FedSel: Federated SGD under Local Differential Privacy with Top-k Dimension Selection

Ruixuan Liu[1], Yang Cao[2], Masatoshi Yoshikawa[2], Hong Chen[1(✉)]

[1]Renmin University of China, Beijing, China
{ruixuan.liu, chong}@ruc.edu.cn
[2]Kyoto University, Kyoto, Japan
{yang, yoshikawa}@i.kyoto-u.ac.jp

Abstract. As massive data are produced from small gadgets, federated learning on mobile devices has become an emerging trend. In the federated setting, Stochastic Gradient Descent (SGD) has been widely used in federated learning for various machine learning models. To prevent privacy leakages from gradients that are calculated on users' sensitive data, local differential privacy (LDP) has been considered as a privacy guarantee in federated SGD recently. However, the existing solutions have a dimension dependency problem: the injected noise is substantially proportional to the dimension d. In this work, we propose a two-stage framework *FedSel* for federated SGD under LDP to relieve this problem. Our key idea is that not all dimensions are equally important so that we privately select Top-k dimensions according to their contributions in each iteration of federated SGD. Specifically, we propose three private dimension selection mechanisms and adapt the gradient accumulation technique to stabilize the learning process with noisy updates. We also theoretically analyze privacy, accuracy and time complexity of *FedSel*, which outperforms the state-of-the-art solutions. Experiments on real-world and synthetic datasets verify the effectiveness and efficiency of our framework.

Keywords: Local Differential Privacy · Federated Learning.

1 Introduction

Nowadays, massive peripheral gadgets, such as mobile phones and wearable devices, produce an enormous volume of personal data, which boosts the process of Federated Learning (FL). In the cardinal FL setting, Stochastic Gradient Descent (SGD) has been widely used [1–4] for various machine learning models, such as Logistic Regression, Support Vector Machine and Neural Networks. The server iteratively updates a global model for E epochs based on gradients of the objective function, which are collected from batches of m clients' d-dimensional local updates. Nevertheless, users' data are still under threats of privacy attacks [5–8] if the raw gradients are transmitted to an untrusted server. Local

Table 1. Overview Comparison (the number of epochs E, the dimension of a gradient vector d, the compressed dimension q, the privacy budget for training ϵ, the amount of participants in one aggregation m. See detailed analyses in Sects.3.1 and 4.2).

LDP-SGD solutions	upper bound of noise	lower bound of batch size
flat solution [12–14]	$O(\frac{E\sqrt{d\log d}}{\epsilon\sqrt{m}})$	$\Omega(\frac{E^2 d\log d}{\epsilon^2})$
compressed solution [9]	$O(\frac{E\sqrt{q\log q}}{\epsilon\sqrt{m}})$	$\Omega(\frac{E^2 q\log q}{\epsilon^2})$
our solution	$O(\frac{E\sqrt{\log d}}{\epsilon\sqrt{md}})$	$\Omega(\frac{E^2\log d}{d\epsilon^2})$

differential privacy (LDP) provides a rigorous guarantee to perturb users' sensitive data before sending to an untrusted server. Many LDP mechanisms have been proposed for different computational tasks or data types, such as matrix factorization [9], key-valued data [10, 11] and multidimensional data [12–16].

However, applying LDP to federated SGD faces a nontrivial challenge when the dimension d is large. First, although some LDP mechanisms [12–14] proposed for multidimensional data are shown to be applicable to SGD (i.e., flat solution in Table 1), the injected noise is substantially proportional to the dimension d. Besides, in order to obtain an acceptable accuracy, the required batch size of clients (i.e., m) linearly depends on d. As the clients in federated learning have the full autonomy for their local data and can decide when and how to join the training process, a large required batch size impedes the model's applicability in practice. Second, a recent work [9] (i.e., compressed solution in Table 1) attempts to solve this problem by reducing the dimension from d to q with random projection [17], which is a well-studied dimension reduction technique. However, as this method randomly discards some dimensions, it introduces a large recovery error which may damage the learning performance.

Our idea is that not all the dimensions are equally "important" so that we privately select Top-k dimensions according to their contributions (i.e., the absolute values of gradients) in each iteration of SGD. A simple method for private Top-1 selection is to employ exponential mechanism [18] that returns a private dimension with a probability proportional to its absolute value of gradient. However, the challenge is how to design Top-k selection mechanisms under LDP for federated SGD with better selection strategies. Besides the absence of such private Top-k selection mechanisms, another challenge is that discarding delayed gradients causes the convergence issues, especially in our private setting where extra noises are injected. Although we can accumulate delayed gradients with momentum [19], it still requires additional design to stabilize the learning process with noisy updates.

Contributions: In this work, we take the first attempt to mitigate the dimension dependency problem in federated learning with LDP. We design, implement and evaluate a two-stage ϵ-LDP framework for federated SGD. As shown in Table 1, comparing to the state-of-the-art techniques, our framework relieves the effect of the number of dimensions d on the injected noise and the required batch size (i.e., size of clients in each iteration; the lower, the better). Our contributions are summarized below.

- First, we propose a two-stage framework *FedSel* with *Dimension Selection* and *Value Perturbation*. With our privacy analysis in Sect.3.2, this framework satisfies ϵ-LDP for any client's local vector. Our theoretical utility analysis in Sect.3.3 shows that it significantly reduces the dimensional dependency on LDP estimation error from $O(\sqrt{d})$ to $O(1/\sqrt{d})$. We also enhance the framework to avoid the loss of accuracy by accumulating delayed gradients. Intuitively, delayed gradients can improve the empirical performance and fix the convergence issues. In order to further stabilize the learning in the private setting with noises injected, we modify an existing accumulation [19]. Our analysis and experiments validate that this modification reduces the variance of noisy updates. (Section 3)
- Second, we instantiate the *Dimension Selection* stage with three mechanisms, which are general and independent of the second stage of value perturbation. The privacy guarantee for the selection is provided. Besides, we show the advance of utility and computation cost by extending Top-1 to Top-k case with analysis and experiments. (Section 4)
- Finally, we perform extensive experiments on synthetic and real-world datasets to evaluate the proposed framework and the private Top-k dimension selection mechanisms. We also implement a *hyper-parameters-free* strategy to automatically allocate the privacy budgets between the two stages with better utility. Significant improvements are shown in test accuracy comparing with the existing solutions [9, 12–14]. (Section 5)

The remainder of this paper is organized as follows. Section 2 presents the technical background. Section 3 illustrates the two-stage privatized framework with analyses. Section 4 proposes the Exponential Mechanism (EXP) for Top-1 selection and extends to Top-k case with Perturbed Encoding Mechanism (PE) and Perturbed Sampling Mechanism (PS). Section 5 provides results on both synthetic and real-world datasets and a *hyper-parameters-free* strategy. Section 6 gives an overview of related works. Section 7 concludes the paper. Due to the limited space, we put the complete pseudocodes and proofs in a full version of this paper.

2 Preliminaries

2.1 Federated SGD

Suppose a learning task defines the objective loss function $L(w; x, y)$ on example (x, y) with parameters $w \in \mathbb{R}^d$. The goal of learning is to construct an empirical minimization as $w^* = \arg\min_w \frac{1}{N} \sum_{i=1}^{N} L(w; x_i, y_i)$ over N clients' data. For a single iteration, a batch of m clients updates local models in parallel with the distributed global parameters. Then they transmit local model updates to the server for an average mean gradient to update the global model as: $w^t \leftarrow w^{t-1} - \alpha \frac{1}{m} \sum_{i=1}^{m} \bigtriangledown L(w^{t-1}; x_i, y_i)$. Without loss of generality, we describe our framework with the classic setting with one local update for each round.

2.2 Local Differential Privacy

Local differential privacy (LDP) is proposed for collecting sensitive data through local perturbations without any assumption on a trusted server. $\mathcal{M}$ is a randomized algorithm that takes a vector v as input and outputs a perturbed vector v^*. ϵ-LDP is defined on $\mathcal{M}$ with a privacy budget ϵ as follows.

Definition 1 (Local Differential Privacy [20]). *A randomized algorithm $\mathcal{M}$ satisfies ϵ-LDP if and only if the following is true for any two possible inputs $v, v' \in \mathcal{V}$ and output v^*: $Pr[\mathcal{M}(v) = v^*] \leq e^\epsilon \cdot Pr[\mathcal{M}(v') = v^*]$.*

2.3 Problem Definition

This paper studies the problem of federated SGD with LDP. Note that in the practical non-private setting, the original gradient $g^t \leftarrow \bigtriangledown L(w^{t-1}; x, y)$ can be sparsified [21] or quantized [22] before a transmission. For generality, we do not limit the form of local gradient and use v_i to denote the local gradient calculated from client u_i's record (x_i, y_i) and the global parameters w^{t-1}.

Suppose the global model iterates for E epochs with a learning rate α and a total privacy budget ϵ for each client. For a single epoch, clients are partitioned into batches with size m. Then the privatized mechanism $\mathcal{M}$ privatizes m local updates before they are aggregated by the untrusted server for one iteration. The global model is updated as: $w^t \leftarrow w^{t-1} - \alpha \frac{1}{m} \sum_{i=1}^{m} \mathcal{M}(v_i)$. We aim to propose an ϵ-LDP framework with a specialized mechanism $\mathcal{M}$ for private federated training against the untrusted server. Moreover, we attempt to mitigate the dimension dependency problem for a higher accuracy.

3 Two-stage LDP Framework: FedSel

In this section, we propose a two-stage framework *FedSel* with dimension selection and value perturbation as shown in Fig.1. The framework and differences from existing works are presented in Sect.3.1. We prove the privacy guarantee in Sect.3.2 and analyze the stability improvement in Sect.3.3.

3.1 Overview

We now illustrate the proposed framework in Algorithm 1 and compare it with the flat solutions [12–14] and the compressed solution [9]. In our framework, the server first initiates the ratio $\mu \in [0, 1]$ for privacy budget allocation and starts the iteration. For the procedure on a local device: (i) Current gradient g^t is accumulated with previously delayed gradients r^{t-1} (line 15). (ii) An important dimension index is privately selected by **Dimension Selection** (line 16). (iii) The value of the selected dimension plus its momentum (line 17) is perturbed by **Value Perturbation** (line 18). The accumulation in step(i) and the momentum in step(iii) derive from the work of Sun et al. [19] to compress local gradient with little loss of accuracy and memory cost. We adapt the accumulation in

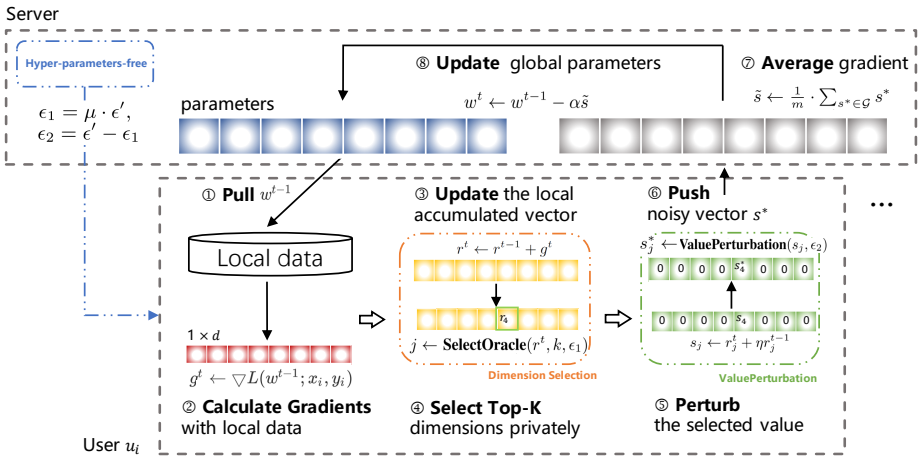


Fig. 1. Two-stage LDP framework.

Algorithm 1 Two-stage LDP framework of federated learning

$E, N, m, \mu, \alpha, \epsilon$ for the server; $\epsilon_1, \epsilon_2, \eta, k$ for N clients;

GlobalUpdate:

initialize $t = 1, w^0, r_i^0 \leftarrow \{0\}^d$ for $i \in [N]$
$\mu \leftarrow$ **HyperParametersFree**(m, ϵ, d)
initialize $\epsilon' = \epsilon / E, \epsilon_1 = \mu \cdot \epsilon', \epsilon_2 = \epsilon' - \epsilon_1$
for each epoch $1, \cdots, E$ **do**
  for each sample batch with size m **do**
    initialize $\mathcal{G}$ as an empty set
    for each client **do**
      $s^* \leftarrow$ LocalUpdate$(w^{t-1}, \epsilon_1, \epsilon_2, \eta, k)$
      add s^* to $\mathcal{G}$
    $\tilde{s} \leftarrow \frac{1}{m} \cdot \sum_{s^* \in \mathcal{G}} s^*$ # aggregation
    $w^t \leftarrow w^{t-1} - \alpha\tilde{s}$, $t = t + 1$
return final global model w

LocalUpdate: $(w^{t-1}, \epsilon_1, \epsilon_2, \eta, k)$

initialize $s^* \leftarrow \{0\}^d$
$g^t \leftarrow \nabla L(w^{t-1}; x, y)$
$r^t \leftarrow r^{t-1} + g^t$ # adapted accumulation
$j \leftarrow$ **SelectOracle**(r^t, k, ϵ_1)
$s_j \leftarrow r_j^t + \eta r_j^{t-1}$
$s_j^* \leftarrow$ **ValuePerturbation**$(s_j, \epsilon_2), r_j^t \leftarrow 0$
return s^*

our framework to stabilize the iteration with noises. The step(ii) is designed to alleviate the dimensional bottleneck and we analyze the improvement of accuracy in Sect.4.2.

Comparison with existing works. The most significant difference from existing works is the way of deciding which dimension to upload. In the flat solution, each client randomly samples and perturbs c dimensions from d. the perturbed value is enlarged by d/c for an unbiased mean estimation. Thus, the injected noise is also amplified. For the compressed solution, d dimensions of the gradient g is reduced to q dimensions by multiplying the vector g with a public random matrix $\Phi_{d\times q}$ drawn from the Gaussian distribution with mean 0 and variance $1/q$. Then an index is randomly sampled from q dimensions with its value perturbed. The estimated mean of compressed vector is then approximately recovered with the pseudo-inverse of Φ. Even if the random projection has the strict isometry property, it ignores the meaning of the gradient magnitude and brings recovery error. Our framework differs the compressed solution because we utilize the magnitude property for gradient value.

3.2 Privacy Guarantee

In Algorithm 1, both the local accumulated vector r and the vector with momentum s are true local vectors that are directly calculated from the private data. So we abuse $v \in \mathbb{R}^d$ to denote them in the following statement. As previously defined by Shokri et al. [23], there are two sources of information that we intend to preserve for the local vector: (i) how a dimension is selected and (ii) the value of the selected dimension. Let $z = \{0,1\}^d$ indicate the ground-truth Top-k status. We decompose the protection goal into following privacy definitions for dimension selection and value perturbation. When combining the two stages together, we can provide an ϵ-LDP guarantee with Theorem 1.

Definition 2 (LDP Dimension Selection). *A randomized dimension selection algorithm $\mathcal{M}_1$ satisfies ϵ_1-LDP if and only if for any two status vectors $z, z' \in \{0,1\}^d$ and any output $j \in [d]$: $Pr[\mathcal{M}_1(z) = j] \leq e^{\epsilon_1} \cdot Pr[\mathcal{M}_1(z') = j]$.*

Definition 3 (LDP Value Perturbation). *A randomized value perturbation algorithm $\mathcal{M}_2$ satisfies ϵ_2-LDP if and only if for any two numeric values v_j, v'_j and any output v_j^*: $Pr[\mathcal{M}_2(v_j) = v_j^*] \leq e^{\epsilon_2} \cdot Pr[\mathcal{M}_2(v'_j) = v_j^*]$.*

Theorem 1. *For a true local vector v, if the two-stage mechanism $\mathcal{M}$ first selects a dimension index j with $\mathcal{M}_1$ and then perturbs v_j with $\mathcal{M}_2$ under ϵ_1-LDP and ϵ_2-LDP respectively, $\mathcal{M}$ satisfies $(\epsilon_1 + \epsilon_2)$-LDP.*

proof. Theorem 1 stands only when for any two possible local vectors v, v', the conditional probabilities for $\mathcal{M}$ to give the same output v^* satisfy the following condition: $Pr[v^*|v] \leq e^{\epsilon_1+\epsilon_2} Pr[v^*|v']$. Let z, z' denote selection status vectors of v, v'. As we are considering the case where v, v' have the same output v^*, we end the proof with:

$$\frac{Pr[v^*|v]}{Pr[v^*|v']} = \frac{Pr[z|v]Pr[j|z]Pr[v_j^*|v_j]}{Pr[z'|v']Pr[j|z']Pr[v_j^*|v'_j]} \leq \frac{Pr[z|v]}{Pr[z'|v']} \cdot e^{\epsilon_1} e^{\epsilon_2} = e^{(\epsilon_1+\epsilon_2)}.$$

3.3 Variance Analysis for Accumulation

In the existing non-private gradients accumulation [19,21,24,25], local vectors are accumulated as $r^t = r^{t-1} + \alpha g^t$. We adapt it to $r^t = r^{t-1} + g^t$ and scale it with the learning rate in line 11. This aims to reduce the variance of noisy local updates and stabilize the iteration. Suppose the j^{th} dimension of a gradient is selected after T rounds of delay, and $s_{i,j}$ denotes the gradient value. In each iteration of Algorithm 1, the update for the global parameter w_j from user u_i is denoted as $\triangle_T w_{i,j} = \frac{\alpha}{m} s^*_{i,j}$. If αg is accumulated [19], the update is $\triangle'_T w_{i,j} = \frac{1}{m}(\alpha s_{i,j})^*$.

For LDP mechanisms of mean estimation, the input range is $[-1, 1]$. Thus all inputs for a perturbation mechanism should be clipped to the defined input range. When the value for the selected dimension is in the input range, we can easily have: $Var[\triangle_T w_{i,j}] = Var[\triangle'_T w_{i,j}]$. It should be noted that our selected gradient value has significantly larger magnitude. When initial values of both cases are within the input domain, we define the clipped input as ξ. Then we have: $Var[\triangle_T w_{i,j}] = \frac{\alpha^2}{m^2} Var[\xi^*], Var[\triangle'_T w_{i,j}] = \frac{1}{m^2} Var[\xi^*]$. As the learning rate α is typically smaller than 1, such as 0.1, we then have Theorem 2. Therefore, the slight adaption of scaling procedure can reduce the variance of single client's noisy update. The advance of a smaller variance will lead to a more accurate learning performance as validated in the experimental part, Fig.3(a).

Theorem 2. *Accumulating g instead of αg for the dimension j of user u_i has an update on the global model's parameter with a less variance:*

$$Var[\triangle_T w_{i,j}] \leq Var[\triangle'_T w_{i,j}], \text{ where } \alpha \in (0, 1).$$

4 Private Dimension Selection

To instantiate the ϵ_1-LDP dimension selection in Algorithm 1 line 16, we design LDP mechanisms from Top-1 to Top-k case. Then, we analyze the accuracy and time complexity of proposed LDP mechanisms.

4.1 Selection Mechanisms

Exponential Mechanism (EXP): Exponential mechanism [18] is a natural building block for selecting private non-numeric data in the centralized differential privacy. We modify this classic method to meet the selection requirement. A client's accumulated vector r is first sorted in ascending order of its absolute value. As a special case for Top-1 selection, the status vector in EXP is defined as: $z = \{1, \cdots, d\}^d$ instead of a binary vector. Intuitively, the dimension with the largest magnitude of its absolute value should be output with the highest probability. For the j^{th} dimension, we assign the selection status as its ranking z_j. Thus, the index $j \in [d]$ is sampled unevenly with the probability $\frac{\exp(\frac{\epsilon_1 z_j}{d-1})}{\sum_{i=1}^{d} \exp(\frac{\epsilon_1 z_i}{d-1})}$. The privacy guarantee is shown Lemma 1.

Lemma 1. *EXP selection is ϵ_1-locally differentially private.*

proof. Given any two possible ranking vectors as $z, z' \in \{1, \cdots, d\}^d$. j denotes any output index of EXP. The following conditional probability ends the proof:

$$\frac{Pr[j|z]}{Pr[j|z']} = \frac{\exp(\frac{\epsilon_1 z_j}{d-1})}{\sum_{i=1}^{d}\exp(\frac{\epsilon_1 z_i}{d-1})} \Big/ \frac{\exp(\frac{\epsilon_1 z'_j}{d-1})}{\sum_{i=1}^{d}\exp(\frac{\epsilon_1 z'_i}{d-1})} \leq \frac{\exp(\frac{\epsilon_1 \cdot d}{d-1})}{\exp(\frac{\epsilon_1 \cdot 1}{d-1})} = e^{\frac{\epsilon_1 (d-1)}{d-1}} = e^{\epsilon_1}.$$

In order to fit various learning tasks, k should be tunable. Thus, we propose two private Top-k methods to better control the selection.

Perturbed Encoding Mechanism (PE): The sorting step for the vector of absolute values $|r|$ in PE is the same as in EXP. Besides, a binary Top-k status vector z is derived. Then we perturb the vector z with the randomized response. Specifically, each status has a large probability p to retain its value and a small probability $1-p$ to flip. For the privacy guarantee, $p = \frac{e^{\epsilon_1}}{e^{\epsilon_1}+1}$. Let $\acute{z}$ denote the privatized status vector. Since indices of non-zero elements in $\acute{z}$ are more likely to be Top-k dimensions, we gather these elements as the sample set $\mathbb{S}$. If $\mathbb{S}$ is empty, the client uploads $\perp$ and the server regards it as receiving a zero vector $s^* = \{0\}^d$. Elsewise the client randomly samples one dimension index from $\mathbb{S}$. The privacy guarantee is shown in Lemma 2.

Lemma 2. *PE selection is ϵ_1-locally differentially private.*

proof. $\acute{z}$ denotes the perturbed status vector. The expected sparsity of $\acute{z}$ is:

$$l = \mathbb{E}[||\acute{z}||_0] = \sum_{j=1}^{d} Pr[\acute{z}_j = 1|z_j] = \sum_{j=1}^{k} Pr[\acute{z}_j = 1|z_j = 1] + \sum_{j=k+1}^{d} Pr[\acute{z}_j = 1|z_j = 0]$$

$$= k \cdot p + (d-k)\cdot(1-p), \text{ where } p = \frac{e^{\epsilon_1}}{e^{\epsilon_1}+1}.$$

Given any two possible selection status vectors as $z, z' \in \{0,1\}^d$ with k non-zero elements, there are two cases for the output j:(i) If the sample set $\mathbb{S}$ is not empty, $j \in \{1, \cdots, d\}$. (ii) If the sample set $\mathbb{S}$ is empty, $j = \perp$. For the first case, we have:

$$\frac{Pr[j|z]}{Pr[j|z']} = \frac{\frac{1}{l}Pr[\acute{z}_j = 1|z]}{\frac{1}{l}Pr[\acute{z}'_j = 1|z']} \leq \frac{\frac{1}{l}Pr[\acute{z}_j = 1|z_j = 1]}{\frac{1}{l}Pr[\acute{z}'_j = 1|z'_j = 0]} = \frac{e^{\epsilon_1}}{e^{\epsilon_1}+1} \Big/ \frac{1}{e^{\epsilon_1}+1} = e^{\epsilon_1}.$$

For the second case, we end the proof with the conditional probability:

$$\frac{Pr[j=\perp|z]}{Pr[j=\perp|z']} = \frac{k^{1-p}\cdot(d-k)^p}{k^{1-p}\cdot(d-k)^p} = 1 \leq e^{\epsilon_1} \text{ (for } \epsilon_1 \geq 0)$$

Perturbed Sampling Mechanism (PS): PS selection has the same criterion as PE that regards Top-k as important dimensions. Intuitively, we define a higher probability p to sample an index j from the Top-k indices set $\{j \in [d] | z_j = 1\}$ and elsewise, sample an index j from non-Top-k dimensions $\{j \in [d] | z_j = 0\}$ with a smaller probability $1-p$. With the privacy guarantee in Lemma 3, we define $p = \frac{e^{\epsilon_1}\cdot k}{d-k+e^{\epsilon_1}\cdot k}$.

Lemma 3. *PS selection is ϵ_1-locally differentially private.*

proof. Given any two possible Top-k status vector z, z' and the output index $j \in \{1, \cdots, d\}$, the following conditional probability ends the proof:

$$\frac{Pr[j|z]}{Pr[j|z']} \leq \frac{Pr[j|z_j = 1]}{Pr[j|z'_j = 0]} = \frac{p\frac{1}{k}}{(1-p)\frac{1}{d-k}} = e^{\epsilon_1}, \text{where } p = \frac{e^{\epsilon_1}k}{d-k+e^{\epsilon_1}\cdot k}.$$

4.2 Analyses of Accuracy and Time Complexity

We analyze the accuracy improvement of the proposed two-stage framework by evaluating the error bound in Theorem 3 which stands independently of value perturbation algorithms in the second stage. The amount of noise in the average vector is $O(\frac{\sqrt{\log d}}{\epsilon_2\sqrt{md}})$ and the acceptable batch is $|m| = \Omega(\frac{\log d}{d\epsilon_2^2})$ which does not increase linearly with d. Since $\epsilon_2 = \epsilon'(1-\mu)$ and $\epsilon' = \epsilon/E$, it is evident that $\Omega(E^2 \log d/d\epsilon^2) < \Omega(E^2 d \log d/\epsilon^2)$ for $\mu < 0.5$. Hence, we can improve the accuracy while keeping the same private guarante. It also reminds us to allocate a small portion of privacy budget to the dimension selection.

Theorem 3. *For any $j \in [d]$, let $\tilde{s} = \frac{1}{m}\sum_{s^* \in \mathcal{G}} s^*$. $X = \frac{1}{m}\sum_{s^* \in \mathcal{G}} s$ denotes the mean of true sparse vectors without perturbations. With $1-\beta$ probability,*

$$\max_{j \in [1,d]} |\tilde{s}_j - X_j| = O(\frac{\sqrt{\log d/\beta}}{\epsilon_2\sqrt{md}}).$$

Compared with the non-private setting, LDP brings extra computation costs for local devices. For EXP, different utility scores and the summation can be initialized offline. Sorting a $d-$dimensional vector consumes $O(d \log d)$. Mapping all d dimensions to according utility values consumes $O(d^2)$ and sampling requires $O(d)$. Thus, each local device has extra time cost $O(d \log d + d^2 + d) = O(d^2)$ for EXP. With a similar analysis, the extra time cost for PE is $O(d \log d + d + l) = O(d \log d)$ which is less than the time complexity $O(d^2)$ of EXP. Since PS avoids the perturbation for each dimension, it has a slightly less computation cost than PE with the magnitude of $O(d \log d)$. We validate this conclusion in experiments.

5 Experiments

In this section, we assess the performance of our proposed framework on real-world and synthetic datasets. We first evaluate our selection methods without the second stage of value perturbation. To evaluate the improvement of reducing injected noises, we compare the learning performance with the state-of-the-art works [9, 13, 14] and validate theoretical conclusions in Sects.3.3 and 4.2. Moreover, we implement a *hyper-parameters-free* strategy that automatically initiates the budget allocation ratio μ to fit scenarios with dynamic population sizes.

5.1 Experimental Setup

Datasets and Benchmarks. For the convenience to control data sparsity, the synthetic data is generated with the existing procedure [26] with two parameters $C_1 = \{0.01, 0.1\}, C_2 = \{0.6, 0.9\}$ and dimensions {100(syn-L), 300(syn-H)} over {60,000, 100,000} records. The over real-world benchmark datasets

Table 2. Frameworks and Variants for Comparisons.

solution	abbreviation	sparsification	perturbation	budget
non-private	NP	full/random/topk	-	∞
flat [13, 14]	PM/HM/Duchi	random sampling	ϵ'	ϵ'
compressed [9]	-RP	random projection	ϵ'	ϵ'
two-stage	EXP/PE/PS-	$\epsilon_1 = \mu \cdot \epsilon'$	$\epsilon_2 = \epsilon' - \epsilon_1$	ϵ'

includes BANK, ADULT, KDD99 which have {32, 123, 114} dimensions over {45,211, 48,842, 70,000} records. We follow a typical pre-process procedure in machine learning with one-hot-encoding every categorical attribute. We test on $l2$-regularized Logistic Regression and Support Vector Machine.

Choices of Parameters. Since we observe that models on the above datasets can converge within 100 rounds of iterations, we set the batch size for one global model's iteration as $m = 0.01 \cdot N$. We report the average accuracy or misclassification rates of 10 times 5-folds cross-validations for one epoch unless otherwise stated. The discounting factor η and learning rate α are same in each case for a fair comparison. We set $k = 0.1d$, $\mu = 0.1$, $\lambda = 0.0001$ by default.

Comparisons with Competitors. The proposed *FedSel* framework is prefixed with selection mechanisms EXP/PE/PS. We compare it with non-private baselines (NP) of three different transmitting methods (-/RS/K): full gradient, random sampled dimension, Top-k($k = 1$) selection. We also compare with the flat solution [13, 14] and the compressed solution [9] with random sampling and random projection respectively before perturbing the value. Due to limited space and variant baselines, we mainly demonstrate comparisons with the optimal competitor PM and show comparisons with other competitors in Fig.2(i) and Fig.2(h). Abbreviations of different variants are listed in Table 2.

5.2 Evaluation of the Dimension Selection

Convergence and Accuracy. We compare EXP/PE/PS with non-private baselines of NP, NP-RS and NP-K by visualizing the misclassification rate and accuracy of test set in Fig.2(a) to Fig.2(d). Note that this comparison only focuses on selection without the second stage of value perturbation. For NP-RS, we enlarge the value randomly sampled by each user for an unbiased estimation. This follows the same principle in the flat or compressed solution.

The advance of NP-K compared with NP-RS shows our essential motivation that Top-k is a more effective and accurate way to reduce the transmitted dimension. On 100-dimensional dataset, NP-K even approaches the full-gradient-uploading baseline NP in Fig.2(a). Besides, EXP/PE/PS converge more stable and faster than NP-RS in Fig.2(a) and Fig.2(b) with $\epsilon_1 = 4$. With a larger privacy budget, there is a trend for EXP/PE/PS to approach the same accuracy performance as the NP-K($k = 1$) in Fig.2(c) and Fig.2(d). Moreover, even a small budget in dimension selection helps to increase the learning accuracy. We can also observe that PE and PS methods which intuitively intend to select from

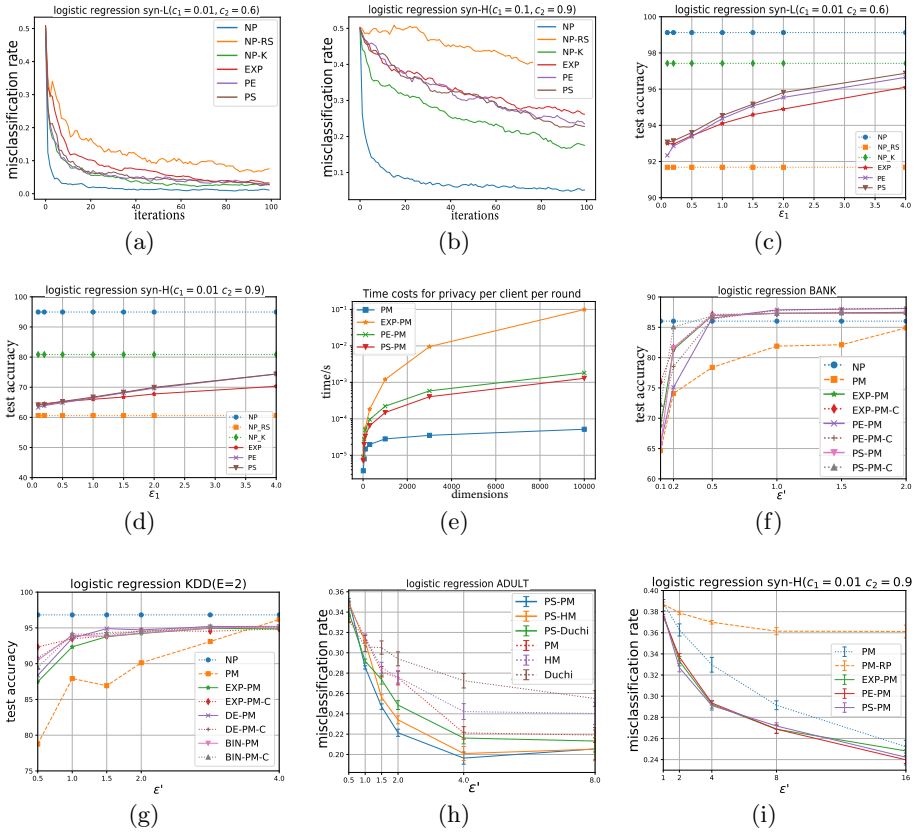


Fig. 2. Improvements of the two-stage framework and dimension selection.

the Top-k list have a better performance than EXP. Thus, our extension from Top-1 to Top-k is necessary.

Validation of Complexity Analysis. Here we analyze the time consumption for each client per transmission in Fig.2(e). The time is counted by iterating over synthetic datasets with variant dimensions from 10 to 10,000. We observe that the selection stage indeed incurs extra computation cost. Consistent with previous analysis in Sect.4.2, PS has the lowest computation cost.

5.3 Effectiveness of the Two-stage Framework

Comparison with Existing Solutions. We compare the learning performance for EXP/PE/PS-PM with the flat solution PM/HM/Duchi [13,14] and the compressed solution [9]. Remark that we set control groups with postfix "-C" in Fig.2(f) and Fig.2(g) by allocating ϵ_1 for dimension selection and ϵ' for value perturbation. To further elucidate the trade-off between what we gain and what

Table 3. Gains and Losses on Accuracy for Private Selection with $\epsilon = 2$ (%).

dataset	model	EXP-gain	EXP-loss	PE-gain	PE-loss	PS-gain	PS-loss
syn-L-0.01-0.9	logistic	**8.6074**	0.3517	**5.410**	1.192	**5.975**	0.4970
syn-L-0.01-0.9	SVM	**7.1950**	2.1593	**3.7704**	0.8533	**5.065**	2.0816
BANK	logistic	**2.4197**	-0.157	**3.2338**	0.0464	**2.5525**	0.1463
BANK	SVM	**4.3823**	0.4436	**3.4369**	0.2530	**4.0244**	0.0164
KDD	logistic	**2.0471**	0.5091	**2.5148**	0.2322	**2.0171**	0.3428
KDD	SVM	**1.85629**	-0.1625	**2.2168**	0.2288	**1.8291**	0.4465
ADULT	logistic	**5.5745**	0.2935	**5.6445**	1.3096	**6.0535**	0.8091
ADULT	SVM	**5.5361**	0.1949	**5.6057**	0.9550	**5.1442**	0.3852

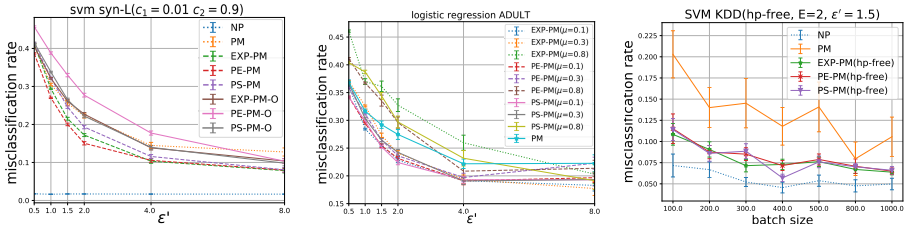
we lose, we qualify the benefit of private selection with the gap between the accuracy as the following, which is shown as Table 3. It is evident that what we gain is much larger than what we lose. This is because when we have enough privacy budget for value perturbation, increasing budget for value perturbation is not comparable to allocating surplus budget to privacy selection.

$$\text{gain} = \text{acc}(\text{EXP/PE/PS-PM-C}) - \text{acc}(\text{PM}),$$
$$\text{loss} = \text{acc}(\text{EXP/PE/PS-PM-C}) - \text{acc}(\text{EXP/PE/PS-PM}).$$

From Fig.2(f) and Fig.2(g), we observe that proposed two-stage solutions have higher test accuracy than the optimal private baseline PM on both models and all datasets. Given enough privacy budget on relatively low-dimensional datasets, proposed solutions even outperform the non-private baseline in Fig.2(f). The key to this success is the inherent randomness in SGD. The slightly introduced stochasticity for privacy-preserving prevents the overfitting problem. In Fig.2(g) with results of two epochs, our adapted local accumulation with momentum helps to reduce the impact of noisy gradients compared with the private baseline PM, especially when ϵ' is small.

From Fig.2(h), we show a comparison with flat solutions of other perturbation methods which have the same optimal error bound as PM. Note that the value perturbation algorithms for each pair of comparison in Fig.2(h) is the same and only differ in the selection stage. It is evident that our framework has a lower misclassification rate and standard deviation over 50 times tests. Therefore, we can conclude that this improvement is independent of value perturbation methods. We omit the comparison of EXP and PE for the same conclusion.

In Fig.2(i), we compare with the compressed solution [9] with the same comparison ratio 0.1. The originally apply random projection in gradients of the Matrix Factorization and use another value perturbation method [12]. Since the dimension reduction idea is independent of value perturbation, for fairness comparison, we adopt the random projection idea and use the same method PM [13] to perturb value when implementing the competitor PM-RP. Our result shows that, even the error bound is reduced by random projection, the recovery error ruins the accuracy while our dimension selection still works.



(a) (b) (c)

Fig. 3. Improvement of the adapted accumulation and impacts of μ

Effectiveness of the Adapted Accumulation. We validate the improvement of stability consistent with our analysis in Sect.3.3 in Fig.3(a). We set the variant of accumulating αg as the competitor (EXP/PE/PS-PM-O). The result shows accumulating αg directly will not improve the learning performance because what we gain by selection is offset by the larger turbulence. Thus, our adaptation is necessary for better compatibility with the private context.

Impacts of μ. As $\mu \in [0, 1]$ controls the privacy budget allocation in our two-stage framework, we evaluate μ in Fig.3(b) with ADULT dataset while the same trends are found in other datasets. From Fig.3(b), we observe that our framework with a small μ works no worse than the flat competitor, even when the total budget is small. In addition, $\mu = 0.1$ gives an optimal learning accuracy, and $\mu = 0.8$ leads to a worse performance with a higher misclassification rate and a significant standard deviation as expected. Thus, μ is an essential parameter that controls the trade-off between what we gain and what we lose.

The divergence of a large μ reminds us that a safe maximum threshold θ is required to guarantee this trade-off always benefits the model's accuracy. It is much easier to tune θ than μ as θ can be tested on synthetic datasets independently of total privacy budget and batch size. At the beginning of training, given the privacy budget per epoch ϵ', the model's dimension d, the available batch size m, our principle is to first allocate at least $\epsilon_2 = \Omega(\sqrt{d \log d / m})$ to the second stage. Then extra privacy budget can be allocated for dimension selection to improve the accuracy as a bonus.

In Fig.3(c), we set a safe threshold as $\theta = 0.2$ empirically and validate the effectiveness of the proposed hyper-parameters-free strategy. Different batch sizes shown in the x-axis simulate the dynamic amount of participants when initiating a distributed learning task in practice. For the fairness to compare the test set accuracy with different batch sizes, we stop the learning process within the same number of iterations. We observe that the proposed solution with three dimension selection methods under this strategy significantly improve the model's accuracy. Besides, the proposed *hyper-parameters-free* strategy works steadily for dynamic batch sizes as the deviations among all 50 times tests are smaller than the private baseline.

6 Related Works

How we select the dimension and accumulate gradients are based on the well-studied gradient sparsification in the non-private setting. Strom et al. [27] propose only to upload dimensions with absolute values larger than a threshold. Instead of a fixed threshold, Aji et al. [25] introduce Gradient Dropping (GD) which sorts absolute values first and dropping a fixed portion of gradients. Wang et al. [26] drop gradients by trading off between the sparsity and variance. Alistarh et al. [21] show the theoretical convergence for Top-k selection. However, even if the gradient update is compressed, there still exist privacy risks because it is calculated directly with local data.

If local gradients are transmitted in clear, the untrusted server threatens clients' privacy. Nasr et al. [6] present the membership inference by only observing uploads or controlling the view of each participant. Wang et al. [8] propose a reconstruction attack in which the server can recover a specific user's local data with Generative adversarial nets (GANs). Secure attack [28] in FL is also an important topic but we focus on private issues in this paper.

Cryptography technologies face a bottleneck of heavy communication and computation costs. Bonawits et al. [29] present an implementation of Secure Aggregation, which entails four rounds interacts per iteration and several costs grow quadratically with the number of users. As for differential privacy(DP) [18] in distributed SGD, Shokri et al. [23] propose a asynchronous cooperative learning with privately selective SGD by sparse vector technique. It may lose accuracy as it drops delayed gradients instead of accumulating as our works. Agarwal et al. [30] combine gradient quantization and differential private mechanisms in synchronous setting, but it requires a higher communication cost for d-dimensional vector. It should be noticed that the privacy definition in the above works differentiate from LDP as it provides the plausible deniability for only single gradient value while LDP guarantees the whole gradient vector to be indistinguishable.

Many LDP techniques are proposed for categorical or numeric values. Randomized response (RR) method [31] is the classic method to perturb binary variables. Kairouz et al. [32] introduce a family of extremal privatization mechanisms k-RR to categorical attributes. With LDP mechanisms for mean estimation, Duchi et al. [14] suggest that LDP leads to an effective degradation in batch size. Wang et al. [13] show that compared with Duchi et al.'s work, their mean estimation mechanisms with lower worst-case variance lead to a lower misclassification rate when applied in SGD. Considering the required batch size is linearly dependent on the dimension, Bhowmick et al. [33] design LDP mechanisms for reconstruction attack with a large magnitude of privacy budget to get rid of the utility limitation of a normal locally differentially private learning.

7 Conclusions

This paper proposes a two-stage LDP privatization framework *FedSel* for federated SGD. The key idea takes the first attempt to mitigate the dimension

problem in injected noises by delaying unimportant gradients. We further stabilize the global iteration by modifying the accumulation with a smaller variance on the noisy update. The improvement of proposed methods is theoretically analyzed and validated in experiments. The framework with *hyper-parameters-free* also outperforms baselines over variant batch sizes. In future work, we plan to formalize the optimal trade-off for utility and accuracy and extend *FedSel* to a more general case.

Acknowledgements. This work is supported by the National Key Research and Development Program of China (No. 2018YFB1004401), National Natural Science Foundation of China (No. 61532021, 61772537, 61772536, 61702522), JSPS KAKENHI Grant No. 17H06099, 18H04093, 19K20269, and Microsoft Research Asia (CORE16).

References

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B. A.: Communication-Efficient Learning of Deep Networks from Decentralized Data. In: Artificial Intelligence and Statistics, pp. 1273-1282. (2017)
2. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Van Overveldt, T.: Towards federated learning at scale: System design. arXiv preprint arXiv:1902.01046. (2019)
3. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19. (2019)
4. McMahan, H. B., Moore, E., Ramage, D., y Arcas, B. A.: Federated learning of deep networks using model averaging. CoRR abs/1602.05629. arXiv preprint arXiv:1602.05629. (2016)
5. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: NeurIPS, pp. 14747-14756. (2019)
6. Nasr, M., Shokri, R., Houmansadr, A. Comprehensive Privacy Analysis of Deep Learning. In: IEEE SP. (2019)
7. Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: SIGSAC CCS, pp. 1322-1333. (2015)
8. Wang, Z., Song, M., Zhang, Z., Song, Y., Wang, Q., Qi, H.: Beyond inferring class representatives: User-level privacy leakage from federated learning. In IEEE INFOCOM, pp. 2512-2520. (2019)
9. Shin, H., Kim, S., Shin, J., Xiao, X.: Privacy enhanced matrix factorization for recommendation with local differential privacy. IEEE TKDE, 30(9), 1770-1782. (2018)
10. Gu, X., Li, M., Cheng, Y., Xiong, L., Cao, Y.: PCKV: Locally Differentially Private Correlated Key-Value Data Collection with Optimized Utility. In: USENIX Security Symposium. (2020).
11. Ye, Q., Hu, H., Meng, X., Zheng, H.: PrivKV: Key-value data collection with local differential privacy. In: IEEE SP, pp. 317-331. (2019)
12. Nguyên, T. T., Xiao, X., Yang, Y., Hui, S. C., Shin, H., Shin, J. (2016). Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053. (2016)

13. Wang, N., Xiao, X., Yang, Y., Zhao, J., Hui, S. C., Shin, H., Yu, G.: Collecting and analyzing multidimensional data with local differential privacy. In: IEEE ICDE, pp. 638-649. (2019)
14. Duchi, J. C., Jordan, M. I., Wainwright, M. J.: Minimax optimal procedures for locally private estimation. Journal of the American Statistical Association, 113(521), 182-201. (2018)
15. Gu, X., Li, M., Cao, Y., Xiong, L.: Supporting both range queries and frequency estimation with local differential privacy. In: IEEE Conference on Communications and Network Security (CNS), pp. 124-132. (2019)
16. Gu, X., Li, M., Xiong, L., Cao, Y.: Providing input-discriminative protection for local differential privacy. In: IEEE ICDE. (2020)
17. Johnson, W. B., Lindenstrauss, J.: Extensions of Lipschitz mappings into a Hilbert space. Contemporary mathematics, 26(189-206), 1. (1984)
18. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3–4), 211-407. (2014)
19. Sun, H., Shao, Y., Jiang, J., Cui, B., Lei, K., Xu, Y., Wang, J.: Sparse gradient compression for distributed SGD. In: DASFAA, pp. 139-155. Springer. (2019)
20. Duchi, J. C., Jordan, M. I., Wainwright, M. J.: Local privacy and statistical minimax rates. In: Annual Symposium on Foundations of Computer Science, pp. 429-438, IEEE. (2013)
21. Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., Renggli, C.: The convergence of sparsified gradient methods. In: NeurIPS, pp. 5973-5983. (2018)
22. Alistarh, D., Hoefler, T., Johansson, M., Konstantinov, N., Khirirat, S., Renggli, C.: The convergence of sparsified gradient methods. In NeurIPS, pp. 5973-5983. (2018)
23. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: SIGSAC CCS, pp. 1310-1321, ACM. (2015)
24. Lin, Y., Han, S., Mao, H., Wang, Y., Dally, W. J.: Deep gradient compression: Reducing the communication bandwidth for distributed training. In ICLR. (2018)
25. Aji, A. F., Heafield, K.: Sparse communication for distributed gradient descent. In: EMNLP, pages 440–445. (2017)
26. Wangni, J., Wang, J., Liu, J., Zhang, T.: Gradient sparsification for communication-efficient distributed optimization. In: NeurIPS, pp. 1299-1309. (2018)
27. Strom, N.: Scalable distributed DNN training using commodity GPU cloud computing. In: INTERSPEECH. (2015)
28. Fang, M., Cao, X., Jia, J., Gong, N. Z.: Local model poisoning attacks to Byzantine-robust federated learning. In: USENIX Security Symposium. (2020)
29. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Seth, K.: In: SIGSAC CCS, pp. 1175-1191, ACM. (2017)
30. Agarwal, N., Suresh, A. T., Yu, F. X. X., Kumar, S., McMahan, B.: cpSGD: Communication-efficient and differentially-private distributed SGD. In: NeurIPS, pp. 7564-7575. (2018)
31. Warner, S. L.: Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association, 60(309), 63-69. (1965)
32. Kairouz, P., Oh, S., Viswanath, P.: Extremal mechanisms for local differential privacy. In: NeurIPS, pp. 2879-2887. (2014)
33. Bhowmick, A., Duchi, J., Freudiger, J., Kapoor, G., Rogers, R.: Protection against reconstruction and its applications in private federated learning. arXiv preprint arXiv:1812.00984. (2018)