DP-LSSGD: A Stochastic Optimization Method to Lift the Utility in Privacy-Preserving ERM

Bao Wang
Department of Mathematics
University of California, Los Angeles
wangbaonj@gmail.com

Quanquan Gu
Department of Computer Science
University of California, Los Angeles
qgu@cs.ucla.edu

March Boedihardjo
Department of Mathematics
University of California, Los Angeles
march@math.ucla.edu

Farzin Barekat
Department of Mathematics
University of California, Los Angeles
fbarekat@math.ucla.edu

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

June 28, 2019

Abstract

Machine learning (ML) models trained by differentially private stochastic gradient descent (DP-SGD) has much lower utility than the non-private ones. To mitigate this degradation, we propose a DP Laplacian smoothing SGD (DP-LSSGD) for privacy-preserving ML. At the core of DP-LSSGD is the Laplace smoothing operator, which smooths out the Gaussian noise vector used in the Gaussian mechanism. Under the same amount of noise used in the Gaussian mechanism, DP-LSSGD attains the same differential privacy guarantee, but a strictly better utility guarantee, excluding an intrinsic term which is usually dominated by the other terms, for convex optimization than DP-SGD by a factor which is much less than one. In practice, DP-LSSGD makes training both convex and nonconvex ML models more efficient and enables the trained models to generalize better. For ResNet20, under the same strong differential privacy guarantee, DP-LSSGD can lift the testing accuracy of the trained private model by more than 8% compared with DP-SGD. The proposed algorithm is simple to implement and the extra computational complexity and memory overhead compared with DP-SGD are negligible. DP-LSSGD is applicable to train a large variety of ML models, including deep neural nets. The code is available at https://github.com/BaoWangMath/DP-LSSGD.

1 Introduction

Many released machine learning (ML) models are trained on sensitive data that are often crowdsourced or contains personal private information [42, 14, 25]. With a large number of parameters, deep neural nets (DNNs) can memorize the sensitive training data, and it is possible to recover the sensitive data and break the privacy by attacking the released models [33]. For example, Fredrikson et al. demonstrated a model-inversion attack can recover training images from a facial recognition system [15]. Protecting the privacy of sensitive training data is one of the most critical tasks in ML.

Differential privacy (DP) [11, 10] is a theoretically rigorous tool for designing algorithms on aggregated databases with a privacy guarantee. The basic idea is to add a certain amount of noise to randomize the output of a given algorithm such that the attackers cannot distinguish outputs of any two adjacent input

datasets that differ in only one entry. Two types of noises are typically injected to the algorithm for DP guarantee: Laplace noise and Gaussian noise [11].

For repeated applications of additive noise based mechanisms, many tools are invented to analyze the DP guarantee for the model obtained at the final stage. These include the basic composition theorem [9, 8], the strong composition theorem and their refinements [13, 23], the momentum-accountant [1], etc. Beyond the original definition of DP, there are also many other ways to define the privacy, e.g., local DP [7], concentrated/zero-concentrated DP [12, 4], and Rényi-DP (RDP) [26].

Differentially private stochastic gradient descent (DP-SGD) reduces the utility of the trained model severely compared with SGD. As shown in Fig. 1, the training and validation loss of the logistic regression increase when the DP guarantee becomes stronger. The ResNet20 trained by DP-SGD has much lower testing accuracy than non-private ResNet20 on the CIFAR10. A natural question is:

Can we improve DP-SGD, with negligible extra computational complexity and memory cost, such that it can be used to train general ML models with better utility?

We answer the above question affirmatively by proposing differentially private Laplacian smoothing SGD (DP-LSSGD). It gives both theoretical and empirical advantages compared with DP-SGD.

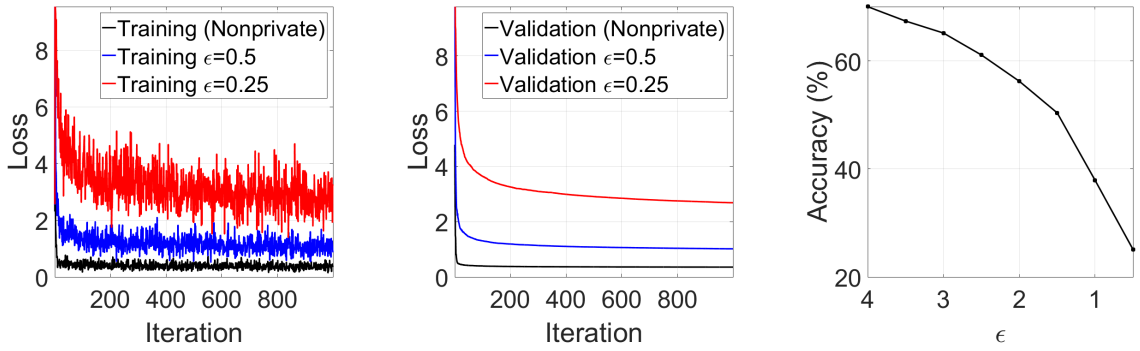

Figure 1: Training (left) and validation (middle) loss of the logistic regression on the MNIST trained by DP-SGD with $(\epsilon, \delta = 2 \times 10^{-5})$-DP guarantee. (right): testing accuracy of ResNet20 on the CIFAR10 trained by DP-SGD with $(\epsilon, \delta = 10^{-5})$-DP guarantee.

1.1 Our Contributions

The main contributions of our work are highlighted as follows:

- We propose DP-LSSGD and prove its privacy and utility guarantees for convex/nonconvex optimizations. We prove that under the same privacy budget, DP-LSSGD achieves better utility, excluding an intrinsic term that usually dominated by the other terms, than DP-SGD by a factor that is much less than one for convex optimization.
- We perform a large number of experiments on logistic regression, SVM, and ResNet to verify the utility improvement by using DP-LSSGD. These results show that DP-LSSGD remarkably reduces training and validation loss and improves the generalization of the trained private models.

In Table 1, we compare the privacy and utility guarantees of DP-LSSGD and DP-SGD. For the utility, the notation $\tilde{O}(\cdot)$ hides the same constant and log factors for each bound. The constants d and n denote the dimension of the model's parameters and the number of training points, respectively. The numbers γ and β are positive constants that are strictly less than one, and D_0, D_σ, G are positive constants, which will be defined later.

1.2 Additional Related Work

There is a massive volume of research over the past decade on designing algorithms for privacy-preserving ML. Objective perturbation, output perturbation, and gradient perturbation are the three major approaches

Table 1: Utility and Privacy Guarantees.

Algorithm	DP Guarantee	Assumption	Utility	Measurement	Reference
DP-SGD	(ϵ, δ)	convex	$\tilde{\mathcal{O}}\left(\sqrt{(D_0 + G^2)d}/(\epsilon n)\right)$	optimality gap	[3]
DP-SGD	(ϵ, δ)	nonconvex	$\tilde{\mathcal{O}}\left(\sqrt{d}/(\epsilon n)\right)$	ℓ_2-norm of gradient	[43]
DP-LSSGD	(ϵ, δ)	convex	$\tilde{\mathcal{O}}\left(\sqrt{\gamma(D_\sigma + G^2)d}/(\epsilon n)\right)$	optimality gap	**This Work**
DP-LSSGD	(ϵ, δ)	nonconvex	$\tilde{\mathcal{O}}\left(\sqrt{\beta d}/(\epsilon n)\right)$ [1]	ℓ_2-norm of gradient	**This Work**

[1] Measured in the norm induced by $\mathbf{A}_\sigma^{-1}$, we will discuss this in detail in Section 4.

to perform empirical risk minimization (ERM) with DP guarantee. We discuss some related works in this part. There are many more exciting works that cannot be discussed here.

Chaudhuri et al. considered both output and objective perturbations for privacy-preserving ERM, and gave theoretical guarantees for both privacy and utility for logistic regression and SVM [5, 6]. Song et al. numerically studied the effects of learning rate and batch size in DP-ERM [34]. Wang et al. studied stability, learnability and other properties of DP-ERM [39]. Lee et al. proposed an adaptive per-iteration privacy budget in concentrated DP gradient descent [24]. Variance reduction techniques, e.g., SVRG, have also been introduced to DP-ERM [37]. The utility bound of DP-SGD has also been analyzed for both convex and nonconvex smooth objectives [3, 43]. Jayaraman et al. analyzed the excess empirical risk of DP-ERM under the distributed setting [21]. Besides ERM, many other ML models have been made differentially private. These include: clustering [35, 41, 2], matrix completion [20], online learning [19], sparse learning [36, 38], and topic modeling [30]. Gilbert et al. exploited the ill-conditionedness of inverse problems to design algorithms to release differentially private measurements of the physical system [17].

Shokri et al. proposed distributed selective SGD to train deep neural nets (DNNs) with a DP guarantee in a distributed system; they achieved quite a successful trade-off between privacy and utility [32]. Abadi et al. considered applying DP-SGD to train DNNs in a centralized setting. They clipped the gradient to bound the sensitivity and invented the momentum accountant to get better privacy loss estimation [1]. Papernot et al. proposed Private Aggregation of Teacher Ensembles/PATE based on the semi-supervised transfer learning to train DNNs and to protect the privacy of the private data [28]. Recently Papernot et al. introduced new noisy aggregation mechanisms for teacher ensembles that enable a tighter theoretical DP guarantee. The modified PATE is scalable to the large dataset and applicable to more diversified ML tasks [29]. Geyer et al. considered general ML with a DP guarantee under federated settings [16]. Rahman et al. numerically studied the vulnerability and privacy-utility trade-off of DNNs trained with a DP guarantee to adversarial attacks [31].

1.3 Notation

We use boldface upper-case letters $\mathbf{A}$, $\mathbf{B}$ to denote matrices and boldface lower-case letters $\boldsymbol{x}$, $\boldsymbol{y}$ to denote vectors. For vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ and positive definite matrix $\mathbf{A}$, we use $\|\boldsymbol{x}\|_2$ and $\|\boldsymbol{x}\|_{\mathbf{A}}$ to denote the ℓ_2-norm and the induced norm by $\mathbf{A}$, respectively; $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denotes the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$; and $\lambda_i(\mathbf{A})$ denotes the i-th largest eigenvalue of $\mathbf{A}$. We denote the set of numbers from 1 to n by $[n]$.

1.4 Organization

This paper is organized in the following way: In Section 2, we introduce the DP-LSSGD algorithm, which merely injects an appropriate Gaussian noise to guarantee the privacy of LSSGD. In Section 3, we analyze the privacy and utility guarantees of DP-LSSGD for both convex and nonconvex optimizations. We numerically verify the efficiency of DP-LSSGD in Section 4. We conclude this work and point out some future directions in Section 5.

Algorithm 1 DP-LSSGD

Input: $f_i(\mathbf{w})$ is G-Lipschitz for $i = 1, 2, \cdots, n$.
$\mathbf{w}^0$: initial guess of $\mathbf{w}$, (ϵ, δ): the privacy budget, η: the step size, T: the total number of iterations.
Output: (ϵ, δ)-differentially private classifier $\mathbf{w}_{\text{priv}}$.
for $k = 0, 1, \cdots, T-1$ **do**
$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \mathbf{A}_\sigma^{-1} \left(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n} \right)$, where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \nu^2 \mathbf{I})$ and ν is defined in Theorem 1.
return $\mathbf{w}^T$

2 Problem Setup and Algorithm

2.1 Laplacian Smoothing Stochastic Gradient Descent (LSSGD)

Consider the following finite-sum optimization

$$\min_{\mathbf{w}} F(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{w}), \tag{1}$$

where $f_i(\mathbf{w}) \doteq f(\mathbf{w}, \boldsymbol{x}_i, y_i)$ is the loss of a given ML model on the training data $\{\boldsymbol{x}_i, y_i\}$. This finite-sum optimization problem is the mathematical formulation for training many ML models that are mentioned above. The LSSGD [27] for solving this finite-sum optimization is given by

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \mathbf{A}_\sigma^{-1} \nabla f_{i_k}(\mathbf{w}^k), \tag{2}$$

where η is the learning rate, and i_k is a random sample from $[n]$. Let $\mathbf{A}_\sigma = \mathbf{I} - \sigma \mathbf{L}$ where $\mathbf{I} \in \mathbb{R}^{d \times d}$ and $\mathbf{L} \in \mathbb{R}^{d \times d}$ are the identity and the discrete one-dimensional Laplacian matrix, respectively. Therefore,

$$\mathbf{A}_\sigma := \begin{bmatrix} 1+2\sigma & -\sigma & 0 & \dots & 0 & -\sigma \\ -\sigma & 1+2\sigma & -\sigma & \dots & 0 & 0 \\ 0 & -\sigma & 1+2\sigma & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -\sigma & 0 & 0 & \dots & -\sigma & 1+2\sigma \end{bmatrix} \tag{3}$$

for $\sigma \geq 0$ being a constant. When $\sigma = 0$, LSSGD reduces to SGD.

This Laplacian smoothing can help to avoid spurious minima, reduce the variance of SGD on-the-fly, and lead to better generalization in training many ML models including DNNs. Computationally, we use the fast Fourier transform (FFT) to perform gradient smoothing in the following way

$$\mathbf{A}_\sigma^{-1} \mathbf{v} = \text{ifft} \left(\frac{\text{fft}(\mathbf{v})}{\mathbf{1} - \sigma \cdot \text{fft}(\mathbf{d})} \right),$$

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

2.2 DP-LSSGD

We propose the following DP-LSSGD algorithm to resolve the finite-sum optimization in Eq. (1)

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta \mathbf{A}_\sigma^{-1} \left(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n} \right), \tag{4}$$

where ∇f_{i_k} denotes the gradient of the total loss function F evaluated from the database $\{\boldsymbol{x}_{i_k}, y_{i_k}\}$ and $\mathbf{n}$ is the injected Gaussian noise. In this scheme, we first add the noise $\mathbf{n}$ to the stochastic gradient vector $\nabla f_{i_k}(\mathbf{w}^k)$, and then apply the operator $\mathbf{A}_\sigma^{-1}$ to smooth the noisy stochastic gradient on-the-fly. We assume that each component function f_i in Eq. (1) is G-Lipschitz. The DP-LSSGD algorithm for finite-sum optimization is summarized in Algorithm 1.

3 Main Theory

In this section, we present the privacy and utility guarantees for DP-LSSGD. The technical proofs are provided in the appendix.

Definition 1 ((ϵ, δ)-DP)**.** *([11]) 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 one element, and any output subset $O \subseteq \mathcal{R}$, it holds that*

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

Theorem 1 (Privacy Guarantee)**.** *Suppose that each component function f_i is G-Lipschitz. Given the total number of iterations T, for any $\delta > 0$ and privacy budget $\epsilon^2 \leq 20T\log(1/\delta)G^2/n^2$, DP-LSSGD, with injected Gaussian noise $\mathcal{N}(0, \nu^2)$ for each coordinate, satisfies (ϵ, δ)-differential privacy with $\nu^2 = 8T\alpha G^2/(n^2\epsilon)$, where $\alpha = 2\log(1/\delta)/\epsilon + 1$.*

Remark 1. *It is straightforward to show that the noise in Theorem 1 is in fact also tight to guarantee the (ϵ, δ)-differential privacy for DP-SGD, since the same amount of Gaussian noise guarantees the same differential privacy for both DP-SGD and DP-LSSGD.*

For convex ERM, DP-LSSGD guarantees the following utility bound in terms of the gap between the ergodic average of the points along the DP-LSSGD path and the optimal solution $\mathbf{w}^*$.

Theorem 2 (Utility Guarantee for convex optimization)**.** *Suppose F is convex and each component function f_i is G-Lipschitz. Given any $\epsilon^2 \leq 20T\log(1/\delta)G^2/n^2$ and $\delta > 0$, if we choose $\eta_k = 1/\sqrt{T}$ and $T = (D_\sigma + G^2)n^2\epsilon^2/\big(24dG^2\log(1/\delta)\big)$, where $D_\sigma = \|\mathbf{w}^0 - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma}$ and $\mathbf{w}^*$ is the global minimizer of F, the DP-LSSGD output $\tilde{\mathbf{w}} = \sum_{k=0}^{T-1}\eta_k/\big(\sum_{i=0}^{T-1}\eta_i\big)\mathbf{w}^k$ satisfies the following utility*

$$\mathbb{E}\big(F(\tilde{\mathbf{w}}) - F(\mathbf{w}^*)\big) \leq \frac{2G\sqrt{6\gamma(D_\sigma + G^2)d\log(1/\delta)}}{n\epsilon},$$

where $\gamma = 1/d\sum_{i=1}^{d} 1/[1 + 2\sigma - 2\sigma\cos(2\pi i/d)]$.

Proposition 1. *In Theorem 2, $\gamma = \frac{1+\alpha^d}{(1-\alpha^d)\sqrt{4\sigma+1}}$, where $\alpha = \frac{2\sigma+1-\sqrt{4\sigma+1}}{2\sigma}$.*

Remark 2. *Compared with the extra utility bound of DP-SGD $\mathcal{O}\left(\frac{G\sqrt{G^2 d\log(1/\delta)}}{(n\epsilon)}\right)$, DP-LSSGD has a strictly better extra utility bound $\mathcal{O}\left(\frac{G\sqrt{\gamma G^2 d\log(1/\delta)}}{n\epsilon}\right)$ by a factor of $\sqrt{\gamma}$, except for the term $\mathcal{O}\left(\frac{G\sqrt{\gamma D_\sigma d\log(1/\delta)}}{n\epsilon}\right)$. In practice, for both logistic regression and SVM, $\mathcal{O}\left(\frac{G\sqrt{\gamma D_\sigma d\log(1/\delta)}}{n\epsilon}\right)$ is dominated by $\mathcal{O}\left(\frac{G\sqrt{\gamma G^2 d\log(1/\delta)}}{n\epsilon}\right)$, and DP-LSSGD improves the utility of both models.*

For nonconvex ERM, DP-LSSGD has the following utility bound measured in gradient norm.

Theorem 3 (Utility Guarantee for nonconvex optimization)**.** *Suppose that F is nonconvex and each component function f_i is G-Lipschitz and has L-Lipschitz continuous gradient. Given any $\epsilon^2 \leq 20T\log(1/\delta)G^2/n^2$ and $\delta > 0$, if we choose $\eta = 1/\sqrt{T}$ and $T = (D_F + L\nu^2)n^2\epsilon^2/\big(12dLG^2\log(1/\delta)\big)$, where $D_F = F(\mathbf{w}^0) - F(\mathbf{w}^*)$ with $\mathbf{w}^*$ being the global minimum of F, then the DP-LSSGD output $\tilde{\mathbf{w}} = \sum_{k=0}^{T-1}\mathbf{w}^k/T$ satisfies the following utility*

$$\mathbb{E}\|\nabla F(\tilde{\mathbf{w}})\|^2_{\mathbf{A}_\sigma^{-1}} \leq 4\frac{G\sqrt{6\beta dL(2D_F + LG^2)\log(1/\delta)}}{n\epsilon},$$

where $\beta = 1/d\sum_{i=1}^{d} 1/[1 + 2\sigma - 2\sigma\cos(2\pi i/d)]^2$.

Proposition 2. *In Theorem 3, $\beta = \frac{2\alpha^{2d+1} - \xi\alpha^{2d} + 2\xi d\alpha^d - 2\alpha + \xi}{\sigma^2\xi^3(1-\alpha^d)^2}$, where $\alpha = \frac{2\sigma+1-\sqrt{4\sigma+1}}{2\sigma}$ and $\xi = -\frac{\sqrt{1+4\sigma}}{\sigma}$.*

The number β is also strictly between 0 and and 1. It is worth noting that if we use the ℓ_2-norm instead of the induced norm, we have the following utility guarantee

$$\mathbb{E}\|\nabla F(\tilde{\mathbf{w}})\|_2^2 \leq \frac{\mathbb{E}\|\nabla F(\tilde{\mathbf{w}})\|_{\mathbf{A}_\sigma^{-1}}^2}{\lambda_{\min}(\mathbf{A}_\sigma^{-1})} \leq (1+4\sigma)\mathbb{E}\|\nabla F(\tilde{\mathbf{w}})\|_{\mathbf{A}_\sigma^{-1}}^2 \leq 4\zeta\frac{G\sqrt{6dL(2D_F+LG^2)\log(1/\delta)}}{n\epsilon}$$

where $\zeta = \sqrt{\frac{1}{d}\sum_{i=1}^{d}\frac{(1+4\sigma)^2}{(1+2\sigma-2\sigma\cos(2\pi i/d))^2}} > 1$. In the ℓ_2-norm, DP-LSSGD has a bigger utility upper bound than DP-SGD (set $\sigma = 0$ in ζ). However, this does not mean that DP-LSSGD has worse performance. To see this point, let us consider the following simple nonconvex function

$$f(x,y) = \begin{cases} \frac{x^2}{4}+y^2, & \text{for } \frac{x^2}{4}+y^2 \leq 1 \\ \sin\left(\frac{\pi}{2}\left(\frac{x^2}{4}+y^2\right)\right), & \text{for } \frac{x^2}{4}+y^2 > 1 \end{cases} \tag{5}$$

For two points $\mathbf{a}_1 = (2,0)$ and $\mathbf{a}_2 = (1, \frac{\sqrt{3}}{2})$, the distance to the local minima $\mathbf{a}^* = (0,0)$ are 2 and $\frac{\sqrt{7}}{2}$, while $\|\nabla f(\mathbf{a}_1)\|_2 = 1$ and $\|\nabla f(\mathbf{a}_2)\|_2 = \frac{\sqrt{13}}{2}$. So $\mathbf{a}_2$ is closer to the local minima $\mathbf{a}^*$ than $\mathbf{a}_1$ while its gradient has a larger ℓ_2-norm. This example shows $\|\nabla F\|_2$ is not the optimal measure in comparing the utility bound for nonconvex optimization. We will further verify this in Section 4.

4 Numerical Results

In this section, we verify the efficiency of the proposed DP-LSSGD in training multi-class logistic regression, SVM, and ResNet20. We perform ResNet20 experiments on the CIFAR10 dataset with standard data augmentation [18], logistic regression and SVM experiments on the benchmark MNIST classification. Based on the range of gradient values of each model, we use the formula $\mathbf{v} \leftarrow \frac{\mathbf{v}}{\max(1, \|\mathbf{v}\|_2/C)}$ [1] to clip the gradient ℓ_2-norms of logistic regression and ResNet20 to 3 and clip the SVM's gradient ℓ_2-norm to 1. These gradient clippings guarantee the Lipschitz condition for the objective functions. For all experiments, we train both logistic regression and SVM with $(\epsilon, 2\times 10^{-5})$-DP guarantee, and ResNet20 with $(\epsilon, 10^{-5})$-DP guarantee. We regard the DP-SGD as the benchmark.

4.1 Multi-class Logistic Regression and SVM

For MNIST classification, we ran 50 epochs of DP-LSSGD with learning rate scheduled as $\frac{1}{t}$ with t being the index of the iteration to train the ℓ_2-regularized multi-class logistic regression and SVM (the objective function of both models are strongly convex), using an ℓ_2 penalty with regularization coefficient $\lambda = 1e-4$. We split the training data into 50K/10K for cross-validation. The models with best validation accuracy are used for testing. The batch size is set to 128.

First, we show that DP-LSSGD converges faster than DP-SGD and makes the training and validation loss much smaller than DP-SGD. We plot the evolution of training and validation loss over iterations for logistic regression (Fig. 2) and SVM (Fig. 3) with DP guarantee. Figures 2 and 3 show that the training loss curve of DP-SGD ($\sigma = 0$) is much higher and more oscillatory (due to the log-scale in y-axis) than that of DP-LSSGD ($\sigma = 1, 3$). The validation loss of both logistic regression and SVM trained by both DP-SGD and DP-LSSGD decrease as iteration goes. The validation loss of the model trained by DP-LSSGD decays faster and has a much smaller loss value than that of the model trained by DP-SGD. For both training and validation, DP-LSSGD with $\sigma = 3$ gives better results than $\sigma = 1$.

Second, consider the validation accuracy of the models trained by DP-SGD and DP-LSSGD. Figure 4 depicts the evolution of the validation accuracy of the trained logistic regression and SVM by DP-SGD and DP-LSSGD. We plot validation accuracy after every training epoch. It shows that DP-LSSGD is almost always better than DP-SGD in the sense that DP-LSSGD gives better validation accuracy. Different σ in DP-LSSGD give different level of improvement. For these experiments, larger σ is usually better than the smaller one.

Third, consider the testing accuracy of logistic regression and SVM trained in different scenarios. The corresponding testing accuracy are listed in Tables. 2, and 3. All the numbers reported in the above tables

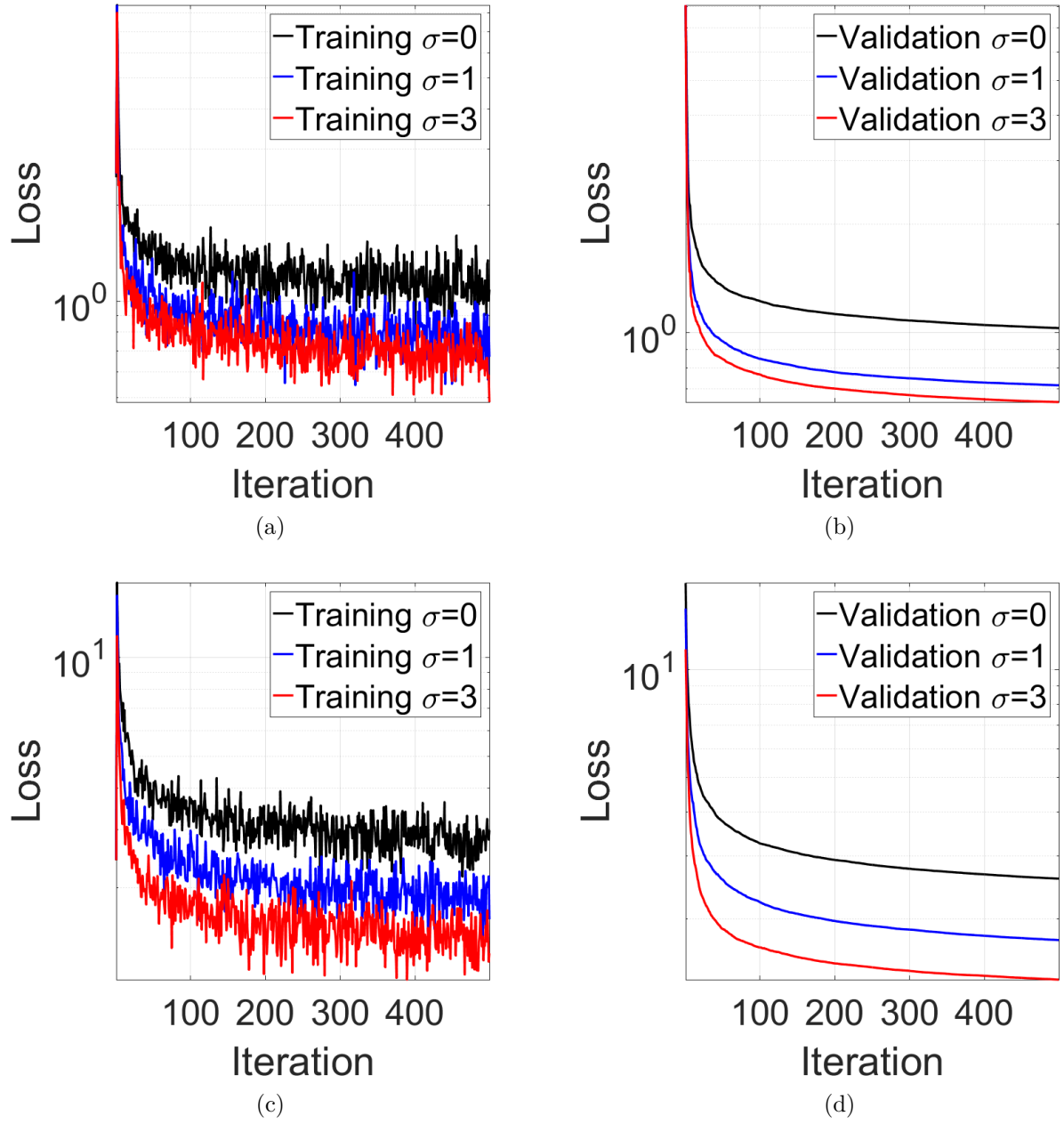

Figure 2: Training and validation losses of the multi-class logistic regression model trained by SGD with different noise injection. (a) and (b): training and validation curves with $(0.5, 2 \times 10^{-5})$-DP guarantee; (c) and (d): training and validation curves with $(0.25, 2 \times 10^{-5})$-DP guarantee.

and the tables below are the results averaged over three independent experiments. These results reveal that the multi-class logistic regression model is remarkably more accurate than SVM for various levels of DP-guarantee. Both logistic regression and SVM trained by DP-LSSGD with $\sigma = 1, 2, 3$ are more accurate than that trained by DP-SGD over different levels of DP-guarantee.

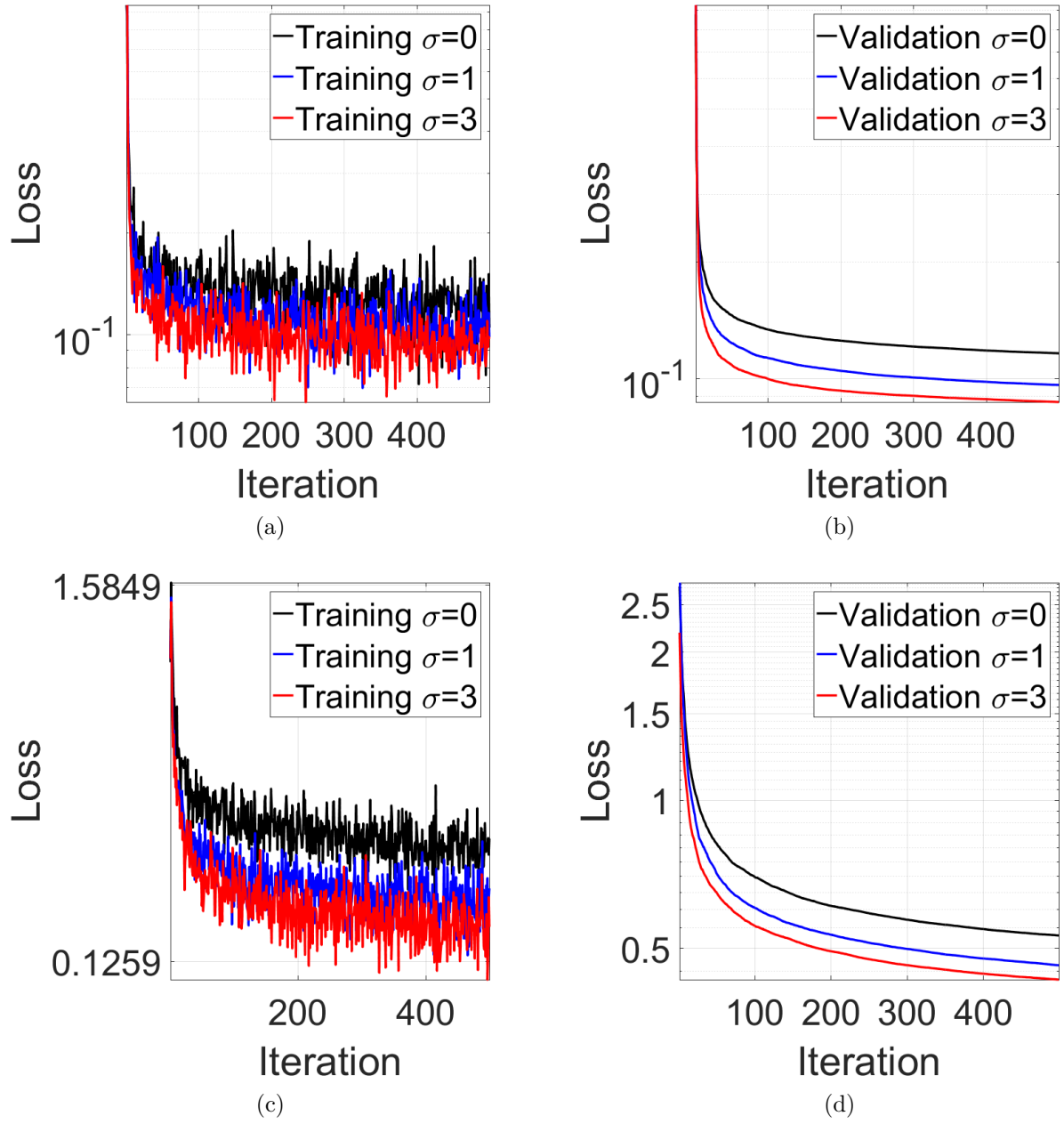

Figure 3: Training and validation losses of the SVM trained by SGD with different noise injection. (a) and (b): training and validation curves with $(0.5, 2 \times 10^{-5})$-DP guarantee; (c) and (d): training and validation curves with $(0.25, 2 \times 10^{-5})$-DP guarantee.

4.1.1 The Choice of σ

Table 5 lists the testing accuracy (averaged over three runs) of both private logistic regression and SVM trained by DP-LSSGD with different σ. It shows that accuracy improvement is stable to σ. As σ increases, the testing accuracy increases initially and then decays. In practice, DP-LSSGD is as fast as DP-SGD, so for a given objective function we might try a few different σ to find the optimal one.

Table 2: Testing accuracy of multi-class logistic regression ($\lambda = 1e-4$) trained by DP-LSSGD with $(\epsilon, \delta = 2 \times 10^{-5})$-DP guarantee and smoothing parameter σ. Unit: %.

ϵ	0.50	0.45	0.40	0.35	0.30	0.25	0.20
$\sigma = 0$	81.59	81.52	80.07	79.30	78.71	77.80	76.02
$\sigma = 1$	83.64	83.70	**82.91**	82.33	**82.25**	79.53	78.01
$\sigma = 2$	**84.41**	83.45	81.88	**83.06**	81.39	79.03	78.86
$\sigma = 3$	84.14	**83.99**	82.17	82.08	81.74	**80.90**	**80.21**

Table 3: Testing accuracy of SVM ($\lambda = 1e-4$) trained by DP-LSSGD with $(\epsilon, \delta = 2 \times 10^{-5})$-DP guarantee and smoothing parameter σ. Unit: %.

ϵ	0.50	0.45	0.40	0.35	0.30	0.25	0.20
$\sigma = 0$	78.28	77.41	76.07	74.09	72.98	72.47	70.25
$\sigma = 1$	80.53	79.53	77.77	77.09	75.37	75.89	72.94
$\sigma = 2$	**81.72**	79.69	**79.59**	**77.99**	**77.09**	**76.19**	73.94
$\sigma = 3$	80.57	**80.11**	78.85	77.44	76.92	75.97	**73.97**

Table 4: Testing accuracy of the ResNet20 trained by DP-LSSGD with different $(\epsilon, \delta = 10^{-5})$-DP guarantee and smoothing parameter σ. Unit: %.

ϵ	4.0	3.5	3.0	2.5	2.0	1.5	1.0	0.5
$\sigma = 0$	70.08	67.41	65.19	61.13	56.27	51.41	37.92	25.12
$\sigma = 1$	72.20	**71.25**	68.42	65.32	**62.70**	58.32	45.05	31.35
$\sigma = 2$	72.06	70.66	**68.97**	65.59	61.30	**58.62**	**46.28**	**32.11**
$\sigma = 3$	**73.61**	70.06	68.33	**66.96**	60.77	57.37	45.14	32.07

Table 5: Testing accuracy of different models trained by DP-LSSGD with different σ. Unit: %.

σ	0	2	4	6	8	10	12	15
Logistic Regression $(0.5, 2 \times 10^{-5})$-DP	81.59	84.41	84.17	84.15	**85.20**	83.71	83.63	83.26
Logistic Regression $(0.3, 2 \times 10^{-5})$-DP	78.71	81.39	80.97	**82.75**	82.02	81.01	80.94	80.89
SVM $(0.5, 2 \times 10^{-5})$-DP	78.28	81.72	80.97	81.11	**81.67**	81.35	80.80	80.56
SVM $(0.3, 2 \times 10^{-5})$-DP	72.98	77.09	77.18	77.02	**77.54**	77.01	76.05	75.82

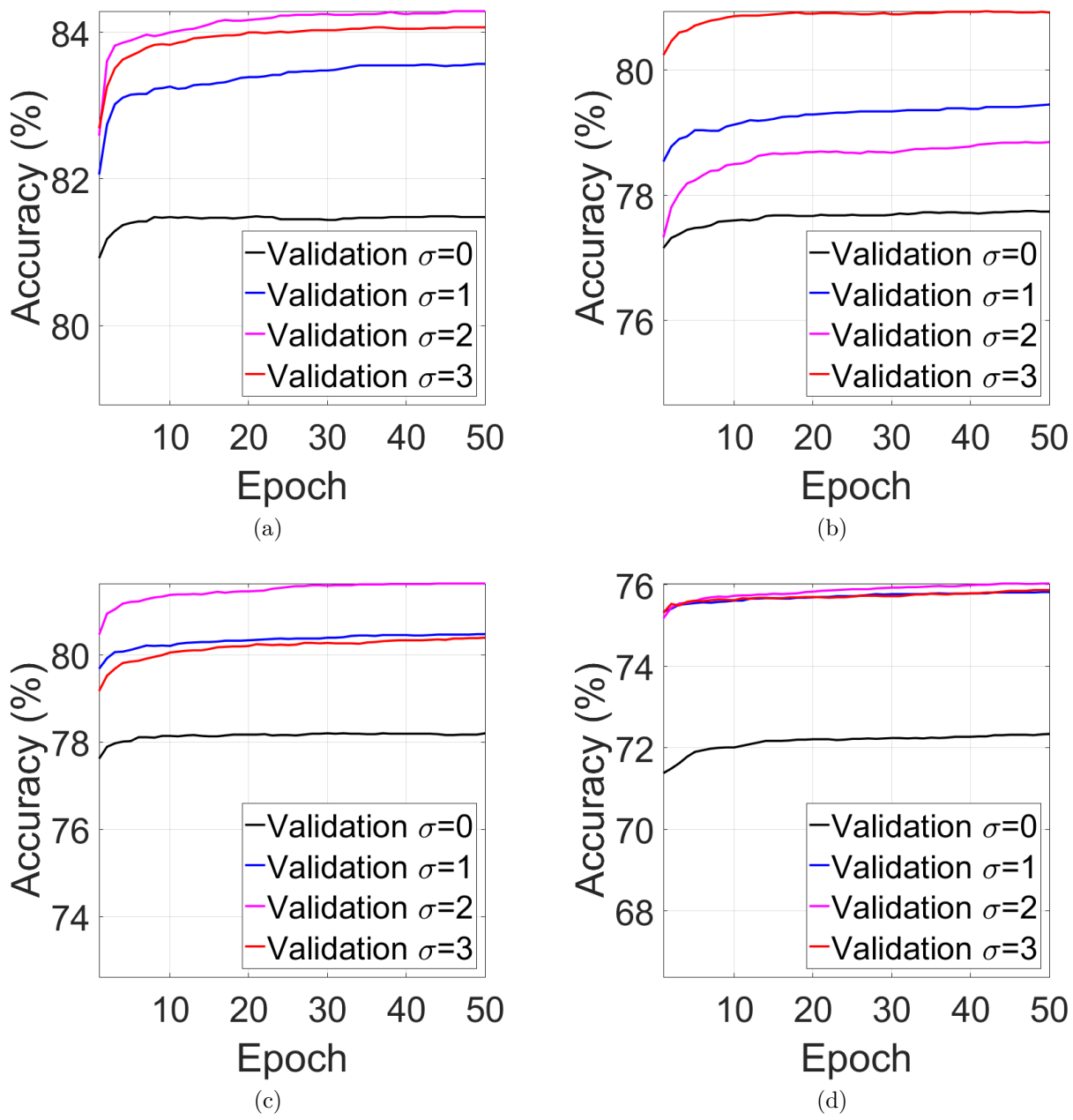

Figure 4: Epoch v.s. validation accuracy. (a): multi-class logistic regression with $(0.5, 2 \times 10^{-5})$-DP; (b): multi-class logistic regression with $(0.25, 2 \times 10^{-5})$-DP; (c): SVM with $(0.5, 2 \times 10^{-5})$-DP; (d): SVM with $(0.25, 2 \times 10^{-5})$-DP.

4.2 Deep Learning

We run 100 epochs of DP-LSSGD with batch size 128 to train ResNet20 on the CIFAR10. To justify our theoretical results, we apply DP-LSSGD without momentum, and no weight decay is used during the training. It is known that Nesterov momentum and weight decay, i.e., the ℓ_2 regularization, are helpful to accelerate the convergence and improve the generalization of the trained model. In our future work, we will integrate these techniques into DP-LSSGD. We split the training data into 45K/5K for crosss validation. During training, we decay the learning rate by a factor of 10 at the 40th and 80th epoch, respectively. Figure 5

shows the evolution of epoch v.s. training (Fig. 5(a)) and validation losses (Fig. 5(b)) of ResNet20 trained by DP-LSSGD with different Laplacian smoothing parameters $\sigma = 0, 1, 2, 3$, and with $(1.5, 10^{-5})$-DP guarantee. We conclude from these two plots that: (i) learning rate decay is still very helpful for DP-LSSGD in training DNNs which is well-known in SGD, as we see that there is a sharp training and validation loss decay at the 40th epoch; (ii) the Laplacian smoothing can reduce both the training and validation losses significantly.

We plot the evolution of epoch v.s. validation accuracy in Fig. 5 (c) which is generally consistent with the evolution of epoch v.s. validation loss. In Fig. 5 (d) we plot the testing accuracy of the trained model by DP-LSSGD with different Laplacian smoothing parameters σ and different ϵ with fixed $\delta = 10^{-5}$, where the corresponding values of the testing accuracy are listed in Table 4. DP-LSSGD can improve testing accuracy up to $\sim 8\%$ when the strong DP is guaranteed. The accuracy improvement is much more significant than that of convex optimization scenario.

DP-LSSGD is a complement to the privacy mechanisms proposed in [1] and [29]. In future work, we will integrate DP-LSSGD into the algorithms proposed in [1] and [29] to further boost private model's utility.

4.2.1 Is Gradient Norm the Right Metric for Measuring Utility?

In Section 3 we gave a simple nonconvex function and showed that a point having a smaller gradient ℓ_2-norm does not indicate it is closer to the local minima. Now, we will show experimentally that for ResNet20, a smaller gradient norm does not indicate more proximity to the local minima. Figure 6 depicts the epoch (k) v.s. $\|\nabla F(\mathbf{w}^k)\|_2$ (a), validation accuracy (b), training (c) and validation (d) losses. These plots show that during evolution, though DP-LSSGD has a larger gradient norm than DP-SGD, it has much better utility in terms of validation accuracy, and training and validation losses.

5 Conclusions

In this paper, we proposed a new differentially private stochastic optimization algorithm, DP-LSSGD, inspired by the recently proposed LSSGD. The algorithm is simple to implement and the extra computational cost compared with the DP-SGD is almost negligible. We show that DP-LSSGD can lift the utility of the trained private ML models both numerically and theoretically. It is straightforward to combine LS with other variance reduction technique, e.g., SVRG [22].

A Proof of the Main Theorems

A.1 Privacy Guarantee

To prove the privacy guarantee in Theorem 1, we first introduce the following ℓ_2-sensitivity.

Definition 2 (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.

We will adapt the concepts and techniques of Rényi differential privacy (RDP) to prove the DP-guarantee of the proposed DP-LSSGD.

Definition 3 (RDP)**.** *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*

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

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

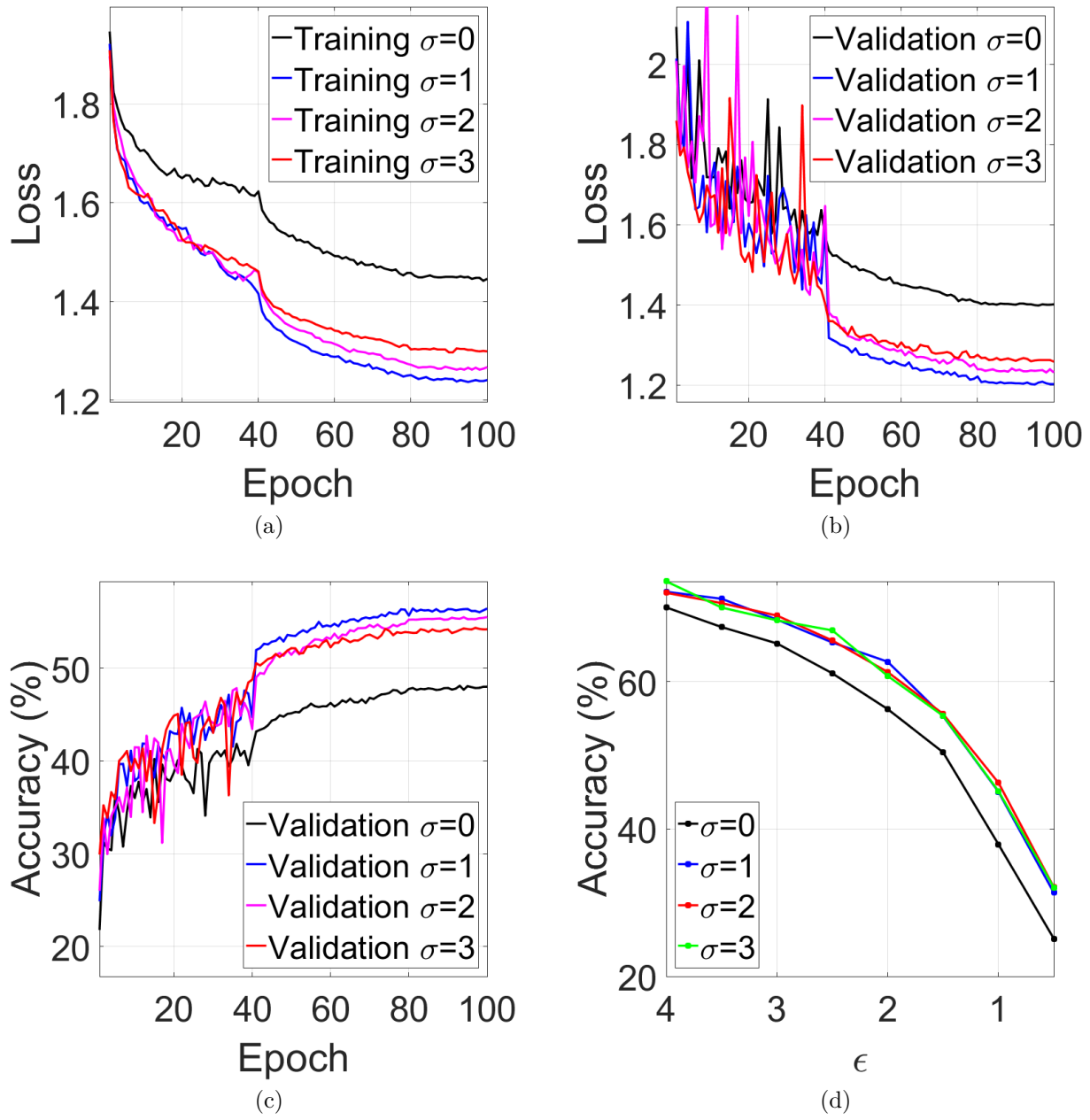

Figure 5: Training (a) and validation (b) losses of ResNet20 trained by DP-LSSGD with $(1.5, 10^{-5})$-DP guarantee and different Laplacian smoothing parameter σ. (c): epoch v.s. validation of the ResNet20 trained by DP-LSSGD with $(1.5, 10^{-5})$-DP guarantee and different σ. (d): Testing accuracy of ResNet20 trained by DP-LSSGD with different $(\epsilon, 10^{-5})$-DP guarantee.

Lemma 1. *[40] Given a function $q : \mathcal{S}^n \to \mathcal{R}$, the Gaussian Mechanism $\mathcal{M} = q(S) + \mathbf{n}$, where $\mathbf{n} \sim N(0, \nu^2\mathbf{I})$, satisfies $(\alpha, \alpha\Delta^2(q)/(2\nu^2))$-RDP. In addition, if we apply the mechanism $\mathcal{M}$ to a subset of samples using uniform sampling without replacement, $\mathcal{M}$ satisfies $(\alpha, \tau^2\Delta_2^2(q)\alpha/\nu^2)$-RDP when $\nu^2 \geq 1/1.25$, with τ denoting the subsample rate.*

Lemma 2. *[26] 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*

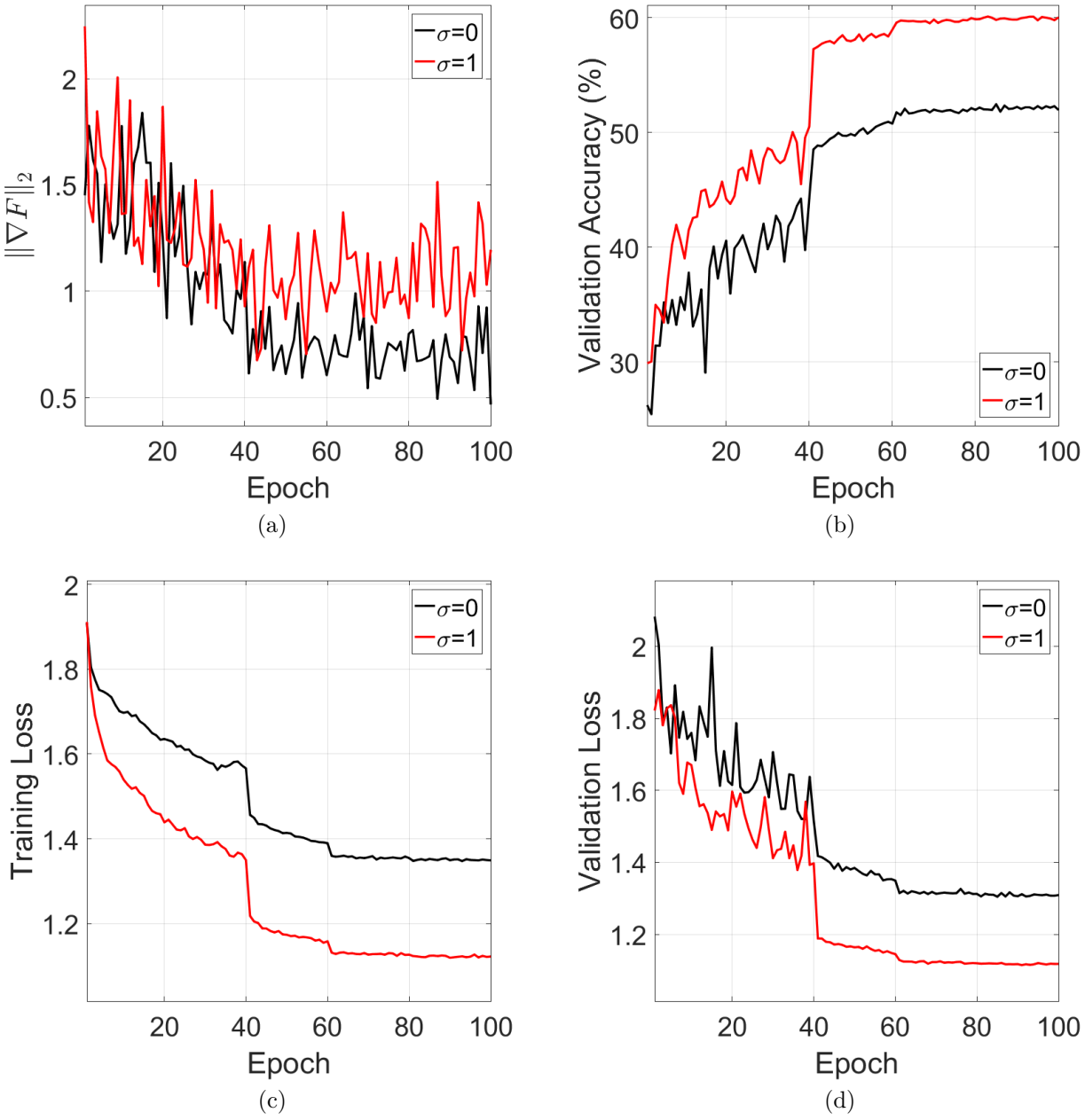

Figure 6: Comparisons between DP-SGD and DP-LSSGD $\sigma = 1$) on ResNet20 with $(1.5, 10^{-5})$-DP guarantee. Epoch v.s. $\|\nabla F(\mathbf{w}^k)\|_2$ (a), validation accuracy (b), training loss (c), validation loss (d).

be based on outputs of the previous $(i-1)$ *mechanisms.*

Lemma 3. *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* $\delta \in (0, 1)$*.*

With the definition (Def. 3) and guarantees of RDP (Lemmas 1 and 2), and the connection between RDP and (ϵ, δ)-DP (Lemma 3), we can prove the following DP-guarantee for DP-LSSGD.

Proof of Theorem 1. Let us denote the update of DP-SGD and DP-LSSGD at the k-th iteration starting

from any given points $\mathbf{w}^k$ and $\tilde{\mathbf{w}}^k$, respectively, as

$$\mathbf{w}^{k+1} = \mathbf{w}^k - \eta_k(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n}), \tag{6}$$

and

$$\tilde{\mathbf{w}}^{k+1} = \tilde{\mathbf{w}}^k - \eta_k \mathbf{A}_\sigma^{-1}(\nabla f_{i_k}(\tilde{\mathbf{w}}^k) + \mathbf{n}), \tag{7}$$

where i_k are drawn uniformly from $[n]$.

We will show that with the aforementioned Gaussian noise $\mathcal{N}(0, \nu^2)$ for each coordinate of $\mathbf{n}$, the output of DP-SGD, $\tilde{\mathbf{w}}$, after T iterations is (ϵ, δ)-DP. Let us consider the mechanism $\hat{\mathcal{M}}_k = \nabla F(\mathbf{w}^k) + \mathbf{n}$ with the query $\mathbf{q}_k = \nabla F(\mathbf{w}^k)$. We have the ℓ_2-sensitivity of $\mathbf{q}_k$ as $\Delta(\mathbf{q}_k) = \|\nabla f_{i_k}(\mathbf{w}^k) - \nabla f_{i'_k}(\mathbf{w}^k)\|_2/n \leq 2G/n$. According to Lemma 1, if we add noise with variance

$$\nu^2 = \frac{T\alpha(\alpha-1)\Delta^2(\mathbf{q}_k)}{\log(1/\delta)} = \frac{4T\alpha(\alpha-1)G^2}{n^2\log(1/\delta)},$$

the mechanism $\hat{\mathcal{M}}_k$ will satisfy $\big(\alpha, n^2\log(1/\delta)/\big(2(\alpha-1)T\big)\big)$-RDP. By post-processing theorem, we immediately have that under the same noise, $\mathcal{M}_k = \mathbf{A}_\sigma^{-1}(\nabla F(\mathbf{w}^k) + \mathbf{n})$ also satisfies $\big(\alpha, n^2\log(1/\delta)/\big(2(\alpha-1)T\big)\big)$-RDP. According to Lemma 1, $\mathcal{M}_k$ will satisfy $\big(\alpha, \log(1/\delta)/(\alpha-1)T\big)$-RDP provided that $\nu^2 \geq 1/1.25$. Let $\alpha = 2\log(1/\delta)/\epsilon + 1$, we obtain that $\mathcal{M}_k$ satisfies $\big(2\log(1/\delta)/\epsilon + 1, \epsilon/(2T)\big)$-RDP as long as we have

$$\nu^2 = \frac{4T\alpha(\alpha-1)G^2}{n^2\log(1/\delta)} = \frac{4T\big(2\log(1/\delta)+\epsilon\big)2\log(1/\delta)G^2}{n^2\log(1/\delta)\epsilon^2} \geq \frac{1}{1.25},$$

which implies that

$$\epsilon^2 \leq \frac{20TG^2\log(1/\delta)}{n^2}.$$

Therefore, according to Lemma 2, we have $\mathbf{w}^k$ satisfies $\big(2\log(1/\delta)/\epsilon + 1, k\epsilon/(2T)\big)$-RDP. Finally, by Lemma 3, we have $\mathbf{w}^k$ satisfies $\big(k\epsilon/(2T) + \epsilon/2, \delta\big)$-DP. Therefore, the output of DP-SGD, $\tilde{\mathbf{w}}$, is (ϵ, δ)-DP. □

Remark 3. *In the above proof, we used the following estimate of the ℓ_2 sensitivity*

$$\Delta(\mathbf{q}_k) = \|\mathbf{A}_\sigma^{-1}\nabla f_i(\mathbf{w}^k) - \mathbf{A}_\sigma^{-1}\nabla f_{i'}(\mathbf{w}^k)\|_2/n \leq 2G/n.$$

Indeed, let $\mathbf{g} = \nabla f_i(\mathbf{w}^k) - \nabla f_{i'}(\mathbf{w}^k)$ and $\mathbf{d} = \mathbf{A}_\sigma^{-1}\mathbf{g}$, then according to [27] we have

$$\|\mathbf{d}\|_2 + 2\sigma\frac{\|\mathbf{D}_+\mathbf{d}\|_2^2}{d} + \sigma^2\frac{\|\mathbf{L}\mathbf{d}\|_2^2}{d} = \|\mathbf{g}\|_2,$$

where d is the dimension of $\mathbf{d}$, and

$$\mathbf{D}_+ = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ 0 & 0 & -1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & 0 & \dots & 0 & -1 \end{bmatrix}.$$

Moreover, if we assume the $\mathbf{g}$ is randomly sampled from a unit ball in a high dimensional space, then a high probability estimation of the compression ratio of the ℓ_2 norm can be derived from Lemma. 5.

Numerical experiments show that $\|\mathbf{A}_\sigma^{-1}\nabla f_i(\mathbf{w}^k) - \mathbf{A}_\sigma^{-1}\nabla f_{i'}(\mathbf{w}^k)\|_2$ is much less than $\|\nabla f_i(\mathbf{w}^k) - \nabla f_{i'}(\mathbf{w}^k)\|_2$, so for the above noise, it can give much stronger privacy guarantee.

A.2 Utility Guarantee – Convex Optimization

To prove the utility guarantee for convex optimization, we first show that the Laplacian smoothing operator compresses the ℓ_2 norm of any given Gaussian random vector with a specific ratio in expectation.

Lemma 4. *Let $\boldsymbol{x} \in \mathbb{R}^d$ be the standard Gaussian random vector. Then*

$$\mathbb{E}\|\boldsymbol{x}\|^2_{\mathbf{A}_\sigma^{-1}} = \sum_{i=1}^{d} \frac{1}{1+2\sigma-2\sigma\cos(2\pi i/d)},$$

where $\|\boldsymbol{x}\|^2_{\mathbf{A}_\sigma^{-1}} \doteq \langle \boldsymbol{x}, \mathbf{A}_\sigma^{-1}\boldsymbol{x}\rangle$ is the square of the induced norm of $\boldsymbol{x}$ by the matrix $\mathbf{A}_\sigma^{-1}$.

Proof of Lemma 4. Let the eigenvalue decomposition of $\mathbf{A}_\sigma^{-1}$ be $\mathbf{A}_\sigma^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with $\Lambda_{ii} = \frac{1}{1+2\sigma-2\sigma\cos(2\pi i/d)}$ We have

$$\begin{aligned}
\mathbb{E}\|\boldsymbol{x}\|^2_{\mathbf{A}_\sigma^{-1}} &= \mathbb{E}[\mathrm{Tr}(\boldsymbol{x}^\top \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top \boldsymbol{x})] \\
&= \sum_{i=1}^{d} \Lambda_{ii} \\
&= \sum_{i=1}^{d} \frac{1}{1+2\sigma-2\sigma\cos(2\pi i/d)}.
\end{aligned}$$

□

Proof of Theorem 2. Recall that we have the following update rule $\mathbf{w}^{k+1} = \mathbf{w}^k - \eta_k \mathbf{A}_\sigma^{-1}(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n})$, where i_k are drawn uniformly from $[n]$, and $\mathbf{n} \sim \mathcal{N}(0, \nu^2\mathbf{I})$. Observe that

$$\begin{aligned}
\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} &= \|\mathbf{w}^k - \eta_k \mathbf{A}_\sigma^{-1}(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n}) - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} \\
&= \|\mathbf{w}^k - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} + \eta_k^2\big(\|\mathbf{A}_\sigma^{-1}\nabla f_{i_k}(\mathbf{w}^k)\|^2_{\mathbf{A}_\sigma} + \|\mathbf{A}_\sigma^{-1}\mathbf{n}\|^2_{\mathbf{A}_\sigma} + 2\langle \mathbf{A}_\sigma^{-1}\nabla f_{i_k}(\mathbf{w}^k), \mathbf{n}\rangle\big) \\
&\quad - 2\eta_k\langle \nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n}, \mathbf{w}^k - \mathbf{w}^*\rangle.
\end{aligned}$$

Taking expectation with respect to i_k and $\mathbf{n}$ given $\mathbf{w}^k$, we have

$$\begin{aligned}
\mathbb{E}\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} &= \mathbb{E}\|\mathbf{w}^k - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} - 2\eta_k\mathbb{E}\langle \nabla F(\mathbf{w}^k), \mathbf{w}^k - \mathbf{w}^*\rangle + \eta_k^2\mathbb{E}\|\nabla f_{i_k}(\mathbf{w}^k)\|^2_{\mathbf{A}_\sigma^{-1}} + \eta_k^2\mathbb{E}\|\mathbf{n}\|^2_{\mathbf{A}_\sigma^{-1}} \\
&\le \mathbb{E}\|\mathbf{w}^k - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} - 2\eta_k\mathbb{E}\big(F(\mathbf{w}^k) - F(\mathbf{w}^*)\big) + \eta_k^2\big(G^2 + \gamma d\nu^2\big),
\end{aligned}$$

where the second inequality is due to the convexity of F, and Lemma 4. It implies that

$$2\eta_k\mathbb{E}\big(F(\mathbf{w}^k) - F(\mathbf{w}^*)\big) \le \big(\mathbb{E}\|\mathbf{w}^k - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma} - \mathbb{E}\|\mathbf{w}^{k+1} - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma}\big) + \eta_k^2(G^2 + \gamma d\nu^2).$$

Now taking the full expectation and summing up over T iterations, we have

$$\sum_{k=0}^{T-1} 2\eta_k\mathbb{E}\big(F(\mathbf{w}^k) - F(\mathbf{w}^*)\big) \le D_\sigma + \sum_{k=0}^{T-1} \eta_k^2(G^2 + \gamma d\nu^2),$$

where $D_\sigma = \|\mathbf{w}^0 - \mathbf{w}^*\|^2_{\mathbf{A}_\sigma}$. Let $v_k = \eta_k / \big(\sum_{k=0}^{T-1}\eta_k\big)$, we have

$$\sum_{k=0}^{T-1} v_k\mathbb{E}\big(F(\mathbf{w}^k) - F(\mathbf{w}^*)\big) \le \frac{D_\sigma + \sum_{k=0}^{T-1}\eta_k^2(G^2 + \gamma d\nu^2)}{2\sum_{k=0}^{T-1}\eta_k}.$$

According to the definition of $\tilde{\mathbf{w}}$ and the convexity of F, we obtain

$$\begin{aligned}
\mathbb{E}\big(F(\tilde{\mathbf{w}}) - F(\mathbf{w}^*)\big) &\le \frac{D_\sigma + \sum_{k=0}^{T-1}\eta_k^2(G^2 + \gamma d\nu^2)}{2\sum_{k=0}^{T-1}\eta_k} \\
&\le \frac{D_\sigma + \sum_{k=0}^{T-1}\eta_k^2 G^2}{2\sum_{k=0}^{T-1}\eta_k} + \frac{\sum_{k=0}^{T-1}\eta_k^2}{2\sum_{k=0}^{T-1}\eta_k} \cdot \frac{24\gamma d T G^2 \log(1/\delta)}{n^2\epsilon^2}.
\end{aligned}$$

Let $\eta = 1/\sqrt{T}$ and $T = (D_\sigma + G^2)n^2\epsilon^2/\big(24\gamma d G^2 \log(1/\delta)\big)$, we can obtain that

$$\mathbb{E}\big(F(\tilde{\mathbf{w}}) - F(\mathbf{w}^*)\big) \leq \frac{2G\sqrt{6\gamma(D_\sigma + G^2)d\log(1/\delta)}}{n\epsilon}.$$

□

A.3 Utility Guarantee – Nonconvex Optimization

To prove the utility guarantee for nonconvex optimization, we need the following lemma, which shows that the Laplacian smoothing operator compresses the ℓ_2 norms of any given Gaussian random vector with a specific ratio in expectation.

Lemma 5. *Let $\boldsymbol{x} \in \mathbb{R}^d$ be the standard Gaussian random vector. Then*

$$\mathbb{E}\|\mathbf{A}_\sigma^{-1}\boldsymbol{x}\|_2^2 = \sum_{i=1}^{d} \frac{1}{(1 + 2\sigma - 2\sigma\cos(2\pi i/d))^2}.$$

Proof of Lemma 5. Let the eigenvalue decomposition of $\mathbf{A}_\sigma^{-1}$ be $\mathbf{A}_\sigma^{-1} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with $\Lambda_{ii} = \frac{1}{1+2\sigma-2\sigma\cos(2\pi i/n)}$ We have

$$\begin{aligned}
\mathbb{E}\|\mathbf{A}_\sigma^{-1}\boldsymbol{x}\|_2^2 &= \mathbb{E}[\mathrm{Tr}(\boldsymbol{x}^\top\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top\boldsymbol{x})] \\
&= \mathbb{E}[\mathrm{Tr}(\boldsymbol{x}^\top\mathbf{U}\mathbf{\Lambda}^2\mathbf{U}^\top\boldsymbol{x})] \\
&= \sum_{i=1}^{d}\Lambda_{ii}^2 \\
&= \sum_{i=1}^{d} \frac{1}{(1 + 2\sigma - 2\sigma\cos(2\pi i/d))^2}.
\end{aligned}$$

□

Proof of Theorem 3. Recall that we have the following update rule $\mathbf{w}^{t+1} = \mathbf{w}^k - \eta_k\mathbf{A}_\sigma^{-1}(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n})$, where i_k are drawn uniformly from $[n]$, and $\mathbf{n} \sim \mathcal{N}(0, \nu^2\mathbf{I})$. Since F is L-smooth, we have

$$\begin{aligned}
F(\mathbf{w}^{k+1}) &\leq F(\mathbf{w}^k) + \langle\nabla F(\mathbf{w}^k), \mathbf{w}^{k+1} - \mathbf{w}^k\rangle + \frac{L}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|_2^2 \\
&= F(\mathbf{w}^k) - \eta_k\langle\nabla F(\mathbf{w}^k), \mathbf{A}_\sigma^{-1}(\nabla f_{i_k}(\mathbf{w}^k) + \mathbf{n})\rangle \\
&\quad + \frac{\eta_k^2 L}{2}\Big(\|\mathbf{A}_\sigma^{-1}\nabla f_{i_k}(\mathbf{w}^k)\|_2^2 + \|\mathbf{A}_\sigma^{-1}\mathbf{n}\|_2^2 + 2\langle\mathbf{A}_\sigma^{-1}\nabla f_{i_k}(\mathbf{w}^k), \mathbf{A}_\sigma^{-1}\mathbf{n}\rangle\Big).
\end{aligned}$$

Taking expectation with respect to i_k and $\mathbf{n}$ given $\mathbf{w}^k$, we have

$$\begin{aligned}
\mathbb{E}F(\mathbf{w}^{k+1}) &\leq \mathbb{E}F(\mathbf{w}^k) - \eta_k\mathbb{E}\langle\nabla F(\mathbf{w}^k), \mathbf{A}_\sigma^{-1}\nabla f_{i_k}(\mathbf{w}^k)\rangle + \frac{\eta_k^2 L}{2}\Big(\mathbb{E}\|\mathbf{A}_\sigma^{-1}\nabla f_{i_k}(\mathbf{w}^k)\|_2^2 + \mathbb{E}\|\mathbf{A}_\sigma^{-1}\mathbf{n}\|_2^2\Big) \\
&\leq \mathbb{E}F(\mathbf{w}^k) - \eta_k\Big(1 - \frac{\eta_k L}{2}\Big)\mathbb{E}\|\nabla F(\mathbf{w}^k)\|_{\mathbf{A}_\sigma^{-1}}^2 + \frac{\eta_k^2 L}{2}(G^2 + d\beta\nu^2) \\
&\leq \mathbb{E}F(\mathbf{w}^k) - \frac{\eta_k}{2}\mathbb{E}\|\nabla F(\mathbf{w}^k)\|_{\mathbf{A}_\sigma^{-1}}^2 + \frac{\eta_k^2 L(G^2 + d\beta\nu^2)}{2},
\end{aligned}$$

where the second inequality uses Lemma 5 and the last inequality is due to $1 - \eta_k L/2 > 1/2$. Now taking the full expectation and summing up over T iterations, we have

$$\mathbb{E}F(\mathbf{w}^T) \leq F(\mathbf{w}^0) - \sum_{k=1}^{T-1}\frac{\eta_k}{2}\mathbb{E}\|\nabla F(\mathbf{w}^k)\|_{\mathbf{A}_\sigma^{-1}}^2 + \sum_{k=1}^{T-1}\frac{\eta_k^2 L(G^2 + d\beta\nu^2)}{2}.$$

If we choose fix step size, i.e., $\eta_k = \eta$, and rearranging the above inequality, and using $F(\mathbf{w}^0) - \mathbb{E}F(\mathbf{w}^T) \leq F(\mathbf{w}^0) - F(\mathbf{w}^*)$, we get

$$\frac{1}{T}\sum_{k=1}^{T-1}\mathbb{E}\|\nabla F(\mathbf{w}^k)\|^2_{\mathbf{A}_\sigma^{-1}} \leq \frac{2}{\eta T}\big(F(\mathbf{w}^0) - F(\mathbf{w}^*)\big) + \eta L(G^2 + d\beta\nu^2),$$

which implies that

$$\begin{aligned}\mathbb{E}\|\nabla F(\tilde{\mathbf{w}})\|^2_{\mathbf{A}_\sigma^{-1}} &\leq \frac{2D_F}{\eta T} + \eta L(G^2 + d\beta\nu^2)\\ &\leq \frac{2D_F}{\eta T} + \eta L\bigg(G^2 + \frac{24d\beta TG^2\log(1/\delta)}{n^2\epsilon^2}\bigg).\end{aligned}$$

Let $\eta = 1/\sqrt{T}$ and $T = (2D_F + LG^2)n^2\epsilon^2/\big(24dL\beta G^2\log(1/\delta)\big)$, where $D_F = F(\mathbf{w}^0) - F(\mathbf{w}^*)$, we obtain

$$\mathbb{E}\|\nabla F(\tilde{\mathbf{w}})\|^2_{\mathbf{A}_\sigma^{-1}} \leq 4\frac{G\sqrt{6\beta dL(2D_F + LG^2)\log(1/\delta)}}{n\epsilon}.$$

□

B β and γ

B.1 γ

To prove Proposition 1, we need the following two lemmas.

Lemma 6 (Residue Theorem)**.** *Let $f(z)$ be a complex function defined on $\mathbb{C}$, then the residue of f around the pole $z = c$ can be computed by the formula*

$$\mathrm{Res}(f, c) = \frac{1}{(n-1)!}\lim_{z\to c}\frac{d^{n-1}}{dz^{z-1}}\left((z-c)^n f(z)\right). \tag{8}$$

where the order of the pole c is n. Moreover,

$$\oint f(z)dz = 2\pi i\sum_{c_i} Res(f, c_i), \tag{9}$$

where $\{c_i\}$ be the set of pole(s) of $f(z)$ inside $\{z||z| < 1\}$.

The proof of Lemma 6 can be found in any complex analysis textbook.

Lemma 7. *For $0 \leq \theta \leq 2\pi$, suppose*

$$F(\theta) = \frac{1}{1 + 2\sigma(1 - \cos(\theta))},$$

has the discrete-time Fourier transform of series $f[k]$. Then, for integer k,

$$f[k] = \frac{\alpha^{|k|}}{\sqrt{4\sigma + 1}}$$

where

$$\alpha = \frac{2\sigma + 1 - \sqrt{4\sigma + 1}}{2\sigma}$$

Proof. By definition,

$$f[k] = \frac{1}{2\pi}\int_0^{2\pi} F(\theta)e^{ik\theta}\,d\theta = \frac{1}{2\pi}\int_0^{2\pi} \frac{e^{ik\theta}}{1+2\sigma(1-\cos(\theta))}\,d\theta. \tag{10}$$

We compute Eq. (10) by using Residue theorem. First, note that because $F(\theta)$ is real valued, $f[k] = f[-k]$; therefore, it suffices to compute Eq. (10) for nonnegative k. Set $z = e^{i\theta}$. Observe that $\cos(\theta) = 0.5(z + 1/z)$ and $dz = izd\theta$. Substituting in Eq. (10) and simplifying yields that

$$f[k] = \frac{-1}{2\pi i\sigma}\oint \frac{z^k}{(z-\alpha_-)(z-\alpha_+)}\,dz, \tag{11}$$

where the integral is taken around the unit circle, and $\alpha_\pm = \frac{2\sigma+1\pm\sqrt{4\sigma+1}}{2\sigma}$ are the roots of quadratic $-\sigma z^2 + (2\sigma+1)z - \sigma$. Note that α_- lies within the unit circle; whereas, α_+ lies outside of the unit circle. Therefore, because k is nonnegative, α_- is the only singularity of the integrand in Eq. (11) within the unit circle. A straightforward application of the Residue Theorem, i.e., Lemma 6, yields that

$$f[k] = \frac{-\alpha_-^k}{\sigma(\alpha_- - \alpha_+)} = \frac{\alpha^k}{\sqrt{4\sigma+1}}.$$

This completes the proof. □

Proof of Proposition 1. First observe that we can re-write γ as

$$\frac{1}{d}\sum_{j=0}^{d-1} \frac{1}{1+2\sigma(1-\cos(\frac{2\pi j}{d}))}. \tag{12}$$

It remains to show that the above summation is equal to $\frac{1+\alpha^d}{(1-\alpha^d)\sqrt{4\sigma+1}}$. This follows by lemmas 7 and standard sampling results in Fourier analysis (i.e. sampling θ at points $\{2\pi j/d\}_{j=0}^{d-1}$). Nevertheless, we provide the details here for completeness: Observe that that the inverse discrete-time Fourier transform of

$$G(\theta) = \sum_{j=0}^{d-1} \delta(\theta - \frac{2\pi j}{d}).$$

is given by

$$g[k] = \begin{cases} d/2\pi & \text{if } k \text{ divides } d, \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, let

$$F(\theta) = \frac{1}{1+2\sigma(1-\cos(\theta))},$$

and use $f[k]$ to denote its inverse discrete-time Fourier transform. Now,

$$\begin{aligned}
\frac{1}{d}\sum_{j=0}^{d-1} \frac{1}{1+2\sigma(1-\cos(\frac{2\pi j}{d}))} &= \frac{1}{d}\int_0^{2\pi} F(\theta)G(\theta) \\
&= \frac{2\pi}{d}\,\mathrm{DTFT}^{-1}[F\cdot G][0] \\
&= \frac{2\pi}{d}(\mathrm{DTFT}^{-1}[F] * \mathrm{DTFT}^{-1}[G])[0] \\
&= \frac{2\pi}{d}\sum_{r=-\infty}^{\infty} f[-r]g[r] \\
&= \frac{2\pi}{d}\sum_{\ell=-\infty}^{\infty} f[-\ell d]\frac{d}{2\pi} \\
&= \sum_{\ell=-\infty}^{\infty} f[-\ell d].
\end{aligned}$$

The proof is completed by substituting the result of lemma 7 in the above sum and simplifying. □

We list some typical values of γ in Table 1.

Table 6: The values of γ corresponding to some σ and d.

σ	1	2	3	4	5
$d = 1000$	0.447	0.333	0.277	0.243	0.218
$d = 10000$	0.447	0.333	0.277	0.243	0.218
$d = 100000$	0.447	0.333	0.277	0.243	0.218

B.2 β

The proof of Proposition 2 is similar as the proof of Proposition 1. The only difference is that we need to compute

$$f[k] = \frac{1}{2\pi}\int_0^{2\pi} \frac{e^{ik\theta}}{(1+2\sigma(1-\cos\theta))^2} d\theta. \tag{13}$$

By Residue theorem, for $k > 0$ (note that $f[-k] = f[k]$), we have

$$\begin{aligned} f[k] &= \frac{1}{2\pi}\int_0^{2\pi} \frac{e^{ik\theta}}{(1+2\sigma(1-\cos\theta))^2} d\theta \\ &= \frac{1}{2\pi i}\oint \frac{z^{k+1}}{(z+\sigma(2z-z^2-1))^2} dz \\ &= \lim_{z\to\alpha^-} \frac{d}{dz}\left((z-\alpha^-)^2 \frac{z^{k+1}}{(z+\sigma(2z-z^2-1))^2}\right) \\ &= \lim_{z\to\alpha^-} \frac{d}{dz}\left(\frac{z^{k+1}}{\sigma^2(z-\alpha^+)^2}\right) \\ &= \frac{(k+1)\alpha^k}{4\sigma+1} + \frac{2\sigma\alpha^{k+1}}{(4\sigma+1)^{3/2}}, \end{aligned}$$

where $\alpha_- = \frac{2\sigma+1-\sqrt{4\sigma+1}}{2\sigma}$. Therefore, we have

$$\beta = \frac{2\alpha^{2d+1} - \xi\alpha^{2d} + 2\xi d\alpha^d - 2\alpha + \xi}{\sigma^2\xi^3(1-\alpha^d)^2}.$$

We list some typical values of β in Table 2.

Table 7: The values of β corresponding to some σ and d.

σ	1	2	3	4	5
$d = 1000$	0.268	0.185	0.149	0.128	0.114
$d = 10000$	0.268	0.185	0.149	0.128	0.114
$d = 100000$	0.268	0.185	0.149	0.128	0.114

Acknowledgments

This material is based on research sponsored by the Air Force Research Laboratory under grant numbers FA9550-18-0167 and MURI FA9550-18-1-0502, the Office of Naval Research under grant number N00014-18-1-2527, the U.S. Department of Energy under grant number DOE SC0013838, and by the National Science Foundation under grant number DMS-1554564, (STROBE). QG is partially supported by the National Science Foundation under grant number SaTC-1717950.

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] M. Balcan, T. Dick, Y. Liang, W. Mou, and H. Zhang. Differentially private clustering in high-dimensional euclidean spaces. In *34th International Conference on Machine Learning (ICML 2017)*, 2017.

[3] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, 2014.

[4] M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. *ArXiv:1605.02065*, 2016.

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

[6] K. Chaudhuri, C. Monteleoni, and A. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12, 2011.

[7] J. Duchi, M. Jordan, and M. Wainwright. Privacy aware learning. *Journal of the Association for Computing Machinery*, 61(6), 2014.

[8] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. *Eurocrypt*, 2006.

[9] C. Dwork and J. Lei. Differential privacy and robust statistics. In *STOC*, 2009.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography (TCC)*, 2006.

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

[12] C. Dwork and G. Rothblum. Concentrated diffentially privacy. *ArXiv:1603.01887*, 2016.

[13] C. Dwork, G. Rothblum, and S. Vadhan. Boosting and differential privacy. In *51th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, 2010.

[14] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou. A survey on security, privacy and trust in mobile crowdsourcing. *IEEE Internet of Things Journal*, 2017.

[15] 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.

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

[17] A. Gilbert and A. McMillan. Local differential privacy for physical sensor data and sparse recovery. *arXiv:1706.05916*, 2017.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[19] P. Jain, P. Kothari, and A. Thakurta. Differentially private online learning. In *25th Conference on Learning Theory (COLT 2012)*, 2012.

[20] P. Jain, O. Thakkar, and A. Thakurta. Differentially private matrix completion. In *35th International Conference on Machine Learning (ICML 2018)*, 2018.

[21] B. Jayaraman, L. Wang, D. Evans, and Q. Gu. Distributed learning without distress: Privacy-preserving empirical risk minimization. In *Advances in Neural Information Processing Systems (NIPS 2018)*, 2018.

[22] R. Johoson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 2013.

[23] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. In *32nd International COnference on Machine Learning (ICML 2015)*, 2015.

[24] J. Lee and D. Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. *ArXiv:1808.09501*, 2018.

[25] Y. Liu, K. Gadepalli, M. Norouzi, G. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Nelson, G. Corrado, and et al. Detecting cancer metastases on gigapixel pathology images. *arXiv:1703.02442*, 2017.

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

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

[28] N. Papernot, M. Abadi, U. Erlingsson, I. Goodfellow, and K. Talwar. Semisupervised knowledge transfer for deep learning from private training data. In *5th International Conference on Learning Representation (ICLR 2017)*, 2017.

[29] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and U. Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations (ICLR 2018)*, 2018.

[30] M. Park, J. Foulds, K. Chaudhuri, and M. Welling. Private topic modeling. *arXiv:1609:04120*, 2016.

[31] M. A. Rahman, T. Rahman, R. Laganiere, N. Mohammed, and Y. Wang. Membership inference attack against differentially private deep learning model. *Transactions on Data Privacy*, 2018.

[32] R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *22nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2015)*, 2015.

[33] 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.

[34] S. Song, K. Chaudhuri, and A. Sarwate. Stochastic gradient descent with differentially private updates. In *GlobalSIP Conference*, 2013.

[35] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin. Differentially private k-means clustering. *arXiv:1504.05998*, 2015.

[36] K. Talwar, A. Thakurta, and L. Zhang. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pages 3025–3033, 2015.

[37] D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems (NIPS 2017)*, 2017.

[38] L. Wang and Q. Gu. Differentially private iterative gradient hard thresholding for sparse learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

[39] Y. Wang, J. Lei, and S. Fienberg. Learning with differential privacy: Stability, learnability and the sufficiency and necessity of ERM principle. *ArXiv:1502.06309*, 2016.

[40] Yu-Xiang Wang, Borja Balle, and Shiva Kasiviswanathan. Subsampled r\'enyi differential privacy and analytical moments accountant. *arXiv preprint arXiv:1808.00087*, 2018.

[41] Y. Wang Y. Wang and A. Singh. Differentially private subspace clustering. In *Advances in Neural Information Processing Systems (NIPS 2015)*, 2015.

[42] M. Yuen, I. King, and K. Leung. A survey of crowdsourcing systems. In *Proceedings of the IEEE international conference on social computing (Socialcom 2011)*, 2011.

[43] J. Zhang, K. Zheng, W. Mou, and L. Wang. Efficient private ERM for smooth objectives. In *The Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 2017.