designed with a privacy-oriented setting, where the privacy budget serves as an input. This budget directly influences the number of bits used for quantization in each round (see Eq. (15)). Next, we will shift our focus to a communication-oriented perspective, which is also common in distributed learning, particularly when bandwidth is limited. Specifically, we restrict each dimension of the update to be represented using only b bits. By summing over all dimensions d, communication rounds K, and selected clients B, the total communication cost of the system is capped at $KBdb$ bits. In this case, the achievable privacy level is dictated by the communication budget. Specifically, the noise introduced by the b quantization bits, denoted as σ_b, is calculated as $\frac{\eta_l Q G_I}{\sqrt{2\ln 2}(2^b-1)}$ (Lemma 3). Using Lemma 1 again, we can determine that the algorithm can achieve a $(\frac{2S(2^b-1)\sqrt{2\ln 2KB\ln[1/\delta]}}{N\eta_l Q G_I}, \delta)$-level of CLDP. When both privacy and communication budgets are posed, we need to choose the more stringent of them to determine the noise level, i.e., $\max\{\sigma_\epsilon, \sigma_b\}$, to meet the requirements of both budgets simultaneously. The impact of different budgets on the performance of Alg. 1 is summarized in Table I.

TABLE I
PERFORMANCE OF LRQ-SGD WITH DIFFERENT [PRIVACY BUDGET, COMMUNICATION BUDGET] SETTINGS.

	$[(\epsilon,\delta),-]$	$[-,b]$	$[(\epsilon,\delta),b]$
σ_ϵ	$\sigma_\epsilon = \frac{2S\sqrt{KB\ln[1/\delta]}}{N\epsilon}$	$\sigma_b = \frac{\eta_l QG_I}{\sqrt{2\ln 2}(2^b-1)}$	$\max\{\sigma_\epsilon, \sigma_b\}$
CLDP-level	(ϵ,δ)	$(\frac{2S(2^b-1)\sqrt{2\ln 2KB\ln[1/\delta]}}{N\eta_l QG_I}, \delta)$	(ϵ,δ)
Comm. Cost	$KBd\log_2\left[\frac{\eta_l QG_I N\epsilon}{2S\sqrt{2\ln 2KB\ln[1/\delta]}}+1\right]$	$KBdb$	$KBdb$
$\mathcal{E}_{privacy}{}^1$	$O(d/\epsilon^2)$	$O(d/(2^b-1)^2)$	$O(\max\{d/\epsilon^2, d/(2^b-1)^2\})$

Remark 2. *In Table I, we only compare the convergence error of $\mathcal{E}_{privacy}$ because privacy and communication budget only affect this term. Additionally, when $S \geq \eta_l QG_T$, the $\mathcal{E}_{clip}$ in Theorem 1 becomes zero. Therefore, the following sections of this paper will focus on this simplified version of the convergence error.*

$$\mathcal{E} \leq \mathcal{E}_{LSGD} + \frac{4S^2\nu\eta_g dK\ln[1/\delta]}{\eta_l QN^2\epsilon^2} \tag{17}$$

which mitigates the influence of the first two terms.

C. Discussion with related works

In this subsection, we compare our proposed algorithm with another method that can achieve both communication reduction and privacy protection, specifically focusing on the theoretical convergence guarantee. Particularly, we select [19], [34], which naively integrates two techniques: initially injecting continuous Gaussian noise into the raw updates to meet privacy requirements (See Gaussian Mechanism in Lemma 1), followed by quantizing the perturbed updates to meet the communication constraints. To ensure a fair comparison of algorithm performance, we follow the conclusions drawn from Theorem 1. For the two algorithms mentioned in this section, we set a fixed privacy budget of (ϵ,δ) and a communication budget of $b_\epsilon = \log_2\left[\frac{\eta_l QG_I N\epsilon}{2S\sqrt{2\ln 2KB\ln[1/\delta]}}+1\right]$, and then compare their convergence errors. In our paper, we refer to this approach as Quantized Gaussian mechanism SGD (QG-SGD), summarized in Alg. VIII-B. To comply with the privacy budget, we add Gaussian noise sampled from $\mathcal{N}(0, \sigma^2 = \frac{4S^2KB\ln(1/\delta)}{N^2\epsilon^2})$ to the clipped model updates to achieve (ϵ,δ)-CLDP. Subsequently, each private model update is quantized to b_ϵ bits using a stochastic quantizer $\mathcal{Q}_{b_\epsilon}[\cdot]$ [4] as per [4] to meet the communication constraint.

Lemma 4 (Performance of QG-SGD)**.** *For an N-client distributed learning problem with clipping threshold $S \geq \eta_l QG$, by setting privacy budget (ϵ,δ) and communication budget b_ϵ, the convergence error of Alg. VIII-B for smooth objective is upper bounded by:*

$$\mathcal{E} \leq \mathcal{E}_{LSGD} + \underbrace{\frac{4S^2\nu\eta_g dK\ln[1/\delta]}{\eta_l QN^2\epsilon^2}}_{\mathcal{E}_{Privacy}} + \underbrace{\frac{2\ln 2\nu\eta_g dS^2KB\ln(1/\delta)}{N^2\epsilon^2 B\eta_l Q}}_{\mathcal{E}_{Quantization}}$$
$$+ \underbrace{\frac{8dB\nu\eta_g\ln 2S^4K^2B^2\ln(1/\delta)^2}{N^4\epsilon^4\eta_l QG_I^2}}_{\mathcal{E}_{Coupling}}, \tag{18}$$

It is important to note that the potential privacy benefits of quantization alone are not considered in these types of approaches. We find the $\mathcal{E}_{LSGD}$ and the privacy error in Eq. (18) are the same as those in Eq. (14). There exist extra quantization error, determined by the stochastic quantizer $\mathcal{Q}_b[\cdot]$, and the coupling error between quantization and privacy. Since our LRQ-SGD realizes communication reduction and privacy protection simultaneously **only** through quantization operation, the quantization error does not appear in Eq. (14), which means a smaller convergence error. Although the idea of combining quantization and continuous noise is not novel, our work is the first to analyze the convergence of such algorithms and to quantify the impact of noise and quantization bits. The full proof is shown in Appendix VIII-B.

V. DYNAMIC LAYERED RANDOMIZED QUANTIZED LOCAL SGD

In LRQ-SGD, the privacy budget ϵ is equally distributed among K communication rounds, resulting in a constant σ for each round throughout the training process. However, this approach is relatively idealized. In this section, we consider two features of the distributed AI training tasks: 1) the dynamic nature of gradients and 2) varying communication conditions within the training process. By taking two factors into account, we aim to develop dynamic allocation strategies for privacy budgets to improve convergence performance.

A. Effect of dynamic parameters of LRQ

Based on Definition 3, to implement a dynamic privacy budget allocation, we need to adjust the variance σ_k of the Gaussian distribution used for sampling q_{step} in each round k. Specifically, instead of taking the fixed σ_ϵ for the input of LRQ, we use a set of noise variance $\{\sigma_0, ..., \sigma_{K-1}\}$ for communication round $k \in 0, K-1$, leading to a dynamic version of LRQ, termed as DLRQ. We relax privacy budget ϵ to explore the impact of DLRQ on the privacy and convergence performance of Alg. 1. The results are presented in the following two lemmas:

Lemma 5 (Dynamic Moment Accountant)**.** *With a set of $\{\sigma_0, ..., \sigma_{K-1}\}$, the DLRQ-SGD satisfied (ϵ',δ)-CLDP with:*

$$\epsilon' = \frac{2S\sqrt{B\ln[1/\delta]}}{N} \cdot \sqrt{\sum_{k=0}^{K-1}\frac{1}{\sigma_k^2}} \tag{19}$$

proof outline: We define the privacy loss at O as

$$c(O, D, D') \triangleq \ln \frac{\Pr(\mathcal{M}(D) = O)}{\Pr(\mathcal{M}(D') = O)}$$

where $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\{\mathcal{M}_0, \mathcal{M}_1, ..., \mathcal{M}_{K-1}\}$. Then the moment generating function evaluated at the value λ: $\beta_{\mathcal{M}}(\lambda) \triangleq \max_{D,D'} \ln \mathbb{E}_{O\sim\mathcal{M}(D)}[\exp(\lambda c(O, D, D'))]$. Using the tail bound of $\beta_{\mathcal{M}}(\lambda)$:

$$\delta = \min_{\lambda} \exp(\beta_{\mathcal{M}}(\lambda) - \lambda\epsilon) \overset{(a)}{\leq} \min_{\lambda} \exp(\sum_{k=0}^{K-1} \beta_{\mathcal{M}_k}(\lambda) - \lambda\epsilon)$$

$$\overset{(b)}{\leq} \min_{\lambda} \exp(\sum_{k=0}^{K-1} \frac{\frac{B^2}{N^2}\lambda^2}{\frac{\sigma_k^2}{B}\frac{B^2}{S^2}} - \lambda\epsilon) \leq \exp(-\frac{N^2\epsilon^2}{4S^2B\sum_{k=0}^{K-1}\frac{1}{\sigma_k^2}})$$

where (a) considers the sequence of mechanisms is independent, and (b) using the Lemma 3 in [30].

Lemma 6. *With a set of* $\{\sigma_0, ..., \sigma_{K-1}\}$*, the convergence error of DLRQ-SGD for the smooth objective is upper bounded by*

$$\mathcal{E} \leq \mathcal{E}_{LSGD} + \underbrace{\frac{\nu\eta_g d}{B\eta_l Q\sum_{i=0}^{K-1}\tau^{-i}} \sum_{k=0}^{K-1} \tau^{-k}\sigma_k^2}_{\mathcal{E}_{privacy}} \tag{20}$$

The full proof of Lemma 5 and Lemma 6 are shown in Appendix VIII-C and VIII-A. If we fix $\sigma_k = \sigma$ for each round k, Lemma 5 degrades to Lemma 1 with $\epsilon' = \frac{2S\sqrt{KB\ln[1/\delta]}}{N\sigma}$; and Eq. (20) degrades to Eq. (17). We observe that recent update information has been assigned greater weight, suggesting that more recent model updates are considered more significant. This is particularly valid in distributed AI training tasks, where, as the training progresses, the model progressively learns and accumulates more knowledge. Next, we will use this insight to determine the values of σ_k for each round.

B. Dynamic privacy budget allocation with learning process

Next, we reintroduce the same (ϵ, δ) privacy budget as utilized in Alg. 1 to develop a theoretically optimal allocation strategy for σ_k. To facilitate this, we formulate the allocation of the privacy budget as a problem of minimizing convergence error, while adhering to the overall privacy constraints. Given that the privacy allocation specifically impacts the privacy error component $\mathcal{E}_{\text{privacy}}$, we further simplify the problem to focus solely on minimizing the privacy error:

$$\begin{aligned} \min_{\{\sigma_k^2\}} \quad & \sum_{k=0}^{K-1} \tau^{-k}\sigma_k^2 \\ s.t. \quad & \frac{2S\sqrt{B\ln[1/\delta]}}{N} \cdot \sqrt{\sum_{k=0}^{K-1}\frac{1}{\sigma_k^2}} = \epsilon, \end{aligned} \tag{21}$$

By solving the above optimization problem, we can determine the vector $\{\sigma_k^2\}_{k=0}^{K-1}$ for every communication round:

$$\boxed{\sigma_k^2 = \frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}(\sum_{i=0}^{K-1}\tau^{-i/2})\tau^{k/2}} \tag{22}$$

Then, we can modify Alg. 1 to a dynamic version by determining σ_k for the LRQ at the beginning of each communication round. Note that, if we take $\tau = 1$, then σ_k^2 will be fixed as $\frac{4S^2KB\ln[1/\delta]}{N^2\epsilon^2}$ for all K rounds, degrading to a fixed LRQ quantizer. If we consider a more general case with $\tau < 1$, we have

$$\sigma_k^2 = \frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}\frac{\tau^{-K/2}-1}{\tau^{-1/2}-1}\tau^{k/2} \tag{23}$$

Three factors determine the σ_k: (i) the total privacy budget (ϵ, δ), less privacy budget requires setting larger σ_k at each round; (ii) the total number of communication rounds K, more communication rounds causes less privacy budget at each round, and larger σ_k; (iii) the current communication round k, σ_k decreases as the training process goes on. This approach aligns with the conclusion reached by [29], which suggests that smaller compression ratios should be used during early training stages and gradually increased.

Theorem 2 (Performance of DLRQ-SGD)**.** *For an* N*-client distributed learning problem in Eq.* (1)*,the Dynamic LRQ-SGD satisfies the following.*
Privacy: *DLRQ-SGD is* (ϵ, δ)*-CLDP;*
Communication: *DLRQ-SGD incurs communication cost*

$$dB\sum_{k=0}^{K-1}\log_2\Big[\frac{\eta_l QG_I N\epsilon\tau^{-k/4}}{2S\sqrt{2\ln 2KB\ln[1/\delta]\sum_{i=0}^{K-1}\tau^{-i/2}}} + 1\Big]$$

Convergence: *By letting the learning rate satisfied* $\frac{\eta_l^2\nu^2Q(Q-1)}{2} + 2\nu\eta_g\eta_l Q \leq 1$*, the convergence error for the smooth objective is upper bounded by*

$$\mathcal{E} \leq \mathcal{E}_{LSGD} + \underbrace{\frac{4S^2\nu\eta_g dK\ln[1/\delta]}{\eta_l QN^2\epsilon^2} \cdot \frac{AM_K^2(\tau^{-k/2})}{QM_K^2(\tau^{-k/2})}}_{\mathcal{E}_{privacy}} \tag{24}$$

where $AM_K(\tau^{-k/2}) = \frac{\sum_{k=0}^{K-1}\tau^{-k/2}}{K}$ *is the Arithmetic Mean, and* $QM_K(\tau^{-k/2}) = \sqrt{\frac{\sum_{k=0}^{K-1}\tau^{-k}}{K}}$ *is the Quadratic Mean.*

We derive several observations from the above Theorem.

• **Privacy**. Take σ_k in Eq. (22) into Lemma 5, we find that the private model update at round k satisfied (ϵ_k, δ)-CLDP with

$$\epsilon_k = \frac{2S\sqrt{B\ln[1/\delta]}}{N} \cdot \sqrt{\frac{1}{\sigma_k^2}} = \epsilon\sqrt{\frac{1}{\sum_{i=0}^{K-1}\tau^{-i/2}}} \cdot \tau^{-k/4}$$

We can see that ϵ_k increases as the training process goes on, which means we allocate less privacy budget (i.e., larger σ_k^2) at the early stage of training and increase the privacy budget as the training goes on. This is consistent with the conclusion drawn by [20], [35] through a heuristic algorithm. The dynamic information in the gradients at various training stages can explain the above allocation strategy. Initially, when the model is far from convergence, gradients are large and distinct, allowing for easier signal extraction from attackers. To maintain privacy, significant noise is added to these gradients. As the model approaches convergence, individual models across clients become more similar, reducing gradient distinctiveness and the risk of information leakage. Consequently, a larger privacy budget (ϵ_k) can be allocated, decreasing the need for excessive noise. This reduction not only preserves privacy but also maintains gradient update fidelity, improving model accuracy and convergence.

• **Communication**. When using DLRQ, the corresponding quantization bits also change across training. Specifically, the quantization bits of each element is upper bounded as

$$\begin{aligned} b_k^{(i)} &= \log_2 [\frac{2\eta_l Q G_I}{2\sigma_k \sqrt{2\ln 2}} + 1] \\ &= \log_2 \Big[\frac{\eta_l Q G_I N \epsilon \tau^{-k/4}}{2S\sqrt{2\ln 2 K B \ln[1/\delta] \sum_{i=0}^{K-1} \tau^{-i/2}}} + 1\Big] \end{aligned}$$

We still use fixed-length coding to transmit $\mathbf{m}_k^{(i)}$, which means that each element is represented by $b_k^{(i)}$ bits. To calculate the total communication cost of Dynamic LRQ-SGD, we sum over d dimensional elements in each quantized model update, B clients, and K communication rounds. The number of bits b used for quantization dynamically changes throughout the training process. In the early stages of training, a smaller number of bits is employed for quantization, while in the later stages, a larger number of bits is allocated. This approach is based on the rationale that as the model approaches convergence and the gradients near their optimal values in the later stages, it becomes crucial to maintain the model's accuracy (fidelity) to avoid the loss of valuable information.

• **Convergence**. The convergence error bound has an analogy format with that in Theorem 1. The first item is generated by full-communication and privacy-free local SGD. In comparing the privacy error between Eq. (24) and Eq. (17), we can observe that the latter is multiplied by an additional factor of $\frac{AM_K^2(\tau^{-k/2})}{QM_K^2(\tau^{-k/2})}$. It is worth noting that $0 < \tau < 1$, which means that $AM_K(\tau^{-k/2})$ is always smaller than $QM_K(\tau^{-k/2})$. This suggests that dynamic LRQ-SGD can achieve lower convergence error when compared to fixed LRQ-SGD. The proof is shown in Appendix VIII-A.

C. Dynamic privacy budget allocation with varying communication rounds

Previous discussions on the strategy for allocating σ whether fixed or dynamic, depend on knowing the number of communication rounds K. Our analysis also indicates that frequent communication results in fewer quantization bits available per round, thereby increasing quantization noise to enhance privacy protection. However, in practical wireless distributed learning scenarios, the number of communication rounds K can be changeable during the training process.

Firstly, real-time learning performance can directly influence K. For instance, the early stop technique [21] might be employed to achieve a predetermined accuracy level prematurely, while a significant gap from the target performance may necessitate additional communication rounds from the clients. Secondly, real-time system performance can also affect K; for example, if a client is at a low battery level, the system might opt to reduce the number of communication rounds or pause local training. To accommodate the changeable communication rounds, we have designed an alternative dynamic privacy budget allocation strategy. The following Corollary shows how to adjust to changes in K to maintain privacy protection requirements:

Corollary 1 (Feedback Mechanisms for Learning-oriented Communications)**.** *Assuming the initial set number of communication rounds is K_1. If training progresses to the J-th round, where $J < K_1$, we adjust the number of communication rounds to K_2, the per-round privacy budget is allocated based on the following criteria:*

i) for $k = 1, 2, ..., J$,

$$\sigma_k^2 = \frac{4S^2 B \ln[1/\delta]}{N^2\epsilon^2} (\sum_{i=0}^{K_1-1} \tau^{-i/2}) \tau^{k/2}; \tag{25}$$

ii) $k = J+1, ..., K_2$,

$$\begin{aligned} \sigma_k^{'2} &= \frac{4S^2 B \ln[1/\delta]}{N^2\epsilon^2} (\sum_{i=0}^{K_1-1} \tau^{-i/2}) \frac{\sum_{i=J}^{K_2-1} \tau^{-i/2}}{\sum_{i=J}^{K_1-1} \tau^{-i/2}} \cdot \tau^{k/2} \\ &= \frac{\sum_{i=J}^{K_2-1} \tau^{-i/2}}{\sum_{i=J}^{K_1-1} \tau^{-i/2}} \sigma_k^2, \end{aligned} \tag{26}$$

We observe that, in contrast to σ_k^2, $\sigma_k^{'2}$ introduces an additional correction coefficient $\frac{\sum_{i=J}^{K_2-1} \tau^{-i/2}}{\sum_{i=J}^{K_1-1} \tau^{-i/2}}$. If $K_2 < K_1$, so $\frac{\sum_{i=J}^{K_2-1} \tau^{-i/2}}{\sum_{i=J}^{K_1-1} \tau^{-i/2}} < 1$, which indicates that as the remaining number of communication rounds decreases, the privacy budget for each round increases, which means that the amount of noise required to be added in each round decreases. The proof is shown in Appendix VIII-D.

VI. Experiments

In this section, we conduct experiments on MNIST, CIFAR-10 and CIFAR-100 to empirically validate our proposed LRQ-SGD and DLRQ-SGD methods. The MNIST consists of 70000 $1 \times 28 \times 28$ grayscale images in 10 classes. The CIFAR-10 dataset consists of 60000 $3 \times 32 \times 32$ color images in 10 classes, and the CIFAR-100 dataset consists of 60000 $3 \times 32 \times 32$ color images in 100 classes. We compare our proposed methods with the following baselines: 1) **Gau-SGD** [24]: adds Gaussian DP noise sampled from $\mathcal{N}(0, \sigma_\epsilon^2 = \frac{4S^2 K B \ln(1/\delta)}{N^2\epsilon^2})$ to achieve (ϵ, δ)-CLDP and then sends the private model update using the full-precision floating-point $b_{init} = 32$ to the server. Gau-SGD only considers privacy protection; 2) **QG-SGD**: adds Gaussian noise sampled from $\mathcal{N}(0, \sigma_\epsilon^2 = \frac{4S^2 K B \ln(1/\delta)}{N^2\epsilon^2})$ to achieve (ϵ, δ)-CLDP and then quantizes the private model update to b^* bits; 3) **Local SGD**: as the oracle, clients send noise-free and non-compressed model updates to server.

Experimental Setting. We select the momentum SGD as an optimizer, where the momentum is set to 0.9, and weight decay is set to 0.0005. Following the setup of [18], [24], the samples on one client can overlap with those on the other clients the samples on each client are allocated C categories of samples . Following a procedure proposed by [24], in each

[1] For the MNIST and CIFAR-10 datasets, an IID setting is represented by $C = 10$, while a non-IID setting corresponds to $C < 10$. For the CIFAR-100 dataset, $C = 100$ denotes an IID setting, and $C < 100$ indicates a non-IID setting.

communication round, we calculate the median norm of all unclipped updates and use this as the clipping bound $S = \text{median}\{\|\boldsymbol{\Delta}_k^{(i)}\|_2, i \in \mathcal{B}_k\}$. Following the setup of [29], we estimate τ as $\tau_{est} = \left[\frac{F(\boldsymbol{\theta}_k)}{F(\boldsymbol{\theta}_0)}\right]^{1/k}$. Other experimental details are given in Table II.

TABLE II
EXPERIMENT SETTING.

Dataset	MNIST	CIFAR-10	CIFAR-100
Net	LeNet	Resnet 18	Resnet 34
Model Size	6×10^4	1×10^7	3×10^7
Learning Rate	0.01	0.01	0.01
Batch Size	32	32	32
Number of Clients	1920	9600	48000
Local Data size	500	800	800
Participated Clients	80	80	80

Testing Performance. Figure 4 shows the test accuracy of LRQ-SGD for different privacy constraints on MNIST, CIFAR-10, and CIFAR-100. For MNIST, the Local SGD can achieve a test accuracy of 0.9830 without privacy protection and incur a communication cost of 768 MB. Then we set $\delta = 1 \times 10^{-5}$ and $\epsilon = 1, 2, 3$, and we keep track of privacy loss using the privacy accountant, and training is stopped once δ reaches 1×10^{-5}. The LRQ-SGD can achieve the test accuracy of 0.8711, 0.9487, and 0.9642, respectively, and incurs the communication cost of 16.8 MB, 30 MB, and 36 MB, respectively. We can see that with the relaxation of the privacy budget (i.e., an increase in ϵ), the added noise decreases, the number of communication rounds increases, and the accuracy of model testing accuracy improves. However, the corresponding communication cost will increase. Similar results can be seen in CIFAR-10 and CIFAR-100. For CIFAR-10, the Local SGD can achieve a test accuracy of 0.9215 without privacy protection and incur the communication cost of 1280 GB. Then we set $\delta = 1 \times 10^{-5}$ and $\epsilon = 20, 40, 60$, and the LRQ-SGD can achieve the test accuracy of 0.6281, 0.7337, and 0.7987, respectively. And LRQ-SGD incurs the communication cost of 43.2 GB, 60 GB, and 127.2 GB, respectively. For CIFAR-100, the Local SGD can achieve a test accuracy of 0.9207 without privacy protection and incur the communication cost of 2400 GB. Then we set $\delta = 1 \times 10^{-5}$ and $\epsilon = 30, 50, 70$, and the LRQ-SGD can achieve the test accuracy of 0.4776, 0.6136 and 0.7691, respectively. And LRQ-SGD incurs the communication cost of 81.6 GB, 158.4 GB, and 235.2 GB, respectively.

Privacy-Learning Tradeoff. Figure 5 shows the tradeoff between the privacy budget and the learning performance in terms of test accuracy on different datasets. We compare this tradeoff between our proposed two algorithms and the naive algorithm, QG-SGD. All three algorithms show a privacy-learning tradeoff; that is, the more privacy budget can be used, the higher test accuracy can be achieved. The marginal utility (how much test accuracy is improved from the increased communication budget) is diminishing. That means when the privacy budget is small, increasing the privacy budget can bring significant improvement. When the privacy budget is large (for example, $\epsilon > 1.5$ in MNIST), the improvement of the test accuracy by increasing the privacy budget is limited.

Dynamically Determine the LRQ Parameter σ_k. Figure 6 shows the how σ_k changes for $k = 0, ..., K-1$. As we stated before, a small privacy budget leads to a large σ_k, that is, a large quantization error (i.e., noise) added to the clipped model. We can see that DLRQ-SGD allocates less privacy budget (i.e., add more noise on clipped model) at the early stage of training and reduces noise level as the training goes on. The main reason is that the model update noise in the later stage of training has a greater impact on the convergence error. We have to reduce the variance of model update noise to ensure better convergence of the algorithm. This result is similar to some heuristic work [20], [35]. Specifically, σ_k decreased from 0.19 to 0.08 on MNIST and decreased from 0.020 to 0.012 on CIFAR-10. We can see that the added noise of CIFAR-10 is smaller than that of MNIST. The reason is that the model of CIFAR-10 is more complicated than that of MNIST.

TABLE III
TEST ACCURACY OF LRQ-SGD ON MNIST WITH DIFFERENT NON-IID DEGREES.

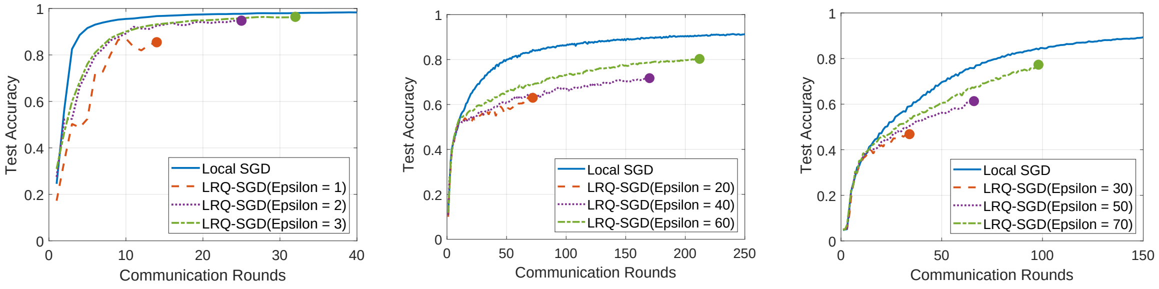
Classes Per Clients	C=1	C=2	C=4	C=10
Local SGD	0.2317	0.9202	0.9701	0.9830
LRQ-SGD	-	0.8481	0.9400	0.9642
DLRQ-SGD	-	0.8613	0.9571	0.9711

TABLE IV
TEST ACCURACY OF LRQ-SGD ON MNIST WITH DIFFERENT CLIENTS.

Number of Clients	960	1920	3840	5760
Test Accuracy	0.8711	0.9488	0.9663	0.9745

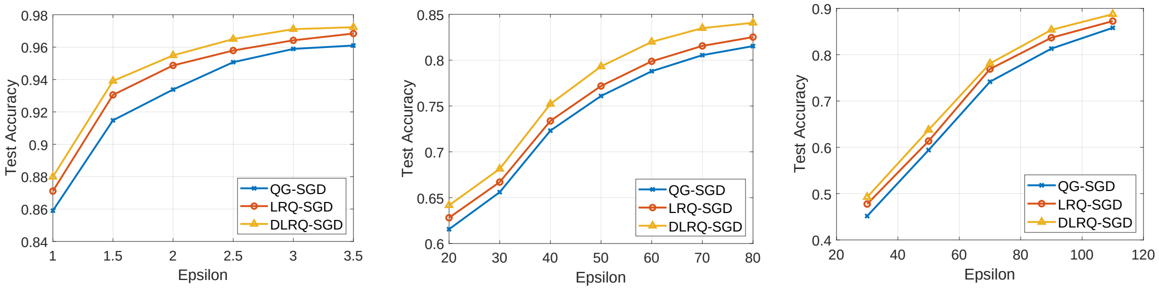
Different Non-IID Degrees. We take the privacy budget as $(3, 1 \times 10^{-5})$, fix the number of participated clients each round as $B = 80$ and the total number of clients $N = 1920$. Then we reduce the value of C to generate data distributions with increasing non-IID level and show the test accuracy on MNIST dataset in Table III. From Table III, we observe a corresponding gradual increase in the gap between LRQ-SGD and Local SGD as the degree of non-IID data increases. Specifically, in an IID setting where $C = 10$, the difference between LRQ-SGD and Local SGD is 0.0188. However, when $C = 2$, this difference escalates to 0.0721. This trend can be attributed to the fact that as the degree of non-IID data increases, the disparity among different clients also increases. Consequently, the bias introduced by the clipping operation increases, which in turn affects the convergence error. This observation also validates Theorem 1. Notably, even when $C = 2$, LRQ-SGD still manages to achieve a testing accuracy of 0.8481. Furthermore, in the non-IID scenario, the application of dynamic compression can further enhance the performance of LRQ-SGD.

The Number of Clients. We take the privacy budget as $(2, 1 \times 10^{-5})$ and fix the number of participated clients each round as $B = 80$. Then we vary the total number of clients N from 960 to 5760 and show the test accuracy on MNIST dataset in Table IV. From Table IV, we can see that



(a) MNIST (b) CIFAR-10 (c) CIFAR-100

Fig. 4. Model Performance on Different Datasets. (Dots at the end of accuracy curves indicate that the δ threshold was reached and training therefore stopped.)

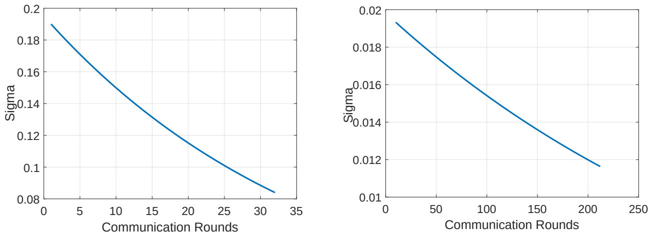


(a) MNIST (b) CIFAR-10 (c) CIFAR-100

Fig. 5. Privacy-Learning Tradeoff on Different Datasets.

TABLE V
PERFORMANCE COMPARISON WITH SOTA ON MNIST, CIFAR-10 AND CIFAR-100. – MEANS THAT THE ALGORITHM CANNOT CONVERGE. CC DENOTES THE ARISING COMMUNICATION COSTS

Algorithm	MNIST $(3, 1\times10^{-5})$		CIFAR-10 $(60, 1\times10^{-5})$		CIFAR-100 $(70, 1\times10^{-5})$	
	CC	Acc	CC	Acc	CC	Acc
Local SGD	768 MB	0.9830	1280 GB	0.9215	2400 GB	0.9207
Gau-SGD	576 MB	0.9642	678.4 GB	0.7987	940.8 GB	0.7691
QG-SGD	36 MB	0.9579	127.2 GB	0.7879	235.2 GB	0.7417
LRQ-SGD (ours)	36 MB	**0.9642**	127.2 GB	**0.7987**	235.2 GB	**0.7691**
DLRQ-SGD (ours)	36 MB	**0.9711**	127.2 GB	**0.8199**	235.2 GB	**0.7811**



(a) MNIST, $(3, 1\times10^{-5})$-CLDP. (b) CIFAR-10, $(50, 1\times10^{-5})$-CLDP.

Fig. 6. Noise level with rounds of Dynamic LRQ-SGD.

more clients can improve the global model's performance. Specifically, when we set the number of clients as 5760, the test accuracy of LRQ-SGD can achieve 0.9745, almost reaching the accuracy of Local SGD. This suggests that for scenarios where many parties are involved, differential privacy comes at nearly no cost in model performance.

Performance Comparison with SOTA. In Table V, we compare the performance of our proposed LRQ-SGD and DLRQ-SGD with some selected algorithms on MNIST, CIFAR-10, and CIFAR-100. The Local SGD, without privacy and communication constraints, provides a benchmark of the testing performance. For MNIST, we set the same privacy budget $(3, 1\times10^{-5})$ for all privacy-preserving algorithms. We first observe that our proposed LRQ-SGD achieves comparable accuracy of 0.9642 as Gau-SGD while the communication cost consumed by LRQ-SGD is only 36MB, far less than the 576MB of Gau-SGD. DLRQ-SGD achieves the highest testing accuracy of 0.9711. Compared to Local SGD, we reduced the cost of communication by 95.3% but only incur a performance impairment of 0.0119. In addition, benefit from the strategy of dynamic privacy budget allocation, the performance of DLRQ-SGD is higher than that of LRQ-SGD. For CIFAR-10 and CIFAR-100, we set the privacy budget $(60, 1\times10^{-5})$ and $(70, 1\times10^{-5})$ for all privacy-preserving algorithms. Similar to the performance on the MNIST dataset, the LRQ-SGD algorithm outperforms GQ-SGD but slightly worse than DLRQ-SGD.

VII. Conclusion

In this paper, we proposed a new quantization-aided distributed SGD algorithm called LRQ-SGD that could simultaneously achieve communication efficiency and privacy protection. We theoretically capture the trade-offs between communication, privacy, and convergence error. To further enhance the convergence performance, we designed a dynamic privacy budget/quantization step allocation strategy by formulating and solving an optimal problem with respect to minimizing the convergence error bound. Experimental evaluation of various machine learning tasks demonstrates our proposed algorithms outperform the benchmarks.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[2] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[3] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

[4] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "Qsgd: Communication-efficient sgd via gradient quantization and encoding," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1709–1720, 2017.

[5] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified sgd with memory," *Advances in neural information processing systems*, vol. 31, 2018.

[6] L. Zhu and S. Han, "Deep leakage from gradients," in *Federated learning*. Springer, 2020, pp. 17–31.

[7] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1322–1333.

[8] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[9] N. Agarwal, A. T. Suresh, F. X. X. Yu, S. Kumar, and B. McMahan, "cpSGD: Communication-efficient and differentially-private distributed sgd," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[10] P. Kairouz, Z. Liu, and T. Steinke, "The distributed discrete gaussian mechanism for federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5201–5212.

[11] N. Agarwal, P. Kairouz, and Z. Liu, "The skellam mechanism for differentially private federated learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 5052–5064, 2021.

[12] G. Yan, T. Li, K. Wu, and L. Song, "Killing two birds with one stone: Quantization achieves privacy in distributed learning," *Digital Signal Processing*, vol. 146, p. 104353, 2024.

[13] M. Hegazy, R. Leluc, C. T. Li, and A. Dieuleveut, "Compression with exact error distribution for federated learning," *arXiv preprint arXiv:2310.20682*, 2023.

[14] Y. Youn, Z. Hu, J. Ziani, and J. Abernethy, "Randomized quantization is all you need for differential privacy in federated learning," in *Federated Learning and Analytics in Practice: Algorithms, Systems, Applications, and Opportunities*, 2023.

[15] D. B. Wilson, "Layered multishift coupling for use in perfect sampling algorithms (with a primer on cftp)," *Monte Carlo Methods 26 (2000): 141-176.*, vol. 26, pp. 141–176, 2000.

[16] M. Hegazy and C. T. Li, "Randomized quantization with exact error distribution," *IEEE Information Theory Workshop (ITW)*, p. 31, 2022.

[17] A. Cheng, P. Wang, X. S. Zhang, and J. Cheng., "Differentially private federated learning with local regularization and sparsification." *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10 122–10 131, 2022.

[18] X. Zhang, X. Chen, M. Hong, Z. S. Wu, and J. Yi, "Understanding clipping for federated learning: Convergence and client-level differential privacy," *International Conference on Machine Learning*, 2022.

[19] H. Zong, Q. Wang, X. Liu, Y. Li, and Y. Shao, "Communication reducing quantization for federated learning with local differential privacy mechanism," in *2021 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2021, pp. 75–80.

[20] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1656–1665.

[21] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007.

[22] S. Shi, Q. Wang, K. Zhao, Z. Tang, Y. Wang, X. Huang, and X. Chu, "A distributed synchronous sgd algorithm with global top-k sparsification for low bandwidth networks," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 2238–2247.

[23] 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 *International Conference on Machine Learning*. PMLR, 2020, pp. 8253–8265.

[24] R. C. Geyer, T. Klein, and M. Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[25] W.-N. Chen, A. Ozgur, and P. Kairouz, "The poisson binomial mechanism for unbiased federated learning with secure aggregation," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3490–3506.

[26] 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*, 2018, pp. 3219–3227.

[27] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *Siam Review*, vol. 60, no. 2, pp. 223–311, 2018.

[28] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu, "Doublesqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6155–6165.

[29] G. Yan, T. Li, S.-L. Huang, T. Lan, and L. Song, "AC-SGD: Adaptively compressed sgd for communication-efficient distributed learning," *IEEE Journal on Selected Areas in Communications*, vol. 40, pp. 2678–2693, 2022.

[30] 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*, 2016, pp. 308–318.

[31] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Transactions on Communication Technology*, vol. 12, no. 4, pp. 162–165, 1964.

[32] R. M. Gray and T. G. Stockham, "Dithered quantizers," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 805–812, 1993.

[33] N. Paladi, M. Tiloca, P. N. Bideh, and M. Hell, "On-demand key distribution for cloud networks." *In 24th Conference on Innovation in Clouds, Internet and Networks and Workshops*, 2021.

[34] N. Mohammadi, J. Bai, Q. Fan, Y. Song, Y. Yi, and L. Liu, "Differential privacy meets federated learning under communication constraints," *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22 204–22 219, 2021.

[35] X. Y. Gong M, Feng J, "Privacy-enhanced multi-party deep learning." *Neural Networks*, vol. 121, pp. 484–496, 2020.

[36] G. Yan, S.-L. Huang, T. Lan, and L. Song, "DQ-SGD: Dynamic quantization in sgd for communication-efficient distributed learning," in *2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*. IEEE, 2021, pp. 136–144.

VIII. Appendix

A. Proof of Theorem 1 and Lemma 6

We have $\boldsymbol{\Delta}_k^{(i)} = \boldsymbol{\theta}_{k,Q}^{(i)} - \boldsymbol{\theta}_{k,0}^{(i)} = -\eta_l \sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)}$, and $\tilde{\boldsymbol{\Delta}}_k^{(i)} = \frac{\boldsymbol{\Delta}_k^{(i)}}{\max\{1, \|\boldsymbol{\Delta}_k^{(i)}\|_2/S\}}$, then we define folloing quantities to simplify notation:

$$\widetilde{\beta}_k^{(i)} := \frac{1}{\max\{1, \eta_l \| \sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)} \|_2/S\}}, \quad \check{\beta}_k := \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\max\{1, \eta_l \|\mathbb{E}[\sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)}]\|_2/S\}},$$

$$\tilde{\boldsymbol{\Delta}}_k^{(i)} = -\eta_l \sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)} \cdot \tilde{\beta}_k, \quad \check{\boldsymbol{\Delta}}_k^{(i)} = -\eta_l \sum_{q=0}^{Q-1} \boldsymbol{g}_{k,q}^{(i)} \cdot \check{\beta}_k, \quad \breve{\boldsymbol{\Delta}}_k^{(i)} = -\eta_l \sum_{q=0}^{Q-1} \nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)}) \cdot \check{\beta}_k,$$

Firstly, we consider function F is ν-smooth, and use Eq. (6), we have:

$$F(\boldsymbol{\theta}_{k+1}) \leq F(\boldsymbol{\theta}_k) + \langle \nabla F(\boldsymbol{\theta}_k), \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k \rangle + \frac{\nu}{2} \|\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k\|^2.$$

For the LRQ-SGD, $\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k = \frac{\eta_g}{B} \sum_{i \in \mathcal{B}_k} [\tilde{\boldsymbol{\Delta}}_k^{(i)} + \boldsymbol{n}_k^{(i)}]$, where $\boldsymbol{n}_k^{(i)} \sim N(0, \sigma_k^2 \boldsymbol{I})$. so:

$$\begin{aligned} \mathbb{E}F(\boldsymbol{\theta}_{k+1}) - F(\boldsymbol{\theta}_k) &\leq \eta_g \Big\langle \nabla F(\boldsymbol{\theta}_k), \frac{1}{B} \sum_{i \in \mathcal{B}_k} \mathbb{E}[\tilde{\boldsymbol{\Delta}}_k^{(i)} + \boldsymbol{n}_k^{(i)}] \Big\rangle + \frac{\nu \eta_g^2}{2} \mathbb{E} \| \frac{1}{B} \sum_{i \in \mathcal{B}_k} [\tilde{\boldsymbol{\Delta}}_k^{(i)} + \boldsymbol{n}_k^{(i)}] \|^2 \\ &= \eta_g \Big\langle \nabla F(\boldsymbol{\theta}_k), \frac{1}{B} \sum_{i \in \mathcal{B}_k} \mathbb{E}[\tilde{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle + \frac{\nu \eta_g^2}{2} \mathbb{E} \| \frac{1}{B} \sum_{i \in \mathcal{B}_k} \tilde{\boldsymbol{\Delta}}_k^{(i)} \|^2 + \frac{\nu \eta_g^2 d \sigma_k^2}{2B} \end{aligned} \tag{27}$$

Firstly, we analyze the first term of the right-hand Eq. (27)

$$\begin{aligned} \Big\langle \nabla F(\boldsymbol{\theta}_k), \frac{1}{B} \sum_{i \in \mathcal{B}_k} \mathbb{E}[\tilde{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle &= \Big\langle \nabla F(\boldsymbol{\theta}_k), \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} \tilde{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle \\ &= \Big\langle \nabla F(\boldsymbol{\theta}_k), \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} (\tilde{\boldsymbol{\Delta}}_k^{(i)} - \check{\boldsymbol{\Delta}}_k^{(i)})] \Big\rangle + \Big\langle \nabla F(\boldsymbol{\theta}_k), \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} \check{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle \\ &\overset{(a)}{\leq} \eta_l G^2 \Big[\mathbb{E} \frac{1}{N} \sum_{i=1}^{N} (\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}) \Big] + \underbrace{\Big\langle \nabla F(\boldsymbol{\theta}_k), \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} \check{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle}_{A} \end{aligned}$$

where (a) comes from Assumption 3. We further upper bound A as

$$
\begin{aligned}
A &= \Big\langle \sqrt{\eta_l \breve{\beta}_k Q} \nabla F(\boldsymbol{\theta}_k), \frac{1}{\sqrt{\eta_l \breve{\beta}_k Q}} \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} \breve{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle \\
&\overset{(a)}{=} \Big\langle \sqrt{\eta_l \breve{\beta}_k Q} \nabla F(\boldsymbol{\theta}_k), \frac{1}{\sqrt{\eta_l \breve{\beta}_k Q}} \mathbb{E}[\frac{1}{N} \sum_{i=1}^{N} \breve{\boldsymbol{\Delta}}_k^{(i)}] \Big\rangle \\
&\overset{(b)}{=} \frac{-\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k)\|^2 - \frac{1}{2\eta_l \breve{\beta}_k Q} \mathbb{E}[\|\frac{1}{N} \sum_{i=1}^{N} \breve{\boldsymbol{\Delta}}_k^{(i)} \|^2] + \frac{\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k) + \frac{1}{Q \eta_l \breve{\beta}_k N} \mathbb{E} \sum_{i=1}^{N} \breve{\boldsymbol{\Delta}}_k^{(i)} \|^2 \\
&= \frac{-\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k)\|^2 - \frac{\eta_l \breve{\beta}_k}{2} \sum_{q=0}^{Q-1} \mathbb{E}[\|\frac{1}{N} \sum_{i=1}^{N} \nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2] + \frac{\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k) - \frac{1}{QN} \sum_{i=1}^{N} \mathbb{E} \sum_{q=0}^{Q-1} \nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2 \\
&= \frac{-\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k)\|^2 - \frac{\eta_l \breve{\beta}_k}{2} [\|\nabla F(\boldsymbol{\theta}_k\|^2 + \sum_{q=1}^{Q-1} \mathbb{E} \|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2] + \frac{\eta_l \breve{\beta}_k Q}{2} \| \frac{1}{QN} \sum_{i=1}^{N} \sum_{q=0}^{Q-1} [\nabla F_i(\boldsymbol{\theta}_k) - \mathbb{E} \nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})]\|^2 \\
&\overset{(c)}{\le} \frac{-\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k)\|^2 - \frac{\eta_l \breve{\beta}_k}{2} [\|\nabla F(\boldsymbol{\theta}_k\|^2 + \sum_{q=1}^{Q-1} \mathbb{E} \|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2] + \frac{\eta_l \breve{\beta}_k}{2N} \sum_{i=1}^{N} \sum_{q=0}^{Q-1} \|\nabla F_i(\boldsymbol{\theta}_k) - \mathbb{E} \nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2 \\
&\overset{(d)}{\le} \frac{-\eta_l \breve{\beta}_k Q}{2} \|\nabla F(\boldsymbol{\theta}_k)\|^2 - \frac{\eta_l \breve{\beta}_k}{2} [\|\nabla F(\boldsymbol{\theta}_k\|^2 + \sum_{q=1}^{Q-1} \mathbb{E} \|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2] + \frac{\eta_l \breve{\beta}_k \nu^2}{2} \sum_{q=0}^{Q-1} \|\boldsymbol{\theta}_k - \mathbb{E} \boldsymbol{\theta}_{k,q}^{(i)}\|^2
\end{aligned}
$$

where (a) comes from $\mathbb{E}\breve{\boldsymbol{\Delta}}_k^{(i)} = \breve{\boldsymbol{\Delta}}_k^{(i)}$, (b) is because $\langle a, b \rangle = -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2 + \frac{1}{2}\|a+b\|^2$, (c) comes from Jensen's inequality, and (d) comes from ν-smothness. For the third term of above inequality, we have

$$
\begin{aligned}
\sum_{q=1}^{Q-1} \|\boldsymbol{\theta}_k - \mathbb{E}\boldsymbol{\theta}_{k,q}^{(i)}\|^2 &= \eta_l^2 \sum_{q=1}^{Q-1} \| \sum_{t=0}^{q-1} \mathbb{E} \boldsymbol{g}_{k,t}^{(i)} \|^2 \\
&\overset{(a)}{\le} \eta_l^2 \sum_{q=1}^{Q-1} q \sum_{t=0}^{q-1} \|\mathbb{E} \boldsymbol{g}_{k,t}^{(i)}\|^2 \\
&\overset{(b)}{\le} \eta_l^2 \sum_{q=1}^{Q-1} [q^2 \frac{\alpha_l^2}{A} + q \sum_{t=0}^{q-1} \|\mathbb{E} \nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2] \\
&\le \eta_l^2 \frac{\alpha_l^2}{A} \frac{Q(2Q-1)(Q-1)}{6} + \eta_l^2 \frac{Q(Q-1)}{2} [\|\nabla F_i(\boldsymbol{\theta}_k)\|^2 + \sum_{q=1}^{Q-1} \mathbb{E}_i \|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2] \\
&\overset{(c)}{\le} \eta_l^2 \frac{\alpha_l^2}{A} \frac{Q(2Q-1)(Q-1)}{6} + \eta_l^2 \frac{Q(Q-1)}{2} \Big[\|\nabla F(\boldsymbol{\theta}_k)\|^2 + \alpha_g^2 + \sum_{q=1}^{Q-1} \mathbb{E}_i \|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2 \Big]
\end{aligned}
$$

where (a) comes from Jensen's inequality, and (b) and (c) come from Assumption 1. Then we analyze the second term of the right-hand Eq. (27)

$$\begin{aligned}
\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}\tilde{\boldsymbol{\Delta}}_k^{(i)}\|^2 &\overset{(a)}{\leq} 2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\boldsymbol{\Delta}}_k^{(i)}-\check{\boldsymbol{\Delta}}_k^{(i)}]\|^2+2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}\check{\boldsymbol{\Delta}}_k^{(i)}\|^2\\
&=2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\boldsymbol{\Delta}}_k^{(i)}-\check{\boldsymbol{\Delta}}_k^{(i)}]\|^2+2\eta_l^2\check{\beta}_k^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}\sum_{q=0}^{Q-1}\boldsymbol{g}_{k,q}^{(i)}\|^2\\
&\overset{(b)}{\leq} 2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\boldsymbol{\Delta}}_k^{(i)}-\check{\boldsymbol{\Delta}}_k^{(i)}]\|^2+2\eta_l^2\check{\beta}_k^2Q\sum_{q=0}^{Q-1}\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}\boldsymbol{g}_{k,q}^{(i)}\|^2\\
&\overset{(c)}{\leq} 2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\boldsymbol{\Delta}}_k^{(i)}-\check{\boldsymbol{\Delta}}_k^{(i)}]\|^2+2\eta_l^2\check{\beta}_k^2Q\sum_{q=0}^{Q-1}\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)}\|^2+\frac{2\eta_l^2\check{\beta}_k^2Q^2\alpha_l^2}{AB}\\
&\overset{(d)}{\leq} 2G^2\eta_l^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)}-\check{\beta}_k^{(i)}]\|^2+2\eta_l^2\check{\beta}_k^2Q\sum_{q=0}^{Q-1}\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)}\|^2+\frac{2\eta_l^2\check{\beta}_k^2Q^2\alpha_l^2}{AB}\\
&=2G^2\eta_l^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)}-\check{\beta}_k^{(i)}]\|^2+2\eta_l^2\check{\beta}_k^2Q[\|\nabla F(\boldsymbol{\theta}_k\|^2+\sum_{q=1}^{Q-1}\|\mathbb{E}_i\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)}\|^2]+\frac{2\eta_l^2\check{\beta}_k^2Q^2\alpha_l^2}{AB}
\end{aligned}$$

where (a) is because $\|a+b\|^2\leq 2\|a\|^2+2\|b\|^2$, (b) come from Jensen's inequality, (c) come from Assumption 1, and (d) come from Assumption 3. Hence, we have

$$\begin{aligned}
&\mathbb{E}F(\boldsymbol{\theta}_{k+1})-F(\boldsymbol{\theta}_k)\\
&\leq \eta_g\eta_l G^2\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)}-\check{\beta}_k^{(i)})\Big]+\frac{-\eta_g\eta_l\check{\beta}_kQ}{2}\|\nabla F(\boldsymbol{\theta}_k)\|^2-\frac{\eta_g\eta_l\check{\beta}_k}{2}[\|\nabla F(\boldsymbol{\theta}_k\|^2+\sum_{q=1}^{Q-1}\mathbb{E}_i\|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2]\\
&+\frac{\eta_g\eta_l\check{\beta}_k\nu^2}{2}\eta_l^2\frac{\alpha_l^2}{A}\frac{Q(2Q-1)(Q-1)}{6}+\frac{\eta_g\eta_l\check{\beta}_k\nu^2}{2}\eta_l^2\frac{Q(Q-1)}{2}\Big[\|\nabla F(\boldsymbol{\theta}_k)\|^2+\alpha_g^2+\sum_{q=1}^{Q-1}\mathbb{E}_i\|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2\Big]\\
&+\frac{\nu\eta_g^2}{2}2G^2\eta_l^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)}-\check{\beta}_k^{(i)}]\|^2+\frac{\nu\eta_g^2}{2}2\eta_l^2\check{\beta}_k^2Q[\|\nabla F(\boldsymbol{\theta}_k\|^2+\sum_{q=1}^{Q-1}\|\mathbb{E}_i\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)}\|^2]+\frac{\nu\eta_g^2}{2}\frac{2\eta_l^2\check{\beta}_k^2Q^2\alpha_l^2}{AB}+\frac{\nu\eta_g^2d\sigma_k^2}{2B}
\end{aligned}$$

Hence

$$\begin{aligned}
&\mathbb{E}F(\boldsymbol{\theta}_{k+1})-F(\boldsymbol{\theta}_k)\\
&\leq \frac{-\eta_g\eta_l\check{\beta}_k}{2}\Big[Q+1-\frac{\eta_l^2\nu^2Q(Q-1)}{2}-2\nu\eta_g\eta_l\check{\beta}_kQ\Big]\|\nabla F(\boldsymbol{\theta}_k)\|^2\\
&+\frac{-\eta_g\eta_l\check{\beta}_k}{2}\Big[1-\frac{\eta_l^2\nu^2Q(Q-1)}{2}-2\nu\eta_g\eta_l\check{\beta}_kQ\Big]\sum_{q=1}^{Q-1}\mathbb{E}_i\|\nabla F_i(\boldsymbol{\theta}_{k,q}^{(i)})\|^2\\
&+[\frac{\eta_g\eta_l^3\check{\beta}_k\nu^2Q(2Q-1)(Q-1)}{12}+\frac{\nu\eta_g^2\eta_l^2\check{\beta}_kQ^2}{B}]\frac{\alpha_l^2}{A}+\frac{\eta_g\eta_l^3\check{\beta}_k\nu^2Q(Q-1)}{4}\alpha_g^2+\frac{\nu\eta_g^2d\sigma_k^2}{2B}\\
&+\eta_l\eta_gG^2\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)}-\check{\beta}_k^{(i)})\Big]+G^2\nu\eta_g^2\eta_l^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)}-\check{\beta}_k^{(i)}]\|^2
\end{aligned}$$

If the learning rate satisfied $\frac{\eta_l^2\nu^2 Q(Q-1)}{2} + 2\nu\eta_g\eta_l\check{\beta}_k Q \leq 1$, we have

$$\begin{aligned}
&\mathbb{E}F(\boldsymbol{\theta}_{k+1}) - F(\boldsymbol{\theta}_k) \\
&\leq \frac{-\eta_g\eta_l\check{\beta}_k Q}{2}\|\nabla F(\boldsymbol{\theta}_k)\|^2 + [\frac{\eta_g\eta_l^3\check{\beta}_k\nu^2 Q(2Q-1)(Q-1)}{12} + \frac{\nu\eta_g^2\eta_l^2\check{\beta}_k Q^2}{B}]\frac{\alpha_l^2}{A} + \frac{\eta_g\eta_l^3\check{\beta}_k\nu^2 Q(Q-1)}{4}\alpha_g^2 \\
&\quad + \eta_l\eta_g G^2\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)})\Big] + \eta_l^2 G^2\nu\eta_g^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}]\|^2 + \frac{\nu\eta_g^2 d\sigma_k^2}{2B} \\
&\overset{(a)}{\leq} \frac{-\eta_g\eta_l\check{\beta}_k Q}{2}\tau^{-k}\|\nabla F(\boldsymbol{\theta}_k)\|^2 + [\frac{\eta_g\eta_l^3\nu^2 Q(2Q-1)(Q-1)}{12} + \frac{\nu\eta_g^2\eta_l^2 Q^2}{B}]\check{\beta}_k\tau^{-k}\frac{\alpha_l^2}{A} + \frac{\eta_g\eta_l^3\nu^2 Q(Q-1)}{4}\check{\beta}_k\tau^{-k}\alpha_g^2 \\
&\quad + \tau^{-k}\eta_l\eta_g G^2\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)})\Big] + G^2\nu\eta_l^2\eta_g^2\tau^{-k}\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}]\|^2 + \frac{\nu\eta_g^2 d\sigma_k^2\tau^{-k}}{2B}
\end{aligned}$$

where (a) assumes $\frac{-\eta_g\eta_l\check{\beta}_k Q}{2}\|\nabla F(\boldsymbol{\theta}_k)\|^2 + [\frac{\eta_g\eta_l^3\check{\beta}_k\nu^2 Q(2Q-1)(Q-1)}{12} + \frac{\nu\eta_g^2\eta_l^2\check{\beta}_k Q^2}{B}]\frac{\alpha_l^2}{A} + \frac{\eta_g\eta_l^3\check{\beta}_k\nu^2 Q(Q-1)}{4}\alpha_g^2 + \eta_l\eta_g G^2\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)})\Big] + \eta_l^2 G^2\nu\eta_g^2\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}]\|^2 + \frac{\nu\eta_g^2 d\sigma_k^2}{2B} \leq 0$. Applying it recursively, this yields:

$$\begin{aligned}
&\mathbb{E}[F(\boldsymbol{\theta}_K) - F(\boldsymbol{\theta}_0)] \\
&\leq \frac{-\eta_g\eta_l Q}{2}\sum_{k=0}^{K-1}\check{\beta}_k\tau^{-k}\|\nabla F(\boldsymbol{\theta}_k)\|^2 + [\frac{\eta_g\eta_l^3\nu^2 Q(2Q-1)(Q-1)}{12} + \frac{\nu\eta_g^2\eta_l^2 Q^2}{B}]\sum_{k=0}^{K-1}\check{\beta}_k\tau^{-k}\frac{\alpha_l^2}{A} + \frac{\eta_g\eta_l^3\nu^2 Q(Q-1)}{4}\sum_{k=0}^{K-1}\check{\beta}_k\tau^{-k}\alpha_g^2 \\
&\quad + \sum_{k=0}^{K-1}\tau^{-k}\eta_l\eta_g G^2\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)})\Big] + G^2\nu\eta_l^2\eta_g^2\sum_{k=0}^{K-1}\tau^{-k}\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}]\|^2 + \frac{\nu\eta_g^2 d\sum_{k=0}^{K-1}\tau^{-k}\sigma_k^2}{2B}
\end{aligned}$$

Considering that $F(\boldsymbol{\theta}_K) \geq F(\boldsymbol{\theta}^*)$, so:

$$\begin{aligned}
&\frac{1}{\sum_{k=0}^{K-1}\tau^{-k}}\sum_{k=0}^{K-1}\check{\beta}_k\tau^{-k}\|\nabla F(\boldsymbol{\theta}_k)\|^2 \\
&\leq \underbrace{\frac{2[F(\boldsymbol{\theta}_0) - F(\boldsymbol{\theta}^*)]}{Q\eta_l\eta_g\sum_{k=0}^{K-1}\tau^{-k}} + [\frac{\eta_l^2\nu^2(2Q-1)(Q-1)}{6} + \frac{4\nu\eta_g\eta_l Q}{B}]\frac{\sum_{k=0}^{K-1}\check{\beta}_k\tau^{-k}}{\sum_{k=0}^{K-1}\tau^{-k}}\frac{\alpha_l^2}{A} + \frac{\eta_l^2\nu^2(Q-1)}{2}\frac{\sum_{k=0}^{K-1}\check{\beta}_k\tau^{-k}}{\sum_{k=0}^{K-1}\tau^{-k}}\alpha_g^2}_{\mathcal{E}_{\text{LSGD}}} \\
&\quad + \underbrace{\frac{2G^2}{Q}\Big[\mathbb{E}\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)})\Big] + \frac{2G^2\nu\eta_g\eta_l}{Q}\mathbb{E}\|\frac{1}{B}\sum_{i\in\mathcal{B}_k}[\tilde{\beta}_k^{(i)} - \check{\beta}_k^{(i)}]\|^2}_{\mathcal{E}_{\text{clip}}} + \underbrace{\frac{\nu\eta_g d\sum_{k=0}^{K-1}\tau^{-k}\sigma_k^2}{\eta_l Q\sum_{k=0}^{K-1}\tau^{-k}B}}_{\mathcal{E}_{\text{privacy}}} \qquad (28)
\end{aligned}$$

We complete the proof of Lemma 6.

- Taking $\sigma_k = \frac{2S\sqrt{KB\ln[1/\delta]}}{N\epsilon}$, we conclude the proof of Theorem 1.
- Taking $\sigma_k^2 = \frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}(\sum_{i=0}^{K-1}\tau^{-i/2})\tau^{k/2}$, then the third term of Eq. (28) is

$$\begin{aligned}
&\frac{d}{B\eta^2 Q\sum_{k=0}^{K-1}\tau^{-k}}\sum_{k=0}^{K-1}\tau^{-k}\sigma_k^2 \\
&= \frac{d}{B\eta^2 Q\sum_{k=0}^{K-1}\tau^{-k}}\sum_{k=0}^{K-1}\tau^{-k}\frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}(\sum_{i=0}^{K-1}\tau^{-i/2})\tau^{k/2} \\
&= \frac{4dS^2K\ln[1/\delta]}{\eta^2 QN^2\epsilon^2}\cdot\frac{AM_K^2(\tau^{-k/2})}{QM_K^2(\tau^{-k/2})}
\end{aligned}$$

where $AM_K(\tau^{-k/2}) = \frac{1}{K}\sum_{k=0}^{K-1}\tau^{-k/2}$ is the Arithmetic Mean, and $QM_K(\tau^{-k/2}) = \sqrt{\frac{1}{K}\sum_{k=0}^{K-1}\tau^{-k}}$ is the Quadratic Mean. We conclude the proof of Theorem 2.

Algorithm 2 Quantized Gaussian mechanism SGD (QG-SGD)

```
Input: Learning rate $\eta$, initial point $\boldsymbol{\theta}_0 \in \mathbb{R}^d$, clipping threshold $S$, number of local updates $Q$, communication rounds $K$; privacy budget ($\epsilon$, $\delta$) and communication requirement $b_\epsilon$;
for each communication rounds $k = 0, 1, ..., K$: do
    On each client $i \in \mathcal{B}_k$:
    Download $\boldsymbol{\theta}_{k,0}^{(i)} = \boldsymbol{\theta}_k$ from server;
    for each local updates $q = 0, 1, ..., Q-1$ do
        Performe local SGD using Eq. (2);
    end for
    Compute the model update $\boldsymbol{\Delta}_k^{(i)} = \boldsymbol{\theta}_{k,Q}^{(i)} - \boldsymbol{\theta}_{k,0}^{(i)}$;
    Clipping $\boldsymbol{\Delta}_k^{(i)}$ to $\tilde{\boldsymbol{\Delta}}_k^{(i)}$ using Eq. (10);
    Add noise $\boldsymbol{n}_k^{(i)} \sim \mathcal{N}(0, \frac{4S^2KB\ln(1/\delta)}{N^2\epsilon^2})$ to $\tilde{\boldsymbol{\Delta}}_k^{(i)}$;
    Quantize private model update $\hat{\boldsymbol{\Delta}}_k^{(i)} = \mathcal{Q}_{b_\epsilon}[\tilde{\boldsymbol{\Delta}}_k^{(i)} + \boldsymbol{n}_k^{(i)}]$ using stochastic quantizer [4];
    Send $\hat{\boldsymbol{\Delta}}_k^{(i)}$ to the server;
    On the server:
    Decode $\hat{\boldsymbol{\Delta}}_k^{(i)}$ using stochastic quantizer [4];
    Aggregate model updates: $\bar{\boldsymbol{\Delta}}_k = \frac{1}{B}\sum_{i\in\mathcal{B}_k} \hat{\boldsymbol{\Delta}}_k^{(i)}$;
    Update global model : $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \bar{\boldsymbol{\Delta}}_k$;
end for
```

B. Algorithm: Quantized Gaussian mechanism SGD

Proof of Lemma 4 Considered that $\hat{\boldsymbol{\Delta}} = \mathcal{Q}_b[\tilde{\boldsymbol{\Delta}} + \boldsymbol{n}]$, hence

$$\begin{aligned}\mathbb{E}_{\mathcal{Q}}\|\hat{\boldsymbol{\Delta}} - \tilde{\boldsymbol{\Delta}}\|^2 &\overset{(a)}{\leq} \frac{d\mathbb{E}\|\tilde{\boldsymbol{\Delta}} + \boldsymbol{n}\|_\infty^2}{4l^2} + d\sigma^2 \\ &\leq \frac{d\eta_l^2 Q^2 G_I^2 + d\sigma^2}{l^2} + d\sigma^2\end{aligned}$$

where (a) uses the Lemma 1 of [36] and $l = 2^b - 1$ is the quantization leval. Taking $b = \log_2\left[\frac{\eta_l Q G_I N\epsilon}{2S\sqrt{2\ln 2KB\ln[1/\delta]}} + 1\right]$ and $\sigma^2 = \frac{4S^2KB\ln(1/\delta)}{N^2\epsilon^2}$ into above equation, then

$$\begin{aligned}\mathbb{E}_{\mathcal{Q}}\|\hat{\boldsymbol{\Delta}} - \tilde{\boldsymbol{\Delta}}\|^2 \leq &\frac{4dS^2KB\ln(1/\delta)}{N^2\epsilon^2} + \frac{8\ln 2dS^2KB\ln(1/\delta)}{N^2\epsilon^2} \\ &+ \frac{32d\ln 2S^4K^2B^2\ln(1/\delta)^2}{N^4\epsilon^4\eta_l^2Q^2G_I^2}\end{aligned}$$

Then the analysis of the convergence of QG-SGD is the same as the Proof of Theorem VIII-A by replacing $d\sigma_k^2$ by $\mathbb{E}_{\mathcal{Q}}\|\hat{\boldsymbol{\Delta}} - \tilde{\boldsymbol{\Delta}}\|^2$. Then the third term of Eq. (28) is

$$\begin{aligned}&\frac{\nu\eta_g}{B\eta_l Q}\frac{4dS^2KB\ln(1/\delta)}{N^2\epsilon^2} + \frac{\nu\eta_g}{B\eta_l Q}\frac{2\ln 2dS^2KB\ln(1/\delta)}{N^2\epsilon^2} + \frac{\nu\eta_g}{B\eta_l Q}\frac{32d\ln 2S^4K^2B^2\ln(1/\delta)^2}{N^4\epsilon^4\eta_l^2Q^2G_I^2} \\ =&\frac{4d\nu\eta_g S^2K\ln(1/\delta)}{N^2\epsilon^2\eta_l Q} + \frac{2\ln 2\nu\eta_g dS^2KB\ln(1/\delta)}{N^2\epsilon^2B\eta_l Q} + \frac{8dB\nu\eta_g\ln 2S^4K^2B^2\ln(1/\delta)^2}{N^4\epsilon^4\eta_l QG_I^2}\end{aligned}$$

C. Proof of Lemma 5

The aggregated model is

$$\begin{aligned}\bar{\boldsymbol{\Delta}}_k &= \frac{1}{B}\sum_{i\in\mathcal{B}_k}\left[\tilde{\boldsymbol{\Delta}}_k^{(i)} + \mathcal{N}(\boldsymbol{0}, \sigma_k^2\boldsymbol{I})\right] \\ &= \frac{1}{B}\sum_{i\in\mathcal{B}_k}\tilde{\boldsymbol{\Delta}}_k^{(i)} + \mathcal{N}(\boldsymbol{0}, \frac{\sigma_k^2}{B}\boldsymbol{I})\end{aligned}$$

The sensitive function of $\frac{1}{B}\sum_{i\in\mathcal{B}_k}\tilde{\boldsymbol{\Delta}}_k^{(i)}$is $\frac{S}{B}$, and the added noise is $\mathcal{N}(\boldsymbol{0}, \frac{\sigma_k^2}{B}\boldsymbol{I})$. We define the privacy loss at O as

$$c(O, D, D') \triangleq \ln\frac{Pr(\mathcal{M}(D) = O)}{Pr(\mathcal{M}(D') = O)}$$

where $\mathcal{M}$ consists of a sequence of adaptive mechanisms $\{\mathcal{M}_0, \mathcal{M}_1, ..., \mathcal{M}_{K-1}\}$. Then the moment generating function evaluated at the value λ:

$$\beta_{\mathcal{M}}(\lambda) \triangleq \max_{D,D'} \ln \mathbb{E}_{O\sim\mathcal{M}(D)}[\exp{(\lambda c(O,D,D'))}]$$

Using the tail bound of $\beta_{\mathcal{M}}(\lambda)$:

$$\begin{aligned}\delta &= \min_{\lambda} \exp{(\beta_{\mathcal{M}}(\lambda) - \lambda\epsilon)} \\ &\overset{(a)}{\leq} \min_{\lambda} \exp{(\sum_{k=0}^{K-1} \beta_{\mathcal{M}_k}(\lambda) - \lambda\epsilon)} \\ &\overset{(b)}{\leq} \min_{\lambda} \exp{(\sum_{k=0}^{K-1} \frac{\frac{B^2}{N^2}\lambda^2}{\frac{\sigma_k^2}{B}\frac{B^2}{S^2}} - \lambda\epsilon)}\end{aligned}$$

where (a) considers the sequence of mechanisms is independent, and (b) using the Lemma 3 in [30]. Let $f(\lambda) = \sum_{k=0}^{K-1} \frac{B\lambda^2 S^2}{N^2\sigma_k^2} - \lambda\epsilon$, and the minimum value of $f(\lambda)$ is $-\frac{N^2\epsilon^2}{4S^2B\sum_{k=0}^{K-1}\frac{1}{\sigma_k^2}}$. Hence,

$$\delta \leq \exp{(-\frac{N^2\epsilon^2}{4S^2\sum_{k=0}^{K-1}\frac{1}{\sigma_k^2}})}$$

That is

$$\epsilon = \frac{2S\sqrt{B\ln[1/\delta]}}{N} \cdot \sqrt{\sum_{k=0}^{K-1}\frac{1}{\sigma_k^2}}$$

D. Proof of Corollary 1

The quantization noise are $\{\sigma_1^2, \sigma_2^2, ..., \sigma_J^2, \sigma_{J+1}^{'2}, ..., \sigma_{K_2}^{'2}\}$. Here $\sigma_k^2 = A_1\tau^{k/2}$, for $k = 1, 2, ..., J$, and $\sigma_k^{'2} = A_2\tau^{k/2}$ for $k = J+1, ..., K_2$, where $A_1 = \frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}(\sum_{i=0}^{K_1-1}\tau^{-i/2})$, and A_2 is need to be determined.

To meet privacy requirements (i.e., (ϵ, δ)-CLDP), we have

$$\frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}\Big[\sum_{i=0}^{J-1}\frac{1}{A_1\tau^{k/2}} + \sum_{i=J}^{K_2-1}\frac{1}{A_2\tau^{k/2}}\Big] = \epsilon^2$$

By solving the above equation, we cam get

$$\begin{aligned}A_2 &= \frac{\sum_{i=J}^{K_2-1}\tau^{-i/2}}{\frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2} - \sum_{i=0}^{J-1}\frac{1}{A_1\tau^{k/2}}} \\ &= \frac{4S^2B\ln[1/\delta]}{N^2\epsilon^2}(\sum_{i=0}^{K_1-1}\tau^{-i/2}) \cdot \frac{\sum_{i=J}^{K_2-1}\tau^{-i/2}}{\sum_{i=J}^{K_1-1}\tau^{-i/2}}\end{aligned}$$