Exploring Private Federated Learning with Laplacian Smoothing

Zhicong Liang
Hong Kong University of Science and Technology
zliangak@connect.ust.hk

Bao Wang
University of California, Los Angeles
wangbao@math.ucla.edu

Quanquan Gu
University of California, Los Angeles
qgu@cs.ucla.dot.edu

Stanley Osher
University of California, Los Angeles
sjo@math.ucla.edu

Yuan Yao
Hong Kong University of Science and Technology
yuany@ust.hk

Abstract

Federated learning aims to protect data privacy by collaboratively learning a model without sharing private data among users. However, an adversary may still be able to infer the private training data by attacking the released model. Differential privacy(DP) provides a statistical guarantee against such attacks, at a privacy of possibly degenerating the accuracy or utility of the trained models. In this paper, we apply a utility enhancement scheme based on Laplacian smoothing for differentially-private federated learning (DP-Fed-LS), where the parameter aggregation with injected Gaussian noise is improved in statistical precision. We provide tight closed-form privacy bounds for both uniform and Poisson subsampling and derive corresponding DP guarantees for differential private federated learning, with or without Laplacian smoothing. Experiments over MNIST, SVHN and Shakespeare datasets show that the proposed method can improve model accuracy with DP-guarantee under both subsampling mechanisms.

1 Introduction

In recent years, we have already witnessed machine learning's great success in handling large-scale and high-dimension data [19, 41, 10, 37]. Most of these models are trained on centralized manner by gathering all data into a single database. However, in fields like medical or financial research, sensitive data are collected by different parties, like hospitals or banks, who are not willing to share their own data with others. Firstly proposed in [25], federated learning provides such a solution that data owners can collaboratively learn a useful ML model without disclosing their private data [25, 22, 39, 38]. In federated learning, multiple data owners, referred as clients, and a server are involved. In each communication round, the server will distribute the latest global model to a random subset of selected clients (active clients), who will perform learning starting with the received global model based on their private data, and then upload the locally updated models back to the server. The server then aggregates these local models into a new global model and start another communication round until convergence.

However, it is still not sufficient to protect the sensitive data by simply decoupling the model training from the need for direct access to the raw training data [25], whose information will be revealed by

Preprint. Under review.

the well-trained model. An adversary may infer the presence of particular records during training [36] or even recover the face images in the training set [16, 17] by attacking the released model. Differential privacy (DP) provides us a solution against the threats above [14, 11]. Differential privacy guarantees privacy in a statistical way that the well-trained models are not sensitive to change of an individual record in the training set. This task is usually fulfilled by adding noise to the outputs or updates calibrated to the sensitivity of the model.

One major issue of differential privacy lies in its possibly significant degeneration of the utility of trained models. Laplacian smoothing (LS) was recently shown to be a good choice for variance reduction and escaping spurious minima in stochastic gradient descent (SGD) [31], and is thus promising for utility improvement in differentially private learning [42].

In this paper, we investigate the differentially-private federated learning (DP-Fed) and charaterize its privacy budget. Then we utilize Laplacian smoothing (DP-Fed-LS) to enhance the utility of training while keeping the differential privacy. The major contributions of our works are:

- We provide tight closed-form privacy bounds for both uniform and Poisson subsampling, which relax the requirements of previous works [43, 6, 29]. Based on these results, we derive DP-guarantee for differential private federated learning, with or without Laplacian smoothing.
- We demonstrate the efficiency of Laplacian smoothing in DP-Fed by training a logistic regression over MNIST, a CNN over extended SVHN, in an **IID** fashion, and we also train a LSTM over Shakespeare dataset in **Non-IID** setting. These experiments show that DP-Fed-LS provides better utility than DP-Fed under different DP-guarantees and subsampling mechanisms.

2 Related Work

One research topic on federated learning focuses on its security and data privacy. Generally speaking, the attackers are in the lead by far. Federated learning increases the risk of privacy leakage by unintentionally allowing malicious clients to participate in the training. Hitaj et al. [20] shows that an adversary client can train a GAN to generate prototypical samples of the private training data owned by other users, and deceive the victims to reveal more information. Melis et al. [27] demonstrate that a curious client may infer the presence of exact data point, or some unintended properties of other clients' data through gradient exchange. Zhu et al. [49] show that it is possible to obtain the private training data from the publicly shared gradients by optimization in distributed learning system. Model poissoning attacks are also introduced in [3, 5]. Even though we can ensure the training is private, the released model may also leak sensitive information about data. Fredrikson et al. [17, 16] introduce the model inversion attack that can infer sensitive features or even recover the input given a model. Membership inference attack can determine whether a record is in the training set by utilizing the ubiquitous overfitting of machine learning models [36, 46, 35].

Simply decoupling the training from direct access to private data is not enough. Dwork et al. [14, 12] consider output perturbation with noise whose standard deviation is calibrated according to the sensitivity of the function. Chaudhuri et al. [8, 9] apply the output perturbation to empirical risk minimization (ERM) and propose objective perturbation. Gradient perturbation [4, 1] receives lots of attention in machine learning applications nowadays since it admits public training process and ensure differential privacy guarantee even for non-convex objective [47]. Wang et al. [45] reveals some intricate relationship between learnability, stability and privacy about ERM. Wang et al. [43] further study privcay-preserving nonconvex ERM and extend it to multi-party computation. Feldman et al. [15] argue that one can amplifies the privacy guarantee by not releasing the intermediate results of contractive iterations. Papernot et al. [32, 33] propose PATE that bridges the target model and training data by multiple teacher models. Mironov [28] proposes a natural relaxation of differential privacy based on Rényi divergence (RDP), which allows tighter analysis of composite heterogeneous mechanisms. Wang et al. [44] provide a tight upper bound on RDP parameters for algorithms that apply a randomized mechanism with uniform subsampling. Furthermore, they extend their bound to the case of Poisson subsampling [50], which derives the same numerical bound as the one in [29].

Differential privacy has already been applied in many distributed learning scenarios. Pathak et al. [34] propose the first differentially-private training protocol in distributed setting. Jayaraman et al. [21]

Algorithm 1 Differentially-Private Federated Learning with Laplacian Smoothing (DP-Fed-LS)

parameters:
 activate client fraction $\tau \in (0, 1]$
 total communication round T
 sensitivity parameter G
 noise level ν

function ClientUpdate(j, w^{t-1})
 $\mathcal{B} \leftarrow$ (split dataset $\mathcal{S}_j$ into batches of size B)
 for i=1 **to** local epoch E **do**
  for $b \in \mathcal{B}$ **do**
   ${w_j}^t \leftarrow {w_j}^t - \eta_t \cdot \frac{1}{B}\sum_{k\in b}\nabla\ell({w_j}^t; b_k)$
   ${w_j}^t \leftarrow w^{t-1} + \text{Clip}\big(w_j^t - w^{t-1}\big)$
  end for
 end for
 return $\Delta_j^t \leftarrow {w_j}^t - w^{t-1}$

function Clip(v, G)
 return $v/\max(1, \|v\|_2/G)$

Server executes:
 initialize w^0
 for $t = 1$ **to** T **do**
  $M_t \leftarrow$ (random subset of m clients selected by uniform or Possion subsampling with ratio τ)
  for client $j \in M_t$ **in parallel do**
   $\Delta_j^t \leftarrow \text{ClientUpdate}(j, w^{t-1})$
  end for
  $\Delta_t \leftarrow \frac{1}{m}\mathbf{A}_\sigma^{-1}(\sum_{j=1}^m \Delta_j^t + \mathcal{N}(\mathbf{0}, \nu^2\mathbf{I}))$
  $w^t \leftarrow w^{t-1} + \Delta_t$
 end for
 Output w^T

reduce the noise needed in [34] by a factor of m by adding the noise inside the secure computation after aggregation . Zhang et al. [48] propose to decouple the feature extraction from the training process, where clients only need to extract features with frozen pre-trained convolutional layers and perturb them with Laplacian noise. However, this method need to introduce extra edge servers besides the center server in standard federated learning. Agarwal et al. [2] take both communication efficiency and privacy into consideration. They derive a new Binomial Mechanism to accommodate to their gradient quantization for communication efficiency. Truex et al. [40] argue that leveraging secure multiparty computation (SMC) can help reduce the noise needed by differential privacy, and they introduce a tunable trust parameter which accounts for various trust scenarios. Geyer et al. [18] and McMahan et al. [26] consider the similar problem setting as this paper, that is applying Gaussian Mechanism in federated learning to ensure differential privacy. Mcmahan el al. [24] use moment accountant in [29, 24], which is a strengthened version of the one in [1] through the notation of Rényi differential privacy (RDP).

3 Preliminaries

In this section, we formulate the basic scheme of private (noisy) federated learning and set the notations for the rest of this paper. Given K clients and the latest model w^{t-1}, the server will randomly select a subset of active clients with or without replacement with subsampling ratio τ to participate in the t-th communication round. In such a communication round, the selected clients will receive the global model w^{t-1}, then perform mini-batch SGD on its own data with a batch size of B for E epochs, and send back the new locally-trained models w_j^ts to the sever. The sever will aggregate them into the latest global model w^t, and start the next communication round. We call that a setting is **IID** if data of each client are sampled from the same distribution, otherwise we call it **Non-IID**.

In each update of the mini-batch SGD, we bound to local model w_j^t within the G-ball centering around w^{t-1} by clipping: clip(v) $\leftarrow v/\max(1, \|v\|_2/G)$, where G is the sensitivity of the update. In each server update, we regards the aggregation of locally-trained models as *gradient*, where we add calibrated Gaussian noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \nu^2\mathbf{I})$ to induce the differetnial privacy. We denote this algorithm as DP-Fed. Furthermore, we apply Laplacian smoothing with smoothing factor σ on the noisy aggregated *gradient*, to stabilize the training while preserving the differential privacy by post-processing theorem. We denote this algorithm as DP-Fed-LS. The detailed implementation of DP-Fed-LS is summarized in Algorithm 1.

4 Methodology

Here we introduce our main methodology, *Private Federated Learning with Laplaician Smoothing (DP-Fed-LS)*. Consider the following stochastic optimization process

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \mathbf{A}_\sigma^{-1} \nabla f_i(\mathbf{w}^k), \tag{1}$$

where $f_i(\mathbf{w}) \doteq f(\mathbf{w}, \mathbf{x}_i, y_i)$ is the loss of a given ML model on the training data $\{\mathbf{x}_i, y_i\}$, η is the learning rate, and i is a random sample from $[n]$. In Laplacian smoothing [31], we let $\mathbf{A}_\sigma = \mathbf{I} + \sigma \mathbf{L}$ where $\mathbf{L} \in \mathbb{B}^{\mathbf{d}\times\mathbf{d}}$ is a 1-dimensional chain graph Laplacian matrix, i.e. a symmetric matrix $\mathbf{A}_\sigma$ whose diagonal elements $\mathbf{A}_\sigma(i,i) = 1 + 2\sigma$, off diagonal $\mathbf{A}_\sigma(i, i+1) = -\sigma$ $(i \neq j)$, and otherwise 0, for some constant $\sigma \geq 0$.

When $\sigma = 0$, Laplacian smoothing gradient descent reduces to SGD. The motivation behind this Laplacian smoothing lies in that when the target parameter $\tilde{v}$ is contaminated by Gaussian noise,

$$\tilde{v} = v + \mathbf{n}, \quad v \in R^d, \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \nu^2 \mathbf{I}),$$

a smooth approximation of $\tilde{v}$ is helpful to reduce the noise. The Laplacian smoothed estimate

$$\hat{v}_{LS} := \arg\min_u \|u - \tilde{v}\|^2 + \sigma \|\nabla u\|^2,$$

where ∇ is a 1-dimensional gradient operator, satisfies the following linear equation

$$\mathbf{A}_\sigma \hat{v}_{LS} = \tilde{v} = v + \mathbf{n}.$$

The following proposition characterizes the prediction error of Laplacian smoothed estimate $\hat{v}_{LS}$.

Proposition 1. *Let the graph Laplacian have eigen-decomposition $\Delta \mathbf{e}_i = \lambda_i \mathbf{e}_i$ with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_d$ and the first eigenvector $\mathbf{e_1} = \mathbf{1}/\sqrt{d}$. Then the risk of estimate $\hat{v}_{LS}$ admits the following decomposition,*

$$\begin{aligned}\mathbf{E}\|\hat{v}_{LS} - v\|^2 &= \|(I - \mathbf{A}_\sigma^\dagger) v\|^2 + \mathbf{E}\|\mathbf{A}_\sigma^\dagger \mathbf{n}\|^2 \\ &= \sum_i \frac{\sigma^2 \lambda_i^2}{(1+\sigma\lambda_i)^2} \langle v, \mathbf{e}_i \rangle^2 + \sum_i \frac{\nu^2}{(1+\sigma\lambda_i)^2},\end{aligned}$$

where the first part is called the **bias** *and the second part is called the* **variance***.*

In the bias-variance decomposition of the risk above, if $\sigma = 0$, the risk becomes bias-free with variance $d\nu^2$; if $\sigma > 0$, bias is introduced while variance is reduced. The optimal choice of σ must depend on an optimal trade-off between the bias and variance in this case. When the true parameter v is smooth, in the sense that its projections $\langle v, \mathbf{e}_i \rangle \to 0$ rapidly as i increases, the introduction of bias can be much smaller compared to the reduction of variance, hence the total risk can be reduced with Laplacian smoothing. In Figure 1, we demonstrate an example where Laplacian smoothing reaches improved estimates of smooth signals (parameters) against Gaussian noise.

Among a variety of usages such that reducing the variance of SGD on-the-fly, escaping spurious minima, and improving generalization in training many machine learning models including neural networks, the Laplacian smoothing in this paper improves the utility when Gaussian noise is injected to federated learning for privacy.

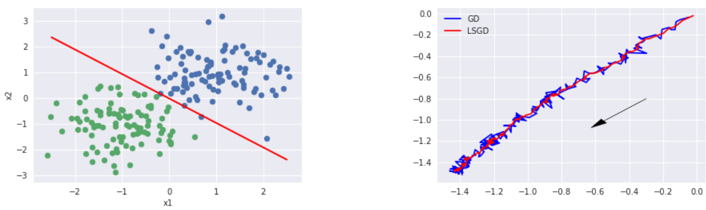
Computationally, we use the fast Fourier transform (FFT) to perform gradient smoothing in the following way

$$\mathbf{A}_\sigma^{-1} \mathbf{v} = \mathrm{ifft}\Big(\frac{\mathrm{fft}(\mathbf{v})}{\mathbf{1} - \sigma \cdot \mathrm{fft}(\mathbf{d})}\Big), \tag{2}$$

where $\mathbf{v}$ is any stochastic gradient vector and $\mathbf{d} = [-2, 1, 0, ..., 0, 1]^T$.

5 Differential Privacy Guarantees

In this section, we provide closed-form differential privacy guarantees for DP-Fed-LS, under both scenarios that activate clients are sampled with uniform subsampling or with Poisson subsampling. Let us recall the definition of (Rényi) differential privacy.



(a) (b)

Figure 1: Demonstration of Laplacian smoothing. We try to use a linear classifier $y = \text{sigmoid}(Wx)$ to separate data points from two distributions, i.e., the blue points ($y = 0$) and the green points ($y = 1$) in (a). We use gradient descent (GD) and Laplacian smoothing gradient descent (LSGD with $\sigma = 1$) with binary cross entropy loss to fulfill this task. Here W is intialized as (0,0) and its perfect solution would be (c,c) for any $c < 0$. Gaussian noise with standard deviation of 0.3 is added on the gradients. Learning rate is set to be 0.1. In (b), we plot the evolution curves of W in 100 updates, where we can find that the curve of LSGD is much smoother than the one of GD.

Definition 1 ((ϵ,δ)-DP [13])**.** *A randomized mechanism $\mathcal{M} : \mathcal{S}^N \to \mathcal{R}$ satisfies (ϵ,δ)-differential privacy if for any two adjacent data sets $S, S' \in \mathcal{S}^N$ differing by only one element, and any output subset $O \subseteq \mathcal{R}$, it holds that*

$$\mathbb{P}[\mathcal{M}(S) \in O] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(S') \in O] + \delta$$

.

Definition 2 (RDP [28])**.** *For $\alpha > 1$ and $\rho > 0$, a randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies (α, ρ)-Rényi differential privacy, i.e., (α, ρ)-RDP, if for all adjacent datasets $S, S' \in \mathcal{S}^n$ differing by one element, we have*

$$D_\alpha\big(\mathcal{M}(S)||\mathcal{M}(S')\big) := \frac{1}{\alpha - 1} \log \left(\frac{\mathcal{M}(S)}{\mathcal{M}(S')}\right)^\alpha \leq \rho,$$

where the expectation is taken over $\mathcal{M}(S')$.

Lemma 1 (From RDP to (ϵ, δ)-DP [28])**.** *If a randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies (α, ρ)-RDP, then $\mathcal{M}$ satisfies $(\rho + \log(1/\delta)/(\alpha - 1), \delta)$-DP for all $\delta \in (0, 1)$.*

Firstly, we consider the case where active clients are selected by uniform subsampling, i.e, in each communication round, a subset of fixed size $m = K \cdot \tau$ of clients are sampled.

Lemma 2 (Uniform Subsampling)**.** *Gaussian mechanism $\mathcal{M} = f(\mathcal{S}) + \mathcal{N}(0, \nu^2)$ applied on a subset of samples that are drawn uniformly without replacement with probability τ satisfies $(\alpha, 3.5\tau^2\alpha/\nu^2)$-RDP given $\nu^2 \geq 0.67$ and $\alpha - 1 \leq \frac{2}{3}\nu^2 \ln\big(1/\alpha\tau(1 + \nu^2)\big)$, where the sensitivity of f is 1.*

Remark 1. *Comparing with the result $(\alpha, 5\tau^2\alpha/\nu^2)$ in [43], and $(\alpha, 6\tau^2\alpha/\nu^2)$ in [6], Lemma 2 provides a tighter bound while relaxing their requirement on ν^2 that $\nu^2 \geq 1.5$ and $\nu^2 \geq 5$ respectively.*

Theorem 1 (Differential Privacy Guarantee For DP-Fed-LS with Uniform Subsampling)**.** *For any $\delta \in (0, 1)$, ϵ, DP-Fed or DP-Fed-LS sampling uniformly without replacement, satisfies (ϵ,δ)-DP when its injected Gaussian noise $\mathcal{N}(0, \nu^2)$ is chosen to be*

$$\nu \geq \frac{\tau G}{\epsilon} \sqrt{\frac{14T}{\lambda}\left(\frac{\log(1/\delta)}{1 - \lambda} + \epsilon\right)}, \tag{3}$$

if there exists $\lambda \in (0, 1)$ such that $\nu^2/4G^2 \geq 0.67$ and $\alpha - 1 \leq \frac{\nu^2}{6G^2} \log(1/(\tau\alpha(1 + \nu^2/4G^2)))$, where $\alpha = \log(1/\delta)/(1 - \lambda)\epsilon + 1$, G is the ℓ_2-bound of clipping map on gradient, $\tau := m/K$ is the subsampling ratio of active clients, T is the total number of communication rounds.

Instead of constructing a subset of active clients of fixed size $m = \tau \cdot K$ uniformly, one can consider Poisson subsampling that includes each clients in the subset with probability τ independently. If we trace back to the definition, this substle difference actually comes from the difference of how we construct the adjacent dataset $\mathcal{S}$ and $\mathcal{S}'$. For uniform subsampling, $\mathcal{S}$ and $\mathcal{S}'$ are adjacent if and only if there exist two samples $a \in \mathcal{S}$ and $b \in \mathcal{S}'$ such that if we replace a in $\mathcal{S}$ with b, then $\mathcal{S}$ is identical with $\mathcal{S}'$ [13]. However, for Poisson subsampling, $\mathcal{S}$ and $\mathcal{S}'$ are said to be adjacent if $\mathcal{S} \cup \{a\}$ or $\mathcal{S}\backslash\{a\}$ is identical to $\mathcal{S}'$ for some sample a [29, 50]. This minor difference actually leads to two different parallel scenarios. The results regarding Poisson subsampling are shown in the following.

Lemma 3 (Poisson Subsampling)**.** *Gaussian mechanism $\mathcal{M} = f(\mathcal{S}) + \mathcal{N}(0, \nu^2)$ applied on a subset of samples that are drawn uniformly without replacement with probability τ satisfies $(\alpha, 2\tau^2\alpha/\nu^2)$-RDP given $\nu^2 \geq 0.53$ and $\alpha - 1 \leq \frac{2}{3}\nu^2 \log\left(1/\alpha\tau(1+\nu^2)\right)$, where the sensitivity of f is 1.*

Remark 2. *Lemma 3's bound equals the boubd in $(\alpha, 2\alpha\tau^2/\nu^2)$-DP in [29]. However, we relax the requirement that $\nu \geq 4$, and simplify multiples requirements over α that $1 < \alpha \leq \frac{\nu^2 L}{2} - 2\ln\nu$ and $\alpha \leq \frac{\nu^2 L^2/2 - \ln 5 - 2\ln\nu}{L + \ln(\tau\alpha) + 1/(2\nu^2)}$. where $L = \ln\left(1 + \frac{1}{\tau(\alpha-1)}\right)$, to one requirement. This makes our closed-form privacy bound below more concise and easily implemented.*

Theorem 2 (Differential Privacy Guarantee For DP-Fed-LS with Poisson Subsampling)**.** *For any $\delta \in (0,1)$, ϵ, DP-Fed or DP-Fed-LS sampling independently with probability τ, satisfies (ϵ,δ)-DP when its injected Gaussian noise $\mathcal{N}(0, \nu^2)$ is chosen to be*

$$\nu \geq \frac{\tau G}{\epsilon}\sqrt{\frac{8T}{\lambda}\left(\frac{\log(1/\delta)}{1-\lambda} + \epsilon\right)}, \tag{4}$$

if there exists $\lambda \in (0,1)$ such that $\nu^2/4G^2 \geq 0.53$ and $\alpha - 1 \leq \frac{\nu^2}{6G^2}\log(1/(\tau\alpha(1+\nu^2/4G^2)))$, where $\alpha = \log(1/\delta)/(1-\lambda)\epsilon + 1$, G is the ℓ_2-bound of clipping map on gradient, $\tau := m/K$ is the subsampling ratio of active clients, T is the total number of communication rounds.

6 Experiments

In this section, we evaluate DP-Fed-LS on three classification tasks. For all three tasks, we compare the utility of DP-Fed-LS ($\sigma > 0$) and DP-Fed ($\sigma = 0$) with varying ϵ in (ϵ, δ)-DP, where $\delta = 1/K^{1.1}$ [26]. These three tasks include training differentially-private federated logistic regression on MNIST dataset [23], CNN on SVHN dataset [30] and LSTM over the Shakespeare dataset [7, 25]. Details about datasets and tasks will be discussed in the corresponding subsections. For logistic regression, we apply the privacy budget in Theorem 1 and 2. For CNN and LSTM models, we apply the moment accountants in [44] and [50, 29] for uniform subsampling and Poisson subsampling, respectively. We report the average loss and average accuracy based on 3 independent runs.

6.1 Logistic Regression

We train a differentially-private federated logistic regression on MNIST dataset [23]. MNIST is a dataset of 28×28 grayscale images of digit from 0 to 9, containing 60K training samples and 10K testing samples. We split 50K training samples into 1000 clients each containing 50 examples in an **IID** fashion [25]. The remaining 10K training samples are left for validation. We set the batch size $B = 10$, local epoch $E = 5$, sensitivity $G = 0.3$, number of communication round $T = 30$, activate client fraction $\tau = 0.05$ and weight decay $\lambda = 4e-5$. We use a initial local learning rate $\eta = 1e-2$ and decay it by a factor of $\gamma = 0.99$ each communication round.

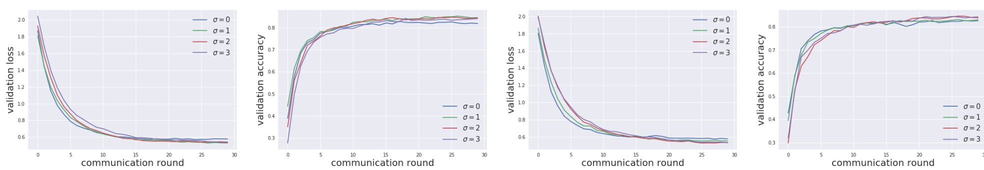
We notice that DP-Fed-LS outperforms DP-Fed in all the settings. And the gap between DP-Fed-LS and DP-Fed is relatively large when the epsilon is small. We notice that DP-Fed-LS converges slower than DP-Fed in both subsampling scenarios. However, DP-Fed-LS will generalize better than DP-Fed at the later stage of training.

6.2 Convolutional Neural Network

In this section, we train a differentially-private federated CNN on the extended SVHN dataset [30]. SVHN is a dataset of 32×32 colored images of digits from 0 to 9, containing 73,257 trainiing samples and 26,032 testing samples. We enlarge the training set with another 531,131 extended samples and split them into 2,000 clients each containing about 300 examples in an **IID** fashion [25]. We also split the testing set by 10K/16K for validation and testing respectively. Our CNNs stacks two 5×5 convolutional layers with max-pooling, two fully-connected layers with 384 and 192 units

Table 1: Testing accuracy of logistic on MNIST with DP-Fed($\sigma = 0$) and DP-Fed-LS($\sigma = 1, 2, 3$) under different $(\epsilon, 1/1000^{1.1})$-DP guarantees and subsampling mechanisms.

Uniform Subsampling					Poisson Subsampling				
ϵ	6	7	8	9	ϵ	6	7	8	9
$\sigma = 0$	78.41	81.85	83.24	84.62	$\sigma = 0$	80.46	83.10	83.60	84.39
$\sigma = 1$	82.44	**85.12**	85.22	84.69	$\sigma = 1$	82.60	84.16	84.32	84.98
$\sigma = 2$	83.33	84.65	**85.31**	85.27	$\sigma = 2$	83.83	**85.15**	**85.35**	**85.25**
$\sigma = 3$	**83.60**	83.53	85.18	**85.35**	$\sigma = 3$	**84.26**	84.29	85.34	85.17



(a) (b) (c) (d)

Figure 2: Training curves of logistic regression on MNIST with DP-Fed($\sigma = 0$), DP-Fed-LS($\sigma = 1, 2, 3$). (a), (b): validation loss and accuracy with uniform subsampling and $(7, 1/1000^{1.1})$-DP (c), (d): validation loss and accuracy with Poisson subsampling and $(7, 1/1000^{1.1})$-DP.

.

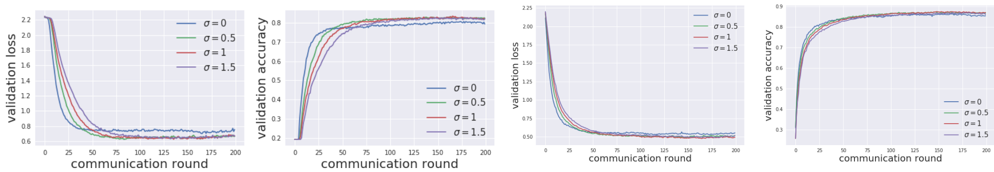
respectively, and a final softmax output layer (about 3.4M parameters in total) [32]. For both the uniform or Poisson subsampling scenarios, we use the same parameter settings. We set the batch size $B = 50$, local epoch $E = 10$, sensitivity $G = 0.7$, number of communication round $T = 200$, activate client fraction $\tau = 0.05$ and weight decay $\lambda = 4e - 5$. Initial learning rate $\eta = 0.1$ and will decay by a factor of $\gamma = 0.99$ each communication round. We vary the privacy budget by setting the noise multiplier z=1, 1.1, 1.3, 1.5.

Table 2: Testing accuracy of CNN on SVHN with DP-Fed($\sigma = 0$) and DP-Fed-LS($\sigma = 0, 5, 1, 1.5$) under different $(\epsilon, 1/2000^{1.1})$-DP guarantees and subsampling mechanisms.

Uniform Subsampling					Poisson Subsampling				
ϵ	5.23	6.34	7.84	8.66	ϵ	2.56	3.19	4.24	5.07
$\sigma = 0.0$	81.40	82.46	85.18	85.84	$\sigma = 0.0$	82.29	83.82	85.53	86.56
$\sigma = 0.5$	82.72	**84.65**	**86.49**	86.32	$\sigma = 0.5$	84.27	**85.47**	**87.00**	**87.50**
$\sigma = 1.0$	**82.39**	84.13	85.88	**86.39**	$\sigma = 1.0$	**84.65**	85.38	86.37	87.26
$\sigma = 1.5$	82.19	83.97	86.03	85.66	$\sigma = 1.5$	84.23	85.12	86.58	87.35

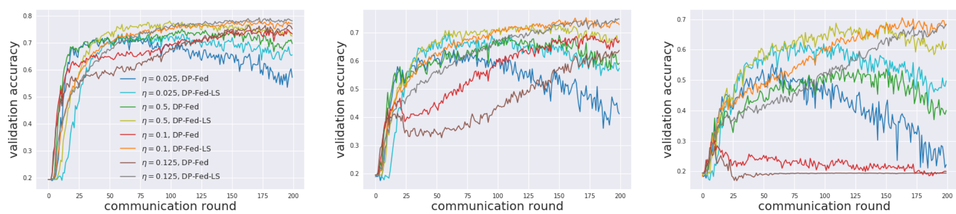
In Table 2, we report the average testing accuracy over 3 independent runs. It demonstrates that DP-Fed-LS yields higher accuracy than DP-Fed with both subsampling mechanisms and different DP guarantees. We show the training curves in Figure 3, which are similar to the ones of logistic regression. The training curves of DP-Fed-LS converges slower than that of DP-Fed, especially when uniform subsampling is used. However, DP-Fed-LS can still provide a better results than DP-Fed at the later stage.

In Figure 4, we show the training curves where relatively large noise multipliers are applied with Possion subsampling and different learning rate. When the noise level is large, the training curves fluctuate a lot. We can observe that, in these extreme cases, DP-Fed-LS outperforms DP-Fed by a large margin. In some cases, for example, when $z = 3$ and $\eta = 0.5$, validation accuracy of DP-Fed start to drop at the 150th epoch while DP-Fed-LS can still converges. When the learning rate increase to 0.125, validation accuracy of DP-Fed drops below 0.2 after the 25th epoch while DP-Fed-LS approach 0.7 at the end. Overall speaking, DP-Fed-LS can suffer large noise level and is less sensitive to learning rate.



(a) (b) (c) (d)

Figure 3: Training curves of CNN on SVHN with DP-Fed($\sigma = 0$), DP-Fed-LS($\sigma = 0.5, 1, 1.5$). (a), (b): validation loss and accuracy with uniform subsampling, where $(5.23, 1/2000^{1.1})$-DP is applied. (c), (d): validation loss and accuracy with Poisson subsampling, where $(5.07, 1/2000^{1.1})$-DP is applied.



(a) (b) (c)

Figure 4: Training curves of CNN on SVHN where large noise levels are applied, with Poisson subsampling and different learning rates η. From left tor right, noise multiplier $z = 2$, 2.5 and 3. For DP-Fed-LS, we set $\sigma = 1$. We can find that DP-Fed-LS is less sensitive to large noise and the change of learning rates than DP-SGD.

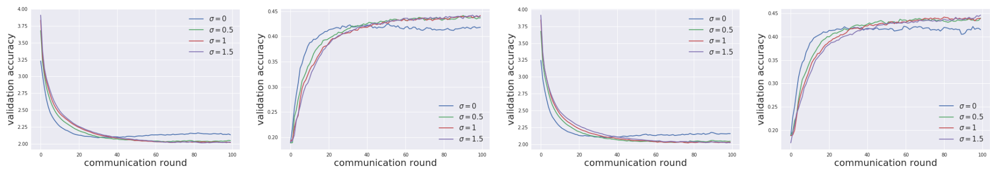
6.3 Long Short Term Memory Network

In this section, we train a differentially-private LSTM on Shakespeare dataset [7, 25]. Shakespeare dataset is built from all the works of William Shakespeare, where each speaking role is consider as a client, whose local database consists of all her/his lines, which will be a **Non-IID** setting. The full dataset contains 1,129 clients and 4,226,158 samples. Here, each sample consists of 80 successive characters and the task is to predict the next character [7, 25]. In our setting, we remove the clients owning less than 64 samples for stabilizing the training, which reduces the total client number to 975. We split the training, validation and testing set chronologically [7, 25], with fractions of 0.7, 0.1, 0.2, respectively. Our LSTM model firstly embeds each input character into a 8 dimensional space, after which two LSTM layers are stacked, each having 256 nodes. The outputs will be then fed into a linear layer, of which the number of output nodes equal the number of distinct characters [25]. In this experiment, we set batch size $B = 50$, local epoch $E = 5$, sensitivity $G = 5$, number of communication round $T = 100$, activate client fraction $\tau = 0.2$ and weight decay $\lambda = 4e - 5$. Initial learning rate $\eta = 1.47$ [25] and will decay by a factor $\gamma = 0.99$ each communication round. We vary the privacy budget by setting the noise multiplier $z = 1, 1.2, 1.4, 1.6$.

Table 3: Testing accuracy of LSTM on Shakespeare with DP-Fed($\sigma = 0$) and DP-Fed-LS($\sigma = 0, 5, 1, 1.5$) under different $(\epsilon, 1/975^{1.1})$-DP guarantees and subsampling mechanisms.

Uniform Subsampling					Poisson Subsampling				
ϵ	14.94	17.69	22.43	27.24	ϵ	6.78	8.22	10.41	14.04
$\sigma = 0.0$	38.22	38.47	39.96	41.87	$\sigma = 0.0$	38.81	39.42	40.19	41.55
$\sigma = 0.5$	39.14	40.27	41.95	43.76	$\sigma = 0.5$	39.07	40.02	42.02	43.59
$\sigma = 1.0$	39.18	**40.94**	**42.60**	43.90	$\sigma = 1.0$	**39.45**	**41.07**	42.09	**43.78**
$\sigma = 1.5$	**40.16**	40.89	42.50	**43.95**	$\sigma = 1.5$	39.38	40.99	**42.19**	43.67

The testing accuracy in Table 3 are comparable to the one in [7]. We can also conclude that DP-Fed-LS provides better utility than DP-Fed. The training curves are plotted in Figure 5. Generally speaking, the training curves in **Non-IID** setting suffer from larger fluctuation than the ones in **IID** setting we show above. And the curves of DP-Fed-LS are smoother than DP-Fed, which further demonstrates the potential of DP-Fed-LS in real-world applications.



(a) (b) (c) (d)

Figure 5: Training curves of LSTM on Shakespeare with DP-Fed($\sigma = 0$), DP-Fed-LS($\sigma = 0.5, 1, 1.5$). (a), (b): validation loss and accuracy with uniform subsampling, where $(27.24, 1/975^{1.1})$-DP is applied. (c), (d): validation loss and accuracy with Poisson subsampling, where $(14.04, 1/975^{1.1})$-DP is applied.

7 Conclusion

In this paper, we introduce DP-Fed-LS and prove tight closed-form privacy guarantees regrading this algorithm under uniform or Poisson subsampling mechanisms. We show by several experiments that DP-Fed-LS outperforms DP-Fed in both **IID** and **Non-IID** settings, which demonstrates its potential in practical applications.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *23rd ACM Conference on Computer and Communications Security (CCS 2016)*, 2016.

[2] 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*, pages 7564–7575, 2018.

[3] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. *ArXiv*, abs/1807.00459, 2018.

[4] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[5] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 634–643, 2019.

[6] Mark Bun, Cynthia Dwork, Guy N Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 74–86, 2018.

[7] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. Leaf: A benchmark for federated settings. *arXiv preprint arXiv:1812.01097*, 2018.

[8] K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. In *Advances in Neural Information Processing Systems (NIPS 2008)*, 2008.

[9] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Noar. Calibrating noise to sensitivity in private data analysis. *TCC*, 2009.

[12] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Noar. Calibrating noise to sensitivity in private data analysis. *TCC*, 2009.

[13] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and trends in Theoretical Computer Science*, 9(3-4), 2014.

[14] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Annual International Cryptology Conference*, pages 528–544. Springer, 2004.

[15] Vitaly Feldman, Ilya Mironov, Kunal Talwar, and Abhradeep Thakurta. Privacy amplification by iteration. *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 521–532, 2018.

[16] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *22nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2015)*, 2015.

[17] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 17–32, 2014.

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

[19] 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*, pages 770–778, 2016.

[20] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 603–618. ACM, 2017.

[21] Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems*, pages 6343–6354, 2018.

[22] 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.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[24] H. Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *NeurIPS 2018 workshop on Privacy Preserving Machine Learnin*, 2018.

[25] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2016.

[26] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *International Conference on Learning Representation*, 2018.

[27] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. *arXiv preprint arXiv:1805.04049*, 2018.

[28] I. Mironov. Rényi differential privacy. In *Computer Security Foundations Symposium (CSF), 2017 IEEE 30th*, pages 263–275. IEEE, 2017.

[29] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv:1908.10530*, 2019.

[30] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[31] S. Osher, B. Wang, P. Yin, X. Luo, M. Pham, and A. Lin. Laplacian smoothing gradient descent. *ArXiv:1806.06317*, 2018.

[32] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *International Conference on Learning Representation*, 2017.

[33] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *International Conference on Learning Representation*, 2018.

[34] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems*, pages 1876–1884, 2010.

[35] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Herve Jegou. White-box vs black-box: Bayes optimal strategies for membership inference. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 5558–5567, 2019.

[36] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. *Proceedings of the 2017 IEEE Symposium on Security and Privacy*, 2017.

[37] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[38] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[39] Ananda Theertha Suresh, Felix X Yu, Sanjiv Kumar, and H Brendan McMahan. Distributed mean estimation with limited communication. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3329–3337. JMLR. org, 2017.

[40] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, and Rui Zhang. A hybrid approach to privacy-preserving federated learning. *Informatik Spektrum*, pages 1 – 2, 2018.

[41] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[42] Bao Wang, Quanquan Gu, March Boedihardjo, Farzin Barekat, and Stanley J Osher. Dp-lssgd: A stochastic optimization method to lift the utility in privacy-preserving erm. *arXiv preprint arXiv:1906.12056*, 2019.

[43] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

[44] Y. Wang, B. Balle, and S. Kasiviswanathan. Subsampled Rényi differential privacy and analytical moments accountant. *arXiv preprint arXiv:1808.00087*, 2018.

[45] Yu-Xiang Wang, Jing Lei, and Stephen E. Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of erm principle. *Journal of Machine Learning Research*, 17(183):1–40, 2016.

[46] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[47] Da Yu, Huishuai Zhang, and Wei Chen. Improve the gradient perturbation approach for differentially private optimization. `https://ppml-workshop.github.io/ppml/ppml18/papers/70.pdf`, 2018.

[48] Jiale Zhang, Junyu Wang, Yanchao Zhao, and Bing Chen. An efficient federated learning scheme with differential privacy in mobile edge computing. In *International Conference on Machine Learning and Intelligent Communications*, pages 538–550. Springer, 2019.

[49] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems*, pages 14747–14756. 2019.

[50] Yuqing Zhu and Yu-Xiang Wang. Poission subsampled rényi differential privacy. In *International Conference on Machine Learning*, pages 7634–7642, 2019.

A Proof of Lemma 2

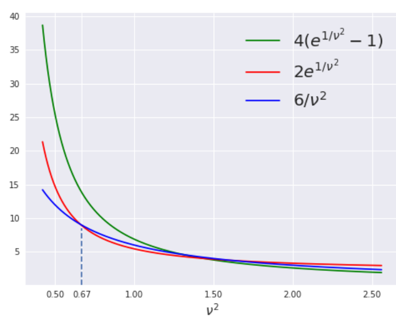
Proof. This proof basically follows the one of Lemma 3.7 of [43], while we relax their requirement and get a tighter bound. According to Theorem 9 in [44], Gaussian mechanism applied on a subset of size $m = \tau \cdot K$, whose samples are drawn uniformly satisfies (α, ρ')-RDP, where

$$\rho'(\alpha) \leq \frac{1}{\alpha - 1} \log \left(1 + \tau^2 \binom{\alpha}{2} \min \left\{ 4(e^{\rho(2)} - 1), 2e^{\rho(2)} \right\} + \sum_{j=3}^{\alpha} \tau^j \binom{\alpha}{j} 2e^{(j-1)\rho(j)} \right)$$

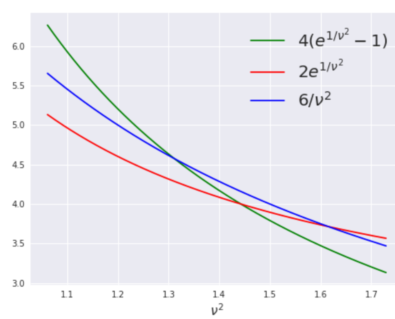
where $\rho(j) = j/2\nu^2$. As mentioned in [44], the dominant part in the summation on the right hand side arises from the term $\min\left\{4(e^{\rho(2)} - 1), 2e^{\rho(2)}\right\}$ when ν^2 is relatively large. We will bound this term as a whole instead of bounding it firstly by $4(e^{\rho(2)} - 1)$ [43]. For $\nu^2 \geq 0.67$, we have

$$\min \left\{ 4(e^{\rho(2)} - 1), 2e^{\rho(2)} \right\} = \min \left\{ 4(e^{1/\nu^2} - 1), 2e^{1/\nu^2} \right\} \leq 6/\nu^2, \tag{5}$$

which can be prove by numerical comparison shown in Figure 6.



(a)



(b)

Figure 6: Numerical comparison of Eq. (5). In (a), we demonstrate the $\min\left\{4(e^{1/\nu^2} - 1), 2e^{1/\nu^2}\right\} \leq 6/\nu^2$ when $\nu^2 \geq 0.67$. In (b), we zoom in the range where $\nu \in [1.1, 1.7]$ of (a).

For the term summing from $j = 3$ to α, we have

$$\begin{aligned}
\sum_{j=3}^{\alpha} \tau^j \binom{\alpha}{j} 2e^{(j-1)\rho(j)} &= \sum_{j=3}^{\alpha} \tau^j \binom{\alpha}{j} 2e^{\frac{(j-1)j}{2\nu^2}} \leq \sum_{j=3}^{\alpha} \tau^j \frac{\alpha^j}{j!} 2e^{\frac{(j-1)j}{2\nu^2}} \\
&\leq \sum_{j=3}^{\alpha} \tau^j \frac{\alpha^j}{3!} 2e^{\frac{(\alpha-1)j}{2\nu^2}} = \tau^2 \frac{\alpha^2}{3} \sum_{j=3}^{\alpha} \tau^{j-2} \alpha^{j-2} e^{\frac{(\alpha-1)j}{2\nu^2}} \\
&\leq \tau^2 \binom{\alpha}{2} \sum_{j=3}^{\alpha} \tau^{j-2} \alpha^{j-2} e^{\frac{(\alpha-1)j}{2\nu^2}} \\
&\leq \tau^2 \binom{\alpha}{2} \frac{\tau \alpha e^{\frac{3(\alpha-1)}{2\nu^2}}}{1 - \tau \alpha e^{\frac{\alpha-1}{2\nu^2}}} \\
&\leq \tau^2 \binom{\alpha}{2} \frac{\tau \alpha e^{\frac{3(\alpha-1)}{2\nu^2}}}{1 - \tau \alpha e^{\frac{3(\alpha-1)}{2\nu^2}}}
\end{aligned} \tag{6}$$

where the first inequality follows from the the fact that $\binom{\alpha}{j} \leq \frac{\alpha^j}{j!}$, and the last inequality follows from the condition that $\tau\alpha\exp{(\alpha-1)}/(2\nu^2) < 1$. In this case, given that

$$\alpha - 1 \leq \frac{2}{3}\nu^2 \ln \frac{1}{\tau\alpha(1+\nu^2)}, \tag{7}$$

we have

$$\sum_{j=3}^{\alpha} \tau^j \binom{\alpha}{j} 2e^{(j-1)\rho(j)} \leq \tau^2 \binom{\alpha}{2} \frac{1}{\nu^2} \tag{8}$$

Combining the results in Eq. (5) and Eq. (8), we have

$$\rho'(\alpha) \leq \frac{1}{\alpha-1} \log\left(1 + \binom{\alpha}{2}\frac{6\tau^2}{\nu^2} + \binom{\alpha}{2}\frac{\tau^2}{\nu^2}\right) \leq \frac{1}{\alpha-1}\tau^2\binom{\alpha}{2}\frac{7}{\nu^2} = 3.5\alpha\tau^2/\nu^2.$$

And conditon $\tau\alpha\exp{(\alpha-1)}/(2\nu^2) < 1$ directly follows from Eq.(7). □

B Proof of Theorem 1

We firstly introduce the notation of ℓ_2-sensitivity and composition theorem of RDP.

Definition 3 (ℓ_2-Sensitivity)**.** *For any given function $f(\cdot)$, the ℓ_2-sensitivity of f is defined by*

$$\Delta(f) = \max_{\|S-S'\|_1=1} \|f(S) - f(S')\|_2,$$

where $\|S - S'\|_1 = 1$ means the data sets S and S' differ in only one entry.

Lemma 4 (Composition Theorem of RDP [28])**.** *If k randomized mechanisms $\mathcal{M}_i : \mathcal{S}^n \to \mathcal{R}$, for $i \in [k]$, satisfy (α, ρ_i)-RDP, then their composition $\big(\mathcal{M}_1(S), \ldots, \mathcal{M}_k(S)\big)$ satisfies $(\alpha, \sum_{i=1}^k \rho_i)$-RDP. Moreover, the input of the i-th mechanism can be based on outputs of the previous $(i-1)$ mechanisms.*

Here we are going to provide privacy upper bound for FedAvg with SGD,

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \frac{1}{m}\left(\sum_{j\in M_t} \mathbf{w}_j^t - m\cdot\mathbf{w}^t + \mathbf{n}\right), \tag{9}$$

and with LSSGD,

$$\tilde{\mathbf{w}}^{t+1} = \tilde{\mathbf{w}}^t + \frac{1}{m}A_\sigma^{-1}\left(\sum_{j\in M_t} \tilde{\mathbf{w}}_j^t - m\cdot\tilde{\mathbf{w}}^t + \mathbf{n}\right), \tag{10}$$

where $\mathbf{n} \sim \mathcal{N}(0, \nu^2 I)$, and $\mathbf{w}_j^t$ is the updated model from client j, based on the previous global model $\mathbf{w}^t$.

Proof. In the following, we will show that the Gaussian noise $\mathcal{N}(0,\nu^2)$ in Eq. (9) for each coordinate of $\mathbf{n}$, the output of DPFed-SGD, $\mathbf{w}$, after T iteration is (ϵ,δ)-DP.

Let us consider the mechanism $\mathcal{M}_t = \frac{1}{m}\sum_{j=1}^K \mathbf{w}_j^t - \mathbf{w}^t + \frac{1}{m}\mathbf{n}$ with the query $\mathbf{q}_t = \frac{1}{m}\sum_{j=1}^K \mathbf{w}_j^t - \mathbf{w}^t$ and its subsampled version $\hat{\mathcal{M}}_t = \frac{1}{m}\sum_{j\in M_t} \mathbf{w}_j^t - \mathbf{w}^t + \frac{1}{m}\mathbf{n}$. Define the query noise $\mathbf{n}_q = \mathbf{n}/m$ whose variance is $\nu_q^2 := \nu^2/m^2$.

We will firstly evaluate the sensitivity of $\mathbf{w}_j^t$. For each local iteration

$$\mathbf{w}_j{}^t \leftarrow \mathbf{w}_j{}^t - \eta_t \cdot \frac{1}{B}\sum_{i\in b} \nabla\ell(\mathbf{w}_j{}^t; b_i)$$
$$\mathbf{w}_j{}^t \leftarrow \mathbf{w}^{t-1} + \mathrm{clip}\big(\mathbf{w}_j^t - \mathbf{w}^{t-1}\big),$$

where $\mathrm{clip}(v) \leftarrow v/\max(1, \|v\|_2/G)$. All the local output $\Delta_j^t \leftarrow \mathbf{w}_j^t - \mathbf{w}^{t-1}$ will be inside the l_2-norm ball centering around $\mathbf{w}^{t-1}$ with radius G. Therefore, after local iterations,

$$\|\mathbf{w}_j^t - \mathbf{w}_j^{t'}\| \leq 2G.$$

We have l_2-sensitivity of $\mathbf{q}_t$ as $\Delta\mathbf{q} = \|\mathbf{w}_j^t - \mathbf{w}_j^{t'}\|_2/m \leq 2G/m$.

According to [28], if we add noise with variance,

$$\nu^2 = m^2\nu_q^2 = \frac{14\tau^2\alpha T G^2}{\lambda\epsilon}, \tag{11}$$

the mechanism $\mathcal{M}_t$ will satisfy $(\alpha, \alpha\Delta^2(\mathbf{q})/(2\nu_q^2)) = (\alpha, \lambda\epsilon/7\tau^2 T)$-RDP. According to Lemma 2, $\hat{\mathcal{M}}_t$ will satisfy $(\alpha,\lambda\epsilon/T)$-RDP provided that $\nu_q^2/\Delta^2(\mathbf{q}) = \nu^2/(m^2\Delta^2(\mathbf{q})) \geq 0.67$ and $\alpha - 1 \leq \frac{2\nu_q^2}{3\Delta^2(\mathbf{q})}\log\left(1/\tau\alpha(1+\nu_q^2/\Delta^2(\mathbf{q}))\right)$. By post-processing theorem, $\tilde{\mathcal{M}}_t = A_\sigma^{-1}\left(\frac{1}{m}\sum_{j\in M_t}\mathbf{w}_j^t - \mathbf{w}^t + \frac{1}{m}\mathbf{n}\right)$ will also satisfy $(\alpha, \lambda\epsilon/T)$-RDP.

Let $\alpha = \log(1/\delta)/(1-\lambda)\epsilon+1$, we obtain that $\hat{\mathcal{M}}_t$ (and $\tilde{\mathcal{M}}_t$) satisfies $(\log(1/\delta)/(1-\lambda)\epsilon+1, \lambda\epsilon/T)$-RDP as long as we have

$$\frac{\nu_q^2}{\Delta^2(\mathbf{q})} = \frac{\nu^2}{m^2\Delta^2(\mathbf{q})} = \frac{\nu^2}{4G^2} \geq 0.67 \tag{12}$$

and

$$\alpha - 1 \leq \frac{\nu^2}{6G^2}\ln\frac{1}{\tau\alpha(1+\nu^2/4G^2)}, \tag{13}$$

Therefore, according to Lemma 4, we have $\mathbf{w}^t$ (and $\tilde{\mathbf{w}}^t$) satisfies $(\log(1/\delta)/(1-\lambda)\epsilon + 1, \lambda t\epsilon/T)$-RDP. Finally, by Lemma 1, we have $\mathbf{w}^t$ (and $\tilde{\mathbf{w}}^t$) satisfies $(\lambda t\epsilon/T + (1-\lambda)\epsilon, \delta)$-DP. Thus, the output of DP-Fed (and DP-Fed-LS), $\mathbf{w}$ (and $\tilde{\mathbf{w}}$), is (ϵ,δ)-DP.

□

C Proof of Lemma 3

Proof. According to [29, 50], Gaussian mechanism applied on a subset where samples are included into the subset with probability ratio τ independently satifies (α, ρ')-RDP, where

$$\rho'(\alpha) \leq \frac{1}{\alpha-1}\log\left((\alpha\tau-\tau+1)(1-\tau)^{\alpha-1}+\binom{\alpha}{2}(1-\tau)^{\alpha-2}\tau^2 e^{\rho(2)}+\sum_{j=3}^{\alpha}\binom{\alpha}{j}(1-\tau)^{\alpha-j}\tau^j e^{(j-1)\rho(j)}\right)$$

where $\rho(j) = j/2\nu^2$.

We notice that, when σ is relatively large, the sum in right-hand side will be dominated by the first two terms. For the first term, we have

$$(\alpha\tau - \tau + 1)(1-\tau)^{\alpha-1} \leq \frac{\alpha\tau - \tau + 1}{1 + (\alpha-1)\tau} = 1, \tag{14}$$

where the first inequality follows from the inequality that

$$(1+x)^n \leq \frac{1}{1-nx} \text{ for } x \in [-1, 0], n \in \mathbb{N}.$$

And for the second term, we have

$$\tau^2\binom{\alpha}{2}(1-\tau)^{\alpha-2}e^{\frac{1}{\nu^2}} \leq \tau^2\binom{\alpha}{2}e^{\frac{1}{\nu^2}} \leq \tau^2\binom{\alpha}{2}\frac{7}{2\nu^2} \tag{15}$$

given that $\nu^2 \geq 0.53$. The last inequality can be proved by numerical comparison like the one we did in the proof of Lemma 2.

And the summation from $j = 3$ to α follows Eq. (8) given that

$$\alpha - 1 \leq \frac{2}{3}\nu^2 \ln \frac{1}{\tau\alpha(1+\nu^2)}. \tag{16}$$

Combining Eq. (14), (15) and (8), we have

$$\rho'(\alpha) \leq \frac{1}{\alpha-1} \log\left(1 + \tau^2 \binom{\alpha}{2} \frac{7}{2\nu^2} + \tau^2 \binom{\alpha}{2} \frac{1}{2\nu^2}\right) \leq \tau^2 \alpha \frac{4}{2\nu^2} = 2\alpha\tau^2/\nu^2. \tag{17}$$

□

D Proof of Theorem 2

Proof. The proof is actually identical to proof of Theorem 1 except that we use Lemma 3 instead of Lemma 2. We start from the Eq. (11) in the proof of Theorem 1. If we add noise with variance

$$\nu^2 = m^2\nu_q^2 = \frac{8\tau^2\alpha T G^2}{\lambda\epsilon}, \tag{18}$$

the mechanism $\mathcal{M}_t$ will satisfy $(\alpha, \alpha\Delta^2(\mathbf{q})/(2\nu_q^2)) = (\alpha, \lambda\epsilon/4\tau^2 T)$-RDP. According to Lemma 3, $\hat{\mathcal{M}}_t$ will satisfy $(\alpha, \lambda\epsilon/T)$-RDP provided that $\nu_q^2/\Delta^2(\mathbf{q}) = \nu^2/(m^2\Delta^2(\mathbf{q})) \geq 0.43$

and

$$\alpha - 1 \leq \frac{\nu^2}{6G^2} \ln \frac{1}{\tau\alpha(1+\nu^2/4G^2)}. \tag{19}$$

By post-processing theorem, $\tilde{\mathcal{M}}_t = A_\sigma^{-1}\big(\frac{1}{m}\sum_{j\in M_t} \mathbf{w}_j^t - \mathbf{w}^t + \frac{1}{m}\mathbf{n}\big)$ will also satisfy $(\alpha, \lambda\epsilon/T)$-RDP. Let $\alpha = \log(1/\delta)/(1-\lambda)\epsilon + 1$, we obtain that $\hat{\mathcal{M}}_t$ (and $\tilde{\mathcal{M}}_t$) satisfies $(\log(1/\delta)/(1-\lambda)\epsilon + 1, \lambda\epsilon/T)$-RDP. Therefore, according to Lemma 4, we have $\mathbf{w}^t$ (and $\tilde{\mathbf{w}}^t$) satisfies $(\log(1/\delta)/(1-\lambda)\epsilon + 1, \lambda t\epsilon/T)$-RDP. Finally, by Lemma 1, we have $\mathbf{w}^t$ (and $\tilde{\mathbf{w}}^t$) satisfies $(\lambda t\epsilon/T + (1-\lambda)\epsilon, \delta)$-DP. Thus, the output of DP-Fed (and DP-Fed-LS), $\mathbf{w}$ (and $\tilde{\mathbf{w}}$), is (ϵ,δ)-DP.

□