Practical Locally Private Federated Learning with Communication Efficiency

Anonymous authors
Paper under double-blind review

Abstract

Federated learning (FL) is a technique that trains machine learning models from decentralized data sources. We study FL under local differential privacy constraints, which provides strong protection against sensitive data disclosures via obfuscating the data before leaving the client. We identify two major concerns in designing practical privacy-preserving FL algorithms: communication efficiency and high-dimensional compatibility. We then develop a gradient-based learning algorithm called *sqSGD* (selective quantized stochastic gradient descent) that addresses both concerns. The proposed algorithm is based on a novel privacy-preserving quantization scheme that uses a constant number of bits per dimension per client. Then we improve the base algorithm in two ways: first, we apply a gradient subsampling strategy that offers simultaneously better training performance and smaller communication costs under a fixed privacy budget. Secondly, we utilize randomized rotation as a preprocessing step to reduce quantization error. We also initialize a discussion about the role of quantization and perturbation in FL algorithm design with privacy and communication constraints. Finally, the practicality of the proposed framework is demonstrated on benchmark datasets. Experiment results show that sqSGD successfully learns large models like LeNet and ResNet with local privacy constraints. In addition, with fixed privacy and communication level, the performance of sqSGD significantly dominates that of baseline algorithms.

1 Introduction

1.1 Background

Federated learning (FL) Kairouz et al. (2019); Konečný et al. (2016) is a rapidly evolving application of distributed optimization to large-scale learning or estimation scenarios where multiple entities. called *clients*, collaborate in solving a machine learning problem, under the coordination of a *central server*. Each client's raw data is stored locally and not exchanged or transferred. To achieve the learning objective, the server collects minimal information from the clients for immediate aggregation. FL is particularly suitable for mobile and edge device applications since the (sensitive) individual data never directly leave the device and has seen deployments in industries (**?**Hard et al., 2019; Leroy et al., 2019). While FL offers significant practical privacy improvements over centralizing all the training data, it lacks a formal privacy guarantee. As discussed in Melis et al. (2018), even if only model updates (i.e. gradient updates) are transmitted, it is easy to compromise the privacy of individual clients.

Differential privacy (DP) (Dwork et al., 2014) is the state-of-the-art approach to address information disclosure. Differentially private algorithms fuse participation of any individual via injecting algorithm-specific random noise. In FL setting, DP is suitable for protecting against *external adversaries*, i.e. a malicious analyst that tries to infer individual data via observing final or intermediate model results. However, DP paradigms typically assume a *trusted curator*, which corresponds to the server in the FL setting. This assumption is often not satisfied in practical cases, under which users that act as clients may not trust the service provider that acts as the server. Local differential privacy (LDP) (Kasiviswanathan et al., 2011; Dwork et al., 2014) provides privacy protection on the *individual level* via applying randomized mechanisms that obfuscate the data before leaving the client, and is a more natural privacy model for distributed learning scenarios like FL, allowing easier compliance with regulatory strictures (Bhowmick et al., 2018).

1.2 Methodology

We will be focusing on distributed learning procedures that aim to solve an empirical risk minimization problem in a decentralized fashion:

$$\min_{\theta\in\Theta} L(\theta), \quad L(\theta) = \frac{1}{N}\sum_{m=1}^{M} L_m(\theta) \tag{1}$$

where M denotes the number of clients. Let Y_i denotes training data that belongs to the m-th client consisting N_m data points, and $N = \sum_{m=1}^{M} N_m$. The term $L_m(\theta) = \sum_{y\in Y_m} \ell(\theta, y)$ stands for locally aggregated loss evaluated at parameter θ, where $\ell : \Theta \times \mathcal{Y} \mapsto \mathbb{R}_+$ is the loss function depending on the problem context and we use Θ and $\mathcal{Y}$ to describe parameter and data domain respectively. We will seek to optimize (1) via gradient-based methods. In what follows we will use $[n], n \in \mathbb{N}_+$ to denote the set $\{1, 2, \ldots, n\}$. At round t, the procedure iterates as:

1. Server distribute the current value θ_t among a subset S of clients.
2. For each client $s \in S$, a local update is computed $\Delta_s = g(\theta_t, Y_s)$ and transmitted to the server.
3. The server aggregates all $\{\Delta_s\}_{s\in S}$s to obtain a global update Δ and updates the parameter as $\theta_{t+1} = \theta_t + \Delta$.

We distinguish 2 practical scenarios as *cross-silo* FL and *cross-device* FL (Kairouz et al., 2019). In cross-silo FL, M is relatively small (i.e. $M \leq 100$) and usually each client has a moderate or large amount of data (i.e. $\min_m N_m \gg 1$). As a consequence, all the clients participate in the learning process (i.e. $S = [M]$). In each iteration a client computes locally a negative stochastic gradient $g(\theta_t, Y_s) = -\frac{1}{R}\sum_{y\in\mathcal{R}_s} \nabla\ell(\theta_t, y)$ of $L_s(\theta_t)/N_s$, based on a uniform random subsample $\mathcal{R}_s \subset Y_s$ of size R. In cross-device FL S is a uniformly random subset of $[M]$, and $g(\theta_t, Y_s) = -\frac{1}{N_s}\sum_{y\in Y_s} \nabla\ell(\theta_t, y)$ is the average negative gradient over Y_s evaluated at θ_t. In this paper we will be focusing on the `FedSgd` aggregation rule corresponding to a stochastic gradient step with learning rate η, i.e., $\Delta = \eta \sum_{s\in S} \frac{N_s}{N} g(\theta_t, Y_s)$. Generally speaking, there are two sources of data that need privacy protection: the parameter θ_t and the updates $\{\Delta_s\}_{s\in S}$. In this paper we will develop algorithms for protecting $\{\Delta_s\}_{s\in S}$ against inference attack or reconstruction attack, which will be defined later. Note that the protection of θ_ts could be done via applying standard central DP techniques as in Bhowmick et al. (2018). To begin our discussion on suitably defined privacy models, we first review the local model of privacy.

Local differential privacy Kasiviswanathan et al. (2011) we say a randomized algorithm $\mathcal{A}$ that maps the private data $X \in \mathcal{X}$ to some value $Z = \mathcal{A}(X) \in \mathcal{Z}$ is *ϵ-locally differentially private*, if the induced conditional probability measure $\mathbb{P}(\cdot|X = x)$ satisfies that for any $x, x' \in \mathcal{X}$ and any $Z \in \mathcal{Z}$:

$$e^{-\epsilon} \leq \frac{\mathbb{P}(Z|X = x)}{\mathbb{P}(Z|X = x')} \leq e^{\epsilon} \tag{2}$$

In the FL setting, the private data X is typically not the raw data Y but the gradient of the loss function ℓ evaluated at Y. Hereafter we will assume the private data to be a d dimensional euclidean vector, while d could be very large (i.e. to the order of millions). LDP is a very strong privacy protection model that protects individual data against *inference attacks* that aims at inferring the membership of arbitrary data from the data universe. According to the discussion in Bhowmick et al. (2018), allowing adversaries with such strength may be overly pessimistic: to conduct an effective inference attack the adversary shall come up with "the true data" and only use the privatized output to verify his/her belief about the membership. If the prior information is reasonably constrained for the adversary, we may adopt a weaker, but still elegant type of privacy protection paradigm that protects against *reconstruction attacks*. Heuristically, reconstruction attack aims at recovering individual data with respect to a well-defined criterion, given a prior over the data domain that is not too concentrated. Formally, we adopt the definition in Bhowmick et al. (2018):

Reconstruction attack Bhowmick et al. (2018) Let π be a prior distribution over the data domain $\mathcal{Y}$ that encodes the adversary's prior belief. We describe the generation process of privatized user data as $Y \to X \to Z = \mathcal{A}(X)$ for some (privacy-protecting) mechanism $\mathcal{A}$. Additionally let $f : \mathcal{Y} \mapsto \mathbb{R}^k$ be the target of reconstruction (i.e., the adversary wants to evaluate the value $f(Y)$) and $L_{\mathsf{rec}} : \mathbb{R}^k \times \mathbb{R}^k \mapsto \mathbb{R}_+$ be the reconstruction loss serving as the criterion. Then an estimator $\zeta : \mathcal{X} \mapsto \mathbb{R}^k$ provides an (α, p, f)-reconstruction breach for the loss L_{rec} if there *exists* some $z \in \mathcal{Z}$ such that:

$$\mathbb{P}\left(L_{\mathsf{rec}}(f(X), \zeta(z)) | \mathcal{A}(X) = z\right) > p \tag{3}$$

The existence of a reconstruction breach provides an attack that effectively breaks the privatized algorithm for some output z. Hence to provide protection against reconstruction attacks, we need the following to hold:

$$\sup_{\zeta} \sup_{z \in \mathcal{Z}} \mathbb{P}\left(L_{\mathsf{rec}}(f(X), \zeta(z)) | \mathcal{A}(X) = z\right) \leq p \tag{4}$$

When (4) holds, the mechanism $\mathcal{A}$ is said to be (α, p, f)-protected against reconstruction for the loss L_{rec}. In (Bhowmick et al., 2018, Lemma 2.2), the authors relates LDP mechanisms with protection against reconstruction attacks, and the result for several scenarios suggest that using LDP mechanisms with large ϵ may still provide decent protection against reconstruction. As a consequence, we identify two sorts of threats used in this paper:

Inference attacks by powerful adversaries this is the standard setup in LDP scenarios under which the adversary can observe the final learned model, as well as the information about potential participants except for the precise membership list. For such cases, only low ϵ LDP mechanisms provide sufficient protection against inference attacks.

Reconstruction attacks by curious onlookers in this relaxed scenario we assume the adversary is able to observe all model updates and individual communications to conduct reconstruction attacks that target the evaluation of some function f over private individual data. But knowledge about individual data is restricted some prior distribution π_f. For such cases, we will discover mechanisms that are based upon LDP algorithms with large ϵ but provides decent protection against reconstruction.

Moreover, we assume the adversaries to be *semi-honest* (Lyu et al., 2020), i.e. they are curious about individual data but follow the FL protocol correctly.

In the seminal work (Duchi et al., 2018), the authors developed minimax optimal LDP mean estimators under i.i.d generative constraints. Bhowmick et al. (2018) used separate mechanisms to privatize both the direction and the scale of the individual gradients, and was shown to achieve comparable performance with respect to non-private counterparts using high ϵs. Despite the effectiveness of the LDP approach, the communication cost [1] is prohibitive. For models involving deep neural architectures, the number of parameters explodes to the order of millions. For each round, each client transmit $O(d\delta)$ bits to the server, where δ is the minimum number of bits to represent a real number with desired precision. The resulting communication cost is usually not affordable for mobile scenarios. In fact, the primary bottleneck for FL is usually communication, especially for deep neural architectures (Kairouz et al., 2019). It is thus of significant importance to reduce the communication cost to a reasonable level. We identify two main challenges in building practical privacy-preserving FL algorithms:

Communication Efficiency The algorithm shall be communication efficient, measured in terms of bits transferred per client in a single round.

High-dimensional compatibility It was shown in Duchi et al. (2018) that upon estimating a d-dimensional vector, the locally private constraint incurs a multiplicative penalty of $O(d/\epsilon^2)$ in terms of minimax ℓ_2 risk. This linear dependence on the dimension of gradients may result in overly noisy gradient estimates that destroy the learning process. Hence the algorithm shall be able to handle models with high-dimensional inputs with tolerable performance degradation.

[1] In general, there are two sources of communication cost in a distributed learning scenario that requires sequential interactions between the server and the clients: downlink cost that happens during clients downloading the updated model from the central server, and uplink cost that happens when clients sending their updates to the server. In was previously noted in Konečný et al. (2016) that uplink cost usually dominates downlink costs in FL settings. Hence in the scope of this work, we will refer to communication cost as uplink communication cost.

Summary of contributions In this paper, we propose a practical solution to locally private FL setting that addresses the above concerns which we term *selective quantized stochastic gradient descent (sqSGD)*. Our contributions are summarized as follows:

1. We propose a novel algorithm for locally private multivariate mean estimation. The proposed algorithm stochastically quantizes each dimension of each sample to K equally spaced levels and serves as the base algorithm for our gradient-based learning framework.
2. We make several improvements to the basic algorithm under the gradient-based learning scenario. Specifically, we attribute the estimation error of the base algorithm to perturbation error caused by privacy constraints, and quantization error. Then we utilize a gradient subsampling strategy to simultaneously reduce communication cost and improve private estimation utility. We also apply randomized rotation to reduce quantization error.
3. We verify the performance of sqSGD on benchmark datasets using standard neural architectures like LeNet (Lecun et al., 1998) and ResNet (He et al., 2016), and make systematic studies on the impact of both communication and privacy constraints. Specifically, under the same privacy and communication constraints, our model outperforms baseline algorithms that do not involve quantization by a significant margin.

2 Related Work

Communication-efficient FL Since its introduction in (Konečný et al., 2016), communication efficiency has been one of the central topics of FL (Konečǹy et al., 2016; Kairouz et al., 2019). Konečǹy et al. (2016) described two kinds of general approaches for improving communication efficiency: structured updates like randomized masking, and sketched updates like quantization methods. The design of sqSGD could be viewed as a combination of the aforementioned approaches.

Privacy-preserving FL As optimization via stochastic gradient methods remains the most representative task in federated learning, most privacy-preserving FL approaches are based on private SGD type algorithms like Abadi et al. (2016). McMahan et al. (2018) used a slightly tweaked central DP model to train large language models. Wei et al. (2020) derived federated aggregation schemes under the central DP model and discussed convergence properties. Bhowmick et al. (2018) justified using large privacy budgets in LDP settings, while still provide reasonable privacy guarantees against reconstruction attacks. However, most of the previous works on privacy-preserving FL did not address communication efficiency problems.

Gradient sparsification and quantization Gradient dropping methods threshold gradients based on either fixed single threshold (Aji & Heafield, 2017) or adaptively chosen thresholds (Chen et al., 2017) are reported to greatly reduce communication overhead. DGC (Lin et al., 2018) adopted more elegant refinements like accumulation correction and factor masking to achieve even higher compression ratios. FedSel (Liu et al., 2020) performed privatized gradient sparsification via adopting randomized response techniques to select the most significant gradient dimension. Gradient quantization methods like QSGD (Alistarh et al., 2017) or TernGrad (Wen et al., 2017) are able to achieve little performance degradation on large neural architectures using less than 4 quantization bits. Privatized versions of quantized SGD are recently proposed, cpSGD (Agarwal et al., 2018) adopts the central DP model and vqSGD (Gandikota et al., 2019) allows local DP model.

3 Our model

3.1 Base mechanism

As a starting point, we base our communication efficient mean estimation algorithm on the *stochastic k-level quantization*(Suresh et al., 2017) scheme: We assume a uniform upper bound U on the ℓ_∞ norm of any individual gradients over the whole course of the FL process. The quantization procedure is described as follows: Let $K \geq 2$ be the desired quantization level. We quantize the gradient to a sequence of points $-U = B_1 < B_2 < \ldots, < B_{K-1} < B_K = U$, where

$$B_k = -U + \frac{2(k-1)U}{K-1} \tag{5}$$

The case for $K = 2$ corresponds to using only the endpoints $\{-U, U\}$ for quantization. We will denote the quantization range as $\mathcal{B} = \{B_k\}_{k=1}^{K}$. For each dimension $j \in [d]$ of the individual gradient X^j, we first locate X^j to one of the K bins via finding a k^* such that $X^j \in [B_{k^*}, B_{k^*+1})$ and round X^j to the boundary of the bin as:

$$\widehat{X}^j = \begin{cases} B_{k^*+1} & \text{with probability } \frac{k^*(X^j - B_{k^*})}{2U} \\ B_{k^*} & \text{otherwise} \end{cases} \tag{6}$$

Note that $\widehat{X}^j$ is an unbiased estimator of X^j. The quantization scheme reduces the communication cost to $d \log_2 K$ bits per round per client.

Next we privatize $\widehat{X}^j$ by constructing a locally private unbiased estimator of $\widehat{X}^j$ (where we fix the randomness of the quantization step), thereby obtaining a locally private estimator of X with its value resides in $\mathcal{B}^d$. To describe the privatization scheme we introduce some new notations: for $\forall v_1, v_2 \in \mathcal{B}^d$, define $\mathcal{M}(v_1, v_2) = \#\{j : v_1^j = v_2^j\} - \#\{j : v_1^j \neq v_2^j\}$ where $\#C$ stands for the number of elements in the set C. For a given positive integer $\kappa \in \{0, \ldots, d-1\}$, let $\overline{\mathcal{S}}(v; \mathcal{B}^d) := \{u : \mathcal{M}(u, v) > \kappa\}$ and $\underline{\mathcal{S}}(v; \mathcal{B}^d) := \{u : \mathcal{M}(u, v) \leq \kappa\}$. The privatization procedure, which we termed $\mathsf{PrivQuant}_{K,\infty}$, is summarized in Algorithm1. To state the privacy guarantee of $\mathsf{PrivQuant}_{K,\infty}$, we identify two setups which corresponds to protection against inference and reconstruction attacks. The later case requires additional specifications for the priors and evaluation function held by the adversary, which we state as follows: we assume the evaluation function f to be a map from $\mathcal{Y}$ to a compact set $\mathcal{C} \subset \mathbb{R}^r$. For any absolutely continuous prior π on $\mathcal{Y}$, we use π_f to denote the induced prior on f. Let π_0 be the uniform prior over $\mathcal{C}$, we will be focusing on priors that belong to the following set:

$$\mathcal{P}_f(\rho_0) = \left\{ \pi : \sup_{y \in \mathcal{C}} \log \frac{d\pi_f(y)}{d\pi_0(y)} \leq \rho_0 \right\} \tag{7}$$

where ρ_0 is a non-negative number. The set $\mathcal{P}_f(\rho_0)$ characterize prior beliefs that are "not much more certain than uniformly random guessing" over all possible outcomes that belong to the image of the evaluation function. We state the theorem under the *orthogonal reconstruction attack* that uses the evaluation function $f_A(x) = Ax / \|x\|_2$ with $A \in \mathbb{R}^{r \times d}$ an orthonormal matrix, i.e., $AA^T = I_r$, and the reconstruction criterion is chosen as the ℓ_2 error $L_{\mathsf{rec}}(x, x') = \|x / \|x\|_2 - x' / \|x'\|_2\|_2^2$.

Theorem 1. *For any $K \geq 2$, Consider $\mathsf{PrivQuant}_{K,\infty}$ as a mechanism parameterized by (κ, p). Let $\tau := \lceil \frac{d+\kappa+1}{2} \rceil$, and let (κ, p) be chosen such that the following relation holds:*

$$\frac{p}{1-p} \times \frac{\sum_{\ell=0}^{\tau-1} \binom{d}{\ell} (K-1)^{(d-\ell)}}{\sum_{\ell=\tau}^{d} \binom{d}{\ell} (K-1)^{(d-\ell)}} \leq e^{\epsilon} \tag{8}$$

then Z is an unbiased estimator of X, i.e. $\mathbb{E}(Z) = X$, with randomness jointly over the quantization step and sampling step. Moreover, we have the following privacy guarantees:

Inference attack protection *$\mathsf{PrivQuant}_{K,\infty}(\cdot, \kappa, p)$ is ϵ-locally differentially private.*

Reconstruction attack protection *with $k \geq 4$ and $a \in [0, 1]$, $\mathsf{PrivQuant}_{K,\infty}(\cdot, \kappa, p)$ is $(\sqrt{2-2a}, \omega(a), f_A)$-protected against reconstruction for*

$$\omega(a) = \sqrt{8} \exp\left(-\frac{(r-1)a^2}{2} \right) \exp(\epsilon + \rho_0) \tag{9}$$

The proof will be deferred to appendix A.1. Theorem 1 implies that, if the adversary aims at reconstructing a non-vanishing fraction of the private data with $r = O(d)$, for small a that allows coarse reconstruction, we only need ϵ and ρ_0 to be of smaller order than d to ensure that any reconstruction attack will not succeed with reasonable probability. Note for $K = 2$ levels and $U = 1$, algorithm 1 reduces to algorithm $\mathsf{PrivUnit}_{\infty}$ in (Bhowmick et al., 2018).

We may view the estimation error of Z (measured in terms of ℓ_2 risk) as coming from two parties: one from the quantization step, and the other from the privatization step. In the following sections we will take closer looks at the two sources of error, and develop corresponding improvements. Indeed as verified by our empirical study (see section 4), certain improvements are *necessary* to achieve high-dimensional compatibility.

Algorithm 1 Private K-quantized ℓ_∞ Ball $\mathsf{PrivQuant}_{K,\infty}$

```
Require: $X \in [-U, U]^d$, $\kappa \in \{0, \cdots, d-1\}$, $p \geq \frac{1}{2}$, $\tau = \lceil \frac{d+\kappa+1}{2} \rceil$.
Quantize each coordinate of $X$ to $\mathcal{B}$ using (6) to obtain $\widehat{X} \in \mathcal{B}^d$.
Sample a random vector $V$ as follows: with probability $p$, $V$ is sampled uniformly at random from $\overline{\mathcal{S}}(\widehat{X}; \mathcal{B}^d)$; otherwise with probability $1-p$, $V$ is sampled uniformly at random from $\underline{\mathcal{S}}(\widehat{X}; \mathcal{B}^d)$.
Calculate normalizing factor
    $m = p\frac{\binom{d-1}{\tau-1}(K-1)^{d-\tau}}{\sum_{\ell=\tau}^{d}\binom{d}{\ell}(K-1)^{d-l}} - (1-p)\frac{\binom{d-1}{\tau-1}(K-1)^{d-\tau}}{\sum_{\ell=0}^{\tau-1}\binom{d}{\ell}(K-1)^{d-l}}$
return $Z = \frac{1}{m} \cdot V$
```

3.2 Improving privatization utility via selective gradient update

$\mathsf{PrivQuant}_{K,\infty}$ provides an unbiased estimator of individual gradients, but the considerable amount of noise injected would hurt the training process. This defect is amplified in deep learning scenarios: empirically, Lin et al. (2018) observed that for many representative deep architectures, most of the gradient dimensions are nearly sparse. Hence privatizing the whole gradient vector would result in a high privatization error that is even orders of magnitude higher than the norm of the gradient itself.

The (almost) sparse structure of the gradients suggests a modification to the estimation scheme via privatizing and transmitting only a fraction of the gradient dimensions, which could be regarded as significantly reducing variance at the cost of a small amount of bias. Such techniques are closely related to *gradient compression* that sends a few most significant gradient dimensions (Lin et al., 2018). Typically, gradient compression techniques require selecting top-k gradients measured in absolute magnitude (Lin et al., 2018; Aji & Heafield, 2017). A straightforward extension to the local private setting would be to perform private selection methods like *exponential mechanism* (Dwork et al., 2014) or noisy top-k (Ding et al., 2019), for which we need to allocate a certain fraction of privacy budget. However, these solutions are either computationally expensive or "too noisy" for picking high dimensional gradients that are almost sparse. We defer a more detailed discussion to appendix A.2.

Motivated by algorithm designs in distributed learning like random masking (Konečný et al., 2016) and Hogwild! (Niu et al., 2011), we utilize a simple strategy, *gradient subsampling*, by randomly sample $\tilde{d} = \lfloor rd \rfloor$ dimensions, where r is the sampling proportion of gradient dimensions. To further improve training efficiency, we applied local accumulation techniques like momentum correction and factor masking similar to (Lin et al., 2018). It is worth noting that low sampling ratio do *not* necessarily improves model performance as the overall bias of the sparse approximation would become significant when only a few dimensions are selected. A more detailed study of this aspect is provided in appendix A.3.

3.3 Improving quantization performance via randomized rotation

As shown in (Agarwal et al., 2018; Suresh et al., 2017), for a d-dimensional vector X, the quantization error scales with $\left(\max_{j\in[d]} X^j - \min_{j\in[d]} X^j\right)^2$. To reduce quantization error, an effective way is to preprocess the data via *randomized rotation*, so as to make $\max_{j\in[d]} X^j - \min_{j\in[d]} X^j$ small in expectation. Note that we only need to rotate the fraction of gradient that are selected to get transmitted.

To perform randomized rotation we need to assume public randomness between the server and the clients, under which we generate an orthogonal random matrix $R \in \mathbb{R}^{\tilde{d}\times\tilde{d}}$ and apply the linear transform to the individual gradients $\tilde{X} = RX$ on the client side, and apply inverse transform on the server side $\tilde{Z} = R^{-1}Z$. We adopt the method in (Agarwal et al., 2018; Suresh et al., 2017) to generate R that support fast matrix multiplication with $O(d \log d)$ time and $O(1)$ space. Specifically, we generate $R = \frac{1}{\tilde{d}}HA$ where A is a random diagonal matrix with i.i.d. Rademacher entries (i.e.

$\mathbb{P}(A_{ii} = 1) = \mathbb{P}(A_{ii} = -1) = 0.5, \forall i \in [\tilde{d}]$), and H is a Walsh-Hadamard matrix (Horadam, 2012), defined recursively by the formula:

$$H(2^1) = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, H(2^m) = \begin{bmatrix} H(2^{m-1}) & H(2^{m-1}) \\ H(2^{m-1}) & -H(2^{m-1}) \end{bmatrix}, \tag{10}$$

The randomized rotation operation is performed *before* quantization and privatization, therefore we project the vector X to an ℓ_2 ball of radius U before rotation. This would ensure the vector obtained after rotation has ℓ_∞ norm bounded by U as well.

The final selective SGD updating process is summarized in algorithm 2. The randomized rotation and sparsification strategy have no effect on privacy level, hence the local update within a single round is ϵ-LDP. At the same time, communication cost was further reduced from $O(d \log_2 K)$ bits to $O(\tilde{d} \log_2 K)$ bits per client.

Algorithm 2 sqSGD: Selective Quantized SGD with local privacy guarantee

```
Require: Training data $\{Y_i\}_{i=1}^M$, gradient norm bound $U$, privacy budge $\epsilon$, sampling ratio $r \in (0,1)$,
    local correction factor $\alpha, \beta$, learning rate $\eta$, batch size $B$.
Find a $(\kappa_\epsilon, p_\epsilon)$ pair that satisfies relation (8)
Calculate the corresponding normalizing constant $m_\epsilon$ as in algorithm 1
Set $\tilde{d} = 2^{\lfloor \log_2(rd) \rfloor}$, and using public randomness to generate a random matrix $R \in \mathbb{R}^{\tilde{d}\times\tilde{d}}$,
    according to the construction in (10)
Initialize model parameter $\theta_0$ and local residuals $\mathsf{res}_{i,0} = \mathbf{0}, \forall i = 1, \ldots, N$
for $t = 0, \ldots, T-1$ do
    Sample a batch of indices $S_t \in [N]$ with size $B$
    for $s \in S$ do
        //* Client Side Update *//
        Select a uniformly random subset of $[d]$ with cardinality $\tilde{d}$, denoted as $\mathcal{D}_s$,
        also let $\mathcal{D}_s^C$ denote $[d]\backslash\mathcal{D}$.
        Calculate local (stochastic) gradient with respect to client data $X_{s,t} = g(Y_{s,t}, \theta_t)$
        Clip the local gradient to the $\ell_2$ ball with radius $U$
        Accumulate and Apply rotation $\tilde{X}_{s,t} = R\left(\mathsf{res}_{i,t}[\mathcal{D}_s] + \beta X_{s,t}[\mathcal{D}_s]\right)$
        Rotate and update local residual $\mathsf{res}_{s,t+1}[\mathcal{D}_s^C] = \mathsf{res}_{s,t}[\mathcal{D}_s^C] + \alpha X_{s,t}[\mathcal{D}_s^C]$
        and $\mathsf{res}_{s,t+1}[\mathcal{D}_s] = 0$
        Calculate $Z_{s,t} = \mathsf{PrivQuant}_{K,\infty}(\tilde{X}_{s,t}, \kappa_\epsilon, p_\epsilon)$
        Upload $(Z_{s,t}/m_\epsilon, \mathcal{D}_s)$ to the server
    //* Server Side Update *//
    Gradient update step $\theta_{t+1} = \theta_t - R^{-1}\frac{\eta}{B}\sum_{s\in S} Z_{s,t}^{\mathsf{dense}}$, where $Z_{s,t}^{\mathsf{dense}}$ is the $d-$ dimensional
    zero vector with indices in $\mathcal{D}_s$ replaced by $Z_{s,t}$.
return $\theta_T$
```

4 Experiments

In this section, we apply sqSGD to simulated FL environments, constructed via standard datasets MNIST (Lecun et al., 1998), EMNSIT (Cohen et al., 2017), and Fashion MNIST (Xiao et al., 2017) which we abbreviate as FMNIST hereafter. All three datasets are of equal (training) sample size, therefore we randomly partition each dataset into 10 equally sized blocks to simulate a *cross-silo* FL environment consisting of a central server and 10 clients.

We train on MNIST and EMNIST datasets using the LeNet-5 architecture (Lecun et al., 1998) ($119,850$ parameters). We also train a larger network under the ResNet-110 architecture ($1,722,224$ parameters) on the FMNIST dataset. Since the main goal of the empirical study is not to achieve state-of-the-art performance, we adopt the following common strategies without further tuning across all experiments: in each round, we randomly sample a single batch with batch size 32 for each client and compute a single gradient update step. Without gradient subsampling, this would be equivalent to a stochastic gradient descent update over the entire data with a batch size of 320. The training duration is measured via number of epochs (1 epoch $=$ 188 rounds). We use the same configuration

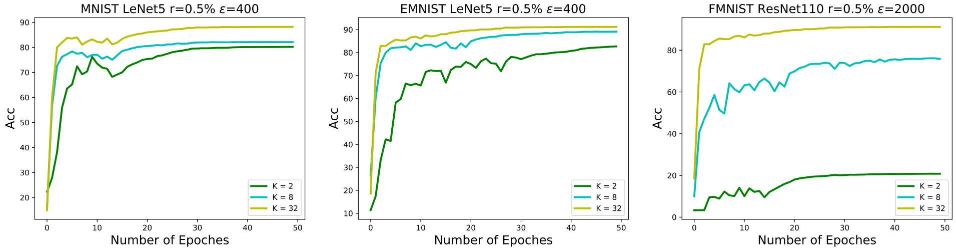

Figure 1: Study on impact of communication constraints

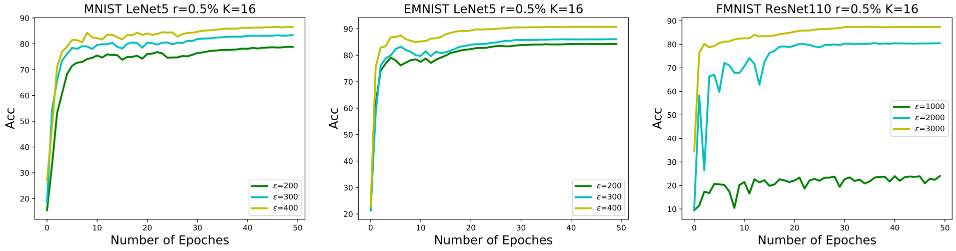

Figure 2: Study on impact of privacy constraints

of learning rate $\eta = 0.001$ and local correction factors $\alpha = 1.0, \beta = 1.0$ across all experiments. We set the ℓ_2 norm bound of the gradients $U = 10$ as it works reasonably well. Note that we could use a more elegant distributed private estimation of the ℓ_2 norm bound via sending privatized magnitude evaluated at different points of the parameter space, i.e. via mechanisms like ScalarDP in Bhowmick et al. (2018). Finally, for a given privacy budget ϵ, we use the following method to determine the value of $(\kappa_\epsilon, p_\epsilon)$: We set κ_ϵ to be the largest integer such that the value of the left hand side of (8) is smaller than or equal to $e^{0.9\epsilon}$, and $p_\epsilon = \frac{e^{0.1\epsilon}}{1+e^{0.1\epsilon}}$.

Choice of privacy budget ϵ across all the experiments we align the choice $\epsilon = O(\sqrt{d})$ with respect to the additional risk factor $O(d/\epsilon^2)$ in private mean estimation (in comparison to non-private case). As the models used in our experiments are all of high dimension, the choice of high-epsilon will basically provide little protection against inference attacks. However, with respect to reconstruction attack specified in section 1.2 and the protection level (4), such choice still provide reasonable protection against reconstruction attacks if the adversary's goal is to evaluate more than $O(\sqrt{d})$ of the entries in the private data.

Impact of communication constraints We evaluate the performance of trained models using the test set accuracy. The privacy budget is chosen as $\epsilon = 400$ for LeNet models and $\epsilon = 2000$ for ResNet models. Sampling ratio is fixed at $r = 0.5\%$. We vary the quantization level in the range $\{2, 8, 32\}$ which corresponds to using $\{1, 3, 5\}$ bits per client per dimension. The results are plotted in figure 1. The results indicate that a while 1-bit sqSGD is efficient in terms of communication cost, the quantization error results in significant degradation in model performance, both in the final accuracy after model convergence, and in the speed of convergence. A reasonable quantization level (i.e. larger than 3 bits) yields competitive performance.

Impact of privacy constraints Next we investigate the performance loss due to privacy constraints. We fix the sampling rate at $r = 0.5\%$ and quantization level at $K = 16$. For LeNet-5 architecture, we select ϵ from the set $\{200, 300, 400\}$, for ResNet-110 architecture, we select ϵ from the set $\{1000, 2000, 3000\}$. The results are plotted in figure 2. The results imply that with limited communication, we need high privacy budgets to ensure successful training on large models.

Baseline comparisons We compare sqSGD against two baselines:

Piecewise mechanism (PM) (Wang et al., 2019) PM is designed for mean estimation under local privacy model that improves accuracy upon the $\mathsf{PrivUnit}_\infty$ algorithm in Bhowmick et al. (2018)

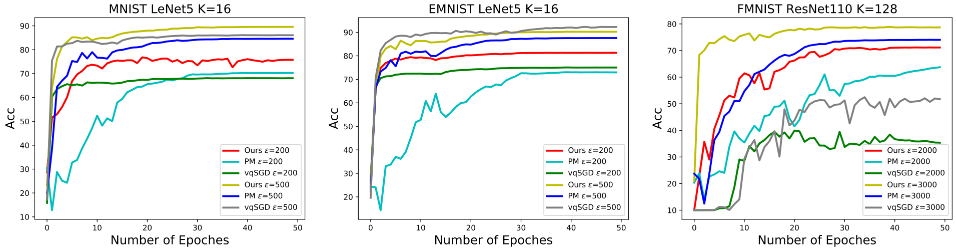

Figure 3: Comparison of sqSGD against piecewise mechanism

$\kappa = 0$. Although PM communicates real values, the multi-dimensional version of PM (Wang et al., 2019, Algorithm 4) samples a fraction of coordinates for perturbation hence we may view PM practically as a limited-communication mean estimation algorithm. For large d settings, we choose reasonably large ϵ values such that PM communicates raw perturbed gradients in $\lfloor \epsilon/2.5 \rfloor$ dimensions.

vqSGD Gandikota et al. (2019) vqSGD uses an communication-optimal distributed mean estimation algorithm that adopts vector quantization strategies. The per client communication cost of vqSGD is $O(\log d)$ for mean estimation task. The algorithm has trivial extensions to locally private settings using randomized response strategies (Gandikota et al., 2019). As discussed by the authors, for distributed optimization settings with high dimensional models. Variance reduction techniques are necessary, specifically one data point is communicated b times and server uses the average of b quantized gradients for aggregation. This makes the communication cost $O(b \log d)$. Note that in private settings this results in a caveat that privacy protection level is degraded under composition. In our experiments we will ignore this caveat and use variance reduction so that the communication of vqSGD matches that of sqSGD and PM.

We compare the three approaches across all tasks mentioned above under the same privacy budget. For sqSGD we use $K = 16$ for LeNet architectures and $K = 128$ for ResNet architectures, and we adjust the sampling ratio in sqSGD such that communication costs remain equal among all methods. The comparisons are presented in figure 3. Figure 3 shows that sqSGD consistently outperforms PM by a significant margin across all tasks under various privacy levels. vqSGD obtains comparable performance with sqSGD in relatively smaller models and high-privacy regime (Note that vqSGD approaches offer much worse privacy protections in this setting), but breaks down when models get larger. Overall, in high-dimensional settings, sqSGD provides much better optimization performance.

Additional experiments We present an experimental study on the effect of gradient subsampling in appendix A.3, and another study on the trade-off between quantization and subsampling in A.4. The results of additional experiments further justify the choice of sampling ratios and quantization levels in the above comparisons.

5 Conclusion

We studied privacy-preserving federated learning under the local differential privacy model. To achieve both communication efficiency and high-dimensional compatibility, we proposed a gradient-based learning framework sqSGD that is based on a novel private multivariate mean estimation scheme with controllable quantization levels. We applied gradient subsampling and randomized rotation to reduce estimation error of the base mechanism that exploits the specific structure of federated learning with large modern neural architectures. Finally, our designed algorithm was shown on standard datasets to be capable of training large models with random initializations, surpassing baseline algorithms by a significant margin.

References

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*, CCS '16, pp. 308–318, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145/2976749.2978318.

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan. cpsgd: Communication-efficient and differentially-private distributed sgd. In *Advances in Neural Information Processing Systems*, pp. 7564–7575, 2018.

Alham Fikri Aji and Kenneth Heafield. Sparse communication for distributed gradient descent. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 440–445, 2017.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1709–1720. Curran Associates, Inc., 2017.

Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

Chia-Yu Chen, Jungwook Choi, Daniel Brand, Ankur Agrawal, Wei Zhang, and Kailash Gopalakrishnan. Adacomp : Adaptive residual gradient compression for data-parallel distributed training, 2017.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

Zeyu Ding, Yuxin Wang, Danfeng Zhang, and Daniel Kifer. Free gap information from the differentially private sparse vector and noisy max mechanisms, 2019.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Venkata Gandikota, Daniel Kane, Raj Kumar Maity, and Arya Mazumdar. vqsgd: Vector quantized stochastic gradient descent, 2019.

Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kathy J Horadam. *Hadamard matrices and their applications*. Princeton university press, 2012.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konečný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning, 2019.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency, 2016.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

D. Leroy, A. Coucke, T. Lavril, T. Gisselbrecht, and J. Dureau. Federated learning for keyword spotting. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6341–6345, 2019.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and Bill Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SkhQHMW0W`.

Ruixuan Liu, Yang Cao, Masatoshi Yoshikawa, and Hong Chen. Fedsel: Federated sgd under local differential privacy with top-k dimension selection, 2020.

Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BJ0hF1Z0b`.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Inference attacks against collaborative learning. *arXiv preprint arXiv:1805.04049*, 13, 2018.

Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. Hogwild! a lock-free approach to parallelizing stochastic gradient descent. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pp. 693–701, Red Hook, NY, USA, 2011. Curran Associates Inc. ISBN 9781618395993.

Ananda Theertha Suresh, X Yu Felix, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *International Conference on Machine Learning*, pp. 3329–3337, 2017.

Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. Collecting and analyzing multidimensional data with local differential privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649. IEEE, 2019.

K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

Wei Wen, Cong Xu, Feng Yan, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Terngrad: Ternary gradients to reduce communication in distributed deep learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1509–1519. Curran Associates, Inc., 2017.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

A Appendix

A.1 Proofs

Proof of theorem 1. Let $u \in \mathcal{B}^d$ and $U \sim \mathsf{Unif}(\mathcal{B}^d)$. The vector $V \in \mathcal{B}^d$ sampled as in algorithm 1, has p.m.f.

$$p(v \mid u) \propto \begin{cases} 1/\mathbb{P}(\mathcal{M}(U,u) > \kappa) & \text{if } \mathcal{M}(v,u) > \kappa \\ 1/\mathbb{P}(\mathcal{M}(U,u) < \kappa) & \text{if} \mathcal{M}(v,u) \leq \kappa. \end{cases}$$

The event that $\mathcal{M}(U,u) = \kappa$ when $\frac{d+\kappa+1}{2} \in \mathbb{Z}$ implies that U and u match in exactly $\frac{d+\kappa+1}{2}$ coordinates; the number of such matches is $\binom{d}{(d+\kappa+1)/2}$. Computing the binomial sum, we have

$$\begin{aligned} \mathbb{P}(\mathcal{M}(U,u) > \kappa) &= \frac{1}{K^d} \sum_{\ell=\lceil \frac{d+\kappa+1}{2} \rceil}^{d} \binom{d}{\ell} (K-1)^{(d-\ell)} \\ \mathbb{P}(\mathcal{M}(U,u) \leq \kappa) &= \frac{1}{K^d} \sum_{\ell=0}^{\lceil \frac{d+\kappa+1}{2} \rceil - 1} \binom{d}{\ell} (K-1)^{(d-\ell)} \end{aligned} \tag{11}$$

Now we show unbiasedness via showing $\mathbb{E}[V \mid u = u] = m \cdot u$. We have

$$\mathbb{E}[V \mid u = u] = p\mathbb{E}[U \mid \mathcal{M}(U,u) > \kappa] + (1-p)\mathbb{E}[U \mid \mathcal{M}(U,u) \leq \kappa]$$

By rotational symmetry, it suffices to show:

$$\mathbb{E}[V^1 \mid u = u] = p \underbrace{\mathbb{E}[U^1 \mid \mathcal{M}(U,u) > \kappa]}_{\Upsilon_1} + (1-p) \underbrace{\mathbb{E}[U^1 \mid \mathcal{M}(U,u) \leq \kappa]}_{\Upsilon_2}$$

For Υ_1, we have:

$$\begin{aligned} \Upsilon_1 &= \frac{1}{K^d \mathbb{P}(\mathcal{M}(U,u) > \kappa)} \times \sum_{\ell=\tau}^{d} \left(u^1 \binom{d}{\ell} (K-1)^{d-\ell} - \sum_{w \in \mathcal{B} \setminus u^1} w \binom{d}{\ell} (K-1)^{d-\ell-1} \right) \\ &= \frac{u^1}{K^d \mathbb{P}(\mathcal{M}(U,u) > \kappa)} \times \binom{d-1}{\tau-1} (K-1)^{(d-\tau)} \end{aligned}$$

Where in the second equality we used the fact that $\sum_{w \in \mathcal{B}} w = 0$. Similar calculations yield:

$$\Upsilon_2 = -\frac{u^1}{K^d \mathbb{P}(\mathcal{M}(U,u) \leq \kappa)} \times \binom{d-1}{\tau-1} (K-1)^{(d-\tau)}$$

Combining the preceding display with (11), we have:

$$\begin{aligned} \mathbb{E}[V^1 \mid u = u] &= \left(p \frac{\binom{d-1}{\tau-1} (K-1)^{d-\tau}}{\sum_{\ell=\tau}^{d} \binom{d}{\ell} (K-1)^{d-l}} - (1-p) \frac{\binom{d-1}{d\tau-1} (K-1)^{d-\tau}}{\sum_{\ell=0}^{\tau-1} \binom{d}{\ell} (K-1)^{d-l}} \right) \times u^1 \\ &= mu^1 \end{aligned}$$

Next we show privacy guarantee.

LDP guarantee As $\mathbb{P}(\mathcal{M}(U,u) > \kappa)$ is decreasing in κ for any $u, u' \in \mathcal{B}^d$ and $v \in \mathcal{B}^d$ we have

$$\frac{p(v \mid u)}{p(v \mid u')} \leq \frac{p}{1-p} \cdot \frac{\mathbb{P}(\mathcal{M}(U,u') \leq \kappa)}{\mathbb{P}(\mathcal{M}(U,u) > \kappa)} = \frac{p}{1-p} \times \frac{\sum_{\ell=0}^{\tau-1} \binom{d}{\ell} (K-1)^{(d-\ell)}}{\sum_{\ell=\tau}^{d} \binom{d}{\ell} (K-1)^{(d-\ell)}}$$

The result follows by relation (8)

Reconstruction protection guarantee the result follows by Bhowmick et al. (2018, Proposition 1) and the LDP guarantee.

□

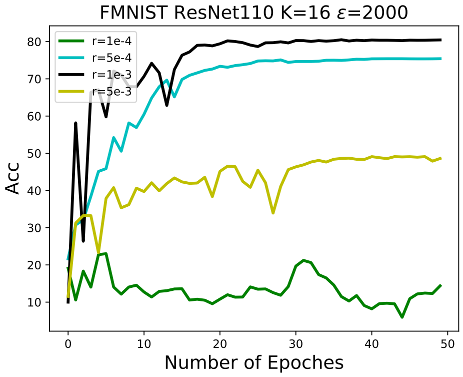

Figure 4: Study on the effect of gradient subsampling

A.2 On the practicality of privacy-preserving selection methods

Performing top-k selection with local privacy constraints could be done via two approaches. The first take is iteratively running the exponential mechanism (Dwork et al., 2014) for k times, each time selecting a single index with gumble noise. The bottleneck of this take is that it requires sampling from a high dimensional distribution for k times, which is computationally heavy. Note that the privatization step is carried out on the client side device, which is usually assumed to be of limited computational power in FL settings (Kairouz et al., 2019), thus using iterative exponential mechanism is not practical for FL scenarios. The second take is the noisy top-k algorithm (Ding et al., 2019), which generalizes the report noisy max mechanism in Dwork et al. (2014). The algorithm requires adding Laplacian noise of scale $2Uk/\epsilon$ to each dimension of the gradient vector. In practice, k is typically chosen at the order of hundreds. Since most of the gradients are very small in magnitude,to ensure reasonable noise requires a high ϵ budget to allocate for the selection step. This would significantly affect the overall privacy level.

A.3 Experiments on the effect of gradient subsampling

In this experiment, we investigate the effect of subsampling under the setup of training a ResNet110 model on the FMNIST dataset. We fix the privacy level at $\epsilon = 2000$ and quantization level at $K = 16$. We vary the subsampling ratio from the set $\{1, 5, 10, 50\} \times 10^{-3}$. The results are plotted in figure 4. It could be seen from the plot that using a high sampling ratio severely hurts the training performance, while also incurs more communication. It is thus necessary to perform subsampling. However, using a very low subsampling ratio also causes training failure. This phenomenon will be further explored in the next experiment.

A.4 Experiments on the trade-off between quantization and sampling

In this experiment, we study the trade-off between quantization and sampling under the setup of training a LeNet-5 model on the MNIST dataset. We fix the privacy level at $\epsilon = 400$, and fix the total communication cost per client, measured using the product $r \log_2 K$. We vary the quantization level in the range $\{2, 8, 32, 128\}$ which corresponds to using $\{1, 3, 5, 7\}$ bits per client per dimension, and the sampling rate are adjusted accordingly. The results are shown in figure 5. The results suggest that increasing the quantization level may not monotonically increase training performance. This is mainly due to the random subsample scheme of sqSGD, under which the structure of gradients is not fully explored.

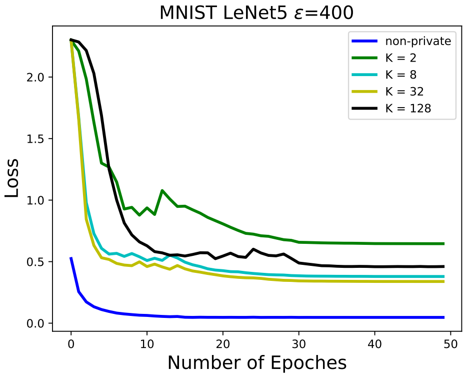

Figure 5: Study on the trade-off between quantization and sampling with fixed communication level