Efficient Privacy Protection via Quantization in Edge and Federated Machine Learning Deployments

Yijin Wang, *Graduate Student Member, IEEE,* Sabreesh Venkitasubramaniam, *Member, IEEE*
Arjun Mehta, Senior Member, IEEE, Emilia Novak , *Member, IEEE*

Abstract— Adoption of machine learning on edge devices and federated networks has surged, leading to complex challenges in safeguarding data privacy alongside the limitations inherent in computational, memory, and transmission resources in such environments. Traditionally, privacy-preserving protocols employ differentially private stochastic gradient descent (DP-SGD) and then apply quantization post-training to curtail model dimensions and minimize communication loads[5]. However, this process brings notable shortcomings—quantization methods generally lack embedded privacy protections and may introduce distortions that impair learning outcomes. To address these dual issues, this work introduces a joint solution where quantization is randomized, embedding privacy directly within the process while simultaneously optimizing system efficiency. Specifically, we propose Randomized Quantizer Projection Stochastic Gradient Descent (RQP-SGD), a novel training method for edge-based models that integrates DP-SGD into a randomized quantization framework. For federated learning, our approach employs Gaussian Sampling Quantization (GSQ), leveraging discrete Gaussian sampling during quantization to enforce local differential privacy without relying solely on conventional additive noise techniques. Comprehensive theoretical analysis and empirical studies on standard datasets confirm that these randomized approaches outperform legacy sequential designs. RQP-SGD delivers, on average, a 10.62% improvement in model utility over deterministic quantization coupled with projected DP-SGD, while sustaining rigorous differential privacy guarantees. For federated learning, GSQ consistently enhances accuracy, exceeding DP-FedPAQ by 11.52% under non-identically distributed data on MNIST and FashionMNIST, and achieving 16.54% and 8.7% gains on CIFAR-10 and FEMNIST, respectively. These findings underscore the efficacy of integrating randomized quantization in both privacy and resource optimization for decentralized machine learning systems.

Index Terms—Privacy, quantization, ML-at-the-edge, differential privacy, federated learning

I. Introduction

As IoT devices proliferate across industries, there is an increasing demand to process data closer to the source [19], [50], enabling real-time insights and decision-making. Machine learning algorithms have achieved tremendous success in a variety of domains in the last several years, due to their ability to extract inferences from data that aid in a variety of tasks. There is now a growing need for machine learning (ML) at-the-edge, where ML models are deployed directly onto IoT devices or gateways. However, developing machine learning models on IoT devices presents two major challenges: privacy and resource constraints.

Machine learning requires substantial training of several layers densely populated with parameters, which is enabled by large datasets often containing sensitive information. As a result, the learned models can be exploited by adversaries to extract the sensitive information in the training datasets. For instance, information about medical procedures can be determined using models built on hospital datasets [48]. It is therefore critical to provide strong and rigorous privacy guarantees for machine learning algorithms.

Differential privacy (DP) is today the cornerstone for privacy in ML, providing a quantifiable measure of privacy protection [17]. A common method to achieve DP in ML is through Differentially Private Stochastic Gradient Descent (DP-SGD), which introduces noise to the gradient updates during training [1]. However, the application of noise in DPSGD can significantly degrade model performance. To mitigate this, recent studies have explored strategies to reduce the impact of noise by applying DP-SGD in a reduced-dimensional space or by leveraging spectral domain perturbations for noise filtering [3], [20], [41], [43], [58]–[60].

In addition to privacy concerns, edge devices are inherently resource-constrained, with limited computational power, memory, and storage capacity. These limitations pose a significant challenge for deploying complex and computationally expensive machine learning models. In contrast to the edge computing environment, the field of machine learning is experiencing a surge in model complexity, driven by the availability of everincreasing datasets and computational power. This growing disparity between the limited capabilities of edge devices and the demands of large-scale ML models necessitates the development of ML models compression. Quantization tackles the resource constraints by reducing the bit-width of the data representation within the ML model. By converting continuous or high-precision numerical values into a finite set of discrete values, quantization effectively reduces the memory and computational demands of ML models.

In particular, the perspective of this work is to study and demonstrate that randomized quantization can be used to address both challenges, compression, and privacy preservation, as opposed to treating these challenges as separate. We consider two application cases of the ML at-the-edge.

The first case is the deployment of machine learning models in resource-constrained environments. For this case, we propose a joint privacy-preserving quantization approach that integrates differential privacy techniques with quantization to train private ML models with quantized parameters. By leveraging quantization, we can reduce the computational and memory requirements of ML models, making them more feasible for deployment on resource-constrained IoT devices. This approach ensures that the models are both lightweight and privacy-preserving, addressing the key challenges of edge computing.

The second application case is private federated learning with communication constraints. Federated learning enables collaborative learning across many edge devices, involving periodic synchronization of local model updates with a central server for aggregation. We propose a randomized quantization scheme to provide differential privacy in federated learning. The randomized quantization incorporates randomness into the local model updates, thereby ensuring that sensitive information is protected during communication between the local devices and the central server. This combination of quantization and privacy preservation aligns well with the decentralized and privacy-focused nature of federated learning, making our method a promising solution for efficient and secure ML at the edge.

Our work is developed from a key idea that quantization and privacy preservation share a common goal: the reduction of information. Quantization achieves this by converting model parameters from high-precision to low-precision representations, thereby removing redundant information. On the other hand, privacy preservation, specifically through differential privacy, involves the removal of sensitive information to protect individual data points in the dataset. Moreover, privacy leakage in machine learning is a consequence of over-learning through highly parameterized architectures, and quantization potentially mitigates privacy leakage by reducing the model's capacity to over-learn sensitive information. The synergies between these two processes present a unique opportunity to develop methods that simultaneously achieve efficient model deployment and robust privacy guarantees. This is particularly relevant in the context of ML at-the-edge, where the limited computational and memory resources of IoT devices necessitate lightweight models, and the sensitive nature of the data demands stringent privacy protections.

Furthermore, our approach is based on randomized quantization, which has been previously employed as a compression mechanism in IoT systems and sensor networks [55]. This technique not only achieves local differential privacy but also compresses data, albeit with a trade-off between privacy, compression, and utility due to the inherent information loss in quantization. Recent work [8] has explored randomized quantization through linear programming to minimize error and preserve privacy. Randomized quantization has also been integrated with Laplace mechanisms to reduce communication overhead and improve efficiency in both local and centralized settings [46]. The Randomized Quantization Mechanism (RQM) [57] further extends this approach to federated learning, ensuring differential privacy without explicit noise addition by using quantization randomness to maintain model accuracy.

A. Overview

Our research extends the use of randomized quantization to deploy machine learning models on edge devices and federated learning systems, addressing two key challenges: privacy and resource constraints. Existing methods apply differential privacy (DP) and quantization separately, treating quantization as a post-processing step and overlooking its potential to enhance privacy during the process. This separation results in suboptimal performance, especially in federated learning, where private model updates must also be compressed for efficient communication.

To address this, we propose:

1) Randomized Quantization Projection with Stochastic Gradient Descent (RQP-SGD):A privacy-preserving training method that embeds randomized quantization into DP-SGD, reducing noise-induced utility loss by incorporating privacy directly into the projection step.
2) Gaussian Sampling Quantization (GSQ): federated learning scheme that integrates discrete Gaussian sampling with stochastic quantization, achieving local differential privacy (LDP) while compressing model updates for enhanced communication efficiency.

These methods offer an integrated framework to privacy and resource challenges. Theoretical analysis and experiments show that RQP-SGD and GSQ outperform existing techniques in balancing privacy, utility, and efficiency. The implementation is available at *github.com/cfeng6/RandomizedQuantizationML*.

B. Contributions

The main contributions of this work are as follows:

- We propose a novel framework that integrates randomized quantization into differential privacy mechanisms, achieving simultaneous model compression and privacy preservation. This approach mitigates the performance degradation typically associated with DP by reducing noise requirements, bridging the gap between privacy guarantees and resource-efficient ML deployment.

- We demonstrate that randomized quantization can be used to address both compression and privacy preservation in two application cases: deployment of the ML models in the resource-constrained environment and private federated learning with communication constraints.
- For the deployment of ML models with resource constraints, we propose RQP-SGD, a randomized quantization projection based SGD in the deployment of ML with a differential privacy guarantee.
  - We theoretically study the utility-privacy trade-off of RQP-SGD for ML with convex and bounded loss functions.
  - Through experiments on two classification datasets: MNIST and Breast Cancer Wisconsin (Diagnostic) dataset [51], the latter being a dataset collected from IoT devices, we demonstrate that RQP-SGD can achieve better utility performance than implementing DP-SGD in machine learning with quantized parameters. Significantly, RQP-SGD achieves a 35% higher accuracy on the Diagnostic dataset while maintaining a $(1.0,0)$-DP, thereby validating its compatibility and efficacy for deployment in IoT systems.
  - We conduct a comprehensive experimental analysis of how various RQP parameters influence the utilityprivacy balance. This includes examining the effects of quantization-induced randomness, noise scale in gradient updates, and quantization bit granularity. Our findings highlight that while quantization-induced randomness can enhance utility, excessive randomness may have a detrimental effect on utility.
- For private federated learning with communication constraints, we propose the Gaussian sampling quantization (GSQ), a novel method that combines the Discrete Gaussian sampling with stochastic quantization to enhance privacy and communication efficiency in federated learning. Unlike RQM [57], which achieves privacy by randomly sub-sampling quantization levels, GSQ employs Discrete Gaussian sampling. The sampling probability in GSQ decays exponentially with the distance between the input and the quantization levels, providing a flexible trade-off between privacy and model performance. This adaptability allows GSQ to align with varying privacy budgets and accuracy requirements.
  - Through theoretical analysis, we demonstrate that the Gaussian sampling quantization provides strong differential privacy at the local edge device.
  - We theoretically study the properties of the Gaussian sampling quantizer and demonstrate its feasibility in the context of federated learning.
  - Through several experiments on four benchmark image classification datasets, namely MNIST, FMNIST, CIFAR10, and Federated EMNIST, we demonstrate the practical effectiveness of GSQ in maintaining high utility while providing strong privacy guarantees and reducing communication costs in federated learning environments.

C. *Paper Organization*

The remainder of the paper is organized as follows. In Section II, we detail the related work to place our work in a broader scientific context. We provide some preliminaries in Section III. We formulate the mathematical model and related formulations of RQP-SGD in Section IV. In Section V, we propose GSQ and provide theoretical analysis in Section V-B. We conduct several experiments of RQP-SGD and analyze it in Section VI. The experimental results of GSQ in federated learning are detailed in Section VII. Section VIII addresses the limitations of the proposed method and discusses potential future research directions. Some concluding remarks are presented in Section IX.

II. Relevant Literature

The subject of differential privacy in ML has attracted significant scientific interest in recent years. Specifically, it has been used in support vector machine [33], linear/logistic regression [11], [62] and risk minimization [6], [7], [12], [26]. In recent years, more works have focused on privacypreserving training for deep learning. Differentially private stochastic gradient descent (DP-SGD) [1], [60] as mentioned earlier perturbs the gradient at each SGD update with random noise drawn from Gaussian distribution during the training. Private Aggregation of Teacher Ensembles (PATE) [42] is one approach that transfers the knowledge from an ensemble of teachers trained on the disjoint subsets of training data to train a student model through the noisy aggregation of teachers' answers. Some recent works aim to improve the privacy-utility trade-off of DP-SGD through dynamical noise addition [16] and gradient dimension reduction [20].

Quantization in ML is categorized into two types: posttraining quantization (PTQ) [28], [38], [40], and quantization aware training (QAT) [15], [39], [45]. PTQ involves quantizing a model after it has been trained, without the need to retrain the model. PTQ typically includes converting weights and activations from a higher resolution to a lower precision, which reduces the model size and speeds up inference. As largescale models show their excellent performance on complex tasks, more advanced PTQ techniques have been proposed. SmoothQuant [53] quantizes only the weights to 8-bit integers for deploying large-scale language models, resulting in significant improvements in latency and memory usage without a substantial loss in accuracy. A mixed-precision PTQ method for vision transformers is proposed in [35], focusing on optimizing the quantization intervals for the linear operations and self-attention layers. GPFQ [61] provides theoretical guarantees on the performance and error bounds of the quantized models.

QAT aims to improve the performance of neural networks by simulating the effects of quantization during the training process. QAT allows the model to learn how to adapt to the quantization error, leading to better performance when the models need to be deployed with quantized weights. A general quantization scheme in QAT is proposed in [25], providing foundational insights into QAT and its benefits for efficient inference. Another work [24] improves the efficiency of QAT by selecting the most important data samples during training. Our proposed RQP-SGD fits into the QAT by modeling QAT as a quantization-constrained optimization problem [4]. Unlike the traditional QAT methods that primarily mitigate quantization errors, RQP-SGD integrates differential privacy directly into the quantization process. By applying randomized quantization during SGD, RQP-SGD simultaneously addresses model compression and privacy preservation. This approach minimizes noise-induced degradation through randomized projections at each gradient step, offering a novel integration of privacy and quantization within the training process.

Differentially private quantization scheme in releasing data has been studied in [55], [63]. We note another application of differential privacy and quantization in ML is federated learning. In recent years, there has been a growing body of research [2], [21], [30] that aims to study privacycommunication trade-off in federated learning. The main objective of private and efficient federated learning is transferring private, low-bandwidth gradient vectors to the server. Some works [22], [27], [30], [64] study the quantization method for achieving privacy preservation and communication efficiency in federated learning. Another work [13] combines gradient quantization with sparsification to improve communication costs while maintaining differential privacy guarantees. However, these prior works primarily employed quantization as a post-processing step for differential privacy mechanism. This sequential design limits the potential for quantization to directly contribute to privacy preservation during training or communication. One notable exception is the Randomized quantization mechanism (RQM) [57], which achieves Renyi differential privacy by randomly sub-sampling quanti-´ zation levels followed by a rounding procedure to a close-by quantization level. In contrast, our proposed GSQ introduces randomness directly into the quantization process through discrete Gaussian sampling, where the sampling probability decays exponentially with the distance between the input and the quantization levels. The scale of the discrete Gaussian serves as a tunable parameter, allowing for a flexible tradeoff between privacy and model performance, enabling GSQ to adapt to varying privacy budgets and accuracy constraints. In Section VII, we numerically compare GSQ with RQM and demonstrate GSQ achieves higher model accuracy in highly heterogeneous FL systems.

III. Preliminary

A. Machine Learning As Empirical Risk Minimization

In the domain of machine learning, model training is often conceptualized within the framework of empirical risk minimization (ERM). Consider a dataset $\mathrm{S} = \{(x_i,y_i) \in \mathrm{X} \times \mathrm{Y}) : i = 1,2,\cdots,n\}$, comprising n pairs of features and labels.
The ML model is denoted by a predictor $f : \mathrm{X} \times \mathrm{W}$ 7→ Y featured by a set of parameters $w \in \mathrm{W}$. The quality of the predictor on training data is quantified through a non-negative loss function $l : \mathrm{Y} \times \mathrm{Y}$ 7→ R. We aim to choose optimal w that minimizes the empirical loss:

$$\min_{w\in\mathcal{W}} \hat{\mathcal{L}}(w;\mathcal{S}) := \frac{1}{n}\sum_{i=1}^{n} l(f(x_i;w),y_i) \qquad (1)$$

Stochastic Gradient Descent (SGD) is a prevalent optimization method in machine learning. It iteratively updates model parameters using a stochastic estimate of the gradient of the loss function. In each iteration, a mini-batch of m training samples is selected, and the stochastic gradient $\sum_{j=1}^{m} \nabla l(f(w_t;x_j);y_j)$ is computed, where w_t represents the parameters at iteration t. This gradient approximation incrementally adjusts the model parameters towards minimizing the empirical loss, thereby enhancing predictive performance.

B. Federated Learning

Federated learning (FL) is a collaborative machine learning technique that aims to construct a unified model by leveraging data from multiple decentralized nodes, namely devices or clients, each holding its local dataset. In an FL system with K clients, the training dataset S is split into K sub-datasets and each client k holds a subdataset S_k. In an FL system with K clients, each client k holds a dataset D_k. FL aims to train a model on the collaborative dataset $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$, while ensuring that the data remains within its originating client, thus eliminating the need for direct data sharing. Let w denote the model parameter, $L_k(w;\mathrm{D}_k)$ be the local loss function of the k-th client evaluated on the local dataset D_k, the collective objective in FL is to find the global model parameters w^* that minimizes the aggregate loss:

$$w^* = \arg\min_{w} \left\{ \frac{1}{K} L_k(w;\mathcal{D}_k) \right\} \qquad (2)$$

In the FL system, each client independently conducts its local model training and computes a model update. The update is subsequently transmitted to a central server. The server's role is to aggregate these updates to improve the global model. Specifically, given the global model parameter at time step t as w_t, each client k performs the local training iterations with its dataset D_k, updating w_t to w_{t+1}^k. The model update $h_t^k = w_{t+1}^k - w_t$ is then transmitted to the server. The server aggregates these model updates from the clients, and updates the global model:

$$w_{t+1} = w_t + \frac{1}{K}\sum_k h_t^k \tag{3}$$

C. Quantization and Machine Learning

In the context of machine learning and federated learning, quantization serves as a crucial technique for enhancing the efficiency of model deployment and reducing communication overhead. Below is an elaboration on the conventional approach to machine learning optimization with quantized parameters, along with an overview of federated learning under communication constraints.

a) Quantized ML optimization: Quantization in machine learning involves mapping continuous or high-precision parameters to a discrete set of values, typically represented with fewer bits. This process is particularly beneficial for deploying models on resource-constrained devices, such as mobile phones or IoT devices, where memory and computational power are limited. With the framework of ERM, the machine learning optimization with quantized parameters is formulated as follows:

$$\min \hat{L}(w;\mathrm{S}) \qquad \text{s.t. } w \in \mathrm{Q} \qquad (4)\ {}_{w\in\mathrm{W}}$$

where $\mathrm{Q} \subseteq \mathrm{R}^d$ represents a discrete, non-convex quantization set.

During the inference, the quantized parameters are used to compute the model's output, leading to increased efficiency due to reduced precision. In the training phase, the gradients are computed based on the quantized parameters. Given that (4) is an integer optimization with non-linear constraints, it necessitates relaxation to a form amenable to solution via projected SGD (Proj-SGD) [32], [56]:

$$\begin{cases} v_{t+1} = w_t - \eta\nabla l(f(w_t;\cdot);\cdot) \\ w_{t+1} = \mathrm{Proj}_{\mathcal{Q}}(v_{t+1}) \end{cases} \tag{5}$$

where $\nabla l(f(w_t;\cdot);\cdot)$ is the sampled mini-batch gradient at the t-th iteration, η is the step size, and $\mathrm{Proj}_{\mathrm{Q}}(v) = \mathrm{argmin}_{u\in\mathrm{Q}} \|u - v\|$ is a projection that projects v onto the quantization set. The Proj-SGD process indicates the backward pass and update in quantized ML, where in the training phase, gradients are computed based on the quantized parameters, and the updates to the parameters are accumulated in a higher precision before being quantized again for the next iteration to maintain accuracy.

b) FL with Communication Constraints: The efficiency of the FL system is significantly affected by communication constraints. These issues mainly arise because the speed at which clients send model updates to the server (the uplink) is slower than the speed at which the server sends updates back to the clients (the downlink). This slow uplink is due to many clients sending updates at the same time, which uses a lot of bandwidth and takes longer. To mitigate these challenges, upstream compression emerges as an indispensable technique in FL. By reducing the size of the updates sent from clients to the server, the technique facilitates a reduction in bandwidth.

The Federated Averaging (FedAvg) algorithm is a widely used method in FL that effectively reduces communication costs through various strategies, including partial client participation and asynchronous updates. In each communication round (time step) t, the central server selects a subset of clients, denoted as P_t for participation. These selected clients then contribute to the model aggregation process. Moreover, FedAvg incorporates asynchronous updates, adding flexibility and efficiency to the training process.

Formally, at time step t, each participating client $j \in \mathrm{P}_t$ conducts E local training iterations. The local update for each client *j* at iteration *i* is calculated as follows:

$$w_{t,i+1}^j = w_{t,i}^j - \eta_t^j \nabla F_i(w_{t,i};\mathcal{D}_j^{t,i}) \tag{6}$$

where η_t^j is the learning rate for client *j* at time step t, $\mathcal{D}_j^{t,i}$ is a randomly chosen subset of client's dataset D_j, and $w_{t,0}$ is initialized as w_t. After completing E local iterations, the local model update from client *j* is:

$$h_t^j = w_{t,E}^j - w_t \tag{7}$$

To further reduce the communication overhead, FedPAQ [44] introduces quantized message-passing: the clients can transmit a quantized version of their local updates to the central server:

$$h_{jt} = Q(w_{t,Ej} \quad - w_t) \tag{8}$$

where $Q(\cdot)$ is a quantization function that compresses the update before transmission. The quantization entails mapping the model update to a discrete set, thereby substantially reducing the E 3e of data transmitted in each communication round.

D. Differential Privacy (DP)

Differential privacy [17] is a quantitative definition of privacy, initially designed in the context of databases. Specifically, it ensures that whether or not an individual's data is included in a dataset does not significantly affect the analysis results on that dataset. A randomized mechanism $\mathrm{M} : \mathrm{D}$ 7→ R satisfies (ϵ,δ)-DP if for any two adjacent sets $d,d' \in \mathrm{D}$ and all possible outputs O of M, it holds that

$$\Pr[\mathrm{M}(d) \in \mathrm{O}] \leqslant e^{\epsilon}\Pr[\mathrm{M}(d') \in \mathrm{O}] + \delta$$

A prevalent approach in applying differential privacy to a real-valued function $f : \mathrm{D}$ 7→ R involves the addition of noise that is carefully calibrated to function's sensitivity S_f. The sensitivity, S_f, is defined as the maximum possible difference $|f(d) - f(d')|$ between the outputs of f for any two adjacent inputs d and d'. A common example is the Gaussian mechanism, which involves adding noise perturbed by a Gaussian distribution directly to the output of the function f. This process can be formulated as:

$$Gauss(f,d,\sigma) = f(d) + \mathrm{N}(0,S_f^2\sigma^2)$$

where $\mathrm{N}(0,S_f^2\sigma^2)$ is the Gaussian distribution with zero mean and variance $S_f^2\sigma^2$. The Gaussian mechanism achieves (ϵ,δ)-DP when $\sigma = \mathrm{p}\overline{2\log(1.25/\delta)}/\epsilon$ [18].

a) Differential Privacy in Machine Learning: In the realm of machine learning, the ability of models to discern intricate patterns from training datasets brings forth significant privacy concerns, particularly regarding the inadvertent memorization and subsequent exposure of individual data points. This issue becomes more pronounced with the advent of sophisticated techniques such as model inversion and membership inference attacks, which can exploit these vulnerabilities to infringe upon individual privacy.

In addressing the privacy concerns inherent in machine learning, the framework of differential privacy (DP) has emerged as a formalized methodology. It is intricately designed to quantify and attenuate the risks associated with the dissemination of information extracted from sensitive datasets. It ensures that machine learning model outputs are carefully calibrated to prevent the inference of sensitive information about any particular individual.

Mathematically, for the machine learning predictor denoted as f, it satisfies (ϵ,δ)-DP if for any two adjacent sets $x,x' \in \mathrm{X}$, the following holds:

$$\Pr[f(w;x)] \leq e^{\epsilon}\Pr[f(w;x')] + \delta \qquad (9)$$

This formula is foundational to the principle of differential privacy in machine learning. It is designed to control the probability distribution of outcomes from the private machine learning model in a manner that is minimally influenced by the presence or absence of any individual data within the training dataset.

A prevalent technique to attain differential privacy in machine learning is differentially private stochastic gradient descent (DP-SGD). DP-SGD modifies the standard SGD update rule to incorporate the Gaussian mechanism, as described by the following formulation:

$$w_{t+1} = w_t - \eta \cdot \frac{1}{m}[\sum_{j=1}^{m} \nabla l(f(w_t;x_j);y_j) + G_t] \qquad (10)$$

where w_{t+1} denotes the updated model parameters, η is the step size, and the gradient $\nabla l(f(w_t;x_j);y_j)$ is calculated at each t-th iteration using the data pair (x_j,y_j). The key to DPSGD is to add Gaussian noise vector G_t which is calibrated to the sensitivity of $\nabla l(f(w_t;x_j);y_j)$. The inclusion of this noisy gradient, processed through the Gaussian mechanism, ensures the privatization of the gradient. The gradient descent serves as the post-processing, which inherently preserves the differential privacy.

However, these techniques typically presume that the model weights are real-valued. Our research pivots from this norm by aiming to achieve DP with quantized weights, essential in resource-constrained environments where model size and computational efficiency are critical. This pursuit addresses the need for privacy-preserving models in environments where resources are limited and hence quantization is essential for reducing computational demands and model size.

b) FL with Local Differential Privacy: While FL is designed to enhance privacy by limiting shared information to local model updates only, this approach is insufficient to secure data privacy. The system remains vulnerable to external risks, especially from malicious adversaries who might intercept the local updates transmitted from the clients to the server. This vulnerability poses a significant risk of sensitive information leakage, allowing these adversaries to determine if certain individual data points were used in the training datasets. Therefore, integrating an advanced privacy-preserving mechanism within the FL framework is essential to safeguard individual data contributions.

To address these privacy concerns, local differential privacy (LDP) emerges as an effective solution in FL. LDP, as a local counterpart of differential privacy (DP), emphasizes protecting individual data within the model updates sent from the user devices. For each client k corresponded to the local dataset D_k, a mechanism M_k is said to be (ϵ_k,δ_k)-LDP w.r.t. the dataset D_k if for any two adjacent subset $\mathrm{D},\mathrm{D}' \in \mathrm{D}_k$ and all possible outputs O of M_k, it holds that

$$\Pr[\mathrm{M}(\mathrm{D}) \in \mathrm{O}] \leqslant e^{\epsilon_k}\Pr[\mathrm{M}(\mathrm{D}') \in \mathrm{O}] + \delta_k \qquad (11)$$

In the context of FL, LDP typically involves adding noise to the model updates before sending them to the central server. A widely used method to implement LDP is the Gaussian mechanism, which adds noise following a Gaussian distribution directly to the local model update:

$$\tilde{h}_t^k = h_t^k + \mathcal{N}(0, \Delta^2\sigma_k^2 I_d) \qquad (12)$$

where $\sigma_k > 0$ is the noise scale, Δ denotes the l_2 norm sensitivity of $h^k{}_t$, and I_d is the $d \times d$ identity matrix. The Gaussian mechanism achieves (ϵ_k,δ_k)-LDP when $\sigma_k = \mathrm{p}$

$2\log(1.25/\delta_k)/\epsilon_k$. This approach effectively integrates LDP into FL, enhancing data privacy by protecting individual contributions within the learning process.

E. Private FL with Quantization

In the field of Federated Learning (FL), the twin objectives of communication efficiency and privacy protection are paramount. To effectively address these concerns, a viable strategy involves the implementation of quantization subsequent to a local differential privacy (LDP) mechanism. At time step t, a participation client j sends the following quantized and noisy update to the server:

$$\begin{cases} \tilde{h}_t^j = h_t^j + \mathcal{N}(0, \Delta^2\sigma_j^2 I_d) \\ \hat{h}_t^j = Q(\tilde{h}_t^j) \end{cases} \tag{13}$$

This approach starts by introducing noise to the data under LDP, thereby obfuscating individual contributions for enhanced privacy. The subsequent step of data quantization serves to compress the data, making it more efficient for transmission across the federated network. However, this integration of quantization with LDP in FL is not without its challenges. Firstly, while quantization acts as a subsequent phase to LDP, it does not add any further privacy protections. Rather, the noise introduced by LDP could interfere with the quantization process, resulting in a compromise between maintaining privacy and retaining the precision or utility of the transmitted data. Secondly, adhering to a stricter privacy budget may require increasing the level of noise, which consequently amplifies the noise error. This highlights the necessity for a meticulously designed quantization function to effectively address these issues.

We next will introduce our proposed method in the following sections. In Section IV, we propose RQP-SGD to address the privacy challenges associated with deploying machine learning models on resource-constrained devices. In Section V, we introduce the GSQ to tackle both privacy and compression challenges in the context of federated learning. Section VI presents the experimental results evaluating the effectiveness of RQP-SGD. The experimental results demonstrating the performance of GSQ is presented in Section VII.

IV. Randomized Quantization in Private Machine Learning

In this section, we introduce randomized quantization into machine learning within the framework of ERM. Specifically, we propose RQP-SGD, a new approach to providing differential privacy in ML with quantized computational models. This method extends the paradigm of randomized quantization to the domain of machine learning, adapting it into a projected stochastic gradient descent (Proj-SGD).

A. Problem setting

Our goal is to develop a differentially private optimization solution to (4). More specifically, we aim to solve the following ERM problem:

$$\min_{w\in\mathcal{W}} \hat{L}(w;\mathcal{S}) := \frac{1}{n}\sum_{i=1}^{n} l(f(x_i;w),y_i) \quad \text{s.t.} \quad \begin{cases} w \in \mathcal{Q} \\ \Pr[f(x;w)] \le e^{\epsilon}\Pr[f(x';w)] + \delta \end{cases} \tag{14}$$

where x,x' are two adjacent subset of X. In this work, we incorporate two key assumptions:

- The parameter space W ⊂ R^d is a closed, convex set bounded by M: $\|w\| \le M$, and the quantization set Q is a discrete subset of W.
- For all data-label pair $(x_i,y_i) \in S$, the loss function, $l(f(x_i;w),y_i)$, is a convex and ρ-Lipschitz with respect to w, for example, binary cross-entropy loss in Logistic regression, and hinge loss in SVM classification.

A conventional approach to solving (14) is adapting DP-SGD (10) to the Proj-SGD (5):

$$\begin{cases} v_{t+1} = w_t - \eta \cdot \frac{1}{m}[\sum_{j=1}^{m} \nabla l(f(x_j;w_t);y_j) + G_t] \\ w_{t+1} = \mathrm{Proj}_{\mathrm{Q}}(v_{t+1}) \end{cases} \tag{15}$$

Here, $G_t \sim \mathrm{N}(0,\sigma^2 \mathrm{I}_d))$ represents noise independently drawn at each SGD update. This adaptation of DP-SGD to Proj-SGD presents certain limitations: First, the deterministic nature of the projection step, although serving as a post-processing phase of DP-SGD, does not contribute additional privacy safeguards. Instead, it induces a ”projection error”, a consequence of aligning model parameters with the nearest point in Q. Second, a stricter privacy budget necessitates scaling up the noise, which in turn leads to an increased noise error.

To address these limitations, we propose randomized projection (RP), a novel methodology that integrates stochastic elements into the projection phase. The crux of this method lies in leveraging the projection phase as a mechanism to bolster privacy protection. By injecting controlled randomness into the projection, we target achieving the designated privacy budget while concurrently lowering the noise error.

B. Randomized Projection

In randomized projection, we consider b-bit quantized parameters with uniformly distributed levels

$$Q_{M,b} = \{Q_0, Q^1, \cdots, Q_{2^b-1}\}$$

where $Q_{M,b}$ denotes the b-bit uniform quantization set with quantization bound $M \in R^+$, and each quantization level is given by

$$Q_i = -M + \frac{2M}{2^b - 1} \cdot i \qquad (16)$$

Randomized projection is a variant of the classical projection of unquantized inputs onto the quantization set. Formally, the classical projection of input Q_{in} onto $Q_{M,b}$ is

$$\mathrm{Proj}^{D}_{\mathcal{Q}_{M,b}}(Q_{in}) = \underset{Q \in Q}{\mathrm{argmin}} \| \lfloor Q_{in}, M \rceil - Q \|_2 \qquad (17)$$

where $\lfloor \cdot, M \rceil$ denotes clipping function that clips the parameters into $[-M, M]$. In contrast to classical projection, randomized projection (outlined in Algorithm 1) adds randomness by using a coefficient $q \in (0,1)$, enhancing privacy by making the input value less deducible from its deterministic projection.

C. RQP-SGD

Based upon randomized projection, we propose an iterative SGD method for solving (14), termed RQP-SGD. This method is detailed in Algorithm 2. RQP-SGD is an adaptation of DP-SGD into a variant of Proj-SGD that incorporates the randomized projection in place of the deterministic one.

```
Algorithm 1 Randomized Projection
Require: $b$-bit uniform quantization set $Q_{M,b}$, randomness
    coefficients $q \in [\frac{1}{2^b-1}, 1)$, quantization input $Q_{in}$
1: Find $Q^* = \mathrm{Proj}^D_{Q_{M,b}}(Q_{in})$ using (17)
2: Project $Q_{in}$ onto $Q_{M,b}$ randomly: $\mathrm{Proj}^R_{\mathcal{Q}_{M,b}}(Q_{in})$
       $\{ Q^*$          with probability $q$
   =   $Q$          for $Q \in Q_{M,b} \setminus \{Q^*\}$ with probability $\frac{1-q}{2^b-1}$
3: Return Output of $\mathrm{Proj}^R_{\mathcal{Q}_{M,b}}(Q_{in})$
```

```
Algorithm 2
RQP-SGD
Require: Training dataset: $S = \{(x_i, y_i) \in X \times Y) : i = 1, 2, \cdots, n\}$, $\rho$-
    Lipschitz, convex loss function $l$, convex parameter space
    $W \subseteq R^d$, step size $\eta$, mini-batch size $m$, number of iterations
    $T$, Quantization set $Q_{M,b}$ which is the discrete subset of W,
    projection randomness coefficient $q$.
1: Choose arbitrary initial point $w_0 \in W$.
2: for $t = 0$ to $T - 1$ do
3:     Sample a batch $B_t = \{(x_j, y_j)\}_{j=1}^m \leftarrow \mathcal{S}$ uniformly with
       replacement.
4:     $v_{t+1} = w_t - \eta \cdot \frac{1}{m}[\sum_{j=1}^m \nabla l(f(x_j; w_t); y_j) + G_t)]$
       where $G_t \sim N(0, \sigma^2 M^2 I_d)$ drawn independently each
       iteration.
5:     $w_{t+1} := \mathrm{Proj}^R_{\mathcal{Q}_{M,b}}(v_{t+1})$ where $\mathrm{Proj}^R_{QM,b}$ denotes the random
       projection onto $Q_r$.
6: end for
7: Return $W_T$.
```

2) Utility of RQP-SGD: In Theorem 2, we provide the utility guarantee of Algorithm 2 based on the convergence analysis of the stochastic oracle model (see [32], [47])

^{1}) *Privacy of RQP-SGD:* The essence of RQP-SGD lies in its strategy for privacy enhancement. In step 4 of Algorithm 2, it upholds DP by incorporating a Gaussian noise vector, which is integral to DP-SGD. Subsequently, step 5 amplifies privacy protection via randomized projections. This dual-pronged method, which effectively combines DP-SGD and randomized projection, is designed to achieve the desired privacy budget. It simultaneously aims to mitigate the adverse effects of noise on the model's accuracy and performance, thereby addressing the inherent limitations of escalated noise levels in traditional DP-SGD applications. The differential privacy guarantee of RQP-SGD is rigorously established in Theorem 1.

Theorem 1. *For any $\epsilon > 0$, there exists mini-batch sampling rate $\frac{m}{n}$, training iterations T, quantization bit b and randomness coefficient q, noise scale σ such that Algorithm 2 achieves $(\epsilon, 0)$-DP.*

Proof. To analyze the privacy guarantee of RQP-SGD, we calculate the differential privacy budget consumed by each training update. We then combine these budgets using a composition method to obtain the overall privacy spent during training. This analysis relies on the ∞-th order Renyi differ´ential privacy for each update. The complete proof is detailed in Appendix A. □

with convex loss. Compared to the utility analysis in [47], the RQPSGD leads to two additional errors: Quantization error (E_Q) and noise error (E_N).

Theorem 2. *Let* $\bar{W}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$. *Suppose the parameter set* W *is convex and M-bounded, and the quantization set* Q_p *is generated by a randomized quantizer with probability q and b-bit. For any* $\eta > 0$*, the excess empirical loss of* $A_{ProjNSGD}$ *satisfies*

$$\begin{aligned} &\mathbb{E}\left[\hat{\mathcal{L}}(\bar{w}_T;\mathcal{S})\right] - \min_{w\in\mathcal{W}} \hat{\mathcal{L}}(w;\mathcal{S}) \\ \leq\quad & \frac{M^2}{2\eta T} + E_Q + \frac{\eta\rho^2}{2} + E_N \end{aligned} \tag{18}$$

$$E_Q = dM^2[\frac{q}{(2^b-1)^2} + \frac{2^{b+1}(2^{b+1}-1)}{3(2^b-1)^2}(1-q)]$$

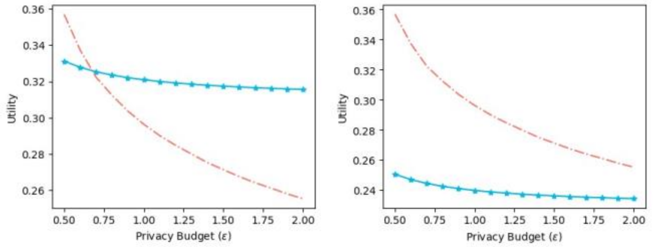
wheredenotes the quantization error and $E_N = \eta\sigma^2 d$ *denotes the noise error.*

Proof. The complete proof is detailed in Appendix B. □

The convergence order for E_Q is expressed as $O(dM^2[\frac{q}{(2^b-1)^2}+(1-q)])$. As the quantization bit b increases, the term $\frac{q}{(2^b-1)^2}$ decreases significantly. Consequently, E_Q becomes primarily dominated by the $(1-q)$ term. This indicates that as q decreases, the randomness in the quantization process increases, resulting in a larger E_Q. In addition to E_Q, the noise error E_N arises from the differential privacy mechanism, with its convergence order of $O(\sigma^2 d)$.

The trade-off between the projection randomness coefficient q and the noise error reflects the dual role of randomized projection in RQP-SGD. As q decreases, the likelihood of projection model parameters to distant quantization levels increases. This additional randomness directly enhances the privacy by increasing uncertainty in the quantized update. This reduces the reliance on the differential privacy mechanism, allowing for a smaller noise scale σ and subsequently decreasing E_N. This interaction highlights how projection randomness serves as a privacy-enhancing mechanism, balancing the quantization and noise errors. Carefully selecting the projection randomness coefficient q is crucial in maintaining model performance within the constraints of the given privacy budget. To further illustrate this trade-off, we next provide an example of the empirical performance of RQP-SGD, demonstrating how randomized projection mitigates performance degradation under strict privacy budgets.

3) RQP-SGD vs Proj-DP-SGD: We employ the utility bound formally characterized in Theorem 2 as a basis to compare RQP-SGD against the adaption of DP-SGD to PSGD (Proj-DP-SGD), which serves as a baseline. This baseline treats privatization and quantization as distinct processes. Our numerical analysis, illustrated in Figure 1, assesses RQP-SGD under a strict $(\epsilon,0)$-DP setting, contrasting it with Proj-DPSGD under a slightly more relaxed $(\epsilon,10^{-7})$-DP. Results in Figure 1b reveal that RQP-SGD has lower utility bounds than Proj-DP-SGD at equivalent privacy levels. However, as shown in Figure 1a, a lower projection randomness coefficient in RQP-SGD, which adds more randomness, can result in a higher utility bound, indicating a critical balance between utility bound and privacy in these SGD frameworks. While these results are based on the theoretical utility bounds, we conduct an empirical comparison of RQP-SGD in Section VI. This section details experiments on the Diagnostic and MNIST datasets to demonstrate the effectiveness of the RQPSGD. Prior to that, we present our approach and analysis for randomized quantization in federated learning.



(a) q = 0.90 (b) q = 0.95

Fig. 1: Utility and privacy trade-off of RQP-SGD (blue) and Proj-DP-SGD. Parameter settings: $b = 4$,$M = 0.3$,$\rho = 0.45, T = 445, \frac{m}{n} = \frac{1}{445}$.

V. Randomized Quantization in Federated Learning

In this section, we study the randomized quantization in federated learning. In RQP-SGD, there is only one parameter controlling the randomness of quantization. However, the utility performance of RQP-SGD is highly sensitive to the quantization randomness. RQP-SGD training with low quantization probability is still challenging, due to high randomness. In this section, we propose Gaussian sampling quantization (GSQ), a novel randomized quantization scheme that infuses stochastic elements into the quantization phase. This approach is grounded in the understanding that quantization inherently involves reducing redundant information and transitioning model parameters from high-precision to low-precision formats. Recognizing that privacy preservation similarly entails the removal of sensitive data, our method capitalizes on the natural overlap between quantization and privacy mechanisms. The crux of this method lies in leveraging the quantization phase as a mechanism to provide privacy. By deliberately incorporating controlled randomness into the quantization process, we aim to meet the specified privacy budget, effectively merging the objectives of data compression and privacy protection.

A. Problem Formulation

The proposed GSQ for federated learning (GSQ-FL) framework aims to address the dual challenge of privacy preservation and communication efficiency in federated learning. The system model of GSQ-FL follows the standard FL setup, similar to the FedAvg framework [36] as described in Section III-B. This setup consists of a central server and K local clients, where each client holds private data and performs local model training. The server coordinates global model aggregation and distribution across the clients.

GSQ-FL introduces discrete Gaussian sampling into the quantization process, embedding privacy directly at the clientside before communication. This approach ensures that model updates are compressed and differentially private prior to transmission, reducing communication overhead without compromising privacy guarantees. Figure 2 illustrates the workflow of the GSQ-FL system. At communication round t, GSQ-FL proceeds through the following steps:

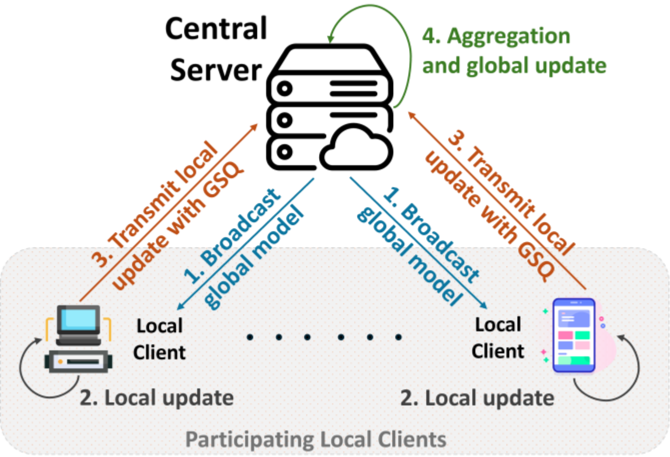

Central Server
4. Aggregation and global update
3. Transmit local update with GSQ
1. Broadcast global model
1. Broadcast global model
3. Transmit local update with GSQ
Local Client
Local Client
2. Local update
2. Local update
Participating Local Clients

Fig. 2: GSQ-FL system workflow at single communication round.

1) Client Selection and Model Broadcasting: The central server selects a subset of clients P_t to participate in the training round and broadcasts the current global model parameters w_t.
2) Local Training: Each selected client $j \in \mathrm{P}_t$ performs E iterations of local training using its dataset D_j, producing local model updates $\Delta w_{j,t}$.
3) Private Compression: The model updates from each client are then processed through the GSQ mechanism. Through GSQ, the model updates are conveyed from the local clients to the central server, with the data compressed to a maximum of b bits per sample.
4) Aggregation and Global Update by the Server: The central server aggregates the quantized updates to refine the global model.

B. Gaussian Sampling Quantization

As a private compression approach, GSQ combines stochastic quantization and the Discrete Gaussian sampling mechanism. Stochastic quantization is introduced to maintain the integrity of the transmitted information. Unlike RQM [57], which achieves Renyi differential privacy through randomly´ sub-sampling quantization levels, GSQ preserves privacy through the use of Discrete Gaussian sampling for the quantization levels. In this work, we provide an experimental comparison between GSQ and RQM in Section VII.

1) Stochastic Quantization: In the work, we consider a $b$bit, stochastic, and uniform quantization approach. In Definition 1, we define a b-bit quantizer.

Definition 1. *(b-bit Stochastic Quantization) Let x be an input with a constraint* $x \in [-C,C]$, *where* $C > 0$. *We define* Q_b *as an b-bit quantization set, where* $\mathrm{Q}_b = \{B(r)|B(r) = -C + \frac{2C}{2^b-1}r\},\ for\ r = 0,1,\cdots,2^b-1$, *a stochastic quantizer* $Q_b(x)$ *randomly maps each* x_i *as:*

$$Q_b(x) = \begin{cases} B(r) & w.\ p.\ \frac{B(r+1)-x}{B(r+1)-B(r)} \\ B(r+1) & w.\ p.\ \frac{x-B(r)}{B(r+1)-B(r)} \end{cases} \quad (19)$$

where $B(r) \leq x \leq B(r+1)$.

The stochastic quantization, employed as an upstream compression method in the context of data transmission, has the advantageous property of transmitting unbiased local updates, i.e. $\mathrm{E}[Q_b(x)] = x$. However, stochastic quantization, while effective for compression, does not inherently preserve the privacy of data points at the extremes of the input range. Although it alters the local update by quantization, this alteration is not primarily aimed at concealing sensitive information. For instance, in the stochastic quantization, an extreme input value like $x = -C$ is mapped deterministically to $B(0)$. Therefore, in scenarios where privacy is a critical concern, relying solely on stochastic quantization is insufficient. Additional LDP mechanisms, such as the Gaussian mechanism are necessary to ensure the individual's privacy. As LDP mechanisms add noise to the model update first, applying stochastic quantization leads to additional variance caused by the noise.

2) The Discrete Gaussian: To address the aforementioned challenge, GSQ utilizes the Discrete Gaussian sampling mechanism as a foundation for achieving differential privacy. In contrast to the noise addition mechanism, the Discrete Gaussian sampling mechanism uses the Discrete Gaussian [10] distribution for differential privacy, defined below:

Definition 2. *(The Discrete Gaussian) Let* $\mu,\sigma \in$ R *with* $\sigma > 0$*. The discrete Gaussian distribution with location* μ *and scale* σ *is denoted by* $\mathrm{N}_{\mathrm{Z}}(\mu,\sigma^2)$*. It is a probability distribution supported on the integers* Z *and defined by*

$$\forall x \in \mathbb{Z}, \underset{X\leftarrow\mathcal{N}_{\mathbb{Z}}(\mu,\sigma^2)}{\mathbb{P}}[X=x] = \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sum_{y\in\mathbb{Z}} e^{-(y-\mu)^2/2\sigma^2}} \tag{20}$$

The Discrete Gaussian is symmetric and centered at μ, sharing many desirable properties of the continuous Gaussian and providing differential privacy. Based on the Discrete Gaussian, we define the Discrete Gaussian sampling mechanism:

Definition 3. *(The Discrete Gaussian Sampling) Let* $\Delta,\sigma > 0$*. Let* q : X^n 7→ Z *satisfies* $|q(x) - q(x')| \leq \Delta$ *for all* $x,x' \in \mathrm{X}^n$ *differing on a single entry. Define the Discrete Gaussian Sampling mechanism* $\mathrm{M}(x,q,\mathrm{Z})$: X^n 7→ Z *outputs an element* $z \in$ Z *with the probability*

$$\mathbb{P}[\mathcal{M}(x,q,\mathbb{Z})=z] = \frac{e^{-(z-q(x))^2/2(\Delta\sigma)^2}}{\sum_{y\in\mathbb{Z}} e^{-(y-q(x))^2/2(\Delta\sigma)^2}} \tag{21}$$

We now provide the full GSQ approach that uses the Discrete Gaussian Sampling mechanism and stochastic quantization.

3) Gaussian Sampling Quantization: We formally present the Gaussian sampling quantization (GSQ) in Algorithm 3. Each communication round within this framework is designed to allow local clients to transmit information at most b bits. The steps of GSQ are as follows:

1) Extension of Output Range: For an input x in the range $[-C,C]$, GSQ first extends the output range of the quantization process. This is achieved by introducing a shift parameter β, resulting in a new range of $[-\tilde{C},\tilde{C}]$ where $\tilde{C} = \frac{2\ -1}{2^b-1-2\beta}C$ $_b$. This extension is crucial as it overcomes a key limitation of standard stochastic quantization, which lacks inherent privacy preservation for data points at the input range's extremes. By enlarging the range, GSQ ensures that inputs at the edges are treated with the same probabilistic approach as other values, thereby bolstering the algorithm's privacy-preserving capabilities.
2) Privacy-Preserving Quantization in the Extended Range: GSQ then applies a two-phase process of quantization within the extended range $[-\tilde{C},\tilde{C}]$:
   a) The Discrete Gaussian Sampling for selecting Quantization Level: After establishing the R quantization levels within the extended range, GSQ identifies the interval $[B(r^*),B(r^*+1)]$ such that $B(r^*) \leq x \leq B(r^*+1)$ for a given input x. The quantization levels are then divided into two sets: $\mathcal{Q}_L = \{B(r)\}_{r=0}^{r^*}$ and $\mathcal{Q}_R = \{B(r)\}_{r=r^*+1}^{R-1}$. GSQ applies the Discrete Gaussian Sampling mechanism separately to Q_L and Q_r, yielding two potential quantization levels $B(r^-)$ and $B(r^+)$.
   b) Stochastic Quantization: In the final step, GSQ performs stochastic quantization as defined in (19) on the sampled quantization levels. This step ensures the achievement of both the privacy guarantee through the Discrete Gaussian Sampling and unbiased quantization through stochastic processing.

Through these procedures, GSQ effectively merges the goals of unbiased quantization and privacy preservation, making it a highly suitable algorithm for federated learning environments where data privacy is of paramount importance.

The hyperparameters in GSQ include the quantization bit b, the quantization shift parameter β, and the Discrete Gaussian parameter σ. Each of these hyperparameters affects the GSQ's behavior and, consequently, its output distribution for a given input. The quantization shift parameter β is particularly influential as it adjusts the range of the output values of GSQ. Different settings of β lead to variations in the quantization levels available for any given input x, thereby altering the optimal threshold r^* for quantization. For example, with an input $x = -0.5C$ and $b = 4$, the choice of β can change the interval within which x falls among the quantization levels. As shown in figure 3, when $\beta = 4$, x falls between $B(5)$ and $B(6)$, and situated between $B(4)$ and $B(5)$ with $\beta = 2$.

The parameter σ of the Discrete Gaussian plays a role in determining the likelihood of selecting a particular quantization interval $[B(r^*),B(r^*+1)]$. A smaller σ implies a higher probability of choosing the quantization interval closely ~~surrounding the input x and, as it influences the steepness~~

~~Algorithm 3 Gaussian Sampling Quantization (GSQ)~~

```
Require: Quantization input $x \in [-C,C]$, the desired quantization bit $b$, a quantization shift parameter $\beta$, and a discrete Gaussian mechanism parameter $\sigma$.
1: Set $\tilde{C} = \frac{2^b-1}{2^b-1-2\beta}C$ ________ , and compute the quantization level:
       $B(r) = -\tilde{C} + \frac{2\tilde{C}}{2^b-1}r\}$ for $r = 0,1,\cdots,2^b-1$.
2: Find $r^*$ such that $B(r^*) \leq x \leq B(r^*+1)$
3: Split the quantization sets: $\mathcal{Q}_L = \{B(r)\}_{r=0}^{r^*}$ and $\mathrm{Q}_R = \{B(r)\}_{r=r^*+1}^{2^b-1}$.
```

```
Discrete Gaussian mechanism: (a)
    Sample $B(r^-)$ from $Q_L$:
```

$$\mathbb{P}[B(r^-) = B(r)] = \frac{e^{-(r^*-r)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}}$$

```
    (b) Sample $B(r^+)$ from $Q_R$:
```

$$\mathbb{P}[B(r^+) = B(r)] = \frac{e^{-(r-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}}$$

```
Stochastic quantization based on $B(r^-)$ and $B(r^+)$:
```

$$Q_R(x) = \begin{cases} B(r^-) & \text{w. p.} \frac{B(r^+)-x}{B(r^+)-B(r^-)} \\ B(r^+) & \text{w. p.} \frac{x-B(r^-)}{B(r^+)-B(r^-)} \end{cases}$$

sampling quantization. At each communication round, a local client partitioning the round first compresses the local updates using the Gaussian sampling quantization, then sends the quantized updates to the server.

1) Privacy in GSQ-FL: Our approach to federated learning with quantization shares similarities with FedPAQ. Both methods integrate quantization to improve efficiency. However, unlike FedPAQ, which relies on the stochastic quantization, we leverage Gaussian Sampling Quantization (GSQ) to guarantee differential privacy for local dege devices. We formulate the differential privacy guarantee of the GSQ in Theorem 3.

Theorem 3. *(Privacy of the Gaussian Sampling Quantization (GSQ)) For any input* $x,x' \in [-C,C]$*, let* b*,* β*, and* σ *be the*

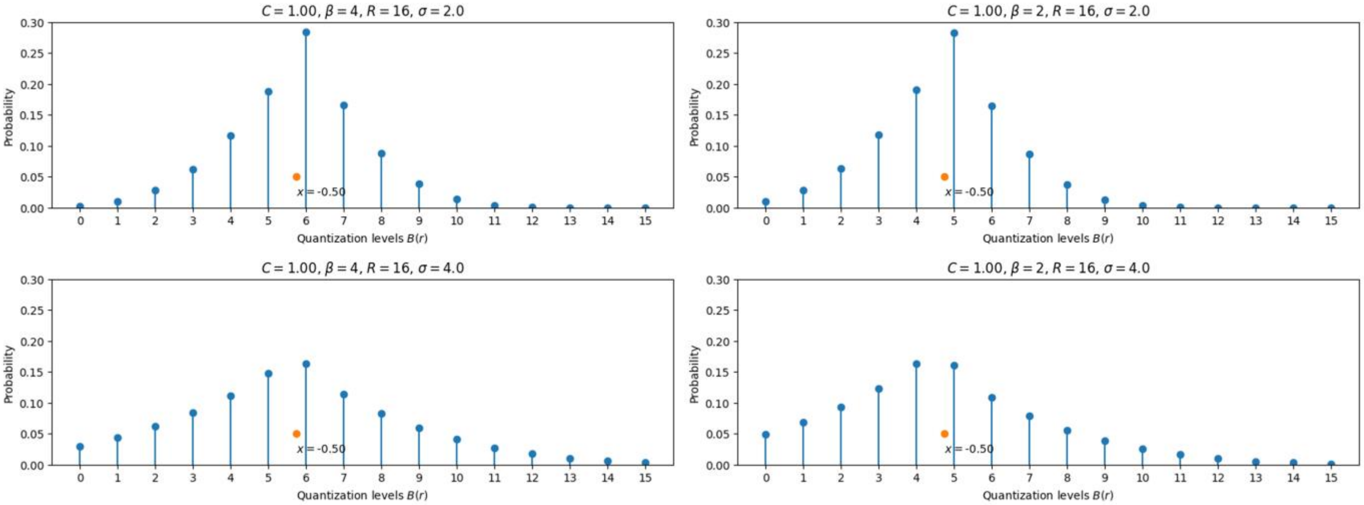


Fig. 3: Examples of quantization levels distributions of GSQ with input $x = -0.5C$ and quantization bit $b = 4$ ($R = 2^b$ in the plot).

of the probability density function of the Discrete Gaussian. Thus, with a smaller σ, GSQ tends to select quantization levels that are closer to the input value. Figure 3 further demonstrates how different settings of β and σ influence the quantization outcome for the same input $x = -0.5C$. Specifically, it shows that for a lower value of σ (i.e. $\sigma = 2.0$), the probability distribution is more concentrated around certain quantization levels (i.e. $[B(5),B(6)]$ in the top left plots) compared to a higher σ (i.e. $\sigma = 4.0$), where the distribution might be more spread out, indicating a broader range of likely quantization intervals.

C. Federated learning with GSQ

In Algorithm 4, we propose GSQ-FL that achieves local differential privacy in federated learning using the Gaussian

Algorithm 4 GSQ-FL: federated learning with the Gaussian sampling quantization

```
Require: $K$ local clients with local dataset $D_k$, local learning rate
    $\eta$, communication rounds $T$, local minibatch size $B$,
    local training iterations $\tau$
for $t = 0,1,\cdots,T-1$ do
    The server select a subset clients $P_t$ uniformly at random
    The server broadcasts global model $w_t$ to selected clients $P_t$
    for each local client $j \in P_t$ do
        Initialize local model: $w_{j,0} \leftarrow w_t$
        for $i = 0,1,\cdots,\tau-1$ do
```

```
            $\nabla\tilde{f}_i(w_{j,i}) = \nabla l(w_{j,i},\xi)$ for a $\xi \in \mathrm{P}_j$ where $\xi$ is a minibatch of $\mathrm{P}_j$ of size $B$
            Perform local model update: $w_{j,i+1} \leftarrow x_{j,i} - \eta\tilde{\nabla} f_i(x_{j,i})$
        end for
        Send quantized local update $\Delta w_j$ to the server:
        $\Delta w_j \leftarrow GSQ(w_{j,\tau} - w_j)$
    end for
    Server aggregates updates:
            compute minibatch gradient:
end for
```

parameter in GSQ, we have

$$D_\infty(\mathrm{P}_{GSQ(x)}\|\mathrm{P}_{GSQ(x')}) \le \log\frac{(2^b-\beta)(2^b-1)}{\beta^2} + \frac{(2^b-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2} \tag{22}$$

And GSQ satisfies $(\log\frac{(2^b-\beta)(2^b-1)}{\beta^2} + \frac{(2^b-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2}, 0)$*-DP.*

Proof. The proof relies on the output distribution of the Gaussian sampling quantization. We formulate the distribution in Lemma 1. The completed proof is detailed in Appendix D. □

Lemma 1. *(Distribution of the GSQ Outputs) For any input* $x \in [-C,C]$*, let* b*,* C_s*, and* σ *be the parameter in GSQ,* r^* *be the integer such that* $B(r^*) \le x \le B(r^*+1)$*, and two sets* $\mathcal{Q}_L = \{B(r)\}_{r=0}^{r^*}$ *and* $\mathcal{Q}_R = \{B(r)\}_{r=r^*+1}^{R-1}$*, the probability of* $GSQ(x) = B(r)$*:*

$$\mathbb{P}(GSQ(x) = B(r)) = \begin{cases} \frac{\phi_L(r)}{\xi_L(r^*)} \sum\limits_{r^+=r^*+1}^{R-1} \frac{\phi_R(r^+)}{\xi_R(r^*)} \frac{B(r^+)-x}{B(r^+)-B(r)} & r \le r^* \\ \frac{\phi_R(r)}{\xi_R(r^*)} [\sum\limits_{r^-=0}^{r^*} \frac{\phi_L(r^-)}{\xi_L(r^*)} \frac{x-B(r^-)}{B(r)-B(r^-)}] & r > r^* \end{cases} \tag{23}$$

where

$$\begin{cases} R = 2^b \\ \phi_L(r) = e^{-\frac{(r^*-r)^2}{2\sigma^2}} \\ \phi_R(r) = e^{-\frac{(r-(r^*+1))^2}{2\sigma^2}} \\ \xi_L(r^*) = \sum\limits_{y=0}^{r^*} e^{-\frac{(r^*-y)^2}{2\sigma^2}} \\ \xi_R(r^*) = \sum\limits_{y=r^*+1}^{R-1} e^{-\frac{(y-(r^*+1))^2}{2\sigma^2}} \end{cases} \tag{24}$$

Proof. The proof is detailed in Appendix C. □

Theorem 3 highlights the influence of the quantization level R, the Discrete Gaussian scale σ, and the shift parameter β on the privacy budget. Specifically, with a fixed quantization level R, an increase in the scale of the Discrete Gaussian (σ) enhances privacy. This is akin to that observed with the continuous Gaussian mechanism, where a higher scale of noise adds more randomness and uncertainty, thereby improving privacy protection. A larger value of β results in a tighter privacy bound. Adjusting upward effectively reduces the potential for privacy leakage in the quantization process.

$$w_{t+1} \leftarrow w_t + \frac{1}{|\mathcal{P}_i|}\sum_{j\in\mathcal{P}_t}\Delta w_j$$

However, while increasing β strengthens privacy, it also introduces the risk of quantization errors that can adversely affect model update quality. Larger values of β may misalign the quantization grid with the underlying distribution of model updates, increasing the likelihood of mapping inputs to suboptimal quantization intervals and thereby amplifying quantization error. This misalignment is particularly pronounced when the variance in local model updates is low, as significant shifts disproportionately impact updates concentrated around narrow regions, resulting in larger deviations. Additionally, in scenarios involving sparse gradients, where only a subset of gradients is non-zero, even minor shifts can lead to substantial quantization errors, ultimately degrading overall model performance. Therefore, careful tuning of the GSQ parameters, particulary β and σ, is crucial to achieving an optimal balance between privacy protection and model update fidelity.

In the GSQ-FL, the privacy composition of a specific local client is dependent on the number of communication rounds between the client and the central serve. Given the parameter in Algorithm 4, and assume the GSQ provides $(\epsilon,0)$-local DP, for any local client k, it holds that the average local differential privacy of the client is $(\frac{r}{K}T\epsilon, 0)$.

2) Convergence of GSQ-FL: The convergence guarantee of GSQ-FL is based on the convergence analysis of FedPAQ. Unlike FedPAQ, GSQ-FL uses the GSQ as the quantization scheme in the federated learning system. The convergence analysis of the FedPAQ assumes the quantizer is unbiased and its variance grows with the squared of L_2-norm of its argument (Assumption 1 in [44]). In addition, other assumptions (Assumption 2 and 3 in [44]) in the FedPAQ convergence analysis, including the property of the loss function and the bias and variance of stochastic gradients, are feasible when employing GSQ in the federated learning system.

Lemma 2. *(Unbiasedness and Variance of the Gaussian Sampling Quantization (GSQ)) For any input* $x \in [-C,C]$*, let* b*,* β*, and* σ *be the parameter in GSQ, and* $t > 0$ *such that* $x \le e^{tx}$ *for* $x > 0$*, we have*

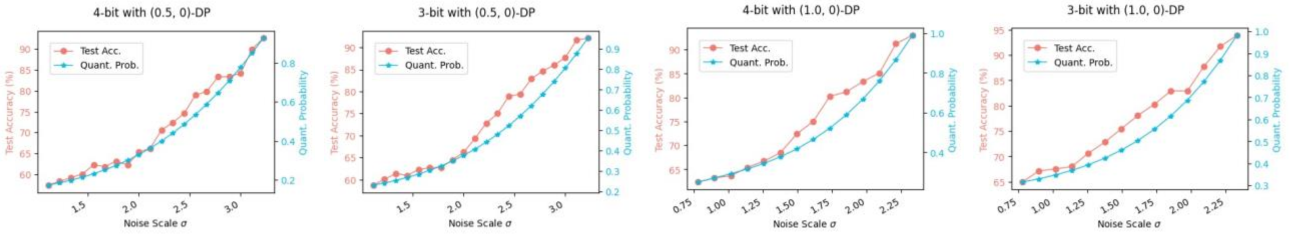
$$\begin{cases} \mathbb{E}_{x\sim\ GSQ(x)}[GSQ(x)] = x \\ \mathbb{E}_{x\sim GSQ}[\|GSQ(x) - x\|^2] \le pC^2 \end{cases} \tag{25}$$

for some positive real constant p.

Proof. The completed proof is detailed in Appendix E. □

In Lemma 2, we demonstrate that the Gaussian sampling quantization is unbiased and its variance grows with the squared of L_2-norm of its argument. These properties ensure that the output of the GSQ is an unbiased estimator of the input with bounded variance, which is aligned with the assumption of the quantizer (Assumption 1 in [44]) in FedPAQ. Consequently, the assumption regarding the quantizer in FedPAQ is satisfied when applying GSQ in the federated learning system. Hence, the convergence guarantee of FedPAQ is applicable for the analysis of GSQ-FL. For a more comprehensive discussion on the convergence analysis, we refer to Theorem 2 in [44].

In the GSQ-FL, to ensure the input of the GSQ lies within the range $[-C,C]$, C an serve as a clipping threshold, directly affecting the variance bound of GSQ as outlined in Lemma 2. Since the variance of GSQ is bounded by pC^2, the selection of C is critical in balancing the trade-off between quantization error and information retention. A larger C allows broader representation of model updates, but increases the quantization error. Conversely, a smaller C reduces the quantization error, but risks discarding informative updates that exceed the clipping threshold C. In the GSQ-FL, C can be determined empirically through a grid search over a predefined range of values. The selection is based on optimizing for the trade-off between model utility and privacy, ensuring that C aligns with distribution of model updates, thereby minimizing utility loss while maintaining efficient quantization. The specific values of C used in the GSQ-FL experiments are provided in the Section VII-B.



(a) 4-bit with $(0.5,0)$-DP

(b) 3-bit with $(0.5,0)$-DP

(c) 4-bit with $(1.0,0)$-DP

(d) 3-bit with $(1.0,0)$-DP

Fig. 4: Noise scale (σ) and projection randomness coefficient (q) trade-off of LogReg classifier on Diagnostic. Test accuracy

VI. Experiments of RQP-SGD

In this section, we implement RQP-SGD on classification tasks using ML algorithms.

a) Setup: We applied RQP-SGD to classification tasks using the Diagnostic [51] and MNIST datasets [31]. The Diagnostic dataset, with 569 30-dimensional instances, was split into 80% training and 20% testing. MNIST contains 70,000 28x28 pixel grayscale images of handwritten digits, split into 60,000 for training and 10,000 for testing.

b) Training Details: We used logistic regression (LogRes) [52] and linear support vector machine (SVM) [23] classifiers on the Diagnostic dataset, implementing DP-SGD, the adaption of DP-SGD to P-SGD (Proj-DP-SGD), and RQPSGD with a mini-batch size of 10, step size of 1, and 46 training iterations. For the MNIST dataset, we implemented LogReg classifier with a mini-batch size of 64, step size of 1.0, and 938 training iterations. For all learning algorithms, we employed the gradient clipping technique with l_2 norm of 0.45. For RQP-SGD and Proj-DP-SGD, we set $[-0.3,0.3]$ as the parameter space bound and set 4 as the quantization bit.

c) Results: We set $(1.0,10^{-7})$ as the privacy budget of DP-SGD and Proj-DP-SGD, and set $(1.0,0)$ as the privacy budget of RQP-SGD. We report the test accuracy of the resulting ML models in Table I. It clearly demonstrates that RQP-SGD achieves better utility performance than Proj-DPSGD. Especially, the SVM classifier using RQP-SGD leads to 35.84% median test accuracy gain among the DP-SGD with deterministic projection on the Diagnostic dataset.

TABLE I: Test accuracy (%) of classifiers with DP-SGD, Proj-DP-SGD, and $\mathrm{A}_{\text{RQP-SGD}}$. We report median and standard deviation values over 10 runs.

	Diagnostic (LogReg Classifier)	Diagnostic (SVM Classifier)	MNIST (LogReg Classifier)
Non-Private	97.37% (0.56%)	98.68% (0.68%)	87.25% (0.07%)
DP-SGD	96.92% (1.80%)	96.49% (1.24%)	86.02% (0.33%)
Proj-DP-SGD	94.30% (1.41%)	69.74% (18.37%)	84.32% (0.29%)
RQP-SGD	95.18% (1.42%)	94.74% (1.53%)	84.81% (0.30%)

*d) Impact of Noise Scale and Projection Randomness:*To better understand the Privacy-Utility trade-off, we adjusted noise scales while maintaining a fixed privacy budget and quantization bits. The quantization randomness coefficient (q) is calculated by Theorem 1. As presented in Fig. 4, the quantization randomness coefficient - noise scale curve values are the median over 10 runs.

(shown in the blue line) illustrates that decreasing quantization probability can enhance privacy while allowing less noise. On the utility front (represented by the red line), a lower projection randomness coefficient (q) leads to degraded test accuracy. For instance, when the projection randomness coefficient (q) is 0.3, the test accuracy decreases to around 62.5%. The utility drop is attributed more to the randomness from quantization than from noise addition. From our observation, the standard deviation of test accuracy is higher when the projection randomness coefficient (q) is lower. This further illustrates the impact of the projection randomness coefficient (q) on utility.

e) Impact of Quantization Bit: We also extend our experiments with different quantization bits to explore the impact of the quantization bits. Based on our observation, the test accuracy does not increase as increasing the quantization bits. This is because the higher the quantization bits, the less randomness is provided by quantization. To maintain the same privacy level, the randomness coefficient (q) is lower which results in higher utility loss.

VII. Experiments of GSQ-FL

In this section, we evaluate the performance of GSQ-FL across multiple datasets. In particular, GSQ-FL is applied to distributed classification tasks with a federated learning work, and compared with the following baseline methods:

- FedAvg [36]: a foundational technique in federated learning aggregates model updates from participating clients.
- FedPAQ [44]: a communication-efficient approach based on FedAvg, utilizing quantization to reduce the data transmitted. The participating clients send quantized local updates to the central server.
- DP-FedAvg [37]: an enhanced FedAvg by ensuring local differential privacy through the Gaussian mechanism applied to local updates. Participating clients send fullprecision updates to the central server.
- DP-FedPAQ: an approach combines the privacy features of DP-FedAvg and the quantization strategy of FedPAQ. It employs the Gaussian mechanism for privacy and sends quantized updates to the central server. While recent studies [27], [64] have proposed various quantization schemes based on the stochastic quantization from FedPAQ, we use stochastic quantization in our experiments.
- RQM [57]: a randomized quantization mechanism in achieving local differential privacy and communication efficiency in federated learning. The RQM randomly sub-samples feasible quantization levels, then employs a randomized rounding procedure using these sub-sampled discrete levels.

A. Dataset Setup

GSQ-FL is tested on four datasets to ensure a diverse evaluation across different data distributions:

- MNIST [31]: A dataset 70,000 grayscale images of handwritten digits (28 × 28 pixels), with 60,000 images used for training and 10,000 used for testing.
- Fashion-MNIST [54]: Similar to MNIST, this dataset contains 70,000 (60,000 images for training and 10,000 for testing) grayscale images of fashion categories (28× 28 pixels).
- CIFAR-10 [29]: This dataset consists of 60,000 RGB images (32 × 32 pixels) divided into 10 classes. The training set contains 50,000 images, while the test set has 10,000.
- FEMNIST [9]: A federated version of the EMNIST [14] dataset, where data is partitioned by writer identity, reflecting real-world client distributions in federated learning environment.

B. Data Distribution and Federated Learning Setup

To simulate federated learning environments, datasets are distributed across clients in both IID (independent and identically distributed) and non-IID settings:

- For MNIST and Fashion-MNIST, the data is distributed as follow:
  - IID Distribution: Data is randomly shuffled and partitioned across 100 clients, with each client receiving 600 samples.
  - Non-IID Distribution:

    (1) Label-Shard Partitioning [36]: Data is sorted by label and split into 200 shards, each containing 300 samples. Each client receives two shards, resulting in 100 clients.

    (2) Dirichlet-based Partitioning [34]: To simulate varying degrees of heterogeneity, data is partitioned by sampling from a Dirichlet distribution $Dir(\alpha)$, where α = 0.1 and α = 0.5 are used to reflect severe and moderate non-IID conditions, respectively. A lower α results in more skewed distributions, while higher α yields more balanced allocations across clients.
- For CIFAR-10, data is partitioned into 100 clients in an IID manner, with each client receiving 500 samples.
- For FEMNIST, data is naturally partitioned by writer, with each writer's data assigned to a unique client, simulating non-IID conditions.

The model architectures for each dataset are summarized in Table II. Table III summarizes the experimental setup, and GSQ-FL parameters. Given that each local client holds varying dataset sizes under Dirichlet-based partitioning, we represent the minibatch size as a ratio. The local minibatch size is computed by multiplying this ratio with the local dataset size. The clipping threshold C bounds the model updates during quantization to minimize utility loss while preserving privacy. The quantization shift parameter β and the noise scale σ are tuned through grid search under the constraints of the privacy budget ϵ. We perform joint optimization over C, β and σ to find the optimal setting for each dataset, ensuring minimal quantization error while maintaining differential privacy guarantees.

C. Results

Tables IV to VII present the test accuracy of GSQ-FL across MNIST, Fashion-MNIST, CIFAR-10, and FEMNIST datasets using CNN and MLP models. GSQ-FL consistently demonstrates strong performance across all datasets, particularly in Non-IID settings, where it achieves notable accuracy improvements over baseline methods. Key Observations:

- MNIST-CNN (Table IV): GSQ-FL achieves a 3.81% accuracy drop compared to the non-private FedPAQ baseline under IID conditions at ϵ = 2.0. In Non-IID scenarios, GSQ-FL surpasses RQM by 8.29% under the Dirichlet-based partitioning with $Dir(0.1)$, reflecting its robustness under highly heterogeneous data distributions. • MNIST-MLP (Table V): In the IID case, GSQ-FL shows only a 3.20% accuracy drop compared to FedPAQ. For $Dir(0.1)$, GSQ-FL achieves a 14.08% improvement over DP-FedPAQ, highlighting its effectiveness across diverse model architectures.
- Fashion-MNIST-CNN (Table VI): GSQ-FL shows significant accuracy gains in Non-IID conditions. Under Dirichlet-based partitioning with $Dir(0.1)$, GSQ-FL achieves a 20.60% improvement over DP-FedPAQ and a 2.93% improvement over RQM. For $Dir(0.5)$, GSQ-FL achieves a 11.82% improvement over DP-FedPAQ and a 2.75% improvement compared to RQM.
- CIFAR-10 and FEMNIST (Table VII): On CIFAR-10, GSQ-FL improves accuracy by 16.54% over DP-FedPAQ and by 10.11% over RQM under ϵ = 8.0. On FEMNISTdigits, GSQ-FL achieves 8.72% higher accuracy than DPFedPAQ and outperforms RQM by 1.27% at ϵ = 2.0.

a) Overall Performance: GSQ-FL consistently surpasses conventional private communication-efficient FL approaches (DP-FedPAQ) and randomized quantization methods (RQM). In all experiments, GSQ achieves accuracy improvements of 0.97% to 27.86% compared to DP-FedPAQ, and 1.06% to 39.56% compared to DP-FedAvg. When compared to RQM, GSQ-FL outperforms it by 1.83% to 10.42%. Despite this minor underperformance in one scenario, GSQ-FL demonstrates the effectiveness in providing a privacy-preserving solution in federated learning across diverse datasets and partitioning strategies.

D. Ablation Study of GSQ-FL

In this section, we study the impact of the GSQ parameters on the performance of the federated learning system. The GSQ parameters include the quantization levels R, the shift parameter β, and the Discrete Gaussian scale σ. The NonIID case of the MNIST-CNN is studied in the section. The hyperparameter setting is as follows: a total of 100 local clients with a participation rate of 0.1 per round, allowing 10 clients to participate in each communication round. The local batch size is set to 60, the local training rounds τ to 1, and the number of communication rounds T to 100.

a) Impact of the Quantization levels: We set the average local differential privacy budget to ϵ = 1.5, and train the MNIST-CNN with different quantization levels (R = 8,16,32). To maintain the output range of the GSQ, we set the GSQ shift parameter to$\beta = \frac{R}{4}$ and compute the Discrete Gaussian scale according to the relevant privacy parameters. After 100 communications rounds, the GSQ-FL achieves a test accuracy of 76.73% with R = 8, 81.16% with R = 16, and 83.28% with R = 32. This demonstrates that the performance of GSQ-FL decreases as the quantization level decreases. Intuitively, a smaller quantization level allows less information transmission in each communication round, thus leading to worse utility performance.

b) Impact of shift parameter and Discrete Gaussian:

Table VIII presents the test accuracy of GSQ-FL on MNISTCNN across different β values, with results reported as the median over 5 runs. The general trend suggests that larger β values lead to reduced performance. However, the test accuracy at β = 4 deviates from this trend, yielding higher accuracy than at β = 3 and β = 5. This anomaly indicates a potential quantization-stability tradeoff at β = 4, where the shift in quantization bins aligns with the variance of model updates, mitigating noise. Additionally, the corresponding noise scale σ = 7.31 may reflect a favorable balance between preserving updates and satisfying privacy constraints. Although β = 2 achieves the highest median accuracy (84.44%), the results at β = 4 highlight the importance of tuning β to balance quantization error and performance. Future work will explore adaptive adjustment mechanisms for β and refinement of noise scale σ to enhance GSQ-FL's stability across datasets and models.

c) GSQ parameter selection: The choice of GSQ parameters, including the shift parameter β, quantization level R, and discrete Gaussian noise scale σ, is critical for balancing privacy, communication efficiency, and model accuracy.

The shift parameter β governs the spacing between quantization bins. Lower values (e.g., β = 2,3) provide tighter quantization intervals, preserving performance by capturing small model update variations. Higher values (e.g., β = 5,6) introduce coarser quantization, increasing noise but potentially

TABLE II: Model Architectures for Different Datasets. *Conv(k × k, n)* represents convolutional layers with a kernel size of $k \times k$, where n indicate the number of output channels for each layer. Each convolutional layer is followed by a 2×2 max pooling operation unless otherwise specified. *FC(n)* denotes a fully connected layer with n output channel.

Dataset	Model Type	Architecture	Activation
MNIST	CNN	Conv(5×5, 10) + Conv(5×5, 20) + FC(50) + FC(10)	ReLU + Softmax
	MLP	FC(256) + FC(10)	ReLU + Softmax
Fashion-MNIST	CNN	Conv(5×5, 16) + Conv(5×5, 32) + FC(10)	ReLU + Softmax
CIFAR-10	CNN	Conv(3×3, 64) + Conv(3×3, 64) + FC(384) + FC(192) + FC(10)	ReLU + Softmax
FEMNIST	CNN	2 Conv (7×7, 32) + (3×3, 64) + FC(10)	ReLU + Softmax

TABLE III: Experimental setup and GSQ-FL parameters choice. The minibatch size for local client is computed by multiplying the minibatch ratio with the local dataset size. τ denotes the local training iteration. C,β,σ are GSQ parameters.

Dataset	Clients	Participation	Minibatch Ratio	Rounds (T)	τ	Privacy (ϵ)	C	β	σ
MNIST	100	10 (0.1)	0.10	100	1	2.0	0.02	5	26.78
Fashion-MNIST	100	10 (0.1)	0.05	200	1	2.0	0.02	5	26.78
CIFAR-10	100	5 (0.05)	0.05	200	8	8.0	0.008	6	11.85
FEMNIST	3,580	40	1.00	100	1	2.0	0.01	6	14.89

TABLE IV: Summary of results for CNN model trained on MNIST.

Algorithm	Privacy	Quantization	MNIST-CNN			
			IID	Label Shard	$Dir(0.1)$	$Dir(0.5)$
FedAvg	∞	full-precision	96.79%	94.32%	93.88%	96.08%
FedPAQ	∞ $\epsilon = 2.0$	4-bit	94.85%	91.54%	93.46%	95.53%
DP-FedAvg		full-precision	83.36%	76.66%	48.48%	80.42%
DP-FedPAQ	ϵ = 2.0	4-bit	84.36%	76.63%	60.79%	74.82%
GSQ-FL	ϵ = 2.0	4-bit	91.04%	83.29%	88.04%	89.12%
RQM	ϵ = 2.0	4-bit	85.27%	75.43%	81.30%	86.51%

TABLE V: Summary of results for MLP model trained on MNIST.

Algorithm	Privacy	Quantization	MNIST-MLP			
			IID	Label Shard	$Dir(0.1)$	$Dir(0.5)$
FedAvg	∞	full-precision	92.05%	90.49%	89.82%	90.70%
FedPAQ	∞	4-bit	91.84%	90.31%	85.10%	90.46%
DP-FedAvg	ϵ = 2.0	full-precision	87.71%	85.75%	67.81%	85.60%
DP-FedPAQ	ϵ = 2.0	4-bit	87.79%	85.47%	73.54%	84.41%
GSQ-FL	ϵ = 2.0	4-bit	88.64%	87.54%	87.62%	89.19%
RQM	ϵ = 2.0	4-bit	86.91 %	80.49%	82.44%	86.78%

mitigating overfitting. Our experiments reveal that $\beta = 2$ achieves the highest accuracy, while $\beta = 4$ yields an unexpected performance stabilization.

Quantization level R is fixed at 16 (4-bit quantization), balancing communication efficiency and accuracy, consistent with prior federated learning research. The discrete Gaussian noise scale σ is adjusted to ensure the same privacy budget $\epsilon = 2.0$ across experiments, aligning with the shift parameter β to maintain differential privacy guarantees.

This parameter selection approach reflects empirical findings and aligns with best practices in quantized federated learning, highlighting the importance of careful tuning for optimal performance.

VIII. Limitations and Discussions

While RQP-SGD and GSQ-FL demonstrate significant improvements in privacy preservation, communication efficiency, and model accuracy, certain limitations merit further investigation.

In this work, we explore the integration of RQP-SGD into DP-SGD in quantized ML tasks with convex objectives. One key limitation of RQP-SGD lies in its sensitivity to projection randomness. Training under low randomness coefficient (q) remains challenging due to the high degree of randomness introduced during projection. This can lead to unstable convergence and hinder model performance. Developing techniques to manage or adaptively control this randomness could contribute to more stable training processes. A deeper exploration into the trade-off between projection randomness and model utility remains an open and promising area for future research. Additionally, the current scope of RQP-SGD is limited to convex optimization problems. Extending the method to nonconvex tasks, such as neural network training, introduces additional complexities, particularly in ensuring convergence under randomized quantization. Investigating how RQP-SGD can be applied to non-convex objectives, or adapting the framework for deep learning models, offers a valuable direction for further analysis and practical deployment.

For GSQ-FL, while it has demonstrated improvements in balancing privacy, utility, and communication efficiency in federated learning, its real-world deployment presents several challenges. One limitation is the scalability of GSQ-FL in large-scale federated learning environments involving a high number of participating clients. As the number of clients increases, the accumulation of quantization error from multiple updates can slow down convergence and degrade overall performance. This issue becomes more pronounced when dealing with non-iid data distributions across clients. Addressing these challenges may require the development of more adaptive quantization strategies that adjust to the heterogeneity of client updates or the implementation of dynamic aggregation techniques to mitigate the compounding effect of quantization noise.

TABLE VI: Summary of results for CNN model trained on Fashion-MNIST.

Algorithm	Privacy	Quantization	Fashion-MNIST-CNN			
			IID	Label Shard	$Dir(0.1)$	$Dir(0.5)$
FedAvg	∞	full-precision	87.12%	82.56%	82.70%	85.63%
FedPAQ	∞	4-bit	87.12%	78.56%	81.26%	85.28%
DP-FedAvg	$\epsilon = 2.0$	full-precision	74.75%	66.85%	56.28%	70.85%
DP-FedPAQ	$\epsilon = 2.0$	4-bit	74.66%	63.25%	59.43%	70.51%
GSQ-FL	$\epsilon = 2.0$	4-bit	81.52%	79.44%	80.03%	82.33%
RQM	$\epsilon = 2.0$	4-bit	78.74%	78.01%	77.28%	79.40%

TABLE VII: Summary of results for CNN model trained on CIFAR-10 and FEMNIST-digits. The ϵ of DP-FedAvg, DP-FedPAQ, and GSQ-FL on CIFAR-10 is 8.0, and 2.0 for FEMNIST-digits.

Algorithm	Quantization	CNN	
		CIFAR-10	FEMNIST-digits
FedAvg	full-precision	74.65%	96.25%
FedPAQ	4-bit	65.98%	96.44%
DP-FedAvg	full-precision	41.41%	85.52%
DP-FedPAQ	4-bit	40.60%	85.02%
GSQ-FL	4-bit	57.14%	93.74%
RQM	4-bit	47.03%	92.47%

TABLE VIII: Test accuracy of the MNIST-CNN with different GSQ parameters. We report median over 5 runs.

β	2	3	4	5	6
σ	50.64	9.92	7.31	6.19	5.59
Test Accuracy	84.44%	83.53%	84.27%	82.36%	80.91%

Another notable limitation of GSQ-FL is that its privacy guarantee depends heavily on the shift parameter β, rather than solely relying on the discrete Gaussian noise scale. While the discrete Gaussian mechanism plays a central role in ensuring differential privacy, β directly modulates the quantization and significantly affects the noise distribution. This reliance on β complicates the tuning of the GSQ, as improper β selection can introduce excessive noise, degrading utility, or provide insufficient privacy protection, undermining the effectiveness of the GSQ-FL. Exploring alternative β selection strategies that are less sensitive to the scale of gradients may enhance the robustness of GSQ-FL across diverse federated learning environments. Potential approaches include adaptive β selection based on parameter variance or dimensionality, and layer-wise GSQ-FL, applying different quantization levels to different layers. Addressing this limitation could stabilize privacy guarantees and improve performance consistency.

IX. Conclusion

In this work, we address privacy challenges in machine learning at the edge. We study two application scenarios: the deployment of machine learning algorithms on resourceconstrained IoT devices and federated learning with communication constraints. For the first scenario, we propose RQP-SGD, a novel approach for providing differential privacy in machine learning using quantized computational models. RQP-SGD combines differentially private noise addition with randomized quantization projection, introducing additional randomness that enables noise reduction and improves utility. We theoretically analyze the feasibility of RQP-SGD for training ML models with convex objectives and validate its effectiveness through experiments on real datasets. For the second scenario, we introduce GSQ-FL, an innovative randomized quantization scheme designed to provide client-level differential privacy in federated learning systems. Extensive experimental results demonstrate that GSQ-FL consistently outperforms conventional private federated learning methods in communication-constrained scenarios.

Appendix A Proof of Theorem 1

Theorem 1. *For any $\epsilon > 0$, there exists mini-batch sampling rate $\frac{m}{n}$, training iterations T, quantization bit b and randomness coefficient q, noise scale σ such that Algorithm 2 achieves $(\epsilon,0)$-DP.*

Proof. The key to the proof is determining the differential privacy budget for each update in the RQP-SGD process. This total budget is then assembled using a composition method. The process involves three main steps:

1) Compute the probability distribution of the quantized weights.

In each update of RQP-SGD, the gradient is calculated from a mini-batch dataset $B_t = \{(x_j, y_j)\}_{j=1}^{m}$, which is uniformly sampled from the training dataset S. The term $f_{sgd}(w_t;B_t)$ represents the standard SGD update employing the stochastic mini-batch B_t, which is defined as

$$f_{sgd}(w_t; B_t) = w_t - \eta \cdot \frac{1}{m} \sum_{j=1}^{m} \nabla l(f(x_j; w_t); y_j)$$

In contrast to $f_{sgd}(w_t;B_t)$, step 4 of Algorithm 2, denote as $f_v(w_t;B_t)$, adds noise drawn from normal distribution. The noise is scaled accordingly, leading to the formula:

$$f_v(w_t; B_t) = f_{sgd}(w_t; B_t) + \frac{\eta}{m} \cdot \mathcal{N}(0, \sigma^2 \mathbb{I}_d)$$

Let the probability density function of $f_v(w_t;B_t) = v_{t+1}$ be denoted as $p_{w_t,B_t}(v_{t+1})$, and $\mathrm{M}^t_{RQP}(w_t,B_t)$ be denoted as the t-th RQP-SGD update with the minibatch B_t. The probability of the quantized weights can be expressed as

$$Pr\{\mathrm{M}^t_{RQP}(w_t,B_t) = Q_i\}$$
$$= \int Pr\{\mathrm{Proj}^R_{\mathcal{Q}_{M,b}}(v) = Q_i\} \cdot p_{w_t,B_t}(v)dv$$

The above equation computes the probability of the weight being reduced to a particular quantization level Q_i.

- Randomized projection:

  For the projection input v, let $i^* = \mathrm{argmin}_i \|Q_i - v\|^2$, the probability of $\mathrm{Proj}^R_{\mathcal{Q}_{M,b}}(v)$ equaling Q_{i^*} is given by

  $$Pr\{\mathrm{Proj}^R_{\mathcal{Q}_{M,b}}(v) = Q_{i^*}\} = q \cdot Pr\{Q^-_{i^*} \leq v < Q^+_{i^*}\}$$

  where $Q^-_{i^*} = Q_{i^*} - \frac{M}{2^b-1}$ and $Q^+_{i^*} = Q_{i^*} + \frac{M}{2^b-1}$.

  For any $Q_j \in \mathrm{Q}_{M,b}$ but not equal to Q_{i^*}, the probability is

  $$Pr\{\mathrm{Proj}^R_{\mathrm{Q}_{M,b}}(v) = Q_j\}$$
  $$= \frac{1-q}{2^b-1} \cdot (1 - Pr\{Q^-_{i^*} \leq v < Q^+_{i^*}\})$$

  where $Q^-_j = Q_j - \frac{M}{2^b-1}$ and $Q^+_j = Q_j + \frac{M}{2^b-1}$.

- Distribution of $f_v(w_t;B_t)$:

  The function $f_v(w_t;B_t)$ introduces noise to $f_{sgd}(w_t;B_t)$. This implies that v adheres to a normal distribution $\mathcal{N}(f_{sgd}(w_t; B_t), (\frac{\eta}{m}\sigma)^2)$.

Based on the probability distributions of randomized projection and $f_v(w_t;B_t)$, the probability distribution of $\mathrm{M}^t_{RQP}(w_t,B_t)$ for any $Q_i \in \mathrm{Q}_{M,b}$ can be formulated as follows:

$$Pr\{\mathcal{M}^t_{RQP}(w_t, B_t) = Q_i\}$$

$$= {}^{\mathrm{R}} Pr\{\mathrm{Proj}^R_{\mathrm{Q}_{M,b}}(v) = Q_i\} \cdot p_{w_t,B_t}(v)dv$$

$$\begin{aligned} = \; & q \cdot [\Phi(\tfrac{Q^+_i - f_{sgd}(w_t;B_t)}{\sigma_l}) - \Phi(\tfrac{Q^-_i - f_{sgd}(w_t;B_t)}{\sigma_l})] + \\ & \tfrac{1-q}{2^b-1} \cdot [1 - \Phi(\tfrac{Q^+_i - f_{sgd}(w_t;B_t)}{\sigma_l}) + \Phi(\tfrac{Q^-_i - f_{sgd}(w_t;B_t)}{\sigma_l})] \\ = \; & \tfrac{2^b q - 1}{2^b-1} [\Phi(\tfrac{Q^+_i - f_{sgd}(w_t;B_t)}{\sigma_l}) - \Phi(\tfrac{Q^-_i - f_{sgd}(w_t;B_t)}{\sigma_l})] \\ & + \tfrac{1-q}{2^b-1} \end{aligned}$$

where $Q^-_i = Q_i - \frac{M}{2^b-1}$, $Q^+_i = Q_i + \frac{M}{2^b-1}$, $\sigma_l = \frac{\eta}{m}\sigma$, and $\Phi(\cdot)$ denotes the cumulative density function (CDF) of the standard normal distribution $\mathrm{N}(0,1)$.

2) Use Renyi divergence to determine the maximum ratio´ of the probabilities of any two input weight values.

$\log \frac{Pr\{\mathcal{M}_{RQP}(w_t,B_t)=Q_i\}}{Pr\{\mathcal{M}^t_{RQP}(w_t,B'_t)=Q_i\}}$ In this step, we compute the upper bound of

t

for any two adjacent sets $B_t,B_t' \in S$. The upper bound of this logarithmic ratio is synonymous with the definition of the ∞-th order of Renyi divergence [49]. Specifically, for two probability´ distributions P_1 and P_2 defined over R, the ∞-th order of Renyi divergence [49] is given as´

$$D_\infty(P_1 \| P_2) = \log \sup_{d \in \mathcal{D}} \frac{P_1(d)}{P_2(d)}$$

Utilizing this definition, the following relation can be established for the RQP-SGD update:

$$\begin{aligned} & \sup_{Q_i \in \mathcal{Q}_{M,b}} \log \tfrac{Pr\{\mathcal{M}^t_{RQP}(w_t,B_t)=Q_i\}}{Pr\{\mathcal{M}^t_{RQP}(w_t,B'_t)=Q_i\}} \\ = \; & D_\infty(Pr\{\mathcal{M}^t_{RQP}(w_t, B_t) = Q_i\} \| \\ & Pr\{\mathcal{M}^t_{RQP}(w_t, B'_t) = Q_i\}) \\ = \; & \{\log \tfrac{\frac{2^b q-1}{2^b-1}[\Phi_+(f(B_t)) - \Phi_-(f(B_t))] + \frac{1-q}{2^b-1}}{\frac{2^b q-1}{2^b-1}[\Phi_+(f(B'_t)) - \Phi_-(f(B'_t))] + \frac{1-q}{2^b-1}}\} \\ \max \; & \Phi_+(f(B_t)) = \Phi(\tfrac{Q^+_i - f_{sgd}(w_t;B_t)}{\sigma_l}) \\ \text{where} \; & \Phi_-(f(B_t)) = \Phi(\tfrac{Q^-_i - f_{sgd}(w_t;B_t)}{\sigma_l}) \\ & \Phi_+(f(B'_t)) = \Phi(\tfrac{Q^+_i - f_{sgd}(w_t;B'_t)}{\sigma_l}) \\ & \Phi_+(f(B'_t)) = \Phi(\tfrac{Q^-_i - f_{sgd}(w_t;B'_t)}{\sigma_l}) \end{aligned} \tag{26}$$

Given that Renyi divergence is quasi-convex [49], (26)´ achieves its maximum at the extreme points.

3) Compute the achieved differential privacy using the maximum ratio of probabilities.

The loss function $l(f(w_t;\cdot);\cdot)$ is ρ-Lipschitz with respect to w_t, according to Lemma 14.7 in [47]. This means for any data-label pair $(x_i,y_i) \in S$, the norm of the gradient is bounded by ρ: $\|\nabla l(f(w_t;x_i);y_i\| \leq \rho$. With the assumption that w_t is bounded by M, for any two adjacent $B_t,{}^{B}{}_t{}' \in$ S, the maximum difference of $f_{sgd}(w_t;\cdot)$ is constrained by C: $\max\|f_{sgd}(w_t;B_t) - f_{sgd}(w_t;B_t)\| \leq C$ where

$C = M - \eta\rho$. Incorporating the extreme value of $f_{sgd}(w_t;B_t)$ into (26), the following inequality is obtained:

$$\log \frac{Pr\{\mathcal{M}^t_{RQP}(w_t, B_t) = Q_i\}}{Pr\{\mathcal{M}^t_{RQP}(w_t, B'_t) = Q_i\}} \leq \epsilon_t$$

where $\epsilon_t \quad = \quad \log \frac{\frac{2^b q-1}{2^b-1}[2\Phi(\frac{a_1}{\sigma_l})-1]+\frac{1-q}{2^b-1}}{\frac{2^b q-1}{b}[\Phi(\frac{a_2}{\sigma})-\Phi(\frac{a_3}{\sigma})]+\frac{1-q}{b}}, \quad a_1 \quad =$ $\frac{M}{2^b-1}$,$a_2 = M + \frac{M}{2^b-1} + C$, and $a_3 = M - \frac{M}{2^b-1} + C$.

This shows that each RQP-SGD update is $(\epsilon_t,0)$-DP with respect to the stochastic mini-batch B_t.

Considering that RQP-SGD performs a total of T iterations, with each iteration involving the uniform sampling of a stochastic mini-batch with replacement, the total privacy budget can be composed. By leveraging the DP composition theorem [18] and amplification of DP via subsampling [5], the overall privacy budget for RQP-SGD is established as $(T\frac{m}{n}\epsilon_t, 0)$-DP. □

APPENDIX B PROOF OF THEOREM 2

Theorem 2. *Let* $\bar{W}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$*. Suppose the parameter set* W *is convex and M-bounded, and the quantization set* Q_p *is generated by a randomized quantizer with probability q and b-bit. For any* $\eta > 0$*, the excess empirical loss of* $\mathrm{A}_{ProjNSGD}$ *satisfies*

$$\begin{aligned} &\mathbb{E}\left[\hat{\mathcal{L}}(\bar{w}_T;\mathcal{S})\right] - \min_{w\in\mathcal{W}} \hat{\mathcal{L}}(w;\mathcal{S}) \\ \leq \quad & \frac{M^2}{2\eta T} + E_Q + \frac{\eta\rho^2}{2} + E_N \end{aligned} \tag{18}$$

where $E_Q = dM^2[\frac{q}{(2^b-1)^2} + \frac{2^{b+1}(2^{b+1}-1)}{3(2^b-1)^2}(1-q)]$ *denotes the quantization error and* $E_N = \eta\sigma^2 d$ *denotes the noise error.*

Proof. The key to the proof is determining the boundary of the randomized projection. The projection process w_{t+1} = Proj $^{R}_{\mathcal{Q}_{M,b}}(v_{t+1})$ involves the randomized projection of $\tilde{w}_{t+1}$ onto $\mathrm{Q}_{M,b}$. For the projection, we have:

$$\begin{aligned} &\mathop{\mathbb{E}}_{w\in\mathcal{Q}_{M,b}}\left[\|v_{t+1} - w_{t+1}\|^2\right] \\ \leq \quad & d(\tfrac{M}{2^b-1})^2 \cdot q + d\sum_{i=1}^{2^b-1}(\tfrac{2M}{2^b-1}i)^2\tfrac{1-q}{2^b-1} \\ \leq \quad & dM^2\left[\tfrac{q}{(2^b-1)^2} + \tfrac{2^{b+1}(2^{b+1}-1)}{3(2^b-1)^2}(1-q)\right] \end{aligned}$$

By the triangle inequality, for $u \in$ W,

$$\|w_{t+1} - u\|_2 \leq \|w_{t+1} - v_{t+1}\|_2 + \|v_{t+1} - u\|_2$$

Let $E_Q \;=\; dM^2\left[\frac{q}{(2^b-1)^2} + \frac{2^{b+1}(2^{b+1}-1)}{3(2^b-1)^2}(1-q)\right]$, we can derive the following inequality from the previous inequalities:

$$\|v_{t+1} - u\|^2 - \|w_{t+1} - u\|^2 \geq -E_Q \tag{27}$$

We next analyze the excess empirical loss under the assumption that the loss function is ρ-Lipschitz and convex over W. According to Lemma 14.7 in [47], for all $w \in$ W and gradient $\nabla \in {}_{\partial w}\frac{\partial l}{}$, the norm of the gradient is bounded: $\|\nabla\| \leq \rho$.

Let w^* be the optimal parameter in the parameter space W, defined as $w^* = \min_{w\in\mathrm{W}} \hat{\mathrm{L}}(w;\mathrm{S})$. Given that w_{t+1} is the projection of v_{t+1} and $w^* \in$ W, the following inequality is obtained:

$$\begin{aligned} & \|w_t - w_*\|_2 - \|w_{t+1} - w_*\|_2 \\ \geq \quad & \|w_t - w_*\|_2 - \|v_{t+1} - w_*\|_2 - E_Q \\ \geq \quad & 2\eta\langle w_t - w^*, \nabla\rangle - \eta^2\|\nabla\|^2 - \eta^2\|G_t\|^2 - E_Q \end{aligned}$$

Taking expectation of both sides, rearranging, and using the fact that $\mathrm{E}[\|\nabla\|^2] \leq \rho^2$ and $\mathrm{E}[\|G_t\|^2] = d\sigma^2$, we have:

$$\begin{aligned} \langle w_t - w^*, \nabla_t\rangle \leq \quad & \tfrac{1}{2\eta}\mathbb{E}[\|w_t - w^*\|^2 - \|w_{t+1} - w^*\|^2] \\ & + \tfrac{\eta}{2}\rho^2 + E_Q + \eta\sigma^2 d \end{aligned}$$

Given the convexity of the loss function l, we can further derive the bound of $\mathrm{E}\left[\hat{\mathrm{L}}(\bar{w}_T;\mathrm{S})\right] - \min_{w\in\mathrm{W}} \hat{\mathrm{L}}(w;\mathrm{S})$:

$$\begin{aligned} &\mathbb{E}\left[\hat{\mathcal{L}}(\bar{w}_T;\mathcal{S})\right] - \min_{w\in\mathcal{W}} \hat{\mathcal{L}}(w;\mathcal{S}) \\ \leq \quad & \frac{M^2}{2\eta T} + E_Q + \frac{\eta\rho^2}{2} + \eta\sigma^2 d \end{aligned}$$

This is achieved through the analytical methods used for SGD applied to Convex-Lipschitz-Bounded functions [47]. □

APPENDIX C PROOF OF LEMMA 1

Lemma 1. *(Distribution of the GSQ Outputs) For any input* $x \in [-C,C]$*, let b,* C_s*, and σ be the parameter in GSQ,* r^* *be the integer such that* $B(r^*) \leq x \leq B(r^*+1)$*, and two sets* $\mathcal{Q}_L = \{B(r)\}_{r=0}^{r^*}$ *and* $\mathcal{Q}_R = \{B(r)\}_{r=r^*+1}^{R-1}$*, the probability of* $GSQ(x) = B(r)$*:*

$$\begin{aligned} & \mathbb{P}(GSQ(x) = B(r)) \\ = \quad & \begin{cases} \frac{\phi_L(r)}{\xi_L(r^*)}\sum\limits_{r^+=r^*+1}^{R-1}\frac{\phi_R(r^+)}{\xi_R(r^*)}\frac{B(r^+)-x}{B(r^+)-B(r)} & r \leq r^* \\ \frac{\phi_R(r)}{\xi_R(r^*)}[\sum\limits_{r^-=0}^{r^*}\frac{\phi_L(r^-)}{\xi_L(r^*)}\frac{x-B(r^-)}{B(r)-B(r^-)}] & r > r^* \end{cases} \end{aligned} \tag{23}$$

where

$$\begin{cases} R = 2^b \\ \phi_L(r) = e^{-\frac{(r^*-r)^2}{2\sigma^2}} \\ \phi_R(r) = e^{-\frac{(r-(r^*+1))^2}{2\sigma^2}} \\ \xi_L(r^*) = \sum\limits_{y=0}^{r^*} e^{-\frac{(r^*-y)^2}{2\sigma^2}} \\ \xi_R(r^*) = \sum\limits_{y=r^*+1}^{R-1} e^{-\frac{(y-(r^*+1))^2}{2\sigma^2}} \end{cases} \tag{24}$$

Proof. Consider an input $x \in [-C,C]$, in the Gaussian sampling quantization with parameter β and R, the initial step involves extending the output range and establishing the quantization set $\mathcal{Q} = \{B(r)\}_{r=0}^{R-1}$. Then given the input $x \in [-C,C]$, the next step is to identify the optimal r^* such that $B(r^*) \leq x < B(r^*+1)$.

Based on the optimal r^*, the quantization set is then divided into two sets: $Q_L = \{B(r)\}_{r=0}^{r^*}$ for the left side and $\mathcal{Q}_R = \{B(r)\}_{r=r^*+1}^{R-1}$ for the right side.

Given the optimal r^*, the Gaussian sampling quantization (GSQ) selects a left-sided quantization level $B(r^-)$ from Q_L with probabilities drawn from a Discrete Gaussian distribution $N_{Q_L}(r^*,\sigma)$, and a right-sided quantization level $B(r^+)$ from Q_R with probabilities drawn from a Discrete Gaussian distribution $N_{Q_R}(r^* + 1,\sigma)$, respectively. Then the GSQ performs the stochastic quantization on the selected $B(r^-)$ and $B(r^+)$. As a result, the output of the GSQ can be either $B(r^-) \in Q_L$ or $B(r^+) \in Q_R$. Given the optimal r^*, we can derive two cases for the probability distribution of the GSQ output: Case 1 for output $B(r^-)$ ($r \le r^*$) and Case 2 for output $B(r^+)$ ($r > r^*$).

Case 1: $r \le r^*$

The probability $P(GSQ(x) = B(r))$ for $r \le r^*$ involves a weighted sum over all possible quantization levels in Q_R, where each term in the sum is the product of three factors:

- The Gaussian probability of selecting $B(r)$ as the quantization level from Q_L, normalized by the sum of Gaussian probabilities of all levels in Q_L.
- The Gaussian probability of selecting a level $B(r^+)$ from Q_R, normalized by the sum of Gaussian probabilities of all levels in Q_R.
- The stochastic quantization probability $\frac{B(r^+)-x}{B(r^+)-B(r)}$ that accounts for the distance of x from the quantization level $b(r)$ relative to $B(r^+)$.

Mathematically, we have

$$\begin{aligned}
&\mathbb{P}(GSQ(x) = B(r)) \\
=\ & \sum_{r^+\in\mathcal{Q}_R} \mathbb{P}(B(r^-) = B(r)) \cdot \mathbb{P}(B(r^+)) \tfrac{B(r^+)-x}{B(r^+)-B(r)} \\
=\ & \sum_{r^+\in\mathcal{Q}_R} \frac{e^{-(r^*-r)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}} \\
& \cdot \frac{e^{-(r^+-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}} \frac{B(r^+)-x}{B(r^+)-B(r)} \\
=\ & \frac{e^{-(r^*-r)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}} \\
& \sum_{r^+=r^*+1}^{R-1} \frac{e^{-(r^+-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}} \frac{B(r^+)-x}{B(r^+)-B(r)}
\end{aligned} \tag{28}$$

where σ^2 represents the variance of the Gaussian distribution in the Discrete Gaussian Sampling. Case 2: $r > r^*$

The probability $P(GSQ(x) = B(r))$ for $r > r^*$ involves a weighted sum over all possible quantization levels in Q_L, where each term in the sum is the product of three factors:

- The Gaussian probability of selecting $B(r)$ as the quantization level from Q_R, normalized by the sum of Gaussian probabilities of all levels in Q_R.
- The Gaussian probability of selecting a level $B(r^-)$ from Q_L, normalized by the sum of Gaussian probabilities of all levels in Q_L.
- The stochastic quantization probability $\frac{x-B(r^-)}{B(r)-B(r^-)}$ that accounts for the distance of x from the quantization level $b(r)$ relative to $B(r^-)$.

Mathematically, we have

$$\begin{aligned}
&\mathbb{P}(GSQ(x) = B(r)) \\
=\ & \sum_{r^-\in\mathcal{Q}_L} \mathbb{P}(B(r^+) = B(r)) \cdot \mathbb{P}(B(r^-)) \tfrac{x-B(r^-)}{B(r)-B(r^-)} \\
=\ & \sum_{r^-\in\mathcal{Q}_L} \frac{e^{-(r-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}} \\
& \cdot \frac{e^{-(r^*-r^-)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}} \frac{x-B(r^-)}{B(r)-B(r^-)} \\
=\ & \frac{e^{-(r-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}} \\
& \sum_{r^-=0}^{r^*} \frac{e^{-(r^*-r^-)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}} \frac{x-B(r^-)}{B(r)-B(r^-)}
\end{aligned} \tag{29}$$

Let $\phi_L(r) = e^{-\frac{(r^*-r)^2}{2\sigma^2}}, \phi_R(r) = e^{-\frac{(r-(r^*+1))^2}{2\sigma^2}}, \xi_L(r^*) = \sum_{y=0}^{r^*} e^{-\frac{(r^*-y)^2}{2\sigma^2}}, \xi_R(r^*) = \sum_{y=r^*+1}^{R-1} e^{-\frac{(y-(r^*+1))^2}{2\sigma^2}}$, we have the desired probability as presented in (23). □

Appendix D Proof of Theorem 3

Theorem 3. *(Privacy of the Gaussian Sampling Quantization (GSQ)) For any input $x,x' \in [-C,C]$, let b, β, and σ be the parameter in GSQ, we have*

$$\begin{aligned}
& D_\infty(\mathbb{P}_{GSQ(x)} \| \mathbb{P}_{GSQ(x')}) \\
\le\ & \log \tfrac{(2^b-\beta)(2^b-1)}{\beta^2} + \tfrac{(2^b-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2}
\end{aligned} \tag{22}$$

And GSQ satisfies $(\log \frac{(2^b-\beta)(2^b-1)}{\beta^2} + \frac{(2^b-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2}, 0)$-*DP.*

Proof. In Lemma 1, we show the distribution of the GSQ output. We next present the differential privacy of GSQ using the Renyi divergence. To obtain an upper bound on the R´ enyi´ divergence between the GSQ probabilities of two input $x,x' \in [-C,C]$, we use the following Theorem from [49].

(Theorem 13 in [49]) For any order $\alpha \in [0,\infty]$ Renyi´ divergence is jointly quasi-convex in its arguments. That is, for any two pairs of probability distributions (P_0,Q_0) and (P_1,Q_1), and any $\lambda \in (0,1)$

$$\begin{aligned}
& D_\alpha((1-\lambda)P_0 + \lambda P_1 \| (1-\lambda)Q_0 + \lambda Q_1) \\
\le\ & \max\{D_\alpha(P_0\|Q_0), D_\alpha(P_1\|Q_1)\}
\end{aligned} \tag{30}$$

The quasi-convexity of the Renyi divergence is demonstrated´ in (30). Based on this, for any input $x,x' \in [-C,C]$, we have

$$D_\infty(P(GSQ(x)\|P(GSQ(x')))$$

$$\leq \quad \max\{D_\infty(\mathrm{P}(GSQ(-C)\|\mathrm{P}(GSQ(C))), \quad (31)$$
$$D_\infty(\mathrm{P}(GSQ(C)\|\mathrm{P}(GSQ(-C)))\}$$

The computation of the Renyi divergence relies on the´

$x = C,\ r^* = \lfloor \frac{C + \frac{R-1}{R-1-2\beta} C}{2C/(R-1-2\beta)} \rfloor = \lfloor \frac{2R-2-2\beta}{2} \rfloor = R - 1 - \beta$

$-C,\ r^* = \lfloor \frac{-C + \frac{R-1}{R-1-2\beta} C}{2C/(R-1-2\beta)} \rfloor = \lfloor \frac{2\beta}{2} \rfloor = \beta$

distribution of the GSQ output with input points $-C,C$. When

$\underline{R-1}$

;

when $x =$. The next step of the proof is to compute the upper bound of two Renyi divergences:´ $D_\infty(\mathrm{P}(GSQ(-C)\|\mathrm{P}(GSQ(C)))$ and $D_\infty(\mathrm{P}(GSQ(C)\|\mathrm{P}(GSQ(-C)))$.

1) Upper bound of $D_\infty(\mathrm{P}(GSQ(-C)\|\mathrm{P}(GSQ(C)))$:

$$\begin{aligned} & D_\infty(\mathbb{P}(GSQ(-C)\|\mathbb{P}(GSQ(C))) \\ = \ & \sup_{r=0,1,\cdots,R-1} \log(\tfrac{\mathbb{P}(GSQ(-C)=B(r))}{\mathbb{P}(GSQ(C)=B(r))}) \\ \leq \ & \log \tfrac{\max_{r=0,1,\cdots,R-1} \mathbb{P}(GSQ(-C)=B(r))}{\min_{r=0,1,\cdots,R-1} \mathbb{P}(GSQ(C)=B(r))} \\ \leq \ & \log(\tfrac{\mathbb{P}(GSQ(C))=B(\beta)}{\mathbb{P}(GSQ(C))=B(0)}) \end{aligned} \quad (32)$$

When $x = C,\ r^* = \lfloor \frac{C+\frac{R-1}{R-1-2\beta}C}{2C/(R-1-2\beta)} \rfloor = \lfloor \frac{2R-2-2\beta}{2} \rfloor =$ $R - 1 - \beta$, we have

$$\log(\tfrac{\mathbb{P}(GSQ(-C))=B(\beta)}{\mathbb{P}(GSQ(C))=B(0)})$$

$$= \log \frac{\frac{1}{\sum_{y\in\mathcal{Q}_L} e^{-(\beta-y)^2/(2\sigma^2)}}}{\frac{e^{-(r^*)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}} \cdot \sum_{r^+=r^*+1}^{R-1} \frac{e^{-(r^+-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}} \frac{B(r^+)-x}{B(r^+)-B(0)}}$$

$$= \log \frac{\frac{1}{\sum_{y\in\mathcal{Q}_L} e^{-(\beta-y)^2/(2\sigma^2)}}}{\frac{e^{-(R-1-\beta)^2/2(\sigma)^2}}{\sum_{y=0}^{R-1-\beta} e^{-(R-1-\beta-y))^2/2(\sigma)^2}} \cdot \sum_{r^+=R-\beta}^{R-1} \frac{e^{-(r^+-(R-\beta))^2/2(\sigma)^2}}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta)))^2/2(\sigma)^2}} \frac{B(r^+)-C}{B(r^+)-B(0)}}$$

$$= \log \frac{\frac{1}{\sum_{y\in\mathcal{Q}_L} e^{-(\beta-y)^2/(2\sigma^2)}}}{\frac{e^{-(R-1-\beta)^2/2(\sigma)^2}}{\sum_{y=0}^{R-1-\beta} e^{-(R-1-\beta-y))^2/2(\sigma)^2}} \cdot \sum_{r^+=R-\beta}^{R-1} \frac{e^{-(r^+-(R-\beta))^2/2(\sigma)^2}}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta)))^2/2(\sigma)^2}} \frac{r^+-R+1+\beta}{r^+}} \quad (33)$$

When $r^+ \in [R-\beta, R-1]$, $\frac{r^+-R+1+\beta}{r^+} = 1 - \frac{R-1-\beta}{r^+} \geq 1 - \frac{R-1-\beta}{R-1} = \frac{\beta}{R-1}$, and $e^{-(r^+-(R-\beta))^2/2(\sigma)^2} \geq e^{-\frac{(\beta-1)^2}{2\sigma^2}}$, we further have

$$\begin{aligned} & \sum_{y=R-\beta}^{R-1} \frac{e^{-(r^+-(R-\beta))^2/2(\sigma)^2}}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta)))^2/2(\sigma)^2}} \frac{r^+-R+1+\beta}{r^+} \\ \geq \ & \frac{\beta}{(R-1)} e^{-\frac{(\beta-1)^2}{2\sigma^2}} \end{aligned} \quad (34)$$

This implies

$$\begin{aligned} & \log(\tfrac{\mathbb{P}(GSQ(-C))=B(\beta)}{\mathbb{P}(GSQ(C))=B(0)}) \\ \leq \ & \log \frac{\frac{1}{\sum_{y=0}^{\beta} e^{-(\beta-y)^2/(2\sigma^2)}}}{\frac{e^{-(R-1-\beta)^2/2(\sigma)^2}}{\sum_{y=0}^{R-1-\beta} e^{-(R-1-\beta-y))^2/2(\sigma)^2}} \frac{\beta}{(R-1)} e^{-\frac{(\beta-1)^2}{2\sigma^2}}} \\ = \ & \log \frac{\sum_{y=0}^{R-1-\beta} e^{-(R-1-\beta-y))^2/2(\sigma)^2} (R-1)}{\beta \cdot e^{-\frac{(R-1-\beta)^2}{2\sigma^2}} e^{-\frac{(\beta-1)^2}{2\sigma^2}} \sum_{y=0}^{\beta} e^{-(\beta-y)^2/(2\sigma^2)}} \\ \leq \ & \log \frac{(R-\beta)(R-1)}{\beta \cdot e^{-\frac{(R-1-\beta)^2}{2\sigma^2}} e^{-\frac{(\beta-1)^2}{2\sigma^2}} \cdot (\beta+1) \cdot e^{\frac{-\beta^2}{2\sigma^2}}} \\ = \ & \log(\frac{(R-\beta)(R-1)}{\beta(\beta+1)} e^{\frac{(R-1-\beta)^2+(\beta-1)^2}{2\sigma^2}}) \\ = \ & \log \frac{(R-\beta)(R-1)}{\beta(\beta+1)} + \frac{(R-1-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2} \end{aligned}$$

2) Upper bound of $D_\infty(\mathrm{P}(GSQ(C)\|\mathrm{P}(GSQ(-C)))$:

$$\begin{aligned} & D_\infty(\mathbb{P}(GSQ(C)\|\mathbb{P}(GSQ(-C))) \\ = \ & \sup_{r=0,1,\cdots,R-1} \log(\tfrac{\mathbb{P}(GSQ(C)=B(r))}{\mathbb{P}(GSQ(-C)=B(r))}) \\ \leq \ & \log \tfrac{\max_{r=0,1,\cdots,R-1} \mathbb{P}(GSQ(C)=B(r))}{\min_{r=0,1,\cdots,R-1} \mathbb{P}(GSQ(-C)=B(r))} \\ \leq \ & \log(\tfrac{\mathbb{P}(GSQ(C)=B(R-\beta))}{\mathbb{P}(GSQ(-C))=B(R-1)}) \end{aligned} \quad (35)$$

When $x = -C,\ r^* = \lfloor \frac{-C+\frac{R-1}{R-1-2\beta}C}{2C/(R-1-2\beta)} \rfloor = \lfloor \frac{2\beta}{2} \rfloor = \beta$, we have

$$\log(\tfrac{\mathbb{P}(GSQ(C)=B(R-\beta))}{\mathbb{P}(GSQ(-C))=B(R-1)})$$

$$\leq \log \frac{\frac{1}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta))^2/(2\sigma^2)}}}{\frac{e^{-(R-1-(r^*+1))^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_R} e^{-(y-(r^*+1)))^2/2(\sigma)^2}} \cdot \sum_{r^-=0}^{r^*} \frac{e^{-(r^*-r^-)^2/2(\sigma)^2}}{\sum_{y\in\mathcal{Q}_L} e^{-(r^*-y))^2/2(\sigma)^2}} \frac{x-B(r^-)}{B(R-1)-B(r^-)}}$$

$$= \log \frac{\frac{1}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta))^2/(2\sigma^2)}}}{\frac{e^{-(R-1-(\beta+1))^2/2(\sigma)^2}}{\sum_{y=\beta+1}^{R-1} e^{-(y-(\beta+1)))^2/2(\sigma)^2}} \cdot \sum_{r^-=0}^{\beta} \frac{e^{-(\beta-r^-)^2/2(\sigma)^2}}{\sum_{y=0}^{\beta} e^{-(\beta-y))^2/2(\sigma)^2}} \frac{-C-B(r^-)}{B(R-1)-B(r^-)}}$$

$$= \log \frac{\frac{1}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta))^2/(2\sigma^2)}}}{\frac{e^{-(R-1-(\beta+1))^2/2(\sigma)^2}}{\sum_{y=\beta+1}^{R-1} e^{-(y-(\beta+1)))^2/2(\sigma)^2}} \cdot \sum_{r^-=0}^{\beta} \frac{e^{-(\beta-r^-)^2/2(\sigma)^2}}{\sum_{y=0}^{\beta} e^{-(\beta-y))^2/2(\sigma)^2}} \frac{\beta-r^-}{R-1-r^-}}$$

(36)

When $r^- \in [0,\beta]$, $\frac{\beta-r^-}{R-1-r^-} = 1 - \frac{R-1-\beta}{R-1-r^-} \geq 1 - \frac{R-1-\beta}{R-1} = \frac{\beta}{R-1}$, and$e^{-(\beta-y))^2/2(\sigma)^2} \geq e^{-\frac{\beta^2}{2\sigma^2}}$, we further have

$$\sum_{r^-=0}^{\beta} \frac{e^{-(\beta-r^-)^2/2(\sigma)^2}}{\sum_{y=0}^{\beta} e^{-(\beta-y))^2/2(\sigma)^2}} \frac{\beta-r^-}{R-1-r^-} \geq \frac{\beta}{(R-1)} e^{-\frac{\beta^2}{2\sigma^2}}$$

This implies

$$\begin{aligned}
& \log\left(\frac{\mathbb{P}(GSQ(C))=B(R-\beta)}{\mathbb{P}(GSQ(-C))=B(R-1)}\right) \\
\leq \quad & \log \frac{\frac{1}{\sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta))^2/(2\sigma^2)}}}{\frac{e^{-(R-1-(\beta+1))^2/2(\sigma)^2}}{\sum_{y=\beta+1}^{R-1} e^{-(y-(\beta+1)))^2/2(\sigma)^2}} \frac{\beta}{(R-1)} e^{-\frac{\beta^2}{2\sigma^2}}} \\
= \quad & \log \frac{\sum_{y=\beta+1}^{R-1} e^{-(y-(\beta+1)))^2/2(\sigma)^2}(R-1)}{\beta \cdot e^{-\frac{(R-1-(\beta+1))^2}{2\sigma^2}} e^{-\frac{\beta^2}{2\sigma^2}} \sum_{y=R-\beta}^{R-1} e^{-(y-(R-\beta))^2/(2\sigma^2)}} \\
\leq \quad & \log \frac{(R-1-\beta)(R-1)}{\beta \cdot e^{-\frac{(R-\beta)^2}{2\sigma^2}} e^{-\frac{\beta^2}{2\sigma^2}} \cdot \beta \cdot e^{-\frac{(\beta-1)^2}{2\sigma^2}}}
\end{aligned}$$

$$\begin{aligned}
& \mathbb{E}_{x\sim\ GSQ(x)}[GSQ(x)] \\
= \quad & \sum_{r^-\in\mathcal{Q}_L} \mathbb{P}(B(r^-)) \sum_{r^+\in\mathcal{Q}_R} \mathbb{P}(B(r^+)) \tfrac{B(r^+)-x}{B(r^+)-B(r^-)} \cdot B(r^-) \\
& + \sum_{r^+\in\mathcal{Q}_R} \mathbb{P}(B(r^+)) \sum_{r^-\in\mathcal{Q}_L} \mathbb{P}(B(r^-)) \tfrac{x-B(r^-)}{B(r^+)-B(r^-)} \cdot B(r^+) \\
= \quad & \log\left(\tfrac{(R-1-\beta)(R-1)}{\beta^2} e^{\frac{(R-\beta)^2+\beta^2+(\beta-1)^2}{2\sigma^2}}\right) \\
= \quad & \log \tfrac{(R-1-\beta)(R-1)}{\beta^2} + \tfrac{(R-\beta)^2+\beta^2+(\beta-1)^2}{2\sigma^2}
\end{aligned} \tag{37}$$

The final step of the proof is to find the maximum Renyi divergence between´ $D_\infty(\mathrm{P}(GSQ(-C)\|\mathrm{P}(GSQ(C)))$ and $D_\infty(\mathrm{P}(GSQ(C)\|\mathrm{P}(GSQ(-C)))$:

$$\begin{aligned}
& D_\infty(\mathrm{P}(GSQ(x)\|\mathrm{P}(GSQ(x')))) \\
\leq \quad & \max\{D_\infty(\mathrm{P}(GSQ(-C)\|\mathrm{P}(GSQ(C))), \\
& D_\infty(\mathrm{P}(GSQ(C)\|\mathrm{P}(GSQ(-C)))\} \\
\leq \quad & \max\{\log \tfrac{(R-\beta)(R-1)}{\beta(\beta+1)} + \tfrac{(R-1-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2} \\
& \log \tfrac{(R-1-\beta)(R-1)}{\beta^2} + \tfrac{(R-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2}\} \\
\leq \quad & \log \tfrac{(R-\beta)(R-1)}{\beta^2} + \tfrac{(R-\beta)^2+(\beta-1)^2+\beta^2}{2\sigma^2}
\end{aligned} \tag{38}$$

,

□

APPENDIX E PROOF OF LEMMA 2

Lemma 2. *(Unbiasedness and Variance of the Gaussian Sampling Quantization (GSQ)) For any input $x \in [-C,C]$, let b, β, and σ be the parameter in GSQ, and $t > 0$ such that $x \leq e^{tx}$ for $x > 0$, we have*

$$\begin{cases} \mathbb{E}_{x\sim\ GSQ(x)}[GSQ(x)] = x \\ \mathbb{E}_{x\sim GSQ}[\|GSQ(x) - x\|^2] \leq pC^2 \end{cases} \tag{25}$$

for some positive real constant p.

Proof. Let Q_L and Q_R denote the left and right subsets of the quantization set determined by GSQ for an input x, respectively. Let $\mathrm{P}(B(r^-))$ represent the probability of selecting a quantization level $B(r^-)$ from Q_L through, and similarly, $\mathrm{P}(B(r^+))$ for selecting $B(r^-)$ from Q_R. The expectation of the GSQ's output is characterized as follows:

$$\begin{aligned}
& \sum_{r^+\in\mathcal{Q}_R} [\mathbb{P}(B(r^+)) \cdot (B(r^+) - B(r^*))] \\
= \quad & \sum_{r^+=r^*+1}^{R-1} \frac{e^{-\frac{(r^+-r^*-1)^2}{2\sigma^2}}}{\sum\limits_{y=r^*+1}^{R-1} e^{-\frac{(y-r^*-1)^2}{2\sigma^2}}} \cdot (B(r^+) - B(r^*)) \\
= \quad & \sum_{r^+=r^*+1}^{R-1} \frac{e^{-\frac{(r^+-r^*-1)^2}{2\sigma^2}}}{\sum\limits_{y=r^*+1}^{R-1} e^{-\frac{(y-r^*-1)^2}{2\sigma^2}}} \frac{2C}{R-1-2\beta} \cdot (r^+ - r^*)
\end{aligned}$$

$$\begin{aligned}
= \quad & \sum_{r^-\in\mathcal{Q}_L} \sum_{r^+\in\mathcal{Q}_R} \mathbb{P}(B(r^-))\mathbb{P}(B(r^+)) \tfrac{B(r^+)-x}{B(r^+)-B(r^-)} \cdot B(r^-) \\
& + \sum_{r^+\in\mathcal{Q}_R} \sum_{r^-\in\mathcal{Q}_L} \mathbb{P}(B(r^+))\mathbb{P}(B(r^-)) \tfrac{x-B(r^-)}{B(r^+)-B(r^-)} \cdot B(r^+) \\
= \quad & \sum_{r^-\in\mathcal{Q}_L} \sum_{r^+\in\mathcal{Q}_R} \mathbb{P}(B(r^-))\mathbb{P}(B(r^+)) (\tfrac{B(r^+)-x}{B(r^+)-B(r^-)} \cdot B(r^-) \\
& + \tfrac{x-B(r^-)}{B(r^+)-B(r^-)} \cdot B(r^+))
\end{aligned}$$

$= \underset{r_-\in Q_L}{\mathrm{P}} \ \underset{r_+\in Q_R}{\mathrm{P}}\ \mathrm{P}(B(r^-))\mathrm{P}(B(r^+)) \cdot x$

$= \quad x$

(39) Next, we show the variance of the GSQ output as follows:

$$\mathbb{E}_{x\sim\ GSQ(x)}[\|GSQ(x)-x\|^2]$$

$$\begin{aligned}
= \quad & \sum_{r^-\in\mathcal{Q}_L}\mathbb{P}(B(r^-))\sum_{r^+\in\mathcal{Q}_R}\mathbb{P}(B(r^+))\tfrac{B(r^+)-x}{B(r^+)-B(r^-)}\cdot \\
& (x-B(r^-))^2 \\
& +\sum_{r^+\in\mathcal{Q}_R}\mathbb{P}(B(r^+))\sum_{r^-\in\mathcal{Q}_L}\mathbb{P}(B(r^-))\tfrac{x-B(r^-)}{B(r^+)-B(r^-)}\cdot \\
& (B(r^+)-x)^2
\end{aligned}$$

$$\begin{aligned}
= \quad & \sum_{r^-\in\mathcal{Q}_L}\sum_{r^+\in\mathcal{Q}_R}\mathbb{P}(B(r^-))\mathbb{P}(B(r^+))[\tfrac{B(r^+)-x}{B(r^+)-B(r^-)}\cdot \\
& (x-B(r^-))^2+\tfrac{x-B(r^-)}{B(r^+)-B(r^-)}\cdot(B(r^+)-x)^2]
\end{aligned}$$

$$\begin{aligned}
= \quad & \sum_{r^-\in\mathcal{Q}_L}\sum_{r^+\in\mathcal{Q}_R}\mathbb{P}(B(r^-))\mathbb{P}(B(r^+))\cdot(B(r^+)-x)\cdot \\
& (x-B(r^-)
\end{aligned}$$

$$\le \sum_{r^-\in Q_L}\sum_{r^+\in Q_R} \mathrm{P}(B(r^-))\mathrm{P}(B(r^+))\cdot(B(r^+)-B(r^*))\cdot (B(r^*+1)-B(r^-))$$

$$= \sum_{r^-\in Q_L}\{\mathrm{P}(B(r^-))\cdot(B(r^*+1)-B(r^-)) \sum_{r^+\in Q_R}[\mathrm{P}(B(r^+))\cdot(B(r^+)-B(r^*))]\} \tag{40}$$

The upper bound of $\sum_{r^+\in Q_R}[\mathrm{P}(B(r^+))\cdot(B(r^+)-B(r^*))]$ is given as follows:

$$= \frac{2C}{R-1-2\beta}\sum_{r^+=r^*+1}^{R-1}\frac{e^{-\frac{(r^+-r^*-1)^2}{2\sigma^2}}}{\sum_{y=r^*+1}^{R-1}e^{-\frac{(y-r^*-1)^2}{2\sigma^2}}}(r^+-r^*) \tag{41}$$

Let $z = r^+ - r^* - 1$:

$$\begin{aligned}
= \quad & \frac{2C}{R-1-2\beta}\sum_{z=0}^{R-2-r^*}\frac{e^{-\frac{z^2}{2\sigma^2}}}{\sum_{y=0}^{R-2-r^*}e^{-\frac{y^2}{2\sigma^2}}}(1+z) \\
\le \quad & \frac{2C}{R-1-2\beta}\sum_{z=0}^{R-2-r^*}\frac{e^{-\frac{z^2}{2\sigma^2}}}{\sum_{y=0}^{R-2-r^*}e^{-\frac{y^2}{2\sigma^2}}}(r^+-\beta) \\
= \quad & \frac{2C}{R-1-2\beta}(r^+-\beta)
\end{aligned}$$

This implies that

$$\mathbb{E}_{x\sim\ GSQ(x)}[\|GSQ(x)-x\|^2]$$

$$\le \quad \frac{2C}{R-1-2\beta}(r^+-\beta)\sum_{r^-\in\mathcal{Q}_L}\mathbb{P}(B(r^-))\cdot(B(r^*+1)-B(r^-) \tag{42}$$

Similarly, the upper bound of $\sum_{r^-\in Q_L}[\mathrm{P}(B(r^-))\cdot(B(r^*+1)-B(r^-))]$ is given as follows

$$\sum_{r^-\in\mathcal{Q}_L}[\mathbb{P}(B(r^-))\cdot(B(r^*+1)-B(r^-))]$$

$$\begin{aligned}
= \quad & \sum_{r^-=0}^{r^*}\frac{e^{-\frac{(r^*-r^-)^2}{2\sigma^2}}}{\sum\limits_{y=0}^{r^*}e^{-\frac{(r^*-y)^2}{2\sigma^2}}}\cdot(B(r^*+1)-B(r^-)) \\
= \quad & \sum_{r^-=0}^{r^*}\frac{e^{-\frac{(r^*-r^-)^2}{2\sigma^2}}}{\sum\limits_{y=0}^{r^*}e^{-\frac{(r^*-y)^2}{2\sigma^2}}}\frac{2C}{R-1-2\beta}\cdot(r^*+1-r^-) \\
= \quad & \frac{2C}{R-1-2\beta}\sum_{r^-=0}^{r^*}\frac{e^{-\frac{(r^*-r^-)^2}{2\sigma^2}}}{\sum\limits_{y=0}^{r^*}e^{-\frac{(r^*-y)^2}{2\sigma^2}}}(r^*+1-r^-)
\end{aligned} \tag{43}$$

Let $z = r^* - r^-$:

$$\begin{aligned}
= \quad & \frac{2C}{R-1-2\beta}\sum_{z=-r^*}^{0}\frac{e^{-\frac{z^2}{2\sigma^2}}}{\sum_{y=0}^{-r^*}e^{-\frac{y^2}{2\sigma^2}}}(1+z) \\
\le \quad & \frac{2C}{R-1-2\beta}\sum_{z=-r^*}^{0}\frac{e^{-\frac{z^2}{2\sigma^2}}}{\sum_{y=-r^*}^{0}e^{-\frac{y^2}{2\sigma^2}}}(R-\beta-r^-) \\
= \quad & \frac{2C}{R-1-2\beta}(R-\beta-r^-)
\end{aligned}$$

By combining two bounds, we have

$$\begin{aligned}
& \mathbb{E}_{x\sim GSQ}[\|GSQ(x)-x\|^2] \\
\le \quad & (\tfrac{2C}{R-1-2\beta})^2(r^+-\beta)(R-\beta-r^-)
\end{aligned} \tag{44}$$

Let $r^+ = R - 1$ and $r- = 0$

$$\le \quad \tfrac{4(R-1-\beta)(R-\beta)}{(R-1-2\beta)^2}C^2$$

Let $p = \frac{4(R-1-\beta)(R-\beta)}{(R-1-2\beta)^2}$, we have the result. □

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Saba Amiri, Adam Belloum, Sander Klous, and Leon Gommans. Compressive differentially private federated learning through universal vector quantization. In *AAAI Workshop on Privacy-Preserving Artificial Intelligence*, pages 2–9, 2021.

[3] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.

[4] Yu Bai, Yu-Xiang Wang, and Edo Liberty. Proxquant: Quantized neural networks via proximal operators. *arXiv preprint arXiv:1810.00861*, 2018.

[5] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in neural information processing systems*, 31, 2018.

[6] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. *Advances in neural information processing systems*, 32, 2019.

[7] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pages 464–473. IEEE, 2014.

[8] Zhongteng Cai, Xueru Zhang, and Mohammad Mahdi Khalili. Privacyaware randomized quantization via linear programming. *arXiv preprint arXiv:2406.02599*, 2024.

[9] Sebastian Caldas, Sai Meher Karthik Duddu, Peter Wu, Tian Li, Jakub Konecnˇ y, H Brendan McMahan, Virginia Smith, and Ameet` Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[10] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," arXiv preprint arXiv:1603.04467, 2016.

[11] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in Proc. ACM SIGSAC Conf. Computer and Communications Security, pp. 308–318, 2016.

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

[13] A. Dwork, "Differential privacy," in Proc. Int. Colloquium Automata, Languages and Programming, pp. 1–12, 2006.

[14] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," Found. Trends Theor. Comput. Sci., vol. 9, no. 3–4, pp. 211–407, 2014.

[15] T. Erlingsson, U. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in Proc. ACM SIGSAC Conf. Computer and Communications Security, pp. 1054–1067, 2014.

[16] Mehta, D. B. (2023). Privacy-Preserving Machine Learning Architecture for Cross-Channel Advertising Optimization in a Large-Scale Social Media Platform. International Journal of Communication Networks and Information Security (IJCNIS), 15(8), 23–32. https://www.ijcnis.org/index.php/ijcnis/article/view/8396

[17] Z. Ghazi, J. Sun, R. Kulkarni, and M. Gagne, "Practical and accurate differential privacy for federated learning," in Proc. NeurIPS, 2021.

[18] I. Sato, K. Nishimura, and K. Takenouchi, "Randomized quantization for communication-efficient, privacy-preserving federated learning," in Proc. ICML, PMLR, 2022.

[19] J. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends® in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.

[20] H. B. McMahan et al., "Communication-efficient learning of deep networks from decentralized data," in Proc. Artificial Intelligence and Statistics (AISTATS), PMLR, pp. 1273–1282, 2017.

[21] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in Proc. MLSys, pp. 429–450, 2020.

[22] Y. Youn, Z. Hu, J. Ziani, and J. Abernethy, "Randomized quantization is all you need for differential privacy in federated learning," arXiv preprint arXiv:2306.11913, 2023.

[23] Y. Cai, X. Xu, C. Zhang, and L. Zhao, "Privacy-aware randomized quantization via linear programming," in Proc. Conf. Uncertainty in Artificial Intelligence, PMLR 244:499–516, 2024.

[24]

[25] M. Colombo et al., "A quantization-based technique for privacy preserving distributed learning," arXiv preprint arXiv:2406.19418, 2024.

[26] M. Colombo, R. Asal, E. Damiani, L. M. AlQassem, A. Almemari, and Y. Alhammadi, "A quantization-based technique for privacy preserving distributed learning," in Proc. ACM Conf. Computer and Communications Security, 2024.

[27] Z. Li, S. Zheng, F. Li, L. Pan, X. Luo, and L. Chen, "Federated learning systems: Vision, case studies and future directions," IEEE Transactions on Service Computing, vol. 16, no. 3, pp. 1568–1585, 2023.

[28]

[29] X. He, Z. Zhang, Z. Wang, J. Jing, "A communication-efficient and differential privacy-preserving FedAvg algorithm," IEEE Access, vol. 9, pp. 70121–70134, 2021.

[30] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In International Conference on Machine Learning, pages 38087–38099. PMLR, 2023.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[32] Sijie Xiong, Anand D Sarwate, and Narayan B Mandayam. Randomized requantization with local differential privacy. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2189–2193. IEEE, 2016.

[33] Penghang Yin, Shuai Zhang, Jiancheng Lyu, Stanley Osher, Yingyong Qi, and Jack Xin. Binaryrelax: A relaxation approach for training deep neural networks with quantized weights. SIAM Journal on Imaging Sciences, 11(4):2205–2223, 2018.

[34] Yeojoon Youn, Zihao Hu, Juba Ziani, and Jacob Abernethy. Randomized quantization is all you need for differential privacy in federated learning. arXiv preprint arXiv:2306.11913, 2023.

[35] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In International Conference on Learning Representations, 2021.

[36] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In International Conference on Machine Learning, pages 12208–12218. PMLR, 2021.

[37] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In 2019 IEEE Symposium on Security and Privacy (SP), pages 332–349, 2019.

[38] Jinjie Zhang, Yixuan Zhou, and Rayan Saab. Post-training quantization for neural networks with provable guarantees. SIAM Journal on Mathematics of Data Science, 5(2):373–399, 2023.

[39] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional mechanism: Regression analysis under differential privacy. Proceedings of the VLDB Endowment, 5(11), 2012.

[40] Ruochi Zhang and Parv Venkitasubramaniam. Optimal local differentially private quantization. IEEE Transactions on Signal Processing, 68:6509–6520, 2020.

[41] Huixuan Zong, Qing Wang, Xiaofeng Liu, Yinchuan Li, and Yunfeng Shao. Communication reducing quantization for federated learning with local differential privacy mechanism. In 2021 IEEE/CIC International Conference on Communications in China (ICCC), pages 75–80. IEEE, 2021.

[42] J. Xu, K. Wang, A. Wang, Y. Liu, and H. Liu, "Efficient federated learning via privacy-preserving quantization," Information Sciences, vol. 570, pp. 260–275, 2021.

[43] X. Luo, H. Wu, S. Zheng, "Privacy-preserving deep learning via random quantization," Computers & Security, vol. 112, p. 102502, 2022.

[44] E. Bagdasaryan, O. Poursaeed, and V. Shmatikov, "Differential privacy has disparate impact on model accuracy," in Proc. NeurIPS, pp. 15453–15461, 2019.

[45] R. Kulkarni et al., "Quantized differentially private federated learning: Analysis and experiments," in Proc. ICASSP, pp. 3326–3330, 2021.

[46] S. Xiong, A. D. Sarwate, N. B. Mandayam, "Randomized requantization with local differential privacy," in Proc. IEEE Int. Conf. Acoustics, Speech Signal Processing (ICASSP), pp. 2189–2193, 2016.

[47] Y. Li, S. Wang, J. Liu, and T. Chen, "Differentially private stochastic quantization for communication-efficient federated learning," in Proc. AAAI, pp. 7593–7600, 2022.

[48] A. Choudhury, S. Soltanpour, "Privacy-preserving federated learning via quantization-based defense," IEEE Transactions on Information Forensics and Security, vol. 17, pp. 1755–1766, 2022.

[49] D. Yu, H. Zhang, W. Chen, T.-Y. Liu, "Gradient embedding perturbation for private learning," in Proc. ICLR, 2021.

[50] D. Yu, H. Zhang, W. Chen, J. Yin, T.-Y. Liu, "Large scale private learning via low-rank reparametrization," in Proc. ICML, pp. 12208–12218, 2021.

[51] J. Zhang, Y. Zhou, R. Saab, "Post-training quantization for neural networks with provable guarantees," SIAM Journal on Mathematics of Data Science, vol. 5, no. 2, pp. 373–399, 2023.

[52] T. Gandikota, S. Bhadane, Z. Hu, P. Li, and J. C. Duchi, "Sparse quantized federated learning with applications to model compression and privacy," in Proc. NeurIPS, 2021.

[53] J. Han, M. Zhu, "Communication-efficient federated learning via privacy-preserving quantized aggregation," IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 5, pp. 1926–1937, 2021.

[54] S. Zhao, W. Wang, X. Li, "Federated learning with local differential privacy: A quantization approach," in Proc. ICML, pp. 4820–4830, 2022.

[55] X. Xu, C. Zhang, Y. Cai, and L. Zhao, “Optimal quantization for differential privacy in distributed machine learning,” Journal of Privacy and Confidentiality, vol. 12, no. 1, 2021.

[56] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, “Smoothquant: Accurate and efficient post-training quantization for large language models,” in Proc. ICML, pp. 38087–38099, 2023.

[57] H. Xiao, K. Rasul, R. Vollgraf, “Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms,” arXiv preprint arXiv:1708.07747, 2017.

[58] C. Song, T. Xiao, Y. Ding, "Quantization as privacy mechanisms for decentralized edge learning," ACM Transactions on Privacy and Security, vol. 25, no. 3, pp. 1–22, 2022.

[59] Y. Youn et al., “Randomized quantization mechanisms for federated learning privacy,” in Proc. OpenReview, 2023.

[60] Raymond E Wright. Logistic regression. 1995.