Input Perturbation: A New Paradigm between Central and Local Differential Privacy

Yilin Kang  
kangyilin@iie.ac.cn  
Institute of Information Engineering,  
Chinese Academy of Sciences

Yong Liu  
liuyong@iie.ac.cn  
Institute of Information Engineering,  
Chinese Academy of Sciences

Ben Niu  
niuben@iie.ac.cn  
Institute of Information Engineering,  
Chinese Academy of Sciences

Xinyi Tong  
tongxinyi@buaa.edu.cn  
Beihang University

Likun Zhang  
2016312357@email.cufe.edu.cn  
Central University of Finance and Economics

Weiping Wang  
wangweiping@iie.ac.cn  
Institute of Information Engineering,  
Chinese Academy of Sciences

ABSTRACT
Traditionally, there are two models on differential privacy: the central model and the local model. The central model focuses on the machine learning model and the local model focuses on the training data. In this paper, we study the input perturbation method in differentially private empirical risk minimization (DP-ERM), preserving privacy of the central model. By adding noise to the original training data and training with the 'perturbed data', we achieve \((\epsilon,\delta)\)-differential privacy on the central model, along with some kind of privacy on the original data. We observe that there is an interesting connection between the local model and the central model: the perturbation on the original data causes the perturbation on the gradient, and finally the model parameters. This observation means that our method builds a bridge between local and central model, protecting the data, the gradient and the model simultaneously, which is more superior than previous central methods. Detailed theoretical analysis and experiments show that our method achieves almost the same (or even better) performance as some of the best previous central methods with more protections on privacy, which is an attractive result. Moreover, we extend our method to a more general case: the loss function satisfies the Polyak-Lojasiewicz condition, which is more general than strong convexity, the constraint on the loss function in most previous work.

KEYWORDS
 differential privacy, machine learning, input perturbation, empirical risk minimization

1 INTRODUCTION
In recent years, machine learning has been shown effective in fields such as pattern recognition and data mining [48], [34], [8], [23] and large quantities of personal data has been collected to support machine learning algorithms. The collection of tremendous data leads a huge problem: the disclosure of personal sensitive information. In real scenarios, not only the leakage of original data will disclose the information of individuals, when training machine learning models, model parameters may reveal sensitive information in an undirect way as well [37], [22].

To solve the problem of information leakage, differential privacy (DP) [17], [18] was proposed and has become a popular way to preserve privacy in machine learning. It preserves sensitive information by adding random noise, making an adversary can not infer any single data instance in the dataset by observing model parameters. Differential privacy has received a great deal of attentions and has been applied to regression [10], [38], [6], boosting [20], [50], PCA [12], [42], GAN [45], [47], transfer learning [31], graph algorithms [35], [39], [3], deep learning [36], [1] and other fields.

Empirical risk minimization (ERM), covering a wide variety of machine learning tasks, is also bothered by privacy problems. There is a long list of works on DP-ERM [43], [4], [11], [49], [29]. According to different ways of adding noise, three approaches were proposed to achieve differential privacy: output perturbation, objective perturbation and gradient perturbation, adding noise to the final model, the objective function and the gradient, respectively. However, the original data is not preserved by perturbation methods mentioned above. In real scenarios, before training, original data is sent to a 'data center', which is trusted in central models, shown in Figure 1 (a). When it comes to the situation that 'data center' is not trusted, local differential privacy (LDP) [5], [27] was proposed to provide plausible deniability, by randomizing the data before releasing it. As shown in Figure 1 (b), LDP focuses on the privacy of the communications between individuals and the 'server', rather than the final machine learning model [15], [40], [41], [16], [44]. However, the noise added for preserving privacy in LDP is always large, compromising predictive performance.

To alleviate the problems mentioned above, in this paper, we study the input perturbation method, achieving \((\epsilon,\delta)\)-differential privacy on the final model. The comparison between our method and previous perturbation methods is shown in Figure 1. It can be observed that our method focuses on the final model and preserves the original data to some extents. Even if the adversaries get the perturbed data in the 'data center', the leakage of sensitive information decreases a lot compared with traditional central models. Actually, adding noise to original data to preserve privacy is commonly used in the field of computer vision [26], [21], [30]. In this way, it is not easy to reconstruct the original data [2].

By adding noise to original data, protections are applied before 'data input', and our method is more reliable than traditional central models. Moreover, we observe that our input perturbation method also perturbs the gradient and the final model parameters, building a bridge between local and central differential privacy.
A More General Condition. Considering that most previous works assume the loss function is strongly convex, we generalize it to the condition that the loss function satisfies the Polyak-Lojasiewicz condition, which is more general than strong convexity.

The rest of the paper is organized as follows. In Section 2, we introduce some works related to our method. We introduce some basic definitions and formulations in Section 3. In Section 4, we propose our method: input perturbation in detail. In Section 5, we give the theoretical analysis of our method and extend it to a more general case. We present the experimental results in Section 6. Finally, we conclude the paper in Section 7.

2 RELATED WORK

In this section, we introduce some work on private ERM methods and list the comparison of their theoretical results.

The first work on DP-ERM was proposed in [11], in which two methods were proposed: output perturbation and objective perturbation. The probability density function of the noise $c(b) = \frac{1}{\alpha} e^{-y^2/b^2}$, where $\alpha$ is a normalizing constant, $y$ is a function of the privacy budget $\epsilon$ and $\|\cdot\|$ denotes $L_2$-norm. In this work, the derivative of the loss function $\nabla f(\cdot)$ was assumed $L$-Lipschitz. Based on these assumptions, it provided theoretical analysis on the noise bound and the excess empirical risk bound. The noise of the method proposed in [11] was improved by [29]. The improved noise is related to the upper bound of $\|\nabla f(\cdot)\|$, $\zeta$ (i.e. $\|\nabla f(\theta)\| \leq \zeta$ for all $\theta$).

Additionally, this work assumed the perturbed objective function is $\Delta$-strongly convex, and gives the excess empirical risk bound, which is related to the noise $b$ and the optimal model $\hat{\theta}$.

By gradient perturbation, [4] added noise to the gradient, guaranteeing differential privacy by assuming that the loss function $f(\cdot)$ is $G$-Lipschitz. Like in [11], [49] proposed an output perturbation method, achieving a better excess empirical bound. Advanced gradient descent method Prox-SVRG [46] was introduced in [43], and a new algorithm DP-SVRG was proposed. DP-SVRG achieved optimal or near optimal utility bounds with less gradient complexity. In this work, the noise bound was related to $m$, the sampling iterations in the algorithm DP-SVRG. Note that in DP-SVRG, better results are because of advanced gradient descent method, rather than advanced perturbation method.

However, all the methods proposed in previous work are based on output perturbation, objective perturbation or gradient perturbation. As a result, privacy preserving is after ‘data input’, which increases the risk of information leakage. Although LDP can solve the problem of ‘untrusted data center’, the theoretical results are much worse, which can be observed in Table 1.

Under these circumstances, input perturbation was proposed in [24], in which although noise is added to data, it achieves differential privacy by constructing a ‘perturbed objective function’. It guarantees $(O(\sqrt{n}),\delta)$-LDP and $(\epsilon,\delta)$-central DP. However, considering that $n$ is always large, the LDP is unsatisfactory. Moreover, its excess empirical risk bound is also much weaker than some central models because the noise added to the original data is large.

Considering the problems mentioned above, in this paper, we focus on input perturbation, adding noise to the original data and the sacrifice is acceptable.

Contributions of Our Method

A Bridge between Local and Central Differential Privacy. By observing a fact that the noise added to data causes perturbation on the gradient and finally the final model, we build a bridge between local and central differential privacy, guaranteeing $(\epsilon,\delta)$-differential privacy on the final model along with some kind of privacy on the original data simultaneously. When comparing with traditional central perturbation methods, in which the privacy of original data is ignored, we provide more privacy. Meanwhile, comparing with LDP, we make a balance on the performance and the privacy of individuals: adding less noise and keeping better performance. Additionally, the privacy on the final model remains.

Superior Theoretical and Experimental Results. Detailed theoretical analysis and experiments show that the performance of our method is similar to (or even better than) some of the best previous methods in central setting. Considering that our method preserves both the original data and the final model and other central methods ignore the security of original training data, the results are attractive. When it comes to LDP, although in our method, the privacy between individuals and the ‘data center’ is weaker, the performance of our method is much better, which is a trade-off and the sacrifice is acceptable.

Figure 1: Different perturbation methods.
Input Perturbation: A New Paradigm between Central and Local Differential Privacy

Table 1: Comparison between our method and other methods on noise bound and excess empirical risk bound.

| Method | \( \delta = 0 \) | Noise Type | Noise Bound | Excess Empirical Risk Bound |
|--------|-----------------|-------------|-------------|-----------------------------|
| [11]   | Output Perturbation | Yes | \( \epsilon(b) = \frac{1}{n} e^{-\gamma/\|b\|} \) | \( \gamma = O(n \lambda \epsilon) \) | \( O(\frac{\log(p/\delta)(L+\lambda)}{\epsilon \lambda n \epsilon^2}) \) |
| [11]   | Objective Perturbation | Yes | \( \epsilon(b) = \frac{1}{n} e^{-\gamma/\|b\|} \) | \( \gamma = O\left(\frac{e - \log(1 + \frac{2L}{n \epsilon^2} + \frac{L^2}{n^2 \lambda^2})}{\epsilon^2}\right) \) | \( O\left(\frac{\log(p/\delta)}{\lambda n \epsilon^2}\right) \) |
| [29]   | Objective Perturbation | No | Gaussian Noise | \( O\left(\frac{\log(n/\delta) \log(\delta/\epsilon)}{\epsilon^2}\right) \) | \( O(\sqrt{n} \epsilon^2 + \Delta \|\hat{\theta}\|^2) \) |
| [4]    | Gradient Perturbation | No | Gaussian Noise | \( O\left(G^2 T \log(2/\delta)\right) \) | \( O\left(G^2 p \log(n/\delta) \log(1/\delta)\right) \) |
| [49]   | Output Perturbation | No | Gaussian Noise | \( O\left(G^2 T \log(1/\delta)\right) \) | None³ |
| [43] (DP-SVRG) | Gradient Perturbation | No | Gaussian Noise | \( O\left(G^2 T \log(1/\delta)\right) \) | \( O\left(G^2 p \log(n/\delta) \log(1/\delta)\right) \) |
| [43] (traditional) | Gradient Perturbation | No | Gaussian Noise | \( O\left(G^2 T \log(1/\delta)\right) \) | \( O\left(G^2 p \log(n/\delta) \log(1/\delta)\right) \) |
| [15]   | LDP | Yes | Randomized response | \( O\left(G_{\rho} \epsilon \sqrt{n}\right) \) | None³ |
| [40]   | LDP | Yes | Laplace Noise² | \( O\left(p\epsilon n^2\right) \) | \( O\left(G^2 \log(16/\delta) + \epsilon\right) \) |
| [24]   | Input Perturbation | No | Gaussian Noise | \( O\left(G^2 T \log(1/\delta)\right) \) | \( O\left(pG^2 \log(16/\delta) + \epsilon\right) \) |
| Our Method | Input Perturbation | No | Gaussian Noise | \( O\left(G^2 T \log(1/\delta)\right) \) | \( O\left(n(2L + G) G^2 \epsilon \log(n) \log(1/\delta)\right) \) |

² The noise bound of the Gaussian and Laplace noise and are represented by the variance, whose means are 0.
³ The noise added by randomized response is complicated, details can be found in [15].
⁴ \( n \) is the size of training set, \( T \) is the number of total iterations, \( p \) is the number of model parameters, input \( x \) has \( d \)-dimensional feature.

Table 1: Comparison between our method and other methods on noise bound and excess empirical risk bound.

- **Input Perturbation:** A New Paradigm between Central and Local Differential Privacy

**3 PRELIMINARIES**

In this section, first, we introduce some basic definitions, including the comparison between central and local differential privacy. Then, we list traditional perturbation methods of central differentially private ERM in detail: output perturbation, objective perturbation and gradient perturbation.

### 3.1 Notations and Basic Definitions

Given a \( d \)-dimensional vector \( \mathbf{x} = [x_1, x_2, ..., x_d]^T \), denotes its \( \ell_2 \)-norm by \( \|\mathbf{x}\| = \left(\sum_{i=1}^{d} |x_i|^2\right)^{1/2} \). Two databases \( D, D' \in \mathbb{D}^n \) differing by one element are denoted by \( D \sim D' \), called adjacent databases.
Definition 1 (Central Differential Privacy [19]). A randomized function \( A \colon \mathcal{D}^n \rightarrow \mathbb{R}^p \) is \((\epsilon, \delta)\)-differential privacy if
\[
\mathbb{P}[A(D) \in S] \leq e^\epsilon \mathbb{P}[A(D') \in S] + \delta,
\]
where \( S \in \text{range}(A) \) and \( p \) is the number of parameters.

Definition 2 (Local Differential Privacy [41]). An algorithm \( Q \) is \((\epsilon, \delta)\)-local differential privacy if for all \( x, x' \in D \), and for all events \( E \) in the output space of \( Q \), we have:
\[
\mathbb{P}[Q(x) \in E] \leq e^\epsilon \mathbb{P}[Q(x') \in E] + \delta.
\]

According to the definitions of central and local differential privacy, in Definition 1, datasets \( D \) and \( D' \) are input to the randomized function \( A \), the privacy of the machine learning model is focused, guaranteeing information cannot be inferred by observing the machine learning model. In Definition 2, records \( x \) and \( x' \) are input to the algorithm \( Q \), data is paid more attention, guaranteeing information cannot be inferred by observing the noisy data. In the local model, ‘untrusted server’ is seen as the malicious adversary.

3.2 Traditional Perturbation Methods

Our method focuses more on the privacy of the machine learning model, similar to the central setting. So, in this part, we introduce three traditional central perturbation methods.

In general, the objective function of ERM without privacy preserving is defined as:
\[
L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, x_i, y_i),
\]
where \((x_i, y_i)\) denotes data instance, \(\ell(\cdot)\) is the loss function.

In the case of binary classification, the data space \( \mathcal{X} = \mathbb{R}^d \) and the label set \( \mathcal{Y} = \{-1, +1\} \), and we assume throughout that \( \mathcal{X} \) is the unit ball so that \( ||x_i|| \leq 1 \).

Output Perturbation. In output perturbation, noise is directly added to the model (in the paper, we denote model by parameters):
\[
\theta_{pri} = \arg\min \{L(\theta)\} + z,
\]
where \( z \) is the noise guaranteeing differential privacy.

Objective Perturbation. In the method of objective perturbation, noise is added to the objective function:
\[
L_{pri}(\theta) = L(\theta) + \frac{1}{n} z^T \theta.
\]

The perturbed objective function \( L_{pri}(\theta) \) is directly optimized:
\[
\theta_{pri} = \arg\min \{L_{pri}(\theta)\}.
\]

Note that in (5), there may be some other terms on the right side of the equality, for example \( \frac{1}{n} \|\theta\|^2 \) in [29]. We only list the most important term \( \frac{1}{n} z^T \theta \) to guarantee differential privacy here.

This method is rarely used in recent years because it is always a trouble to optimize the perturbed objective function and the performance is unsatisfactory.

Gradient Perturbation. In the gradient perturbation method, noise is added to the gradient when training, which leads the gradient descent process at round \( t \) to:
\[
\theta_{t+1} = \theta_t - \alpha (\nabla L(\theta_t) + z),
\]
where \( \alpha \) is the learning rate.

After \( T \) iterations in total, the final model \( \theta_{pri} = \theta_T \).

Because most machine learning algorithms are based on gradient descent method, gradient perturbation is feasible and popular.

4 DIFFERENTIALLY PRIVATE ERM WITH INPUT PERTURBATION

In this section, first, we analyze the weaknesses of traditional central perturbation methods and local models introduced in Section 3, then we propose our method input perturbation in detail.

When training models, original data is always sent to the ‘data center’ in advance, which is shown in Figure 1. By observing three traditional perturbation methods of central DP-ERM, original data is not protected, which means the ‘data center’ is assumed trusted.

However, ‘data center’ is not easy to ‘trust’ because the adversaries always desire to ‘take away’ the original data and the ‘data center’ may be monitored with high probability. As a result, the security of original data instances is of the same importance as (or even more important than) the model parameters. LDP is a superior way to solve the problem of ‘untrusted data center’, guaranteeing differential privacy over the communications (data exchanging) between individuals and the ‘data center’. However, as shown in Table 1, the noise added to data is large, and it is inevitable that the performance is worse than central models.

To solve the problems mentioned above, we propose a new input perturbation method, adding noise to data instances and training the machine learning model by the perturbed data instances, which leads the objective function to:
\[
\tilde{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, x_i + z, y_i).
\]

In order to distinguish with the objective function without privacy consideration \( L(\theta) \) in (3), we denote the objective function of input perturbation by \( \tilde{L}(\theta) \). In (8), ‘noise adding’ has been done in advance and the formulation \( x_i + z \) is for distinguishing the perturbed data and original data.

Our method focuses on achieving \((\epsilon, \delta)\)-differential privacy on the machine learning model with some kind of privacy on original data. As a result, even if the ‘data center’ is not trusted or monitored, the data ‘taken away’ by malicious adversaries is with random noise, which preserves the ‘true original data’ of individuals from some kinds of attacks.

Although in our method, noise is added to the original data, we focus more on the \((\epsilon, \delta)\)-differential privacy of the final model, which is different from the local model: protections between individuals and the ‘server’ are paid more attentions, and the privacy of model parameters is not discussed. Comparing with LDP and input perturbation method in [24], based on the aim to guarantee the quality of the machine learning model, we sacrifice some of the privacy on individuals for the performance. In fact, the sacrifice compared with [24] is not much. In other words, focusing on keeping good
performance, we attempt to preserve the privacy on original data as much as possible. It can be observed that in LDP and previous input perturbation method, the noise added to data is much more than ours. As a result, the privacy preserving on individuals of our method is weaker than in LDP and previous input perturbation method, but still stronger than central methods.

Our method is detailed in Algorithm 1.

Algorithm 1 Differentially Private ERM with Input Perturbation Method

Require: Dataset $D$, iteration rounds $T$, learning rate $\alpha$

1: function INPUTPERTURBATION($D$, $T$, $\alpha$)
2: For all data instances $(x_i, y_i)$ in $D$, add noise $z$ to it:
3: $(x_i, y_i) \leftarrow (x_i + z, y_i)$.
4: New data $(x_i + z, y_i)$ is denoted as ‘perturbed data’.
5: Train model by perturbed data, the objective function is the same as (8), which leads the following process.
6: for $t = 0$ to $T - 1$ do
7: $\theta_{t+1} \leftarrow \theta_t - \alpha \frac{1}{n} \sum_{i=1}^{n} \nabla L(\theta_t)$.
8: end for
9: return $\theta_T$.
10: end function

In Algorithm 1, the random noise $z \sim \mathbb{R}^d$ and each element $z_i \sim N(0, \sigma^2)$, sampled independently. By line 7 in Algorithm 1, it can be seen that the noise added to the original data affects the gradient. The theoretical analysis of our method in Section 5 is based on this observation.

Besides, by observing that our method adds noise to original data instances, leading perturbation on the gradient and eventually causing perturbation on the model parameters, a bridge is built between local and central differential privacy: input perturbation ERM protects the original data, the gradient and the final model parameters simultaneously, providing a higher level protection on privacy in a more reliable way in the field of central DP-ERM.

5 THEORETICAL ANALYSIS OF INPUT PERTURBATION ERM

In this section, first, we give privacy guarantees of our proposed method: input perturbation ERM. Then, we analyze the excess empirical risk bound of our method. Finally, we extend our method to a more general case, in which the loss function is not restricted strongly convex but satisfies the Polyak-Lojasiewicz condition, which is more general than the property ‘strongly convex’.

5.1 Differential Privacy

In this part, we analyze the $(\epsilon, \delta)$-differential privacy of our proposed method: input perturbation in Algorithm 1.

In this paper, we analyze $(\epsilon, \delta)$-differential privacy by Gaussian mechanism proposed in [18] and moments accountant proposed in [1]. Moreover, we assume $\ell(\theta, x, y)$ is $\ell(y\theta^T x)$ like in [11].

Theorem 1. In Algorithm 1, for $\epsilon, \delta > 0$, if $\ell(\theta, x, y)$ is $G$-Lipschitz and $\Delta$-strongly convex over $\theta$ and

$$
\sigma^2 = c \frac{G^2 T \log(1/\delta)}{n(n-1)\sqrt{n}\Delta^2},
$$

it is $(\epsilon, \delta)$-differential privacy for some constant $c$.

The proof is detailed in the Appendix.

5.2 Excess Empirical Risk Bound

In this part, we analyze the utility of our proposed method and give the excess empirical risk bound, denoted by the expectation of $\tilde{L}(\theta_T) - L^*$, where $L^*$ is the value of the objective function over the optimal model without privacy consideration. Formally, $L^* = \min_{\theta \in \mathbb{R}^d} L(\theta)$, where $L(\theta)$ is the same as in (3).

Theorem 2. Suppose that $\ell(\theta, x, y)$ is $G$-Lipschitz, $\nabla \ell(\theta, x, y)$ is $L$-Lipschitz\footnote{L-Lipschitz on $\nabla \ell(\cdot)$ means $L$-smooth on $\ell(\cdot)$.} and the $L_2$-norm of the model parameter has an upper bound $D$ (i.e. $\|\theta\| \leq D$ for all $\theta$), with $\sigma$ is the same as in (9), we have:

$$
\mathbb{E} \left[ \tilde{L}(\theta_T) - L^* \right] \leq O \left( \frac{\alpha(2LD + G)^2 d \log^2(n) \log(1/\delta)}{n(n-1)\sqrt{n}\Delta^2} \right),
$$

where $T = \tilde{O} \left( \log(\frac{m(n-1)\sqrt{n}\Delta^2}{\alpha(2LD + G)d \log(1/\delta)}) \right)$, $\alpha$ represents the learning rate and each data instance $x_i \in X \in \mathbb{R}^d$ has $d$-dimensional features.

The proof is shown in the Appendix.

Remark 1. Considering that the smoothness of the objective function after input perturbation $\tilde{L}(\theta)$ is not easy to achieve because of the existence of the random variable $z$, we assume $L(\theta)$ (without random variables) is $L$-smooth, which is easier to hold, making the utility and the excess empirical risk bound of our method feasible.

It can be observed that the excess empirical risk bound of our method is better than the traditional gradient perturbation method proposed in [4] by a factor of $\frac{\alpha(2LD + G)d \Delta \sqrt{n}}{np}$. Considering that the
variables $L, D, G, \alpha, \Delta$ can be seemed as constants, our method in much better than which proposed in [4] by a factor of $\frac{d \log n}{p}$. When comparing with gradient perturbation methods proposed in [43], the gap on empirical risk bound is by a factor of $\frac{d \log n}{p}$. In some cases that $p \gg d$, which is common in the field such as deep learning, the gap between our method and the gradient perturbation methods proposed in [43] is relatively small and can be ignored. When it comes to the comparison between our method and LDP methods, the excess empirical risk bound of our method is much better, with weaker privacy on individuals.

5.3 More general condition

In this part, we extend our method to a more general condition that the loss function $\ell(\theta, x, y)$ is not restricted $\Delta$-strongly convex, but satisfies the Polyak-Lojasiewicz condition.

**Definition 3.** Given a function $\ell(\cdot)$, if there exists $\mu > 0$ and for all $\theta$, we have:

$$\|\nabla \ell(\theta)\|^2 \geq 2\mu(\ell(\theta) - \ell^*),$$

then $\ell(\cdot)$ satisfies the Polyak-Lojasiewicz condition.

The Polyak-Lojasiewicz condition is much more general than strongly convex. It was shown in [28] that when function $\ell$ is differential and $L$-smooth under $\ell_2$-norm, we have:

Strong Convex $\Rightarrow$ Essential Strong Convexity $\Rightarrow$ Weakly Strongly Convexity $\Rightarrow$ Restricted Secant Inequality $\Rightarrow$ Polyak-Lojasiewicz Inequality $\Leftrightarrow$ Error Bound

**Theorem 3.** In Algorithm 1, for $\epsilon, \delta > 0$, if the loss function $\ell(\theta, x, y)$ is $G$-Lipschitz and satisfies Polyak-Lojasiewicz condition over $\theta$ and

$$\sigma^2 = \frac{G^2 T \log(1/\delta)}{n(n-1)\epsilon^2},$$

it is $(\epsilon, \delta)$-differential privacy for some constant $c$.

Detailed proof is shown in the Appendix.

**Theorem 4.** Suppose that $\ell(\theta, x, y)$ is $G$-Lipschitz, $\nabla \ell(\theta, x, y)$ is $L$-Lipschitz, $L(\theta)$ is $L$-smooth over $\theta$ and the $\ell_2$-norm of the model parameter has an upper bound $D$ (i.e. $\|\theta\| \leq D$), with $\sigma$ is the same as in (52), we have:

$$\mathbb{E}\left[\hat{L}(\theta_T) - L^*\right] \leq O\left(\frac{\alpha(2LD + G)G^3d \log^2(n) \log(1/\delta)}{n(n-1)\epsilon^2}\right),$$

where $T = O\left(\log\left(\frac{n(1-\epsilon)^2}{\alpha(2LD + G)G^3d \log(1/\delta)}\right)\right)$, $\alpha$ is the learning rate and each data instance $x_i$ has $d$-dimensional features.

The proof of Theorem 4 is almost the same as Theorem 2, with replacement of $\sigma$.

By Theorem 3 and Theorem 4, it can be observed that in a more general case: the loss function is not restricted strongly convex but satisfies the Polyak-Lojasiewicz condition, our noise bound and the excess empirical risk bound are almost the same as previous work on central models.
6 EXPERIMENTS

The experiments are performed on the classification task. Considering that our method focuses on the privacy of the final model, the experiments are applied on central methods: the objective perturbation method proposed in [29], the output perturbation method proposed in [49] and the gradient perturbation methods proposed in [4] and [43] (without DP-SVRG). The performance is represented by accuracy and the optimality gap, the latter is defined as $L(\theta_{pri}) - L^*$. Accuracy represents the performance on test data and optimal gap denotes excess empirical risk on training data.

According to the sizes of datasets, we use logistic regression model (LR) and deep learning model on the datasets KDDCup99 [25], Adult [14], Bank [33], where the total number of data instances are 70000, 45222 and 41188, the sizes are larger than 10000. On datasets Breast Cancer [32], Credit Card Fraud [7], Iris [14], only logistic regression model is applied because the sizes are less than 1000, where the total number of data instances are 699, 984 and 150, respectively. In the experiments, deep learning model is denoted by Multi-layer Perceptron (MLP) with one hidden layer whose size is the same as the input layer. The training set and the testing set are chosen randomly.

In all experiments, $T$ and $\alpha$ are chosen by cross-validation. We evaluate the influence over differential privacy budget $\epsilon$, which is set from 0.01 to 0.25. Meanwhile, $\delta$ is set according to the size of datasets and can be seemed as a constant. Note that in logistic regression model, $d = p$ and in deep learning model, $d < p$.

Figure 2 shows that the accuracy of our proposed method is better than the gradient perturbation method proposed in [4] and the objective perturbation method proposed in [29]. And our method is almost the same as the gradient perturbation method proposed in [43] and the output perturbation method proposed in [49] on accuracy, no matter on the LR model or on the MLP model. However, because the variance of the Gaussian noise added to the gradient in the method [4] is large: $O \left( \frac{G^n \log(n/\delta) \log(1/\delta)}{\epsilon^2} \right)$, the accuracy of this method over $\epsilon$ fluctuates sharply in Figure 2.

It can be observed that in Figure 3, the optimality gap of our method is almost the same as the output perturbation method proposed in [43] and is better than other methods mentioned above over most datasets, which is similar to the theoretical analysis. Moreover, it can be observed that the optimality gap of our method on some datasets are close to 0, which means that our method achieves almost the same performance as the ERM model without privacy consideration in some scenarios, on both LR model and MLP model. In addition, like the accuracy in Figure 2, the optimality gap of the gradient perturbation method proposed in [4] fluctuates sharply because of its noise bound.

Figure 4 shows accuracy and optimality gap on small datasets (the sizes are less than 1000), in which only logistic regression model is applied. The results are similar to which in Figure 2 and Figure 3, which means that our method is effective in most cases.
By observing the experimental results, we find that although there are slight differences in experimental results on different datasets, the performance of the gradient perturbation method proposed in [4] and the objective perturbation method proposed in [29] is much weaker than our method, the former is because of its loose noise bound and the latter is because of the perturbation method itself. Our proposed method: input perturbation, is almost the same as (on some datasets, even better than) the output perturbation method in [43] and the traditional gradient perturbation method without DP-SVRG in [49] on both accuracy and optimality gap, which is similar to our theoretical analysis in Section 4. The experimental results on the deep learning model (MLP), are similar to the traditional machine learning (logistic regression) model. Considering that our method preserves the privacy of the original data, the gradient and the final model simultaneously, providing more privacy without decreases on the performance compared with previous central methods, it is an attractive result.

7 CONCLUSIONS
In this paper, we study the input perturbation method in DP-ERM, adding Gaussian noise to original data instances and training the machine learning model by the ‘perturbed data’. By observing that input perturbation leads the perturbation on the gradient and finally the perturbation on the final model, we build a bridge between local and central differential privacy, achieving $(\epsilon, \delta)$-differential privacy on the final machine learning model, along with some kind of privacy on individuals. Through the ‘bridge’, we preserve the original data, the gradient and the final machine learning model simultaneously. Meanwhile, we extend our method to a more general condition, in which the loss function is not considered strongly convex but satisfies the Polyak-Lojasiewicz condition. Theoretical analysis and experiments (applied on both traditional machine learning model: logistic regression, and deep learning model: MLP) on real datasets show that our method achieves almost the same (or even better) performance compared with some of the best previous methods. Additionally, higher level of privacy is achieved, comparing with previous central methods. It is worth emphasizing that our method adds noise to original data, independent of specific optimization methods, which means that our proposed method is a general paradigm. Moreover, detailed analysis of the privacy preserved on individuals of our method and how to improve the privacy of individuals will also be paid attentions in future work.

REFERENCES
[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 308–318.
[2] Rakesh Agrawal and Ramakrishnan Srikant. 2000. Privacy-preserving data mining. In ACM Sigmod Record, Vol. 29. 439–450.
[3] Raman Arora and Jalaj Upadhyay. 2019. On Differentially Private Graph Sparsification and Applications. In Advances in Neural Information Processing Systems 32. H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 13378–13389. http://papers.nips.cc/paper/9494-on-differentially-private-graph-sparsification-and-applications.pdf
A DETAILS OF PROOF

A.1 Theorem 1

Proof. By observing that the noise added to data causes the perturbation on the gradient, we pay our attentions on the gradient descent process:

\[
\theta_{t+1} = \theta_t - \alpha \nabla \tilde{L}(\theta_t) = \theta_t - \alpha \left( \frac{1}{n} \sum_{i=1}^{n} y_i \nabla f(y_i \theta_t^T(x_i + z))(x_i + z) \right).
\]  
(14)

where \( z \sim N(0, \sigma^2) \) and \( \alpha \) denotes the learning rate.

Then, considering about the \( t^{th} \) query which may disclose privacy, the randomized mechanism \( M_t \) is:

\[
M_t = \frac{1}{n} \sum_{i=1}^{n} y_i \nabla f(y_i \theta_t^T(x_i + z))(x_i + z).
\]  
(15)

Denote probability distributions on adjacent databases \( D \) and \( D' \) over mechanism \( M_t \) as \( P \) and \( Q \):

\[
P = \frac{1}{n} \sum_{i=1}^{n-1} y_i \nabla f(y_i \theta_t^T(x_i + z))(x_i + z) + \frac{1}{n} y_n \nabla f(y_n \theta_t^T(x_n + z))(x_n + z),
\]
\[
Q = \frac{1}{n} \sum_{i=1}^{n-1} y_i \nabla f(y_i \theta_t^T(x_i + z))(x_i + z) + \frac{1}{n} y'_n \nabla f(y'_n \theta_t^T(x'_n + z))(x'_n + z),
\]  
(16)

where we suppose that the single different data instance between \( D \) and \( D' \) is the \( n^{th} \) one, denoted as \( (x_n, y_n) \) and \( (x'_n, y'_n) \), respectively.

For simplicity on expression, we set:

\[
A = \frac{1}{n} \sum_{i=1}^{n-1} y_i \nabla f(y_i \theta_t^T(x_i + z))(x_i + z), \quad B = \frac{1}{n} y_n \nabla f(y_n \theta_t^T(x_n + z))(x_n + z),
\]
\[
B' = \frac{1}{n} y'_n \nabla f(y'_n \theta_t^T(x'_n + z))(x'_n + z), \quad C = \frac{1}{n} \sum_{i=1}^{n-1} y_i \nabla f(y_i \theta_t^T(x_i + z)).
\]  
(17)

Then, by (16), (17) and note that \( z \sim N(0, \sigma^2) \), we have:

\[
P = N(A + B, C^2), \quad Q = N(A + B', C^2).
\]  
(18)

In moments accountant method proposed in [1], the \( \lambda^{th} \) moment \( \sigma_M(\lambda; D, D') \) on mechanism \( M \) is defined as:

\[
\sigma_M(\lambda; D, D') = \log \mathbb{E}_{\sigma \sim M(D)} \left[ \exp(\lambda c(\sigma; M, D, D')) \right],
\]  
(19)

where \( c(\sigma; M, D, D') \) is privacy loss at the output \( o \), defined as:

\[
c(\sigma; M, D, D') = \log \frac{P[M(D) = o]}{P[M(D') = o]}.
\]  
(20)

When it comes to privacy preserving, it is necessary to bound all possible \( \sigma_M(\lambda; D, D') \), denoted as \( \sigma_M(\lambda) \), which is defined as:

\[
\sigma_M(\lambda) = \max_{D, D'} \sigma_M(\lambda; D, D').
\]  
(21)

By Definition 2 in [9], \( D_\alpha \) is defined as:

\[
D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \left( \mathbb{E}_{x \sim P} \left[ \left( \frac{P(x)}{Q(x)} \right)^{\alpha - 1} \right] \right).
\]  
(22)

By (19), (20), (21), (22) and \( P, Q \) in (18), we have:

\[
\sigma_M(\lambda) = \log \mathbb{E}_{o \sim P} \left[ \exp \left( \lambda \log \frac{P}{Q} \right) \right] = \log \mathbb{E}_{o \sim P} \left[ \left( \frac{P}{Q} \right)^{\lambda} \right] = \lambda D_{\lambda+1}(P\|Q).
\]  
(23)

By (23) and Lemma 2.5 in [9], we have:

\[
\sigma_M(\lambda) = \lambda D_{\lambda+1}(P\|Q) = \frac{\lambda(\lambda + 1)||A + B - (A + B')||^2}{2\sigma^2}.
\]  
(24)

By definitions of \( B \) and \( B' \) in (17) and note that \( \ell(\theta, x, y) \) is G-Lipschitz \((G, \alpha)\), we have:

\[
\|B - B'\| = \frac{1}{n} \nabla \ell(\theta_t, x_n + z, y_n) - \frac{1}{n} \nabla \ell(\theta_t, x'_n + z, y'_n) \leq \frac{2G}{n}.
\]  
(25)

By [13], if function \( \ell(\theta, x, y) \) is \( \Delta \)-strongly convex \((\Delta)\), we have:

\[
\|\nabla \ell(\theta, x, y)\|^2 \geq 2\Delta (\ell(\theta, x, y) - \ell^*).
\]  
(26)

Combining (26) and the definition of \( C \) in (17), we have:

\[
C = \frac{1}{n} \sum_{i=1}^{n-1} y_i \nabla f(y_i \theta_t^T(x_i + z))(x_i + z) + \frac{1}{n} y_n \nabla f(y_n \theta_t^T(x_n + z))(x_n + z).
\]  
(27)

In general, with the increasing of training iteration, loss of the model decreases. i.e. \( \ell(\theta_t) \leq \ell(\theta_2) \) if \( t_1 \geq t_2 \). So, we have:

\[
C \geq \frac{n}{n - 1} \sqrt{2\Delta (\ell(\theta_T) - \ell^*)}.
\]  
(28)

Considering that \( \ell(\theta_T) - \ell^* \) can be seemed as a constant, by (25) and (28), for some constant \( c_1, c_2 \) can be transferred to:

\[
\sigma_M(\lambda) \leq c_1 \frac{\lambda(\lambda + 1)G^2}{\sqrt{\Delta} \sigma^2(n - 1)}.
\]  
(29)

By Theorem 2.1 in [1], we have:

\[
\sigma_M(\lambda) \leq \sum_{t=1}^{T} \sigma_M(\lambda).
\]  
(30)

By summing over \( T \) iterations on (29), for some constant \( c_2 \):

\[
\sigma_M(\lambda) = \sum_{t=1}^{T} \sigma_M(\lambda) \leq c_1 \frac{\lambda(\lambda + 1)GT \sqrt{\Delta} \sigma^2(n - 1)}{\sqrt{\Delta} \sigma^2(n - 1)} \leq c_2 \frac{\lambda^2G^2}{\sigma^2(n - 1)\sqrt{\Delta}}.
\]  
(31)

Taking \( \sigma^2 = c_2 \frac{G^2T \log(1/\delta)}{n(n - 1)\sqrt{\Delta} \lambda^2} \), for some constant \( c \), we can guarantee:

\[
\sigma_M(\lambda) \leq \frac{c_2 \lambda^2G^2}{\sigma^2(n - 1)\sqrt{\Delta}} \leq \frac{\lambda \varepsilon}{2},
\]  
(32)

and as a result, we have:

\[
\delta \leq \exp(\frac{-\lambda \varepsilon}{2}),
\]  
(33)

leading \((\epsilon, \delta)\)-differential privacy according to Theorem 2.2 in [1].

\[\Box\]

A.2 Theorem 2

Proof. First, considering \( \mathbb{E} \left[ \hat{L}(\theta_{t+1}) - \hat{L}(\theta_t) \right] \) at round \( t \):

\[
\mathbb{E} \left[ \hat{L}(\theta_{t+1}) - \hat{L}(\theta_t) \right] = \mathbb{E}_z \left[ \frac{1}{n} \sum_{i=1}^{n} \ell((y_i \theta_{t+1}^T(x_i + z)) - \ell((y_i \theta_t^T(x_i + z))) \right].
\]  
(34)

Note that \( \ell(\cdot) \) is G-Lipschitz \((G, \alpha)\), then for all \( x, y \):

\[
\ell(x) - \ell(y) \leq G |x - y|.
\]  
(35)

Kang et al.
By the combination of (34) and (35), without loss of generality:
\[
\mathbb{E} \left[ \hat{L}(\theta_{t+1}) - \hat{L}(\theta_t) \right] \leq \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_x \left[ G |y_i| \theta_T^T (x_i + z) - y_i \theta_T^T (x_i + z) \right] 
\]
\[
\leq aG \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_x \left[ y_i (\nabla \hat{L}(\theta_t) x_i + y_i (\nabla \hat{L}(\theta_t) z) \right]. 
\]

Using the definition of \( \hat{L}(\theta) \) in (8), we have:
\[
\nabla \hat{L}(\theta_t) = \frac{1}{n} \sum_{i=1}^{n} y_i \nabla \ell(y_i \theta_T^T (x_i + z))(x_i + z).
\]

Note that the loss function \( \nabla \ell(\cdot) \) is \( L \)-Lipschitz, then we have:
\[
\nabla \ell(y_i \theta_T^T (x_i + z)) - \nabla \ell(y_i \theta_T^T x_i) \geq -L |y_i \theta_T^T z|.
\]

Then, by (37) and (38), we have:
\[
\nabla \hat{L}(\theta_t) \geq \frac{1}{\Delta} \sum_{i=1}^{n} y_i \nabla (\ell(y_i \theta_T^T x_i) - L |y_i \theta_T^T z|)(x_i + z).
\]

Note that \( y_i \in [-1, 1] \) and \( ||x_i|| \leq 1 \), (39) can be transferred to:
\[
\nabla \hat{L}(\theta_t) \geq \frac{1}{\Delta} \sum_{i=1}^{n} y_i \nabla \ell(y_i \theta_T^T x_i) z.
\]

By combining (36) and (40), we have:
\[
\mathbb{E} \left[ \hat{L}(\theta_{t+1}) - \hat{L}(\theta_t) \right] 
\leq aG \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_x \left[ -\nabla L(\theta_t) + L || \theta_t || || z ||^2 \right] 
\leq aG \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_x \left[ L || \theta_t || || z ||^2 - \frac{1}{\Delta} \sum_{i=1}^{n} y_i \nabla \ell(y_i \theta_T^T x_i) z \right].
\]

Then, by combining (44) and (47), we have:
\[
\mathbb{E} \left[ \hat{L}(\theta_T) - L^* \right] 
\leq \frac{aG}{n} \left[ 2L \| \theta_T \| + \frac{aG}{2} T \right] \leq \frac{aG}{n} \left[ 2L \| \theta_T \| + \frac{aG}{2} T \right].
\]

For random variable \( X \), we have:
\[
\mathbb{E}(X^2) = \mathbb{E}^2(X) + \sigma^2(X),
\]
where \( \sigma(X) \) denotes the variance of \( X \).

By (42) and note that the random variable \( z \sim (0, \sigma^2) \), (41) can be transferred to:
\[
\mathbb{E} \left[ \hat{L}(\theta_{t+1}) - \hat{L}(\theta_t) \right] 
\leq -aG \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(\theta_t) + aG(2L || \theta_t || + G) \sigma^2.
\]

By summing (43) over \( T \) iterations and note that \( || \theta || \leq D \):
\[
\mathbb{E} \left[ \hat{L}(\theta_T) - \hat{L}(\theta_0) \right] \leq G^2 aT + aG(2LD + G) \sigma^2 T.
\]

Then, considering the gap between \( \hat{L}(\theta_0) \) and \( L^* \):
\[
\mathbb{E} \left[ \hat{L}(\theta_0) - L^* \right] \leq \frac{1}{n} \sum_{i=1}^{n} \ell(y_i \theta_0^T (x_i + z)) - L^*
\]
\[
\leq \mathbb{E} \left[ G |y_i| \theta_0^T z + (L \theta_0 - L^*) \right]
\]
\[
\leq G \| \theta_0 \| \mathbb{E} \left[ || z ||^2 \right] + (L \theta_0 - L^*)
\]
\[
= L \theta_0 - L^*.
\]

If \( \ell(\cdot) \) is \( L \)-smooth, we have:
\[
L \theta_0 - L^* \leq \mathbb{E} \left[ \nabla \ell(\theta_0^*) \right] \frac{L}{2} + \frac{L}{2} \| \theta_0 - \theta^* \|^2.
\]

where \( \theta^* \) denotes the optimal model and \( \nabla \ell(\theta^*) = 0 \). Then, by (49) and (46), the inequality holds:
\[
\mathbb{E} \left[ \hat{L}(\theta_0) - L^* \right] \leq \frac{L}{2} \| \theta_0 - \theta^* \|^2.
\]

Then, by combination of (44) and (47), we have:
\[
\mathbb{E} \left[ \hat{L}(\theta_T) - L^* \right] \leq O \left[ \frac{(2LD + G) \sigma^2 \log(n \log(1/\delta))}{n(n - 1) \Delta^2} \right].
\]

where \( \Delta \) is similar to \( O(\cdot) \), but hiding factors polynomial in \( \log n \) and \( \log(1/\delta) \).

\[\Box\]

A.3 Theorem 3

Proof. Taking \( M, P, Q, A, B, B', C \) the same as in A.1. Note that the loss function \( \ell(\theta, x, y) \) satisfies the Polyak-Lojasiewicz condition (PL), we have:
\[
C \geq \frac{n - 1}{n} \sqrt{2} \mu(\ell(\theta) - \ell^*).
\]

As a result, in the moments accountant method:
\[
\alpha_{M^2}(\lambda) \leq \frac{2L(\lambda + 1) \sigma^2}{\sqrt{2} \mu(\ell(\theta_T) - \ell^*) m(n - 1) \sigma^2}.
\]

The factor \( \sqrt{2} \mu(\ell(\theta_T) - \ell^*) \) can be seemed as a constant, then:
\[
\alpha_{M^2}(\lambda) \leq c_3 \frac{2L(\lambda + 1) \sigma^2}{m(n - 1) \sigma^2},
\]
for some constant \( c_3 \).

By summing T iterations, for some constant \( c_2 \), we have:
\[
\alpha_M(\lambda) \leq c_2 \frac{\lambda^2 \sigma^2 T}{\sigma^2 n(n - 1)}.
\]

Taking \( \sigma \) the same as in (9), it can be guaranteed that:
\[
\alpha_M(\lambda) \leq \frac{\lambda^2}{2},
\]

and as a result:
\[
\delta \leq \exp(-\frac{\lambda^2}{2}),
\]n

for some constant \( c_1 \), which means \( (\epsilon, \delta) \)-differential privacy due to Theorem 2.2 in [1].

\[\Box\]