Differentially-Private Deep Learning With Directional Noise

Liyao Xiang, *Member, IEEE*, Weiting Li, Jungang Yang, Xinbing Wang, *Senior Member, IEEE*, and Baochun Li, *Fellow, IEEE*

Abstract—With the popularity of deep learning applications, the privacy of training data has become a major concern as the data sources may be sensitive. Recent studies have found that deep learning models are vulnerable to privacy attacks, which are able to infer private training data from model parameters. To mitigate such attacks, differential privacy has been proposed to preserve data privacy by adding randomized noise to these models. However, since deep learning models usually consist of a large number of parameters and complicated layered structures, an overwhelming amount of noise is often inserted, which significantly degrades model accuracy. We seek a better tradeoff between model utility and data privacy, by choosing directions of noise w.r.t. the utility subspace. We propose an optimized mechanism for differentially-private stochastic gradient descent, and derive a closed-form solution. The form of the solution makes the mechanism ready to be deployed in real-world deep learning systems. Experimental results on a variety of models, datasets, and privacy settings show that our proposed mechanism achieves higher accuracies at the same privacy guarantee compared to the state-of-the-art methods. Further, we extend the privacy guarantee to a mutual information bound, and propose a general form to the utility-privacy problem.

Index Terms—Privacy, data mining, machine learning, optimization

1 Introduction

The recent proliferation of deep learning has empowered a wide spectrum of data analytical applications on crowdsourced data, which are collected from a crowd of participants. The data is typically sensitive and thus the deep models are required to preserve privacy. Fed with large volumes of data, deep models capture the intrinsic logic between data and tasks, with a huge number of model parameters and complicated model structures. Unfortunately, these models, trained and stored in smartphones can be sources of severe privacy leakage as shown by many studies, and such privacy leakage poses significant threats to sensitive training data. As examples, the model inversion attack [1] is able to recover class representatives, which can be sensitive facial features; the membership inference attack [2] can be used to infer whether an identity participates in the training or not, which may be able to single out a single training record. Regardless of the specific form of these attacks, it is widely recognized that deep learning models are privacy-leaking, and therefore privacy-preserving mechanisms are required to be implemented.

Differential privacy was proposed as a class of privacy-preserving mechanisms for releasing data statistics, and recently for publishing models. These mechanisms usually introduce randomness so that adversaries cannot distinguish adjacent input distributions when given the output of the mechanism [3]. In the context of deep learning, differential privacy mechanisms are applied with the purpose of 'hiding' a single input instance in the training dataset, i.e., despite the existence of the instance, no attacker can tell the difference in the released features, prediction outputs, or model parameters. Specifically, some mechanisms choose to insert randomized noise at each training iteration to guarantee that model parameters are differentially-private [4], [5], [6], [7]. Some mechanisms train differentially-private models on randomized prediction outcomes of other trained models [8], [9]. And some train the model towards a differentially-private objective function to fulfill the privacy guarantee [10], [11].

A key problem in deep learning with differential privacy lies in the fundamental tradeoff between model utility and data privacy. To guarantee differential privacy, randomized noise is inserted into the model, and such noise may significantly affect the utility of the model. It is likely that an overwhelming amount of noise perturbs the model to a degree that the model is not usable at all. The problem is even more severe with over-parameterized deep neural networks. For example, Abadi *et al.* [5] only achieve an accuracy of 95% on the MNIST dataset with a two-layer ReLU network at a medium privacy level ($\epsilon = 2.0, \delta = 10^{-5}$), whereas an unperturbed model can easily achieve an accuracy of over 99%. Although [6], [7] have shown significant improvement with

- *Liyao Xiang, Weiting Li, Jungang Yang, and Xinbing Wang are with Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {xiangliyao08, liweiting, yangjungang, xwang8}@sjtu.edu.cn.*
- *Baochun Li is with the University of Toronto, Toronto, ON M5S 1A1, Canada. E-mail: bli@ece.toronto.edu.*

Manuscript received 21 December 2020; revised 4 November 2021; accepted 10 November 2021. Date of publication 23 November 2021; date of current version 4 April 2023.
This work was supported in part by National Key R&D Program of China under Grant 2017YFB1003000, NSFC China under Grants 61902245, 62032020, 62136006, 61960206002, 42050105, and 61829201, and the Science and Technology Innovation Program of Shanghai under Grant 19YF1424500.
(Corresponding author: Liyao Xiang.)
Digital Object Identifier no. 10.1109/TMC.2021.3130060

new privacy budget allocation methods, their performance is still inferior to the unperturbed case.

Although the tradeoff between utility and privacy is inherent, we found that a better tradeoff can be obtained by taking the utility subspace of the model into account. Specifically, we observe that when the same amount of noise is inserted, the model would result in different accuracies depending on the noise directions. As a larger amount of noise is added to the less important directions of the utility subspace, the model suffers less accuracy loss. While pursuing a noise direction with optimal utility, it is still critical to guarantee differential privacy at the same time.

We formulate the problem of arbitrating the utility-privacy tradeoff as a constrained optimization problem, which seeks an optimal noise direction w.r.t. the utility subspace while preserving differential privacy. Our proposed mechanism follows the convention of differentially-private stochastic gradient descent [4], [5]. The privacy mechanism and setting are publicly known. The adversary can access the model parameters and any auxiliary information. In our proposed mechanism, we perform stochastic gradient descent on the training data, and add directional noise to the gradients. The directional noise is generated from a distribution which is obtained by solving the differential privacy constrained optimization problem. Due to the large number of model parameters, such a large-scale optimization problem is intrinsically inefficient to solve. Fortunately, we found a closed-form solution to our problem, which can be efficiently deployed in practice.

Since our optimized mechanism is proposed as an additive noise scheme, we further introduce a general form of the utility-privacy problem and establish its connection with the distortion-rate function. The 'utility' describes the perturbation impact to the original model, also known as the 'distortion' of the model. The 'privacy' is defined by the mutual information between the released model and the original one. By properties of the distortion-rate function, we are able to give a theoretical lower bound to the utility-privacy problem.

From an engineering perspective, we have implemented a differential privacy module for the proposed mechanism. The module is composed of an optimization submodule and a noise generation submodule. The former contains tensor operations solving the optimization problem and apply differentially-private noise to the gradients. Tensor operations can be processed by GPUs in batches. The latter is implemented to efficiently generate randomized noise given the distribution solved by the former component. Experimental results show that our implementation achieves higher accuracies compared to the baseline at moderate computational overhead.

Highlights of our original contributions are as follows. *First*, we novelly take into account the utility subspace in the design of differentially-private stochastic gradient descent, with the observation that it is possible to guarantee the same privacy with higher utility. *Second*, we formulate the problem of arbitrating the utility-privacy tradeoff as a constrained optimization problem that seeks the optimal noise distribution, and find a closed-form solution to the problem. *Third*, we extend the problem to a general form and establish a link with the distortion-rate function. *Last*, experiments on a variety of state-of-the-art deep learning models and datasets have shown that our mechanism can significantly improve model accuracy compared to previous works under the same privacy constraints.

2 Related Work

Our work is related to works in the following categories.

2.1 Deep Learning With Differential Privacy

Models trained over sensitive data can be a significant threat to the privacy of such data [1], [2]. To mitigate privacy risks, a number of algorithms have been proposed to achieve deep learning with differential privacy.

Following the principle of differentially-private stochastic gradient descent [12], Shokri *et al.* [4] let participants train their own datasets privately, and selectively share small subsets of their models' key parameters. Even that a small percentage (<0.1) of the parameters are perturbed and shared, their composition method still consumes a large amount of the privacy budget, which is way beyond a meaningful privacy guarantee. By exploiting higher moments of the privacy loss variable, the accounting method proposed by Abadi *et al.* [5] reduces the total amount of additive noise significantly. However, it only achieves an accuracy of 90% (with a privacy budget of $\epsilon = 0.5, \delta = 10^{-5}$) on MNIST.

Inspired by [4], [5], our work characterizes the relationship between model utility and the privacy constraints for the first time. So far, existing differential privacy mechanisms are mostly heuristic. Given the probability distribution function (pdf) of the perturbation noise, one could guarantee the privacy a heuristic can achieve using existing mechanisms, but knows little about its utility performance. This is detrimental to the results since an overly conservative privacy constraint usually requires an overwhelming amount of noise to be added. Another critical drawback in existing works is that their composition methods are suboptimal. In contrast, we adopt the optimal composition theorem [13] and further amplify the privacy guarantee with input sampling [14].

Differential privacy mechanisms may not be directly applied to the model parameters, but rather to the model features [15], prediction outcomes [8], [9] or the objective functions [10], [11]. A common property of all these mechanisms is that the differential privacy property holds no matter the adversary can only query the system as a black-box, or can view the model internals as a white-box. In [15], the authors propose to adaptively inject noise into features based on the contribution of each to the output, while in our work, noise is injected to the gradients w.r.t. their sensitivity to the output. Our work shares a similar principle to [15], but takes a different approach. In [8], a set of teacher models are trained on sensitive data privately while their perturbed predictions are aggregated to train a public student model. Compared to [4], [5], a noisy voting/aggregation mechanism [8], [9] achieves an impressive learning accuracy, but only works for classification tasks and requires non-sensitive data to be present. We show our experimental results achieve similar accuracies at higher privacy regime compared to their works.

2.2 Differentially-Private ERM

When the cost function is convex or strongly convex, some approaches have been proposed to achieve optimal or near

optimal utility bounds with differential privacy. The problem is called differentially-private empirical risk minimization (ERM) as the utility is defined as the worst-case (over inputs) expected excess empirical risk [16]. The approaches include gradient perturbation [16], [17], [18], output perturbation [16], [18], [19], [20], and objective perturbation [19]. In our work, we do not rely on the convexity of the cost function and focus on the algorithm's real-world performance.

2.3 Advanced Mechanisms in Differential Privacy

A wide variety of literature tries to improve the utility of differential privacy mechanisms from different perspectives. Geng *et al.* have proposed the optimal ϵ-differentially private mechanism under the general utility-maximization framework for single real-valued query functions has staircase-shaped noise probability density functions [21], [22]. Geng *et al.* further show that the nearly optimal mechanisms in (ϵ, δ)-differential privacy for integer-valued query functions under the utility-maximization framework. We follow their convention in formulating the general utility-maximization (or cost-minimization) objective function, but give the optimized mechanism for real-valued vectorized queries.

The scale of the perturbation noise typically depends on the global sensitivity of the query over the private datasets. We use the clipping value of the gradients as the global sensitivity for each query, but it is also likely to use local sensitivity [23] to further reduce the noise magnitude. Advanced results include the one introduced by Wu *et al.* in [20] that a new bound on the l_2-sensitivity of the stochastic gradient descent (SGD) algorithm allows better convergence of SGD under the same privacy guarantee. Pichapati *et al.* [24] use a coordinate-wise adaptive clipping of the gradient for differentially-private SGD algorithm. The work shares the same purpose with ours but takes an orthogonal approach.

When accessing datasets multiple times with differential privacy mechanisms, the overall privacy level would degrade on the union of those outputs, which is addressed by the composition theorem [25]. Abadi *et al.* adopt higher moments of the privacy loss variable to obtain tighter estimates [5], and our work takes a similar approach to compose differential privacy mechanisms over iterations. Theoretically optimal composition theorem has been proposed in [13], but is limited in practicality due to some constraints. Rényi differential privacy [26] proposes a relaxation of differential privacy based on the Rényi divergence, which could use more compact composition.

3 Preliminaries

We will introduce some preliminaries for better understanding this work.

3.1 Differential Privacy

Differential privacy is originally introduced to ensure that the ability of an adversary to compromise the privacy of any set of users is independent of whether any individual opts in to, or out of, the dataset [3]. Such an ability prevents any adversary from gaining additional information about any individual. The privacy guarantee is expressed by the logarithmic distance between the posterior probability distributions of two adjacent inputs given the outputs. Adjacent inputs are defined on two sets between which their distance is one unit, e.g., the two sets differ by a single entry. We use ϵ to define the upper bound of the distribution distance and δ to denote the residual probability. Formally, letting $\mathbf{X}$ and $\mathbf{X}'$ be the pair of adjacent inputs, $\mathcal{O}$ be the output set and $\mathcal{K}$ be the private mechanism, we have

Definition 1. *A mechanism $\mathcal{K}$ is (ϵ, δ)-differentially private if for all adjacent inputs $\mathbf{X}$ and $\mathbf{X}'$, and all possible output $\mathcal{O}$,*

$$\Pr[\mathcal{K}(\mathbf{X}) \in \mathcal{O}] \leq e^{\epsilon} \cdot \Pr[\mathcal{K}(\mathbf{X}') \in \mathcal{O}] + \delta. \tag{1}$$

In the special case of $\delta = 0$ we call $\mathcal{K}$ ϵ-differentially private. An intuitive interpretation of the differential privacy concept above is that, with the definition, we are constraining how well an adversary can distinguish the input $\mathbf{X}$ from $\mathbf{X}'$ given only the output of $\mathcal{K}(\mathbf{X})$ and $\mathcal{K}(\mathbf{X}')$. A common paradigm for approximating a function $f(\cdot)$ with differential privacy is to add noise gauged by the sensitivity of $f(\cdot)$, which is defined by the maximum of the distance $\|f(\mathbf{X}) - f(\mathbf{X}')\|$.

3.2 Stochastic Gradient Descent

In general, a deep neural network is denoted by a multi-dimensional function $\mathbf{F} : \mathcal{X} \mapsto \mathcal{Y}$ featured by a set of parameters $\boldsymbol{\theta}$. We use $\boldsymbol{\theta} \in \mathbb{R}^d$ to represent the flattened vector of parameters where d is its dimension. $\mathbf{X} \in \mathcal{X}$ is a training set from the training data space and $\mathbf{Y} \in \mathcal{Y}$ is the corresponding targeted output. Let the cost function C be the discrepancy between the output predicted by $\mathbf{F}$ and the target $\mathbf{Y}$. The training process is to find $\boldsymbol{\theta}$ such that

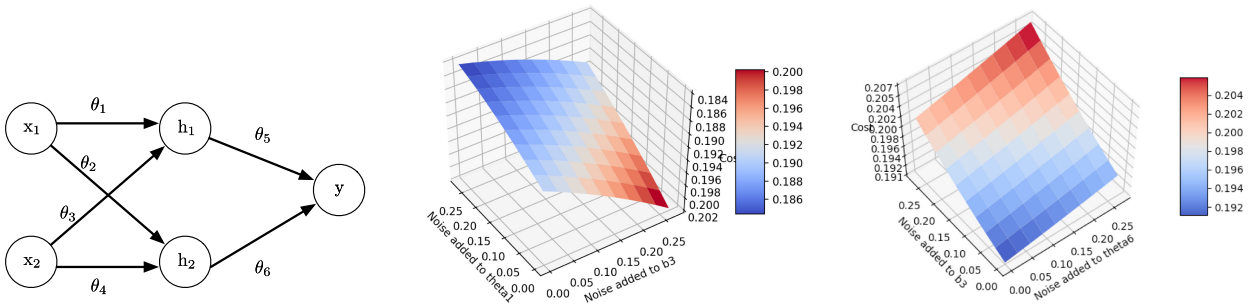
$$\underset{\boldsymbol{\theta}}{\text{minimize}}\, C. \tag{2}$$

Various forms of the cost function can be applied, such as square error for linear regression, or the logistic regression cost function. With SGD, we repeatedly pick training examples (usually in mini batches) and compute the gradient of the cost function with respect to each parameter. The parameters are updated in the opposite direction of the gradients to minimize the total cost.

4 Optimized Additive Noise for DPSGD

General Settings. The solution of Eqn. (2), or the updated parameters $\boldsymbol{\theta}$, may leak the secret of training data. Particularly, as pointed out by [27], there is a clear dependence between membership inference and overfitting. In the threat model, an adversary with auxiliary knowledge may infer about a particular record in the sensitive training set, from the released model parameters.

Differentially-private SGD (DPSGD), or noisy SGD, is an application of differential privacy on stochastic gradient descent which is a common optimization technique in machine learning, and the mechanism has been adopted in [4], [5], [12], [24]. A way of performing DPSGD is to add noise to the gradient, which inhibits the adversary from inferring information about the training data. In particular, randomized noise is generated with respect to the sensitivity of the parameter $\boldsymbol{\theta}$ such that an individual change in the training data alters $\boldsymbol{\theta}$ so little that one could hardly discern



(a) (b) (c)

Fig. 1. (a) A multi-layer perceptron architecture. (b) The cost variation when perturbing θ_1 and b_3. $\frac{\partial C}{\partial \theta_1} \in (-0.048, -0.017)$ and $\frac{\partial C}{\partial b_3} \in (0.017, 0.049)$. (c) The cost when perturbing θ_6 and b_3. $\frac{\partial C}{\partial \theta_6} \in (0.013, 0.022)$ and $\frac{\partial C}{\partial b_3} \in (0.033, 0.056)$.

the difference. We will follow the convention of DPSGD and formulate our problem in this section.

Prior to the problem formulation, we first observe that perturbation on parameters have different impact on the cost which is closely associated with resulting accuracies. Based on the observation, we propose an optimized additive noise scheme for DPSGD, i.e., the randomized noise is generated being aware of the resulting accuracy. The problem is formulated as an optimization one and a closed-form solution is given. Finally, we explicitly provide our privacy mechanism based on the optimized noise scheme.

4.1 Utility Subspace

We ask the question: for a trained model, if inserting the same total amount of perturbation, would different levels of perturbation to each of the model weights yield the same model accuracy? We refer to the accuracy as the utility, and use the cost to gauge utility: the lower the cost, the higher the utility. In this section, we introduce preliminary experiments conducted to explore the utility subspace, i.e., how the change in the model parameters affects the cost.

Similar to the sensitivity analysis by Papernot *et al.* [28], and the influence function of the input by Koh *et al.* [29], we study how the perturbation of model parameters affect the resulting cost. The example in analysis is a tiny multi-layer perceptron architecture (Fig. 1a). The architecture can be considered as a basic unit of a deep neural network, consisting of an input layer $\{x_1, x_2\}$, a hidden layer $\{h_1, h_2\}$, weights across different layers $\{\theta_1, \ldots, \theta_6\}$, and the output y. Neurons in the hidden layers apply the sigmoid function $\phi(t) = \frac{1}{1+e^{-t}}$ to the weighted input layer. Given input $\mathbf{x}$, $h_1(\mathbf{x}) = \phi(\theta_1 x_1 + \theta_3 x_2 + b_1)$ where θ_1, θ_3 are weights and b_1 is a bias, and the output is $y = \theta_5 h_1 + \theta_6 h_2 + b_3$. Weights and biases are tuned during training.

We use the model in Fig. 1a to evaluate a function $f(x_1, x_2) = x_1 \text{AND} x_2$ which outputs a binary number given $x_1, x_2 \in [0, 1]$. When x_1, x_2 are not integers, they are rounded up to the closest integer. For example, $f(0.3, 0.7) = 0$. We train on 1000 samples for 150 epochs to minimize the binary cross-entropy loss. The training accuracy achieves over 98% in the end. Then, we randomly choose any two parameters to perturb by z_1, z_2, and record how the resulting cost changes with respect to varying values of z_1, z_2.

In the experiments, we perturb θ_1 and b_3 by a value in $(0, 0.3)$ and get the resulting costs as shown by Fig. 1b. Likewise, Fig. 1c shows the cost when θ_6 and b_3 are perturbed. From Fig. 1b, one can tell that the least amount of additive noise (coordinate $(0.0, 0.0)$) does not always yield the least cost. Actually, the cost decreases when a larger positive noise is added to θ_1, which is in accordance with the direction indicated by $\frac{\partial C}{\partial \theta_1}$. On the contrary, the cost increases when a larger positive noise is added to b_3. Overall, if we are inserting a given amount of perturbation, we should spread more to θ_1 to keep the cost minimal. We have similar observations in Fig. 1c. As $\frac{\partial C}{\partial \theta_6}, \frac{\partial C}{\partial b_3} > 0$, any positive noise would increase the cost and such increase is even higher if we add a greater amount of noise to b_3 than to θ_6 since $\frac{\partial C}{\partial \theta_6} < \frac{\partial C}{\partial b_3}$.

A lesson learned from the experiment is that, non-uniform perturbation may incur less cost when carefully consider the perturbation impact of different parameters. Further, such impact can be estimated by each parameter's derivatives. In the follows, we will formulate a problem in seek of a directional noise that leads to less cost while guaranteeing differential privacy.

4.2 Problem Formulation

Following the DPSGD framework and our observation in Sec. 4.1, we choose to add carefully calibrated noise to each clipped gradients. The additive noise is sampled from a multi-dimensional distribution that minimizes the total perturbation cost.

Assume that

$$\mathbf{w} = (w_1, w_2, \ldots, w_d) \in \mathcal{D}^d$$

represents the impact of each parameter on the cost, which can be evaluated/approximated through multiple ways. For example, one can use domain knowledge that some features are more critical to the cost than others so that the parameters associated with those features have higher impact. It is also possible to derive the directions from SVD/PCA of the feature matrices to decide the impact. In this work, we simply use parameters' derivatives to estimate $\mathbf{w}$. However, such estimation would require access to the private training data and thus is privacy-leaking. We will later discuss how to compensate such leakage by spending additional privacy budget to turn $\mathbf{w}$ into $\tilde{\mathbf{w}}$.

Our additive noise mechanism is $\mathcal{K}(\mathbf{x}) = \mathbf{x} + \mathbf{z}$ where $\mathbf{z} = (z_1, \ldots, z_d)$ is drawn from a multi-dimensional probability

distribution designed by our optimized noise scheme. Letting $p(\cdot) \in \mathcal{P}$ denote the probability density function for $\mathbf{z}$, we aim to minimize the expected magnitude of the noise weighted by $\tilde{\mathbf{w}}$. For example, the weight of z_i is $|\tilde{w}_i|$: a larger $|\tilde{w}_i|$ indicates that the unit increment in θ_i will lead to a larger perturbation in the output, and thus it is desirable to keep the corresponding additive noise z_i small. Our optimization goal is as follows:

$$\underset{p \in \mathcal{P}}{\text{minimize}} \int_{\mathbf{z} \in \mathbb{R}^d} \|\tilde{\mathbf{w}} \circ \mathbf{z}\|_1 p(\mathbf{z}) \mathrm{d}\mathbf{z}. \tag{3}$$

In the equation above, $\tilde{\mathbf{w}} \circ \mathbf{z}$ represents the entry-wise product of the two vectors. $\mathrm{d}\mathbf{z}$ is short for $\mathrm{d}z_1 \ldots \mathrm{d}z_d$.

Next we study the privacy conditions that $\mathcal{P}$ should satisfy. We identify two leakage sources in our scenario: the utility subspace $\mathbf{w}$, which distinguishes the significance of the weights, as well as the released gradients computed on a randomly sampled input batch. We use $\mathbf{g}^t$ to denote the gradients calculated at the tth iteration. Generally, let $\mathbf{g}^t$ and $\mathbf{g}'^t$ be two gradient vectors respectively computed respectively on $\mathbf{X}$ and $\mathbf{X}'$, where the two differ by a single example instance. The global sensitivity is defined as

$$\alpha = \sup_{\forall \mathbf{X}, \mathbf{X}'} \|\mathbf{g}^t - \mathbf{g}'^t\|_2, \tag{4}$$

where $\|\cdot\|$ is the l_2 norm.

For our mechanism $\mathcal{K}$ to satisfy the differential privacy constraint, we split the total privacy budget (ϵ, δ) up to assign partial budget to $\mathbf{w}$ for optimization, and the rest to the release of the gradients. For example, we assign $(\epsilon/8, \delta/8)$ to $\mathbf{w}$ and $(7\epsilon/8, 7\delta/8)$ to $\mathbf{g}^t$. The noisy copy of $\tilde{\mathbf{w}}$ is used instead of $\mathbf{w}$ in the latter computation. Next we discuss how to release a noisy $\mathbf{g}^t$ which meets the condition of differential privacy.

Our key observation is a sufficient condition for differential privacy:

Lemma 1. *We have two datasets $\mathbf{X}$ and $\mathbf{X}'$ that differ by a single instance and $\mathbf{\Delta} = \mathbf{g}^t(\mathbf{X}) - \mathbf{g}^t(\mathbf{X}')$. For any output set $\mathcal{O}$, if*

$$\Pr\Big[\ln \frac{p(\mathbf{z})}{p(\mathbf{z}+\mathbf{\Delta})} > \epsilon\Big] < \delta,$$

$\mathcal{K}(\mathbf{g}^t) = \mathbf{g}^t(\mathbf{X}) + \mathbf{z}$ is (ϵ, δ)-differentially private.

Proof. We let $\mathbf{g}^t = \mathbf{g}^t(\mathbf{X})$ and $\mathbf{g}'^t = \mathbf{g}^t(\mathbf{X}')$, and $\|\mathbf{\Delta}\| \le \alpha$. We consider two events $\mathcal{S} = \{\mathbf{z} : \ln \frac{p(\mathbf{z})}{p(\mathbf{z}+\mathbf{\Delta})} > \epsilon\}$, and $\mathcal{S}^c = \{\mathbf{z} : \ln \frac{p(\mathbf{z})}{p(\mathbf{z}+\mathbf{\Delta})} \le \epsilon\}$. The sufficient condition can be written as $\Pr[\mathcal{S}] < \delta$. And the event of $\mathcal{S}^c$ represent the case where $\mathcal{K}$ satisfies ϵ-differential privacy in that, for any output set $\mathcal{O}$:

$$\begin{aligned}
&\Pr[\mathcal{K}(\mathbf{g}^t) \in \mathcal{O}] \le e^\epsilon \Pr[\mathcal{K}(\mathbf{g}'^t) \in \mathcal{O}] \\
\Leftrightarrow\ &\Pr[\mathbf{g}^t + \mathbf{z} \in \mathcal{O}] \le e^\epsilon \Pr[\mathbf{g}'^t + \mathbf{z} \in \mathcal{O}] \\
\Leftrightarrow\ &\Pr[\mathbf{z} \in \mathcal{O} - \mathbf{g}^t] \le e^\epsilon \Pr[\mathbf{z} \in \mathcal{O} - \mathbf{g}'^t] \\
\Leftrightarrow\ &\Pr[\mathbf{z} \in \mathcal{O}'] \le e^\epsilon \Pr[\mathbf{z} \in \mathcal{O}' + \mathbf{g}^t - \mathbf{g}'^t] \\
\Leftrightarrow\ &\Pr[\mathbf{z} \in \mathcal{O}'] \le e^\epsilon \Pr[\mathbf{z} \in \mathcal{O}' + \mathbf{\Delta}],
\end{aligned} \tag{5}$$

where $\mathcal{O}' = \mathcal{O} - \mathbf{g}^t \triangleq \{\forall \mathbf{o} : \mathbf{o} - \mathbf{g}^t\}$. Since the inequality has to hold for any $\mathcal{O}$, ϵ-differential privacy is met if and only if $\mathbf{z} \in \mathcal{S}^c$. Therefore, we can deduce the following if the sufficient condition of $\Pr[\mathcal{S}] < \delta$ holds

$$\begin{aligned}
&\Pr[\mathcal{K}(\mathbf{g}^t) \in \mathcal{O}] = \Pr[\mathbf{g}^t + \mathbf{z} \in \mathcal{O}] \\
&= \Pr[\mathbf{g}^t + \mathbf{z} \in \mathcal{O} \cap \mathcal{S}^c] + \Pr[\mathbf{g}^t + \mathbf{z} \in \mathcal{O} \cap \mathcal{S}] \\
&\le e^\epsilon \Pr[\mathbf{g}'^t + \mathbf{z} \in \mathcal{O} \cap \mathcal{S}^c] + \Pr[\mathcal{S}] \\
&\le e^\epsilon \Pr[\mathbf{g}'^t + \mathbf{z} \in \mathcal{O}] + \delta \\
&= e^\epsilon \Pr[\mathcal{K}(\mathbf{g}'^t) \in \mathcal{O}] + \delta,
\end{aligned}$$

which means that $\mathcal{K}$ satisfies (ϵ, δ)-differential privacy. The first inequality holds due to the definition of $\mathcal{S}^c$ and the second inequality holds because of the sufficient condition $\Pr[\mathcal{S}] < \delta$. Proof completes. □

We assume $\mathbf{z}$ is drawn from the probability density function $p(\mathbf{z})$. If we define the privacy loss variable $c = \ln \frac{p(\mathbf{z})}{p(\mathbf{z}+\mathbf{\Delta})}$ as the logarithmic distance between two adjacent noise distributions, it suffices to show $\Pr[c > \epsilon] < \delta$ for any $\mathbf{z}$ and for all $\mathbf{\Delta}$ that $\|\mathbf{\Delta}\| \le \alpha$ to ensure (ϵ, δ)-differential privacy.

To simplify the problem, we assume $\mathbf{z}$ is drawn from a multi-dimensional Gaussian distribution with zero mean and standard deviation $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$. Each dimension is assumed to be independent of each other. That is to say, $z_i \sim \mathcal{N}(0, \sigma_i^2)$. In essence, we search a probability distribution of the noise w.r.t. the cost direction while satisfying the differential privacy property:

$$\begin{aligned}
&\underset{\sigma_1, \ldots, \sigma_d}{\text{minimize}} \int_{\mathbf{z} \in \mathbb{R}^d} \|\tilde{\mathbf{w}} \circ \mathbf{z}\|_1 p(\mathbf{z}) \mathrm{d}\mathbf{z} \\
&\text{s.t.} \Pr\Big[\ln \frac{p(\mathbf{z})}{p(\mathbf{z}+\mathbf{\Delta})} > \epsilon\Big] < \delta, \forall \mathbf{z} \in \mathbb{R}^d, \forall \|\mathbf{\Delta}\| \le \alpha, \mathbf{\Delta} \in \mathbb{R}^d.
\end{aligned} \tag{6}$$

Prior to our work, Geng *et al*. [22] proved that the optimal pdf of the random noise to minimize its l_1 norm is a symmetric and staircase-shaped function when $d = 2$. However, adding the least amount of noise is not exactly what we want according to the observation in Sec. 4.1. Our goal is to release a perturbed model with high accuracy while satisfying differential privacy property at the same time.

4.3 Main Result

We assume a multivariate Gaussian noise p is applied where each of its dimensions is independent of one another (we assume the independence of different dimensions of the noise, not the gradient). Thus we present the pdf in a product form: $p(\mathbf{z}) = \prod_{i=1}^d p_i(z_i)$ with each $p_i(z_i)$ being the pdf of $\mathcal{N}(0, \sigma_i^2)$ from which the random noise z_i is drawn. We will show that Eqn. (6) can be transformed into a form which has a closed-form solution.

First, we rewrite the objective function as

$$\begin{aligned}
&\int_{\mathbf{z} \in \mathbb{R}^d} \|\tilde{\mathbf{w}} \circ \mathbf{z}\|_1 p(\mathbf{z}) \mathrm{d}\mathbf{z} = \int_{\mathbf{z} \in \mathbb{R}^d} \sum_{i=1}^d |\tilde{w}_i z_i| \cdot p(\mathbf{z}) \mathrm{d}\mathbf{z} \\
&= \sum_{i=1}^d \int_{z_i} |\tilde{w}_i z_i| \cdot \frac{1}{\sqrt{2\pi}\sigma_i} \exp\Big(-\frac{z_i^2}{2\sigma_i^2}\Big) \mathrm{d}z_i \\
&= 2 \sum_{i=1}^d \frac{|\tilde{w}_i|}{\sqrt{2\pi}} \int_0^{+\infty} \frac{z_i}{\sigma_i} \exp\Big(-\frac{z_i^2}{2\sigma_i^2}\Big) \mathrm{d}z_i = \sum_{i=1}^d \sqrt{2/\pi} |\tilde{w}_i| \sigma_i.
\end{aligned}$$

The last equation is due to $\int_0^{+\infty} x\exp(-x^2)\mathrm{d}x = 1/2$. Thus the objective is transformed to minimizing over the weighted sum of $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_d)$.

Next we analyze the constraints. We rewrite c with the expression of Gaussian distributions, and let $\boldsymbol{\Delta} = (\Delta_1, \ldots, \Delta_d)$ to get

$$c = \ln \frac{p(\mathbf{z})}{p(\mathbf{z}+\boldsymbol{\Delta})} = \ln \prod_{i=1}^{d} \frac{p(z_i)}{p(z_i+\Delta_i)} = \ln \prod_{i=1}^{d} \frac{\exp\Big(-z_i^2/2\sigma_i^2\Big)}{\exp\Big(-(z_i+\Delta_i)^2/2\sigma_i^2\Big)} = \sum_{i=1}^{d} \frac{2z_i\Delta_i + \Delta_i^2}{2\sigma_i^2}.$$

Since $z_i \sim \mathcal{N}(0, \sigma_i^2)$, we have

$$c \sim \mathcal{N}\Big(\sum_{i=1}^{d} \frac{\Delta_i^2}{2\sigma_i^2}, \sum_{i=1}^{d} \frac{\Delta_i^2}{\sigma_i^2}\Big).$$

We consider the composition of differential privacy mechanisms over k iterations. Due to the additive and scaling properties of Gaussian distributions, the composed privacy loss variable also follows the Gaussian distribution. If we assume the privacy loss variable in each iteration is drawn from the same distribution, we have

$$c \sim \mathcal{N}\Big(k\sum_{i=1}^{d} \frac{\Delta_i^2}{2\sigma_i^2}, k\sum_{i=1}^{d} \frac{\Delta_i^2}{\sigma_i^2}\Big).$$

We adopt the idea from moments accountant [5] to use a higher moment of the privacy loss variable c to give a larger feasible range of the constrained variables, which facilitates the finding of a globally optimal solution to Eqn. (6). By Markov inequality, we transform the constraint as:

$$\begin{aligned} &\Pr[c > \epsilon] = \Pr[\beta c > \beta\epsilon] = \Pr[\exp(\beta c) > \exp(\beta\epsilon)] \\ &\leq \mathbb{E}[\exp(\beta c)]/\exp(\beta\epsilon) \leq \delta, \end{aligned} \tag{7}$$

for any positive integer β.

It is worthwhile to point out that our composition is not a basic composition in which the privacy budget is simply added up. The composition method bears a similar principle to [5] in that a higher order of the privacy loss variable is adopted to derive the differential privacy constraint. However, in moments accountant, the privacy loss variable includes the sampling procedure while ours does not specifically include it. Hence in principle, our composition shares the same tightness bound as [5].

Incorporating the expression of c into the inequality, we have

$$\frac{\mathbb{E}[\exp(\beta c)]}{\exp(\beta\epsilon)} = \exp\Big(\frac{k(\beta+\beta^2)}{2}\sum_{i=1}^{d}\frac{\Delta_i^2}{\sigma_i^2} - \beta\epsilon\Big) \leq \delta,$$

$$\text{i.e.,} \quad \sum_{i=1}^{d} \frac{\Delta_i^2}{\sigma_i^2} \leq 2 \cdot \frac{\epsilon + (\ln\delta)/\beta}{k(1+\beta)}.$$

Letting $\tau(\epsilon, \delta) \triangleq 2 \cdot \frac{\epsilon + (\ln\delta)/\beta}{k(1+\beta)}$, Eqn. (6) can be rewritten as

$$\underset{\boldsymbol{\sigma}}{\text{minimize}} \quad \sum_{i=1}^{d} |\tilde{w}_i| \cdot \sigma_i \tag{8a}$$

$$\text{subject to} \quad \sum_{i=1}^{d} \Delta_i^2/\sigma_i^2 \leq \tau(\epsilon, \delta) \tag{8b}$$

$$\forall \Delta_i, \text{such that} \sum_{i=1}^{d} \Delta_i^2 \leq \alpha^2. \tag{8c}$$

We observe that Eqn. (8a) is an affine function of σ_i, and Eqn. (8b) is a convex function of σ_i. And thus Eqn. (8) satisfies the strong duality condition. The Lagrangian function is:

$$L(\boldsymbol{\sigma}, \lambda) = \sum_{i=1}^{d} |\tilde{w}_i| \cdot \sigma_i + \lambda\left(\sum_{i=1}^{d} \Delta_i^2/\sigma_i^2 - \tau(\epsilon, \delta)\right), \tag{9}$$

and the dual problem can be written as:

$$\max_{\lambda} \quad g(\lambda), \text{s.t.} \quad \lambda \geq 0, \tag{10}$$

where $g(\lambda) = \inf_{\boldsymbol{\sigma}} L(\boldsymbol{\sigma}, \lambda)$. Observing that $L(\boldsymbol{\sigma}, \lambda)$ is convex on $\boldsymbol{\sigma}$, by applying the first-order condition, we have

$$\sigma_i^* = \left(\frac{2\lambda\Delta_i^2}{|\tilde{w}_i|}\right)^{1/3}. \tag{11}$$

Substituting Eqn. (11) into Eqn. (10), we obtain a concave problem about λ. By applying the first-order condition on λ we have

$$\lambda^* = \left[\frac{3\tau(\epsilon, \delta)}{c_1 \sum_{i=1}^{d} (\Delta_i\tilde{w}_i)^{2/3}}\right]^{-3/2}, \tag{12}$$

where c_1 is a constant. By substituting λ^* back to the dual problem and the problem becomes:

$$\max_{\boldsymbol{\Delta}} \sum_{i=1}^{d} (\Delta_i\tilde{w}_i)^{\frac{2}{3}}$$

$$\forall \Delta_i, \text{such that} \sum_{i=1}^{d} \Delta_i^2 \leq \alpha^2.$$

The problem is convex on $\boldsymbol{\Delta}$ and when the optimality is achieved, it must hold that $\sum_{i=1}^{d} \Delta_i^2 = \alpha^2$. Together with the first-order condition, one can obtain

$$\Delta_i^* = \frac{\alpha\tilde{w}_i}{(\sum_{i=1}^{d} \tilde{w}_i^2)^{(1/2)}}. \tag{13}$$

Combining Eqns. (11), (12) and (13), we can obtain the optimal σ_i^* to problem (8) and hence problem (6).

Sampling and composition. In each iteration of training, we sample a batch of training instances to perform stochastic gradient descent. Hence the differential privacy guarantee is amplified according to [16]. Letting the total guarantee be

(ϵ, δ) and sampling rate be q, the probability that each instance is chosen is reduced by q throughout the training process, and hence the amplified privacy budget is $(\epsilon/q, \delta/q)$ overall.

In the previous proof, we compose the privacy loss variables over iterations, which is different from the composition that the moments accountant [5] has adopted. We take advantage of the fact that the privacy loss variable is a Gaussian random variable which is additive, to compose differential privacy mechanisms over k iterations.

4.4 Optimized Additive Noise Mechanism

Our mechanism can be considered as an instantiation of the differentially-private SGD. Overall, to achieve (ϵ, δ)-differential privacy for k iterations of SGD, we split the privacy budget to (ϵ_w, δ_w) and (ϵ_g, δ_g) for $\mathbf{w}$ and $\mathbf{g}$. In each iteration, we sample a batch of data at each iteration with sampling rate q, and the total privacy budget for releasing $\mathbf{g}$ is amplified as $(\epsilon_g/q, \delta_g/q)$. To compute the noise parameters, we first substitute $\tilde{\mathbf{w}}$ in Eqn. (6) with $\mathbf{1}$ to solve $\boldsymbol{\sigma}$ to generate the noise for $\mathbf{w}$. Then we generate randomized noise for $\mathbf{w}$ given $\boldsymbol{\sigma}$ and compute $\tilde{\mathbf{w}}$. Note that $\boldsymbol{\sigma}$ only need to be computed once but fresh randomized noise is required for different $\mathbf{w}$s in each iteration. With $\tilde{\mathbf{w}}$ in Eqn. (6), we can compute the optimized noise parameters for $\mathbf{g}^t$ and generate the randomized noise in each iteration. The detail of the proposed mechanism is shown in Algorithm 1.

Algorithm 1. Optimized Additive Noise Mechanism

Input: Training dataset $(\mathbf{X}, \mathbf{Y})$, total number of training examples n, cost function $C(\cdot)$, clipping value α, learning rate η^t, total iterations k, privacy parameters (ϵ, δ)

Output: kth iteration parameters $\boldsymbol{\theta}^k$

1: Split the privacy budget to (ϵ_w, δ_w) and (ϵ_g, δ_g).
2: Compute $\sigma_{w1}, \ldots \sigma_{wd}$ by substituting $\mathbf{w} = \mathbf{1}, \alpha, \epsilon = \epsilon_w, \delta = \delta_w$ to Eqn. (6).
3: **for** $t \in \{0, \ldots, k-1\}$ **do**
4: Compute $\mathbf{w}^t = \frac{1}{n} \cdot \frac{\partial C(\mathbf{X}, \mathbf{Y})}{\partial \boldsymbol{\theta}^t}$.
5: Generate the randomized noise $\mathbf{z} = (z_1, \ldots, z_d)$ such that $z_i \sim \mathcal{N}(0, \sigma_{wi}^2)$ and compute $\tilde{\mathbf{w}}^t = \mathbf{w}^t / \max(1, \|\mathbf{w}^t\|/\alpha) + \mathbf{z}$.
6: Randomly sample a batch $\mathcal{B}$ of training data $(\mathbf{X}_\mathcal{B}, \mathbf{Y}_\mathcal{B})$ with probability q.
7: **for** $(x, y) \in (\mathbf{X}_\mathcal{B}, \mathbf{Y}_\mathcal{B})$ **do**
8: Compute $\mathbf{g}_x^t = \nabla_{\boldsymbol{\theta}^t} C(\boldsymbol{\theta}^t, x, y)$.
9: Clip by α: $\bar{\mathbf{g}}_x^t = \mathbf{g}_x^t / \max(1, \|\mathbf{g}_x^t\|/\alpha)$.
10: **end for**
11: Compute the average: $\bar{\mathbf{g}}_\mathcal{B}^t = 1/|\mathcal{B}| \sum_{x \in \mathcal{B}} \bar{\mathbf{g}}_x^t$.
12: Compute $\sigma_1, \ldots \sigma_d$ by substituting $\tilde{\mathbf{w}}, \alpha, \epsilon = \epsilon_g/q, \delta = \delta_g/q$ to Eqn. (6).
13: Generate the randomized noise $\mathbf{z} = (z_1, \ldots, z_d)$ such that $z_i \sim \mathcal{N}(0, \sigma_i^2)$ and add them to $\bar{\mathbf{g}}_\mathcal{B}^t$: $\tilde{\mathbf{g}}_\mathcal{B}^t = \bar{\mathbf{g}}_\mathcal{B}^t + \mathbf{z}$.
14: Update $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta^t \tilde{\mathbf{g}}_\mathcal{B}^t$.
15: **end for**
16: Return $\boldsymbol{\theta}^k$.

In practice, we do not need to insert additive noise for every batch, but only apply the mechanism for every *lot*. A *lot* usually consists of multiple batches such that gradients are released every lot. We only consider the composition of the privacy mechanisms between different lots. We show that

Theorem 1. *Algorithm 1 satisfies (ϵ, δ)-differential privacy.*

Proof. To prove that, we need to show the definition of differential privacy holds true for any pair of adjacent inputs. Let $\mathbf{X}$ and $\mathbf{X}'$ be any pair of training dataset with a single entry difference, and w.l.o.g., we have $\mathbf{X} = \mathbf{X}' \cup x$. One can easily derive from Lemma 1 that $\tilde{\mathbf{w}}$ satisfies (ϵ_w, δ_w)-differential privacy. In each iteration of training, the sampled batch on the two datasets $\mathcal{B}, \mathcal{B}'$ differ by x with probability q. By the privacy amplification rule, the privacy budget consumed is (ϵ_g, δ_g) in total for the release of $\tilde{\mathbf{g}}_\mathcal{B}^t$ over k iterations. Hence Algorithm 1 satisfies (ϵ, δ)-differential privacy overall. □

Convergence. The convergence of Algorithm 1 is similar to that of the moments accountant [5] or stochastic gradient descent, since the noise added is of zero mean. We only manipulate the standard deviation of the noise to be added. Intuitively, the generated noise follows a distribution which in expectation yields the least impact to the output whereas satisfying differential privacy.

5 A General Form and a Lower Bound

In this section, we extend the optimized additive noise problem (Eqn. (6)) to a more general form. The general form extends the differential privacy concept to the well-understood mutual information, which shares the same spirit with [30], and provides insight to the utility-maximization framework in Sec. 4.2. In fact, the problem interestingly links to the celebrated distortion-rate function in communications, where the minimum distortion (maximum utility) can be derived under rate (privacy) constraints.

5.1 Privacy as Mutual Information Constraints

We consider the privacy constraint from an information theoretic perspective. Assuming $\mathbf{g}$ and $\tilde{\mathbf{g}}$ are the gradients vector before and after perturbation, i.e., $\mathcal{K}(\mathbf{g}) = \tilde{\mathbf{g}}$, we would like to directly constrain how much information $\tilde{\mathbf{g}}$ reveals about $\mathbf{g}$.

Conventionally, *mutual information* $I(Y; Z)$ represents the reduction in the uncertainty of Y given the knowledge of Z, and is defined as the relative entropy between the joint distribution $p(y, z)$ and the product distribution $p(y)p(z)$. Thus $I(\mathbf{g}; \tilde{\mathbf{g}})$ is a measure of the information leakage of the real gradients due to the release of $\tilde{\mathbf{g}}$. To link to our setup, we show that the differential privacy constraint on $\mathbf{g}$ implies that mutual information $I(\mathbf{g}; \tilde{\mathbf{g}})$ is bounded, i.e.,

Theorem 2 (differential privacy $\Longrightarrow$ mutual information bound). *If mechanism $\mathcal{K}(\mathbf{g})$ is ϵ-differentially private, then $I(\mathbf{g}; \tilde{\mathbf{g}}) \leq \epsilon(e^\epsilon - 1)/2$.*

We first introduce several notions of divergence between distributions and some properties.

Definition 2 (KL-Divergence). *The KL-Divergence, or relative entropy, between two random variables Y and Z is defined as*

$$D_{KL}(Y \| Z) = \mathbb{E}_{Y \sim \mathcal{Y}} \Big[\ln \frac{\Pr(Y = y)}{\Pr(Z = y)} \Big],$$

where if the support of Y is not equal to the support of Z, then $D_{KL}(Y\|Z)$ is not defined.

Definition 3 (Max Divergence). *The Max Divergence between two random variables Y and Z is defined as*

$$D_{\infty}(Y\|Z) = \max_{\mathcal{O}\subseteq \mathrm{Supp}(Y)} \Big[\ln \frac{\Pr(Y\in\mathcal{O})}{\Pr(Z\in\mathcal{O})}\Big],$$

where if the support of Y is not equal to the support of Z, then $D_{\infty}(Y\|Z)$ is not defined.

With the Max Divergence, we can rewrite the definition of ϵ-differential privacy as follows:

Definition 4. *A mechanism $\mathcal{M}$ is ϵ-differentially private if for all adjacent inputs I and I', and all possible output O,*

$$D_{\infty}(O|I\|O|I'), D_{\infty}(O|I'\|O|I) \leq \epsilon.$$

The relation between the Max Divergence and KL-Divergence is proven by [31] such that:

Lemma 2 ([31], Lemma 3.8). *For any two random variables Y and Z such that $D_{\infty}(Y\|Z), D_{\infty}(Z\|Y) \leq \epsilon$,*

$$D_{KL}(Y\|Z) \leq \epsilon(e^{\epsilon}-1)/2.$$

Lemma 2 indicates that when the Max Divergence between two distributions are bounded, the KL-Divergence indicating the average distance between the two is also bounded. Now we formally prove Theorem 2.

Proof (Theorem 2). We start by assuming $\mathcal{K}(\mathbf{g})$ satisfies ϵ-differential privacy. By Definition 4 and Lemma 2, the KL-Divergence between the conditional probabilities $\Pr(\tilde{\mathbf{g}}|\mathbf{g})$ and $\Pr(\tilde{\mathbf{g}}|\mathbf{g}')$ is bounded. On the other hand, we have

$$I(\mathbf{g};\tilde{\mathbf{g}}) = I(\tilde{\mathbf{g}};\mathbf{g}) = \mathbb{E}_{\mathbf{g}}[D_{KL}(\tilde{\mathbf{g}}|\mathbf{g}\|\tilde{\mathbf{g}})].$$

The second equality is obtained from the relation between mutual information and KL-Divergence. Next, we bound $D_{KL}(\tilde{\mathbf{g}}|\mathbf{g}\|\tilde{\mathbf{g}})$ for each instance $\mathbf{g}$. For any $\mathbf{g}'$ such that $\|\mathbf{g}-\mathbf{g}'\| \leq \alpha$, we have

$$\tilde{\mathbf{g}} = \mathbb{E}_{\mathbf{g}'}[\tilde{\mathbf{g}}|\mathbf{g}'].$$

Hence, for each $\mathbf{g}$,

$$\begin{aligned} D_{KL}(\tilde{\mathbf{g}}|\mathbf{g}\|\tilde{\mathbf{g}}) &= D_{KL}(\tilde{\mathbf{g}}|\mathbf{g}\|\mathbb{E}_{\mathbf{g}'}[\tilde{\mathbf{g}}|\mathbf{g}']) \\ &\leq \mathbb{E}_{\mathbf{g}'}[D_{KL}(\tilde{\mathbf{g}}|\mathbf{g}\|\tilde{\mathbf{g}}|\mathbf{g}')] \\ &\leq \epsilon(e^{\epsilon}-1)/2. \end{aligned}$$

The first inequality is due to $D_{KL}(\cdot)$ is convex in the second argument; and the second one follows from the KL-Divergence between each pair of instances $\tilde{\mathbf{g}}|\mathbf{g}$ and $\tilde{\mathbf{g}}|\mathbf{g}'$ is bounded. Therefore, we prove that the average of the above KL-Divergence over all instances of $\mathbf{g}$ is bounded, thus we have

$$I(\mathbf{g};\tilde{\mathbf{g}}) \leq \epsilon(e^{\epsilon}-1)/2.$$

To summarize the proof, we have: ϵ-differential privacy implies KL-Divergence is bounded, and the latter implies mutual information is bounded. □

The conclusion is similar to Lemma 1 of [30] except that we define the neighboring inputs based on their l_2-norm distance rather than the Hamming distance. The mathematical intuition is the same: differential privacy is defined on 'pairwise' requirements on distinguishability which can be considered as 'worst-case' privacy, but the mutual information measures the 'average' amount of information about $\mathbf{g}$ in $\tilde{\mathbf{g}}$, and thus defines an 'average-case' privacy. Cuff *et al.* prove the equivalence between differential privacy and conditional mutual information in [30]. However, they only show the case where the inputs are from a finite set, and the conclusion cannot be directly extended to continuous variables.

5.2 A General Form

Now we give a general form of the privacy-constrained cost optimization by relaxing its differential privacy constraint to the mutual information constraint: $I(\mathbf{g};\tilde{\mathbf{g}}) \leq R$ where $R \triangleq \epsilon(e^{\epsilon}-1)/2$. Instead of searching the optimal probability density function $p(\mathbf{z})$ that minimizes the total projected random noise, we assume the probability distribution of $\mathbf{g}$ — $\Pr[\mathbf{g}]$ is given and try to find the optimal posterior probability distribution $\Pr[\tilde{\mathbf{g}}|\mathbf{g}]$ over which the weighted perturbation is minimal. Note that this expression over $\tilde{\mathbf{g}}$ is more general than the additive noise mechanism since we do not impose on how $\tilde{\mathbf{g}}$ is obtained. Letting $\mathbf{w}$ be the impact of each parameter on the cost, we express the objective function as the expectation over the distributions of $\mathbf{g}$ and $\tilde{\mathbf{g}}$ such that:

$$\mathbb{E}[\hat{D}(\mathbf{g},\tilde{\mathbf{g}})] = \int_{\mathbf{g}\in\mathbb{R}^d}\int_{\tilde{\mathbf{g}}\in\mathbb{R}^d} \Pr[\mathbf{g}]\Pr[\tilde{\mathbf{g}}|\mathbf{g}]\cdot\|\mathbf{w}\circ(\tilde{\mathbf{g}}-\mathbf{g})\|_2^2 \mathrm{d}\tilde{\mathbf{g}}\mathrm{d}\mathbf{g}.$$

Thus the general form of the privacy-constrained cost optimization turns out to be:

$$\underset{\Pr[\tilde{\mathbf{g}}|\mathbf{g}]}{\text{minimize}} \quad \mathbb{E}[\hat{D}(\mathbf{g},\tilde{\mathbf{g}})] \tag{14a}$$

$$\text{subject to} \quad I(\mathbf{g};\tilde{\mathbf{g}}) \leq R. \tag{14b}$$

For known prior distribution of $\mathbf{g}$, we seek the optimal posterior probability distribution of $\Pr[\tilde{\mathbf{g}}|\mathbf{g}]$ that minimizes the expected distortion while satisfying the mutual information constraint. We interestingly found that the problem has a form of the classic distortion-rate problem if the privacy constraint is considered as the rate constraint. As a convention, we use $D(R)$ to denote the *distortion rate function*, which is the infimum of all distortions D for a given rate R such that (R, D) is in the rate distortion region

$$D(R) = \min_{\Pr[\tilde{\mathbf{g}}|\mathbf{g}]: I(\mathbf{g};\tilde{\mathbf{g}})\leq R} \mathbb{E}[\hat{D}(\mathbf{g},\tilde{\mathbf{g}})].$$

The distortion rate function defines the boundary of the rate distortion region, which contains all achievable rate distortion pairs. Our main result is a lower bound of the

distortion rate function $D(R)$ for the privacy constraint R under the following assumptions: 1) each element in $\mathbf{g}$ is independent of each other; 2) $\tilde{g}_i$ is only associated with g_i, and $\mathbb{E}[\tilde{g}_i] = g_i, \forall i$. Our formal statement is as follows:

Theorem 3 (Distortion Rate). *Let $h(\mathbf{g})$ denote the differential entropy of $\mathbf{g}$ in nats. Then for any R, $D(R) \geq D^*(R)$ such that*

$$\beta_i = w_i^2 e^{2h(g_i)}/(2\pi e), \tag{15a}$$

$$D_i = \begin{cases} \lambda & if\ \lambda < \beta_i \\ \beta_i & if\ \lambda \geq \beta_i \end{cases}, \forall i \in \{1, 2, \ldots, d\}, \tag{15b}$$

$$R = h(\mathbf{g}) - \sum_{i=1}^{d} \frac{1}{2}\left(1 + \ln\left(\frac{2\pi D_i}{w_i^2}\right)\right), \tag{15c}$$

$$D^*(R) = \sum_{i=1}^{d} D_i. \tag{15d}$$

The boundary of the rate distortion region $D(R)$ can be equivalently expressed as the *rate distortion function* $R(D)$. For a given distortion D, the rate is:

$$R(D) = \min_{\Pr[\tilde{\mathbf{g}}|\mathbf{g}]:\mathbb{E}[\hat{D}(\mathbf{g},\tilde{\mathbf{g}})]\leq D} I(\mathbf{g}; \tilde{\mathbf{g}}).$$

An important property of $R(D)$ is its convexity:

Lemma 3 ([32] Lemma 10.4.1.). *The rate distortion function $R(D)$ is a non-increasing convex function of D.*

Likewise, we can also prove $D(R)$ to be a non-increasing convex function of R. To prove Theorem 3, we first prove a lower bound of the rate distortion function:

Lemma 4 (Rate Distortion). *Let $h(\mathbf{g})$ be the differential entropy of $\mathbf{g}$ in nats. Then for any D, $R(D) \geq R^*(D)$ such that*

$$\beta_i = w_i^2 e^{2h(g_i)}/(2\pi e), \tag{16a}$$

$$D_i = \begin{cases} \lambda & if\ \lambda < \beta_i \\ \beta_i & if\ \lambda \geq \beta_i \end{cases}, \forall i \in \{1, 2, \ldots, d\}, \tag{16b}$$

$$D = \sum_{i=1}^{d} D_i, \tag{16c}$$

$$R^*(D) = h(\mathbf{g}) - \sum_{i=1}^{d} \frac{1}{2}\left(1 + \ln\left(\frac{2\pi D_i}{w_i^2}\right)\right). \tag{16d}$$

Proof. Let $\mathbf{g} = \{g_1, \ldots, g_d\}, \tilde{\mathbf{g}} = \{\tilde{g}_1, \ldots, \tilde{g}_d\}$ be random variables before and after distortion. The distortion vector is $\mathbf{z} = \{z_1, \ldots, z_d\}$ of which each dimension is independent. By definition,

$$I(\mathbf{g}; \tilde{\mathbf{g}}) = I(\mathbf{z}; \tilde{\mathbf{g}}) = h(\mathbf{z}) - h(\mathbf{z}|\tilde{\mathbf{g}}) \tag{17a}$$

$$= \sum_{i=1}^{d} h(z_i) - \sum_{i=1}^{d} h(z_i|z_1, \ldots, z_{i-1}, \tilde{\mathbf{g}}) \tag{17b}$$

$$= \sum_{i=1}^{d} h(z_i) - \sum_{i=1}^{d} h(z_i|\tilde{g}_i) \tag{17c}$$

$$= \sum_{i=1}^{d} I(z_i; \tilde{g}_i) = \sum_{i=1}^{d} I(g_i; \tilde{g}_i), \tag{17d}$$

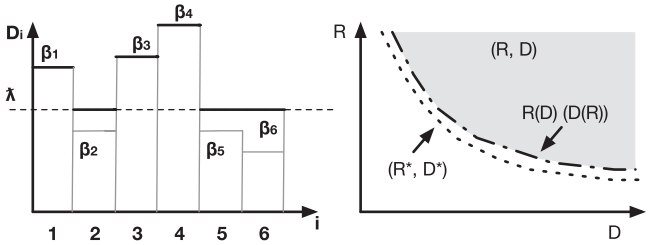


Fig. 2. Left: an illustrative example for the minimum distortion. Right: Rate distortion region, rate distortion function and a lower bound R^*.

where $\forall i = 1, \ldots, d$. We have for each i,

$$I(g_i; \tilde{g}_i) = h(g_i) - h(z_i|\tilde{g}_i) \tag{18a}$$

$$\geq h(g_i) - h(z_i) \tag{18b}$$

$$\geq h(g_i) - h(\mathcal{N}(0, \mathbb{E}[z_i^2])) \tag{18c}$$

$$= h(g_i) - \frac{1}{2}\left(1 + \ln\left(\frac{2\pi D_i}{w_i^2}\right)\right). \tag{18e}$$

The inequalities of (18b) follow from the fact that conditioning reduces entropy. Inequality (18c) is because the normal distribution maximizes the entropy for a given second moment (a given l_2-norm distortion). We also have $\mathbb{E}[\hat{D}(g_i, \tilde{g}_i)] = w_i^2 \mathbb{E}[(g_i - \tilde{g}_i)^2] \triangleq D_i$, which leads to Eqn. (18e). Apparently, the equality of Eqn. (18b) cannot be achieved, hence the rate R^* cannot be within the achievable (R, D) region.

Since $I(g_i; \tilde{g}_i) \geq 0$, the problem of finding the lower bound of rate distortion function becomes the following convex optimization problem:

$$R^*(D) = \min_{\sum D_i = D} \sum_{i=1}^{d} \max\left\{h(g_i) - \frac{1}{2}\left(1 + \ln\left(\frac{2\pi D_i}{w_i^2}\right)\right), 0\right\}.$$

By introducing Lagrange multipliers, we construct:

$$J(D) = \sum_{i=1}^{d}\left(h(g_i) - \frac{1}{2}\left(1 + \ln\left(\frac{2\pi D_i}{w_i^2}\right)\right)\right) + \lambda \sum_{i=1}^{d} D_i. \tag{19}$$

We use the KKT conditions to find the minimum in Eqn. (19), that is:

$$\frac{\partial J}{\partial D_i} = -\frac{1}{2}\frac{1}{D_i} + \lambda,$$

where λ should be chosen such that

$$\frac{\partial J}{\partial D_i}\begin{cases} = 0, if\ D_i < \beta_i, \\ \leq 0, if\ D_i \geq \beta_i. \end{cases} \tag{20}$$

It is easy to verify that Eqn. (16d) satisfies Eqn. (20), and the KKT conditions are satisfied by Eqn. (16). □

Now we prove Theorem 3 with the help of Fig. 2.

Proof. (Theorem 3.) We prove it by three steps. First, as $R^*(D)$ is another expression of $D^*(R)$, any (R^*, D^*) pair on $R^*(D)$ must be on $D^*(R)$. And the opposite is true.

Second, we show that any (R, D) pair on $R(D)$ is also on $D(R)$ by contradiction. Let $(R(D_1), D_1)$ be any point on $R(D)$. We assume $D(R(D_1)) \neq D_1$. If $D(R(D_1)) > D_1$, it does not agree with the definition of distortion rate function, since there exists a point in (R, D) region with smaller distortion when $R = R(D_1)$. If $D(R(D_1)) < D_1$, by the complementary argument of Lemma 3, $R(D_1) \geq D^{-1}(D_1)$ as $D(R)$ is a non-increasing function of R. It turns out $R(D_1)$ has to be larger than $D^{-1}(D_1)$ since $R(D_1) \neq D^{-1}(D_1)$ by our assumption. However, $R(D_1) > D^{-1}(D_1)$ violates the definition of rate distortion function since $(D^{-1}(D_1), D_1)$ is achievable. Hence this assumption is wrong in the first place: for any D_1 on $R(D)$, $D(R(D_1)) = D_1$. We can also prove any point on $D(R)$ is on $R(D)$ as well.

By Lemma 4, we have $R^*(D) \leq R(D), \forall D$. Now we prove $D^*(R)$ is no larger than $D(R)$ for any R by contradiction. Assume $\exists R^*$ on $R^*(D)$ such that $D^*(R^*) > D(R^*)$. By the non-increasingness of $R(D)$, $R(D^*(R^*)) \leq R(D(R^*)) = R^*$. Since R^* is the lower bound of $R(\cdot)$, it has to be $R(D^*(R^*)) = R^*$, which leads to $D(R(D^*(R^*))) = D(R^*)$. Note that we also have $D(R(D^*(R^*))) = D^*(R^*)$ since any (R, D) pair on $R(D)$ is also on $D(R)$. Hence we have $D^*(R^*) = D(R^*)$ and arrive at a contradiction. The theorem is therefore established.

Eqn. (15a) gives a lower bound to the expected distortion in the distortion rate function $D(R)$, which is also a lower bound to the privacy-constrained cost optimization problem. Proof completes. □

Given the distribution of $\mathbf{g}$ and the privacy constraint R, there are $d + 1$ unknown variables in Eqn. (15c) and Eqn. (15b). Hence we can decide the values for $D_1, \ldots, D_d$, from which $D^*(R)$ can be determined. It is obvious that when the privacy constraint is given, $D^*(R)$ gives a lower bound to the minimum of the expected distortion $\mathbb{E}[D(\mathbf{g}, \tilde{\mathbf{g}})]$. Since Eqn. (14b) is a necessary condition on ϵ-differential privacy according to Theorem 2, the mutual information bound is more relaxed than the differential privacy constraint. Thus $D^*(R)$ is also a lower bound to the expected distortion given the differential privacy constraints.

The left subfigure of Fig. 2 shows an illustrative example of the distortion results by Theorem 3. When the privacy is constrained by R, the minimum distortion is akin to reverse water-filling, in that all distortions above a constant λ is expressed, and all that less than λ adopts λ as the distortion. It means for independent Gaussian perturbations, there is a minimum perturbation threshold for each g_i to meet the mutual information privacy constraint.

6 Evaluation

Despite its theoretical guarantee, it remains a challenge to solve Eqn. (6) due to the high-dimensional model parameters. In this section, we introduce our setup on three conventional machine learning datasets and classic models. Implementation details on the privacy mechanisms on `TensorFlow` and`Pytorch` are followed. Finally, we compare the evaluation results of our design against the state-of-the-art.

TABLE 1
Experimental Setup

	MNIST	SVHN	CIFAR-10
train/test instances	55000/5000	73257/26032	60000/10000
model	LeNet	AlexNet	ResNet-18
pruning ratio	0	0.9	0.9
batch size	64	256	256
lot size	512	256	256
clip value	0.1	2.0	2.0
training epochs	60	30	100

6.1 Experimental Setup

The default experimental setting is given in Table 1. MNIST is a standard image dataset for handwritten digit recognition, with each image containing 28×28 gray-level pixels. LeNet is used as the base model and if trained without perturbation, the model reaches 98.32% testing accuracy. SVHN is a real-world dataset which comes from house numbers in Google Street View images. We adopt a modified AlexNet with kernel sizes of all convolutional layers set to 3 and it reaches 93.01% testing accuracy when unperturbed. Each image in CIFAR-10 dataset follows $32 \times 32 \times 3$ RGB format and is trained using ResNet-18. The unperturbed accuracy of CIFAR-10 reaches 91.43%. We perform DPSGD from the start for all model parameters of MNIST. While for SVHN and CIFAR-10, as we have found few working DPSGD methods on large-scale models in the current literature, we compromise by performing DPSGD on compressed (pruned) gradients and update the model by those gradients. In each iteration, we prune the gradients by their magnitude and add differentially-private noise to the left gradients. The pruned gradients are set to zeros. Models are updated only by the non-zero gradients while the pruning error is compensated according to Algorithm 2 of [33]. Pruning errors in each iteration are added back in the next iteration to the gradients between line 8 and 9 of Algorithm 1. The pruning ratio defaults to 0.9. Note that different from [5], we do not pre-train models and all models are trained from scratch.

6.2 Implementation

We have implemented a general-purpose framework on `TensorFlow` and `Pytorch`, and it supports conventional and customized datasets with pluggable models and provides a convenient user interface. The privacy mechanisms have been implemented as a noise generation module in our framework. The module contains four noise generators: 1) *Gaussian*: a Gaussian noise generator following the implementation of moments accountant in [5]; 2) *AdaClip*: a Gaussian noise generator with adaptive gradient clipping schemes from [24]; 3) a version of our optimized noise generator without assigning the privacy budget in computing $\tilde{\mathbf{w}}$ but simply setting it to $\mathbf{1}$, denoted as **Ours ($\mathbf{w} = \mathbf{1}$)**; 4) a full version of Algorithm 1, denoted as *Ours* in the following. For the latter two noise generators, we built in the optimal solutions obtained by Eqn. (11), (12) and (13). The noise generator is implemented as a part of the computation graph using tensor operations

TABLE 2
Average Batch Processing Time (s) on MNIST

Unperturbed	Gaussian	Ours	Ours ($\mathbf{w} = \mathbf{1}$)	AdaClip
1.50	1.53	2.19	1.70	1.95

to take advantage of GPU batch processing. The `Pytorch` version implements gradients pruning and error feedback.

For fair comparison, we adopt the same training hyperparameters as in Table 1 across all experiments. For Gaussian and AdaClip, moments accountant is used as the composition method. For AdaClip, we pick the best γ according to the accuracy performance. On MNIST, γ is set to 0.01 meaning that 99% of all gradients are clipped.

To evaluate the additional computation overhead incurred by the privacy mechanisms, we measured the average batch processing time of each scheme by the `TensorFlow` implementation on `RTX 3090`. The results are shown in Table 2, which suggest that the computation overhead of Ours ($\mathbf{w} = \mathbf{1}$) only increases mildly (11%) over the Gaussian on MNIST, due to the additional distribution parameter generation step, which shows the efficiency of the closed-form solutions. The computation overhead of Ours is comparatively higher since it requires to compute $\tilde{\mathbf{w}}$ in each iteration. The batch processing time of AdaClip is between Ours and Ours ($\tilde{\mathbf{w}} = \mathbf{1}$).

6.3 Comparison

We compare our mechanisms with the unperturbed case and state-of-the-art baselines in this section. The unperturbed accuracy represents the performance of the model without privacy. The Gaussian method implements the moments accountant method which tightly composes privacy budget over iterations, whereas AdaClip improves over conventional differentially-private SGD by adaptively clipping gradients. We compare AdaClip with our method on MNIST since it was previously conducted on MNIST [24].

Accuracies versus Iterations. We first show the testing accuracy per training iteration on the three datasets in Figs. 3a, 3b, and 3c, where we pick three representative curves at different privacy levels. It can be observed for the unperturbed case, accuracies ramp up quickly in the first few epochs. From Fig. 3a, we can tell that our method obtains the highest accuracy among all, with highly stable performance, followed by Ours ($\mathbf{w} = \mathbf{1}$) and Gaussian. AdaClip is weaker than Gaussian and is quite unstable according to our observation. We think this may be because varied clipping values at the lot aggregation step leading to unpredictable gradient descent directions. On SVHN (Fig. 3b) and CIFAR-10 (Fig. 3c), we observe Ours ($\mathbf{w} = \mathbf{1}$) obtains the highest accuracies followed by Ours. The moments accountant method is inferior in accuracy, especially on large models like ResNet-18. As the gradients are pruned on SVHN and CIFAR-10, it is analyzed that each single gradient is more sensitive to the noise and thus Ours performs worse since it inserts higher noise than Ours ($\mathbf{w} = \mathbf{1}$). This also explains the accuracy drop around Epoch 25 in Fig. 3b that the perturbation effect takes over training. But still, the optimized additive noise mechanism is better than the baseline.

Accuracies versus Privacy Parameters. One may argue that it is not fair to argue the superiority of our algorithm's performance since the baselines may not converge at the same number of training epochs. In fact, the increase of training iterations has contradictory effects to accuracy: while it continues to improve training accuracy by minimizing the cost function, the model also suffers from further perturbations as additional noise is inserted. To eliminate the concern about the stopping condition, we conduct experiments with different budget schemes and privacy parameters where we report the highest testing accuracies throughout training in each setting.

We show the results in Figs. 4a, 4b, and 4c. First of all, our method exceeds other baselines by a large margin in the high privacy regime on MNIST: the gap between Ours and Gaussian is up to 9% when $\epsilon = 0.2$. On SVHN, Ours ($\mathbf{w} = \mathbf{1}$) yields 92.47% accuracy when $\epsilon = 5.0$, fairly close to the unperturbed case while the moments accountant reaches less than 40% accuracy at that privacy level, showing the advantage of optimizing noise hyperparameters. On CIFAR-10, our methods show a greater improvement at the low privacy level (e.g., $\epsilon = 15.0$), and this may be because CIFAR-10 on ResNet is more complicated to learn, and our method is less competitive when

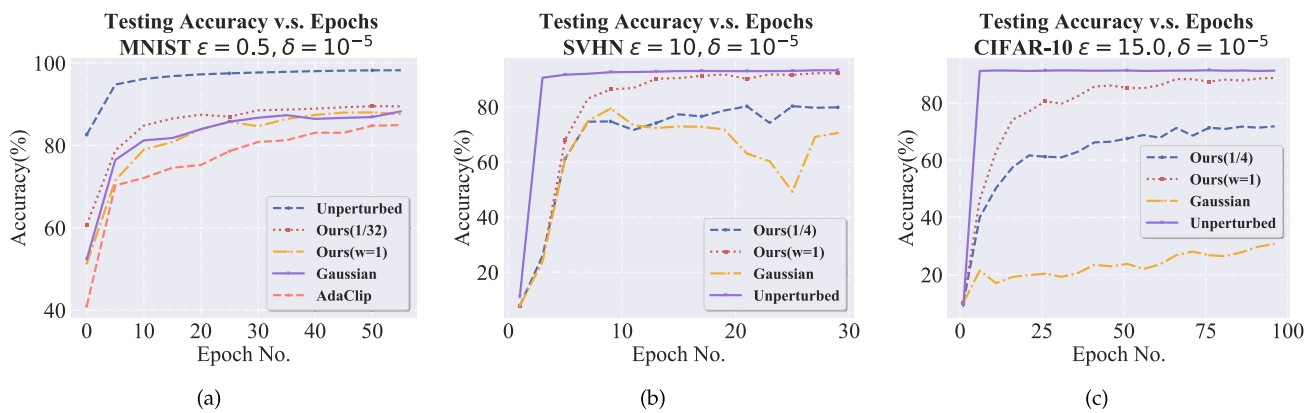


Fig. 3. Testing accuracy versus training epochs on MNIST (a), SVHN (b), and CIFAR-10 (c). Our method obtains the best convergence and accuracy performance among all. Ours (1/4) represents our method in which $\epsilon_w = \frac{1}{4}\epsilon$.

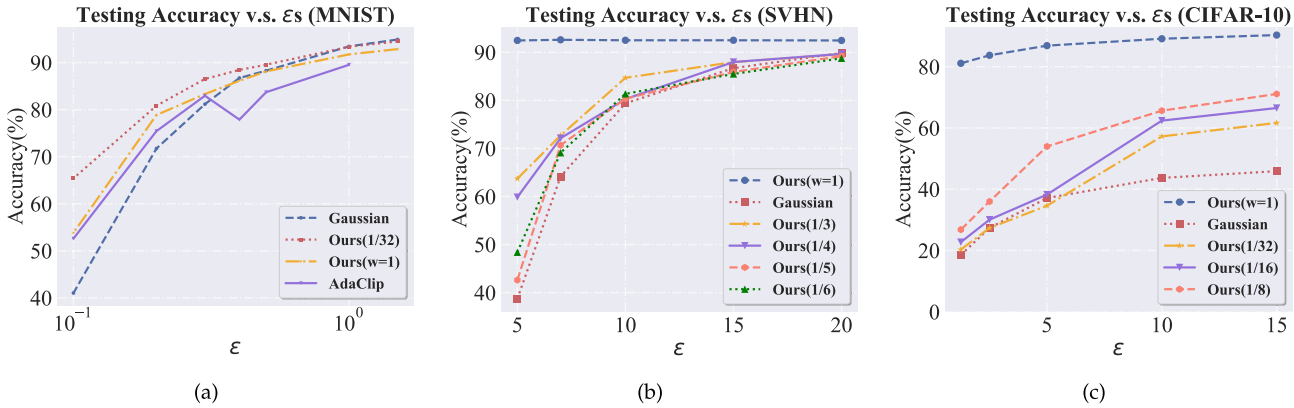


Fig. 4. Testing accuracy versus privacy parameters on MNIST (a), SVHN (b), and CIFAR-10 (c). $\delta = 10^{-5}$ in all cases. On MNIST, the comparison with baselines show our method surpasses others across a variety of privacy settings. On SVHN and CIFAR-10, results under different privacy budget schemes are displayed, e.g., Ours (1/3) denotes the scheme in which $\epsilon_w = \frac{1}{3}\epsilon$. Our methods yields higher accuracy than the Gaussian under all privacy budget schemes.

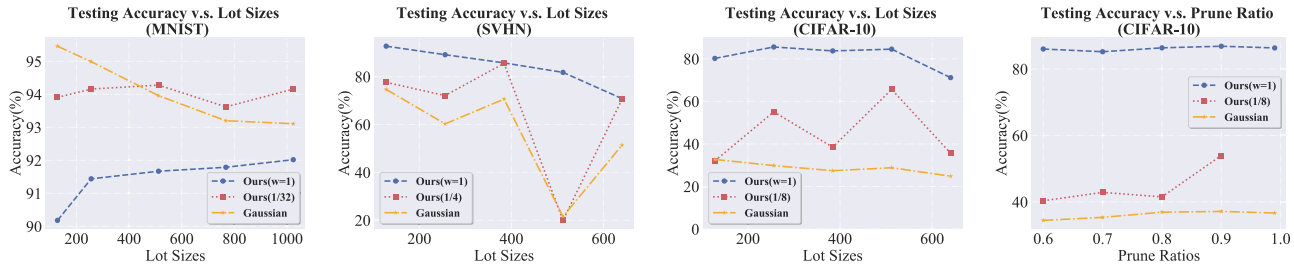


Fig. 5. (a)(b)(c) Testing accuracy versus lot sizes on MNIST, SVHN and CIFAR-10. We set $(\epsilon = 1.0, \delta = 10^{-5})$ on MNIST, $(\epsilon = 10.0, \delta = 10^{-5})$ on SVHN, and $(\epsilon = 5.0, \delta = 10^{-5})$ on CIFAR-10. (d) Testing accuracy versus pruning ratios on CIFAR-10 $(\epsilon = 5.0, \delta = 10^{-5})$.

the privacy level is high. For all mechanisms, the accuracy enhances with a growing ϵ, showing the tradeoff between privacy and utility. We also found that the privacy budget assignment only mildly affects the accuracy performance. In general, when we assign a slightly larger budget to ϵ_w, the accuracy is higher. The results verify that our optimized additive noise mechanisms are effective in both low and high privacy regimes.

6.4 Sensitivity to Hyperparameters

Privacy and accuracy cannot be discussed without mentioning the hyperparameters and the specific neural network structures. We show that our optimized mechanism is robust in most cases. We mainly consider the following hyperparameters: lot sizes, gradients pruning ratios, and l_2 clipping values. We reuse the previous setting except that the batch size is set to 128 on SVHN and CIFAR-10 for the lot size experiments.

By Figs. 5a, 5b, and 5c, we find that the accuracy performance in general decays with the lot size. In particular, the accuracies drop to around 20% on SVHN at lot size 512 for both Ours and the moment accountant method but surge up at a larger lot size. As we analyze, growing lot sizes have contradictory impact to the accuracy: both the inserted noise magnitude and the total training iterations reduce. Hence it is possible to yield deteriorated performance when an inappropriate lot size is chosen. Pruning ratios should also be taken into consideration in the design. In Fig. 5d, it is observed that when the pruning ratio is extremely high (over 0.9), training does not converge in Ours. But even at a high pruning ratio (e.g., 0.99), the training performance of Ours ($\mathbf{w} = \mathbf{1}$) and the moment accountant method is not much different from any lower pruning ratio. This may due to inaccurate approximation of the impact of each parameter. The sensitive ranges of clipping values vary across different datasets. For example, on MNIST (Fig. 6a), the performance is robust to clipping value variation in the range of 0.01 to 0.1 while degrades significantly with a larger clipping value. On SVHN (Fig. 6b), our methods are not sensitive to the change of clipping values in the entire range from 0.5 to 6.0, while the moments accountant decays over larger clipping values.

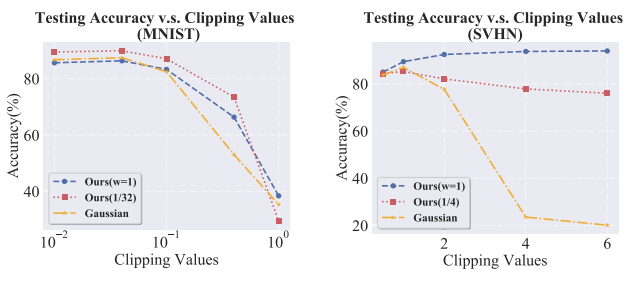


Fig. 6. Testing accuracy versus clipping values on MNIST $(\epsilon = 0.3, \delta = 10^{-5})$ and SVHN $(\epsilon = 10.0, \delta = 10^{-5})$.

7 Conclusion

In this work, we seek an optimized differential privacy mechanism for deep learning with stochastic gradient descent. The problem is formulated as a constrained optimization which minimizes the loss over a set of differential privacy constraints. The high dimensionality of the problem is a major obstacle, so we tackle it from both a theoretical and an engineering perspective. Further, a general form of the problem is introduced, which has a theoretical solution rooted in the distortion-rate problem. Evaluations on a variety of datasets and settings have demonstrated that our proposed privacy mechanism improves the model accuracy at all privacy levels under proper choice of hyperparameters.

References

[1] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1322–1333.

[2] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy*, , 2017, pp. 3–18.

[3] C. Dwork, "A firm foundation for private data analysis," *Commun. ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[4] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1310–1321.

[5] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 308–318.

[6] J. Lee and D. Kifer, "Concentrated differentially private gradient descent with adaptive per-iteration privacy budget," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1656–1665.

[7] L. Yu, L. Liu, C. Pu, M. E. Gursoy, and S. Truex, "Differentially private model publishing for deep learning," in *Proc. 40th IEEE Symp. Secur. Privacy*, 2019, pp. 332–349.

[8] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *Proc. 5th Int. Conf. Learn. Representations (ICLR)*, 2017.

[9] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, Ú. Erlingsson, "Scalable private learning with PATE," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

[10] J. Zhang, Z. Zhang, X. Xiao, Y. Yang, and M. Winslett, "Functional mechanism: Regression analysis under differential privacy," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1364–1375, 2012.

[11] N. Phan, Y. Wang, X. Wu, and D. Dou, "Differential privacy preservation for deep auto-encoders: An application of human behavior prediction," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1309–1316.

[12] S. Song, K. Chaudhuri, and A. D. Sarwate, "Stochastic gradient descent with differentially private updates," in *Proc. IEEE Global Conf. Signal Inf. Process.*, 2013, pp. 245–248.

[13] P. Kairouz, S. Oh, and P. Viswanath, "The composition theorem for differential privacy," *IEEE Trans. Inf. Theory*, vol. 63, no. 6, pp. 4037–4049, Jun. 2017.

[14] A. Beimel, S. P. Kasiviswanathan, and K. Nissim, "Bounds on the sample complexity for private learning and private data release," in *Proc. Theory Cryptogr. Conf.*, 2010, pp. 437–454.

[15] N. Phan, X. Wu, H. Hu, and D. Dou, "Adaptive laplace mechanism: Differential privacy preservation in deep learning," in *Proc. IEEE Int. Conf. Data Mining*, 2017, pp. 385–394.

[16] R. Bassily, A. Smith, and A. Thakurta, "Private empirical risk minimization: Efficient algorithms and tight error bounds," in *Proc. IEEE 55th Annu. Symp. Found. Comput. Sci.*, 2014, pp. 464–473.

[17] D. Wang, M. Ye, and J. Xu, "Differentially private empirical risk minimization revisited: Faster and more general," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, pp. 2719–2728.

[18] J. Zhang, K. Zheng, W. Mou, and L. Wang, "Efficient private ERM for smooth objectives," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 3922–3928.

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

[20] X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton, "Bolt-on differential privacy for scalable stochastic gradient descent-based analytics," in *Proc. ACM Int. Conf. Manage. Data*, 2017, pp. 1307–1322.

[21] Q. Geng and P. Viswanath, "The optimal noise-adding mechanism in differential privacy," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 925–951, Feb. 2015.

[22] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath, "The staircase mechanism in differential privacy," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 7, pp. 1176–1184, Oct. 2015.

[23] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proc. 39th Annu. ACM Symp. Theory Comput.*, 2007, pp. 75–84.

[24] V. Pichapati, A. T. Suresh, F. X. Yu, S. J. Reddi, and S. Kumar, "AdaCliP: Adaptive Clipping for Private SGD," 2019, *arXiv: 1908.07643*.

[25] C. Dwork, G. N. Rothblum, and S. Vadhan, "Boosting and differential privacy," in *Proc. IEEE 51st Annu. Symp. Found. Comput. Sci.*, 2010, pp. 51–60.

[26] I. Mironov, "Rényi differential privacy," in *Proc. IEEE 30th Comput. Secur. Found. Symp.*, 2017, pp. 263–275.

[27] S. Yeom, M. Fredrikson, and S. Jha, "The unintended consequences of overfitting: Training data inference attacks," 2017, *arXiv: 1709.01604*.

[28] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 372–387.

[29] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1885–1894.

[30] P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 43–54.

[31] C. Dwork and G. N. Rothblum, "Concentrated differential privacy," 2016, *arXiv:1603.01887*.

[32] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.

[33] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes sign-SGD and other gradient compression schemes," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3252–3261.

Liyao Xiang (Member, IEEE) received the BEng degree in electrical and computer engineering from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the PhD degree in computer engineering from the University of Toronto, Toronto, ON, Canada, in 2018. She is currently an associate professor with Shanghai Jiao Tong University. Her research interests include security and privacy, privacy analysis in data mining, and mobile computing.

Weiting Li received bachelor's degree from the Department of Electronic Engineering, Shanghai Jiao Tong University, in 2020. He is currently working toward the master's degree at the Department of Electronic Engineering, Shanghai Jiao Tong University. His research interests include the privacy preserving of machine learning.

Jungang Yang received the bachelor's degree in information and computing science from Nanjing University, Nanjing, China, in 2019. He is currently working toward the PhD degree in computer science at Shanghai Jiao Tong University. His research interests include privacy analysis and adversarial learning.

Xinbing Wang (Senior Member, IEEE) received the BS degree (with Hons.) from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 1998, the MS degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001, and the PhD degree major from the Department of electrical and Computer Engineering, minor from the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is currently a professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. He has been an associate editor for *IEEE/ACM Transactions on Networking* and *IEEE Transactions on Mobile Computing*, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, ACM MobiHoc 2012-2014, IEEE INFOCOM 2009-2017.

Baochun Li (Fellow, IEEE) received the BE degree from the Department of Computer Science and Technology, Tsinghua University, China, in 1995, and the MS and PhD degrees from the Department of Computer Science, University of Illinois at Urbana Champaign, Champaign, in 1997 and 2000, respectively. He held the Nortel Networks Junior Chair with Network Architecture and Services from 2003 to 2005. He has been the Bell Canada Endowed chair in computer engineering since 2005. Since 2000, he has been with the Department of Electrical and Computer Engineering, University of Toronto, where he is currently a professor. His research interests include cloud computing, largescale data processing, computer networking, and distributed systems. He was a recipient of the IEEE Communications Society Leonard G. Abraham Award in the Field of Communications Systems in 2000. He was a recipient of the Multimedia Communications Best Paper Award from the IEEE Communications Society in 2009, and a recipient of the University of Toronto McLean Award. He is a member of ACM.