Efficient Privacy-Preserving Stochastic Nonconvex Optimization

Lingxiao Wang[1] **Bargav Jayaraman**[2] **David Evans**[2] **Quanquan Gu**[3]

[1]Toyota Technological Institute at Chicago
[2]Department of Computer Science, University of Virginia
[3]Department of Computer Science, University of California, Los Angeles

Abstract

While many solutions for privacy-preserving convex empirical risk minimization (ERM) have been developed, privacy-preserving nonconvex ERM remains a challenge. We study nonconvex ERM, which takes the form of minimizing a finite-sum of nonconvex loss functions over a training set. We propose a new differentially private stochastic gradient descent algorithm for nonconvex ERM that achieves strong privacy guarantees efficiently, and provide a tight analysis of its privacy and utility guarantees, as well as its gradient complexity. Our algorithm reduces gradient complexity while matching the best-known utility guarantee. Our experiments on benchmark nonconvex ERM problems demonstrate superior performance in terms of both training cost and utility gains compared with previous differentially private methods using the same privacy budgets.

1 INTRODUCTION

For many important domains such as health care and medical research, the datasets used to train machine learning models contain sensitive personal information. There is a risk that models trained on this data can reveal private information about individual records in that training data [Fredrikson et al., 2014, Shokri et al., 2017, Carlini et al., 2019]. This motivates the research on privacy-preserving machine learning, much of which has focused on achieving *differential privacy* [Dwork et al., 2006], a rigorous definition of privacy that provides statistical data privacy for individual records. In the past decade, many differentially private machine learning algorithms for solving the empirical risk minimization (ERM) problem have been proposed (e.g., [Chaudhuri et al., 2011, Kifer et al., 2012, Bassily et al., 2014, Zhang et al., 2017, Wang et al., 2017, Jayaraman et al., 2018, Wang and Gu, 2019, 2020]). Almost all of these are for ERM with convex loss functions, but many important machine learning approaches, including deep learning, are formulated as ERM problems with nonconvex loss functions. Furthermore, these learning problems often involve large training sets, necessitating the use of stochastic optimization algorithms such as stochastic gradient descent (SGD).

Several recent studies have advanced the application of differential privacy in deep learning [Abadi et al., 2016, Papernot et al., 2016, McMahan et al., 2018, Bu et al., 2019]. While these studies prove differential privacy is satisfied, they evaluate utility experimentally. Only a few differentially private algorithms for solving nonconvex optimization problems have proven utility bounds [Zhang et al., 2017, Wang et al., 2017]. For example, Wang et al. [2017] proposed a differentially private gradient descent (DP-GD) algorithm with both privacy and utility guarantees. However, each iteration of DP-GD requires computing the full gradient, which makes it too expensive for use on large training sets. Zhang et al. [2017] proposed a random round private stochastic gradient descent (RRPSGD) that can achieve the same privacy guarantee as DP-GD with reduced runtime complexity but with slightly worse utility bounds. In this paper, we propose a differentially private Stochastic Recursive Momentum (DP-SRM) algorithm for nonconvex ERM. At the core of our algorithm is the stochastic recursive momentum technique [Cutkosky and Orabona, 2019] that can consistently reduce the accumulated variance of the gradient estimator. Our approach is more scalable than stochastic variance-reduced algorithms [Johnson and Zhang, 2013, Reddi et al., 2016a, Allen-Zhu and Hazan, 2016, Lei et al., 2017, Nguyen et al., 2017, Fang et al., 2018, Zhou et al., 2018] since it eliminates the periodical computation of the checkpoint gradient which usually requires a giant batch size. A recent work [Arora et al., 2022] developed a differentially private variant of the stochastic variance-reduced algorithm [Wang et al., 2019c] called Private SpiderBoost. While Private SpiderBoost can achieve the same utility guarantee as our proposed DP-SRM algorithm, Private Spider-

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Boost requires periodic full gradient computation, making it less scalable and results in worse gradient complexity.

Contributions. The main contributions of our paper are summarized as follows:

- We develop a new differentially private stochastic optimization algorithm for nonconvex ERM and provide a sharp analysis of the privacy guarantee using Rényi Differential Privacy (RDP) [Mironov, 2017].
- Our algorithm improves the previous best-known utility guarantee for nonconvex optimization with lower computational complexity. The utility guarantee of our algorithm is $O\big((d\log(1/\delta))^{1/3}/(n\epsilon)^{2/3}\big)$[1], which is better than the previous best-known results of $O\big((d\log(1/\delta))^{1/4}/(n\epsilon)^{1/2}\big)$ established in Wang et al. [2017]. The gradient complexity (i.e., the number of stochastic gradients calculated in total) of our algorithm is $O\big((n\epsilon)^2/(d\log(1/\delta))\big)$, which outperforms the best previous results [Zhang et al., 2017, Wang et al., 2017] when the problem dimension d is large (see Table 1 for more details).
- We evaluate our proposed methods on two nonconvex ERM techniques: nonconvex logistic regression and convolutional neural networks. We report on experiments on several benchmark datasets (Section 7), finding that our method not only produces models that are the closest to the non-private models in terms of model accuracy but also reduces the computational cost.

Notation. We use curly symbol such as $\mathcal{B}$ to denote the index set. For a set $\mathcal{B}$, we use $|\mathcal{B}|$ to denote its cardinality. For a finite sum function $F = \sum_{i=1}^n f_i/n$, we denote $F_{\mathcal{B}}$ by $\sum_{i\in\mathcal{B}} f_i/|\mathcal{B}|$. For a d-dimensional vector $\mathbf{x} \in \mathbb{R}^d$, we use $\|\mathbf{x}\|_2$ to denote its ℓ_2-norm. Given two sequences $\{a_n\}$ and $\{b_n\}$, if there exists a constant $0 < C < \infty$ such that $a_n \leq Cb_n$, we write $a_n = O(b_n)$. Besides, if there exist constants $0 < C_1, C_2 < \infty$ such that $C_1 b_n \leq a_n \leq C_2 b_n$, we write $a_n = \Theta(b_n)$. We use n, d to represent the number of training examples and the problem dimension, respectively. We also use the standard notation for (ϵ, δ)-DP where ϵ is the privacy budget and δ is the failure probability.

2 RELATED WORK

Over the past decade, many differentially private machine learning algorithms for convex ERM have been proposed. There are three main approaches to achieve differential privacy in such settings, including output perturbation [Wu et al., 2017, Zhang et al., 2017], objective perturbation [Chaudhuri et al., 2011, Kifer et al., 2012, Iyengar et al., 2019], and gradient perturbation [Bassily et al., 2014, Wang et al., 2017, Jayaraman et al., 2018]. However, other than the methods using gradient perturbation, it is very hard to generalize these methods to nonconvex ERM because of the difficulty in computing the sensitivity for nonconvex ERM. Thus, most differentially private algorithms for nonconvex ERM are based on the gradient perturbation, including our work. The problem with gradient perturbation approaches is that their iterative nature quickly consumes any reasonable privacy budget. Hence, the main challenge is to develop algorithms for nonconvex ERM that can provide sufficient utility while maintaining privacy with high computational efficiency.

Several recent works [Abadi et al., 2016, Papernot et al., 2016, Xie et al., 2018] studied deep learning with differential privacy. Abadi et al. [2016] proposed a method called moments accountant to keep track of the privacy cost of stochastic gradient descent algorithm during the training process, which provides a strong privacy guarantee. Papernot et al. [2016] established a Private Aggregation of Teacher Ensembles (PATE) framework to improve the privacy guarantee of deep learning for classification tasks. Xie et al. [2018] and Yoon et al. [2019] investigated the differentially private Generative Adversarial Nets (GAN) with different distance metrics. However, none of these works provide utility guarantees for their algorithms.

Table 1 summarizes differentially private nonconvex optimization algorithms that provide utility guarantees for nonconvex ERM. The Random Round Private Stochastic Gradient Descent (RRPSGD) method developed by Zhang et al. [2017] is the first differentially private nonconvex optimization algorithm with the utility guarantee. This method performs the perturbed SGD (adding Gaussian noise to the stochastic gradients), for a random number of iterations [Ghadimi and Lan, 2013]. The gradient complexity of RRPSGD is $O(n^2)$, which makes it impractical for most settings. Zhang et al. [2017] showed that RRPSGD is able to find a stationary point in expectation with a diminishing error $O\big((d\log(n/\delta)\log(1/\delta))^{1/4}/(n\epsilon)^{1/2}\big)$. Their analysis of the privacy guarantee is based on the standard privacy-amplification by subsampling result and strong composition theorem [Bassily et al., 2014]. Although such an analysis can be easily adapted to the nonconvex setting with stochastic optimization algorithms, it results in a large bound on the variance of the added noise compared with relaxed definitions such as the moments accountant [Abadi et al., 2016] and Gaussian differential privacy [Dong et al., 2019].

Wang et al. [2017] proposed the Differentially Private Gradient Descent (DP-GD) algorithm for nonconvex optimization. DP-GD achieves an improved utility guarantee of $O\big((d\log(1/\delta))^{1/4}/(n\epsilon)^{1/2}\big)$ compared to that of RRPSGD, with a reduced gradient complexity of $O\big(n^2\epsilon/(d\log(1/\delta))^{1/2}\big)$. The reason DP-GD can achieve this factor of $O\big((\log(n/\delta))^{1/4}\big)$ improvement, is that it uses the full gradient rather than the stochastic gradient. This

[1] A recent work [Arora et al., 2022] also achieves this utility guarantee with a worse gradient complexity.

Table 1: Comparison of different (ϵ, δ)-DP algorithms for nonconvex optimization. We report the utility bound in terms of $\mathbb{E}\|\nabla F(\boldsymbol{\theta}^p)\|_2$, where $\boldsymbol{\theta}^p$ is the output of the differentially private algorithm, $\mathbb{E}$ is taken over the randomness of the algorithm. We only present results in terms of n, d, ϵ, δ and ignore other parameters for simplicity. *Although Private SpiderBoost and DP-SRM have the same utility guarantee, Private SpiderBoost requires periodic full gradient computation, which makes it less scalable and results in worse gradient complexity.

Algorithm	Utility	Gradient Complexity
RRPSGD [Zhang et al., 2017]	$O\Big(\frac{(d\log(n/\delta)\log(1/\delta))^{1/4}}{(n\epsilon)^{1/2}}\Big)$	$O\big(n^2\big)$
DP-GD [Wang et al., 2017]	$O\Big(\frac{(d\log(1/\delta))^{1/4}}{(n\epsilon)^{1/2}}\Big)$	$O\Big(\frac{n^2\epsilon}{(d\log(1/\delta))^{1/2}}\Big)$
Private SpiderBoost [Arora et al., 2022]	$O\Big(\frac{(d\log(1/\delta))^{1/3}}{(n\epsilon)^{2/3}}\Big)$	$O\Big(\frac{(n\epsilon)^2}{d\log(1/\delta)} + \frac{n^{5/3}\epsilon^{2/3}}{(d\log(1/\delta))^{1/3}}\Big)$
DP-SRM (This paper)	$O\Big(\frac{(d\log(1/\delta))^{1/3}}{(n\epsilon)^{2/3}}\Big)$	$O\Big(\frac{(n\epsilon)^2}{d\log(1/\delta)}\Big)$

makes DP-GD computationally very expensive or even intractable for large-scale machine learning problems (n is big). Recently, Wang et al. [2019a] also proposed a differentially private stochastic algorithm for nonconvex optimization. Their goal is to find the local minima, while we aim to find the stationary point. In addition, their utility guarantee is asymptotic—it provides the desired utility guarantee only if an infinite number of iterations could be run. In contrast, our utility guarantee holds for a finite number of iterations.

Recently, Arora et al. [2022] developed a Private SpiderBoost algorithm for noncovex optimization, which achieves an improved utility guarantee of $O\big((d\log(1/\delta))^{1/3}/(n\epsilon)^{2/3}\big)$ with gradient complexity of $O\big((n\epsilon)^2/(d\log(1/\delta)) + n^{5/3}\epsilon^{2/3}/(d\log(1/\delta))^{1/3}\big)$. Although Private SpiderBoost attains the same improved utility guarantee as our method, it requires periodic full gradient computation, making it less scalable and results in worse gradient complexity when $d \geq O\big(\sqrt{n}\epsilon^2/\log(1/\delta)\big)$.

3 PRELIMINARIES

We consider the empirical risk minimization (ERM) problem: given a training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ drawn from some unknown but fixed data distribution with $\mathbf{x}_i \in \mathbb{R}^D, y_i \in \mathcal{Y} \subseteq \mathbb{R}$, we aim to find a solution $\widehat{\boldsymbol{\theta}} \in \mathbb{R}^d$ that minimizes the following empirical risk

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} F(\boldsymbol{\theta}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta}), \tag{3.1}$$

where $F(\boldsymbol{\theta})$ is the empirical risk function (i.e., training loss), $f_i(\boldsymbol{\theta}) = \ell(\boldsymbol{\theta}; \mathbf{x}_i, y_i)$ is the loss function defined on the i-th training example $(\mathbf{x}_i, y_i)$, and $\boldsymbol{\theta} \in \mathbb{R}^d$ is the model parameter we want to learn.

Here, we provide some definitions and lemmas that will be used in our theoretical analysis.

Definition 3.1. $\boldsymbol{\theta} \in \mathbb{R}^d$ is an ζ-approximate stationary point if $\|\nabla f(\boldsymbol{\theta})\|_2 \leq \zeta$.

Definition 3.2. A function $f : \mathbb{R}^d \to \mathbb{R}$ is G-Lipschitz, if for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$, we have

$$|f(\boldsymbol{\theta}_1) - f(\boldsymbol{\theta}_2)| \leq G\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2.$$

Definition 3.3. A function $f : \mathbb{R}^d \to \mathbb{R}$ has L-Lipschitz gradient, if for all $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{R}^d$, we have

$$\|\nabla f(\boldsymbol{\theta}_1) - \nabla f(\boldsymbol{\theta}_2)\|_2 \leq L\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2.$$

Differential privacy provides a formal notion of privacy, introduced by Dwork et al. [2006]:

Definition 3.4 ((ϵ, δ)-DP [Dwork et al., 2006])**.** A randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent data sets $S, S' \in \mathcal{S}^n$ differing by one element, and any output subset $O \subseteq \mathcal{R}$, it holds that $\mathbb{P}[\mathcal{M}(S) \in O] \leq e^{\epsilon} \cdot \mathbb{P}[\mathcal{M}(S') \in O] + \delta$.

To achieve (ϵ, δ)-DP for a given function $q : \mathcal{S}^n \to \mathcal{R}$, we can use Gaussian mechanism [Dwork and Roth, 2014] $\mathcal{M} = q(S) + \mathbf{u}$, where $\mathbf{u}$ is a standard Gaussian random vector with variance that is proportional to the ℓ_2-sensitivity of the function q, $\Delta(q)$, which is defined as follows.

Definition 3.5 (ℓ_2-sensitivity[Dwork and Roth, 2014])**.** For two adjacent datasets $S, S' \in \mathcal{S}^n$ differing by one element, the ℓ_2-sensitivity $\Delta(q)$ of a function $q : \mathcal{S}^n \to \mathcal{R}$ is defined as $\Delta(q) = \sup_{S,S'} \|q(S) - q(S')\|_2$.

Rényi differential privacy. Although the notion of (ϵ, δ)-DP is widely used in the output and objective perturbation methods, it suffers from the loose composition and privacy-amplification by subsampling results, which makes it unsuitable for the stochastic iterative learning algorithms. In this

work, we will make use of the notion of Rényi Differential Privacy (RDP) [Mironov, 2017] which is particularly useful when the dataset is accessed by a sequence of randomized mechanisms [Wang et al., 2019b].

Definition 3.6 (RDP [Mironov, 2017])**.** For $\alpha > 1, \rho > 0$, a randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ is (α, ρ)-Rényi Differential Privacy, if for all adjacent datasets $S, S' \in \mathcal{S}^n$ differing by one element, we have $D_\alpha\big(\mathcal{M}(S)||\mathcal{M}(S')\big) := \log \mathbb{E}\big[\big(\mathcal{M}(S)/\mathcal{M}(S')\big)^\alpha\big]/(\alpha-1) \le \rho$.

By Definition 3.6, RDP measures the ratio of probability distributions $\mathcal{M}(S)$ and $\mathcal{M}(S')$ by α-order Renyi Divergence with $\alpha \in (1, \infty)$. As $\alpha \to \infty$, RDP reduces to ϵ-DP.

To further improve the privacy guarantee when using the Gaussian mechanisms to satisfy RDP, we establish the following privacy-amplification by subsampling result, which is derived based on the result in [Wang et al., 2019b].

Lemma 3.7. Given a function $q : \mathcal{S}^n \to \mathcal{R}$, the Gaussian Mechanism $\mathcal{M} = q(S) + \mathbf{u}$, where $\mathbf{u} \sim N(0, \sigma^2\mathbf{I})$, satisfies $(\alpha, \alpha\Delta^2(q)/(2\sigma^2))$-RDP. In addition, if we apply the mechanism $\mathcal{M}$ to a subset of samples using uniform sampling without replacement with sampling rate τ, $\mathcal{M}$ satisfies $(\alpha, 3.5\tau^2\Delta^2(q)\alpha/\sigma^2)$-RDP given $\sigma'^2 = \sigma^2/\Delta^2(q) \ge 0.7$, $\alpha \le 2\sigma^2 \log\big(1/\big(\tau\alpha\big(1+\sigma'^2\big)\big)\big)/3 + 1$.

Remark 3.8 (*Comparison with moment accountant*)**.** Suppose $\Delta(q) = 1$, Lemma 3.7 suggests that to achieve $(\alpha, 3.5\tau^2\alpha/\sigma^2)$-RDP of the subsampled Gaussian mechanism, we require $\sigma^2 \ge 0.7$. For the moment accountant based method [Abadi et al., 2016], it can achieve the asymptotic privacy guarantee of $\big(\alpha, \tau^2\alpha/(1-\tau)\sigma^2 + O(\tau^3\alpha^3/\sigma^3)\big)$-RDP when τ goes to zero and $\sigma^2 \ge 1$, $\alpha \le \sigma^2 \log(1/(\tau\sigma))$. In contrast to moment accountant, our result has a closed-form bound on the privacy guarantee and a relaxed requirement of σ^2.

It is worth noting that there exist some other works [Mironov et al., 2019, Zhu and Wang, 2019] also studying the privacy-amplification by subsampling results. However, they consider the Poisson subsampling approach, which is different from our uniform subsampling method.

Based on Lemma 3.7, we can establish a strong privacy guarantee of our method in terms of RDP, and then transfer it to (ϵ, δ)-DP using the following lemma.

Lemma 3.9 (Mironov [2017])**.** If a randomized mechanism $\mathcal{M} : \mathcal{S}^n \to \mathcal{R}$ satisfies (α, ρ)-RDP, then $\mathcal{M}$ satisfies $(\rho + \log(1/\delta)/(\alpha-1), \delta)$-DP for all $\delta \in (0, 1)$.

4 ALGORITHM

Our proposed algorithm for differentially private nonconvex ERM, is illustrated in Algorithm 1.

Algorithm 1 Differentially Private Stochastic Recursive Momentum (DP-SRM)

input $\boldsymbol{\theta}^0, T, G, L, \gamma, \beta, n_0$, privacy parameters ϵ, δ, accuracy for the first-order stationary point ζ

1: Uniformly sample b_0 examples without replacement indexed by $\mathcal{B}_0$

2: Compute $\mathbf{v}^0 = \nabla F_{\mathcal{B}_0}(\boldsymbol{\theta}^0)$, where $\nabla F_{\mathcal{B}_0}(\boldsymbol{\theta}^0) = \sum_{i\in\mathcal{B}_0} \nabla f_i(\boldsymbol{\theta}^0)/b_0$, draw $\mathbf{u}^0 \sim N(0, \sigma_0^2\mathbf{I}_d)$ with $\sigma_0^2 = 14TG^2\alpha/(\beta n^2\epsilon)$, $\alpha = \log(1/\delta)/\big((1-\beta)\epsilon\big) + 1$

3: Release the differentially private gradient estimator $\mathbf{v}_p^0 = \mathbf{v}^0 + \mathbf{u}^0$

4: **for** $t = 0, 1, 2, \ldots, T-1$ **do**

5: $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta_t\mathbf{v}_p^t$, where $\eta_t = \min\big\{\zeta/(n_0L\|\mathbf{v}_p^t\|_2), 1/(2n_0L)\big\}$

6: Uniformly sample b examples without replacement indexed by $\mathcal{B}_{t+1}$

7: Compute $\mathbf{v}^{t+1} = \nabla F_{\mathcal{B}_{t+1}}(\boldsymbol{\theta}^{t+1}) + (1-\gamma)\big(\mathbf{v}_p^t - \nabla F_{\mathcal{B}_{t+1}}(\boldsymbol{\theta}^t)\big)$, draw $\mathbf{u}^{t+1} \sim N(0, \sigma^2\mathbf{I}_d)$ with $\sigma^2 = 14T\big((1-\gamma)\zeta/n_0 + \gamma G\big)^2\alpha/(\beta n^2\epsilon)$, $\alpha = \log(1/\delta)/\big((1-\beta)\epsilon\big) + 1$

8: Release the differentially private gradient estimator $\mathbf{v}_p^{t+1} = \mathbf{v}^{t+1} + \mathbf{u}^{t+1}$

9: **end for**

output $\widetilde{\boldsymbol{\theta}}$ chosen uniformly at random from $\{\boldsymbol{\theta}^t\}_{t=0}^{T-1}$

The main idea is to construct the differentially private gradient estimator $\mathbf{v}_p^t$ iteratively based on the information obtained from the previous updates. We initialize $\mathbf{v}^0$ to be the mini-batch stochastic gradient $\nabla F_{\mathcal{B}_0}(\boldsymbol{\theta}^0)$ and inject Gaussian noise, $\mathbf{u}^0$, with covariance matrix $\sigma_0^2\mathbf{I}_d$ (lines 2, 3), to make it differentially private. Then, we recursively update $\mathbf{v}^t$ (line 7) as $\mathbf{v}^t = \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^t) + (1-\gamma)\big(\mathbf{v}_p^{t-1} - \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1})\big)$, where $\nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^t)$, $\nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1})$ are mini-batch stochastic gradients and $\mathbf{v}_p^{t-1}$ is the private gradient estimator released at the last iteration. The momentum parameter, γ, is used to control the decay rate of the prior information, $\mathbf{v}_p^{t-1} - \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1})$. This is called stochastic recursive momentum [Cutkosky and Orabona, 2019], which can lead to fast convergence. After updating $\mathbf{v}^t$, we again inject Gaussian noise $\mathbf{u}^t$ with covariance matrix $\sigma_2\mathbf{I}_d$ (line 8), to provide differential privacy. The variance σ_0^2, σ^2 of the Gaussian random vectors are determined by our RDP-based analysis. We choose an adaptive step size (line 5) to bound the sensitivity of the gradient estimator $\mathbf{v}_p^t$, which is the key to establish the tight privacy and utility guarantees (Section 6) of our algorithm.

5 MAIN THEORETICAL RESULTS

In this section, we establish formal privacy and utility guarantees for Algorithm 1.

Theorem 5.1. Suppose that each component function f_i

is G-Lipschitz and has L-Lipschitz gradient. Given the total number of iterations T, the momentum parameter γ and the accuracy for the first-order stationary point ζ, for any $\delta > 0$ and the privacy budget ϵ, Algorithm 1 satisfies (ϵ, δ)-differential privacy with $\sigma_0^2 = 14TG^2\alpha/(\beta n^2\epsilon)$ and $\sigma^2 = 14T\big((1-\gamma)\zeta/n_0 + \gamma G\big)^2\alpha/(\beta n^2\epsilon)$ if we have $\alpha - 1 = \log(1/\delta)/\big((1-\beta)\epsilon\big) \leq 2\sigma'^2 \log\big(1/\big(\tau\alpha(1+\sigma'^2)\big)\big)/3$ with $\beta \in (0,1)$ and $\sigma'^2 = \min\{b^2\sigma^2/\big(4((1-\gamma)\zeta/n_0 + \gamma G)^2\big), b_0^2\sigma_0^2/(4G^2)\} \geq 0.7$, where b_0 and b are batch sizes, and $\tau = \max\{b_0/n, b/n\}$.

Remark 5.2. According to Theorem 5.1, there exists a constraint on the parameter α, which is due to the privacy-amplification by subsampling result in Lemma 3.7, and is similar to the constraint given by the moments accountant [Abadi et al., 2016] and other RDP-based analyses with subsampling approaches [Mironov et al., 2019, Zhu and Wang, 2019]. Furthermore, as we mentioned in Remark 3.8, our result relaxes the requirement of the variance σ'^2 compared with the moments accountant based analysis.

Following the previous work [Bassily et al., 2019], we can get rid of the constraints in Theorem 5.1 by using a larger mini-batch size, as states in the following corollary.

Corollary 5.3. Given the total number of iterations T, the momentum parameter γ and the accuracy for the first-order stationary point ζ. Under the same conditions of Theorem 5.1 on $f_i, \sigma_0^2, \sigma^2$, for any $\delta > 0$ and the privacy budget ϵ, Algorithm 1 satisfies (ϵ, δ)-differential privacy if we choose $b_0^2 = b^2 = n^2\epsilon/T$, $\beta = 1/2$, and T is larger than $O\big(\log^4(1/\delta)/\epsilon^3\big)$.

Theorem 5.1 and Corollary 5.3 require that each component function f_i is G-Lipschitz and has L-Lipschitz gradient which will be used to derive the sensitivity of the underlying query function (i.e., the gradient estimator $\mathbf{v}^t$ in Algorithm 1) and thus determine the variance of the Gaussian noise. The G-Lipschitz condition has been widely assumed in the literature of differential privacy [Abadi et al., 2016, Wang et al., 2017, Jayaraman et al., 2018, Bassily et al., 2019], and the L-Lipschitz gradient condition has also been made in several previous works [Zhang et al., 2017, Feldman et al., 2020]. In practice, we can use the clipping technique [Abadi et al., 2016] to ensure that at each iteration, $\|\nabla f_i(\boldsymbol{\theta}^t)\|_2 \leq C_1$ and $\|\nabla f_i(\boldsymbol{\theta}^t) - \nabla f_i(\boldsymbol{\theta}^{t-1})\|_2 \leq C_2$, where C_1, C_2 are some predefined constants. As a result, we can guarantee that the sensitivity of $\mathbf{v}^t$ is bounded by $2\big((1-\gamma)C_2 + \gamma C_1\big)/b$ (see (6.1)). Then, we can replace G and ζ/n_0 with C_1 and C_2 in Algorithm 1 to establish the same privacy guarantee.

The following theorem shows the utility guarantee and the gradient complexity, which is the total number of the stochastic gradients we need to estimate during the training process, of Algorithm 1.

Theorem 5.4. Under the same conditions of Theorem 5.1 on $f_i, \sigma^2, \sigma_0^2, \sigma'^2, \alpha$, if we choose the number of iterations $T = C_1(n\epsilon LD_F)^{4/3}/\big(G^{8/3}(d\log(1/\delta)^{2/3}\big)$, where $D_F = F(\boldsymbol{\theta}^0) - F(\boldsymbol{\theta}^*)$ and $F(\boldsymbol{\theta}^*)$ is a global minimum of F, the accuracy for the first-order stationary point $\zeta = C_2\big(GLD_F d\log(1/\delta)\big)^{1/3}/(n\epsilon)^{2/3}$, batch sizes $b_0 = C_3G^3/(\zeta LD_F)$, $b = C_4G/(n_0\zeta)$, $n_0 = LD_F/G^2$, the momentum parameter $\gamma^2 = C_5\zeta^2/(n_0^2G^2)$ and $n\epsilon \geq C_6\max\{G^8\log^2(1/\delta)/(LD_Fd)^4, \sqrt{G^4d\log(1/\delta)}/(LD_F)\}$, then the output $\widetilde{\boldsymbol{\theta}}$ of Algorithm 1 and satisfies the following

$$\mathbb{E}\|\nabla F(\widetilde{\boldsymbol{\theta}})\|_2 \leq C_7\bigg(\frac{\sqrt{GLD_Fd\log(1/\delta)}}{n\epsilon}\bigg)^{\frac{2}{3}},$$

where $\{C_i\}_{i=1}^7$ are absolute constants, and the expectation is taken over all the randomness of the algorithm, i.e., the random Gaussian noise and the subsample gradient. Since $T = O\big((n\epsilon LD_F)^{4/3}/\big(G^{8/3}(d\log(1/\delta)^{2/3}\big)$, $b_0 = b = O\big(G^{8/3}(n\epsilon)^{2/3}/(LD_F)^{4/3}(d\log(1/\delta))^{1/3}\big)$, the total gradient complexity of Algorithm 1 is $O\big((n\epsilon)^2/(d\log(1/\delta))\big)$.

Remark 5.5 (*Comparison with existing methods*)**.** According to Theorem 5.4, our method can achieve the following utility guarantee $O\big(\big(GLD_Fd\log(1/\delta)\big)^{1/3}/(n\epsilon)^{\frac{2}{3}}\big)$. This result is better than the previous best-known result for differentially private nonconvex optimization method [Wang et al., 2017]. Furthermore, their method is based on gradient descent, which is computationally very expensive in large-scale machine learning problems. Furthermore, the gradient complexity of our method is $O\big((n\epsilon)^2/\big(d\log(1/\delta)\big)\big)$. This result is smaller than $O(n^2)$ gradient complexity provided by Zhang et al. [2017] and $O\big(n^2\epsilon/(d\log(1/\delta))^{1/2}\big)$ gradient complexity provided by Wang et al. [2017] when d is large. Compared with Private SpiderBoost [Arora et al., 2022], our method has better gradient complexity when $d \geq O\big(\sqrt{n}\epsilon^2/\log(1/\delta)\big)$.

Theorem 5.4 shows that our method only requires the computation of minibatch gradients with batch size at the order of $O\big((n\epsilon)^{2/3}/(d\log(1/\delta)^{1/3}\big)$ (ignoring the dependence on other parameters). Therefore, our method is more scalable than existing differentially private stochastic variance-reduced algorithms, such as DP-SVRG [Wang et al., 2017] for convex optimization and Private SpiderBoost [Arora et al., 2022] for nonconvex optimization, which often require the periodic computation of the checkpoint gradient with a giant batch size (full batch in DP-SVRG and Private SpiderBoost).

6 PROOF OUTLINE OF THE MAIN RESULTS

In this section, we present the proof outline of the main results in Section 5. Our proof involves new techniques for the privacy and utility guarantees that are of general use for

variance reduction-based algorithms. The detailed proof can be found in Section B in Appendix.

6.1 PRIVACY GUARANTEE

According to Algorithm 1, the mechanism at t-th iteration is $\mathcal{M}_t$, which is a composition of t Gaussian mechanisms: $\mathcal{G}_0, \ldots, \mathcal{G}_t$, where $\mathcal{G}_0 = \nabla F_{\mathcal{B}_0}(\boldsymbol{\theta}^0) + \mathbf{u}^0$ and $\mathcal{G}_t = \nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^t) - (1-\gamma)\nabla F_{\mathcal{B}_t}(\boldsymbol{\theta}^{t-1}) + \mathbf{u}^t$. Therefore, we want to show that $\mathcal{M}_t$ is differentially private. For the given dataset S, we use S' to denote its neighboring dataset with one different example indexed by i'

There are two main challenges in providing a tight privacy analysis. The first one is to deal with the subsampled mechanisms $\{\mathcal{G}_i\}_{i=0}^{T-1}$. The second one is to control the sensitivity of $\mathcal{G}_t$ when $t > 0$. The first challenge can be addressed by our privacy-amplification by subsampling result (Lemma 3.7), which gives us a tight closed-form bound on the privacy guarantee. We can overcome the second challenge by using an adaptive stepsize, enabling us to use a small amount of random noise to achieve differential privacy.

According to Algorithm 1, $\mathcal{G}_t$ is the application of the following Gaussian mechanism $\widetilde{\mathcal{G}}_t$ to a subset of uniformly sampled examples, indexed by $\mathcal{B}_t$

$$\widetilde{\mathcal{G}}_t = \begin{cases} \frac{1}{b}\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^0) + \mathbf{u}^0, & t = 0 \\ \frac{1}{b}\sum_{i=1}^n \big(\nabla f_i(\boldsymbol{\theta}^t) - \phi \nabla f_i(\boldsymbol{\theta}^{t-1})\big) + \mathbf{u}^t, & t > 0, \end{cases}$$

where $\phi = 1 - \gamma$. For $\widetilde{\mathbf{q}}_0 = \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^0)/b_0$ in $\widetilde{\mathcal{G}}_0$, the sensitivity $\Delta(\widetilde{\mathbf{q}}_0)$ is determined by

$$\|\widetilde{\mathbf{q}}_0(S) - \widetilde{\mathbf{q}}_0(S')\|_2 \le \frac{1}{b}\|\nabla f_i(\boldsymbol{\theta}^0) - \nabla f_{i'}(\boldsymbol{\theta}^0)\|_2 \le \frac{2G}{b_0},$$

where the last inequality is due to G-Lipschitz of each component function. For $\widetilde{\mathbf{q}}_t = \sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^t)/b - (1-\gamma)\sum_{i=1}^n \nabla f_i(\boldsymbol{\theta}^{t-1})/b$ in $\widetilde{\mathcal{G}}_t$ when $t > 0$, the sensitivity $\Delta(\widetilde{\mathbf{q}}_t) = \|\widetilde{\mathbf{q}}_t(S) - \widetilde{\mathbf{q}}_t(S')\|_2$ is determined by

$$\frac{1-\gamma}{b}\|\nabla f_i(\boldsymbol{\theta}^t) - \nabla f_i(\boldsymbol{\theta}^{t-1}) + \nabla f_{i'}(\boldsymbol{\theta}^t) - \nabla f_{i'}(\boldsymbol{\theta}^{t-1})\|_2 + \frac{\gamma}{b}\|\nabla f_i(\boldsymbol{\theta}^t) - \nabla f_{i'}(\boldsymbol{\theta}^t)\|_2. \tag{6.1}$$

Therefore, we have

$$\begin{aligned}\|\mathbf{q}_t(S) - \mathbf{q}_t(S')\|_2 &\le \frac{2L(1-\gamma)}{b}\|\boldsymbol{\theta}^t - \boldsymbol{\theta}^{t-1}\|_2 + \frac{2\gamma G}{b} \\ &= \frac{2L(1-\gamma)}{b}\eta_{t-1}\|\mathbf{v}_p^{t-1}\|_2 + \frac{2\gamma G}{b} \\ &\le \frac{2(1-\gamma)\zeta}{n_0 b} + \frac{2\gamma G}{b},\end{aligned}$$

where the first inequality is due to L-Lipschitz continuous gradient and G-Lipschitz of each component function. The last inequality comes from the adaptive stepsize $\eta_t = \min\big\{\zeta/(n_0 L\|\mathbf{v}_p^t\|_2), 1/(2n_0 L)\big\}$. Note that the proposed adaptive stepsize η_t is the key to control the sensitivity of $\widetilde{\mathbf{q}}_t$. If we choose a fixed stepsize such as $\eta_t = 1/(2L)$, the sensitivity of $\widetilde{\mathbf{q}}_t$ will be in the order of $O(G^2/b)$, which will lead to a much larger random noise to achieve differential privacy and thus deteriorate the utility of our method.

According to Lemma 3.7, if the noise $\mathbf{u}^0$ and $\mathbf{u}^t$ satisfy $\sigma_0^2 = 14T\alpha G^2/(\beta n^2\epsilon)$ and $\sigma^2 = 14T\alpha\big((1-\gamma)\zeta/n_0 + \gamma G\big)^2/(\beta n^2\epsilon)$, the Gaussian mechanism $\widetilde{\mathcal{G}}_t$ satisfies $\big(\alpha, \beta\epsilon n^2/\big(7b_0^2 T\big)\big)$-RDP, and the privacy-amplification by subsampling result shows that $\mathcal{G}_t$ satisfies $(\alpha, \beta\epsilon/T)$-RDP. Therefore, by the composition rule of RDP Mironov [2017], after T' iterations, Algorithm 1 satisfies $(\alpha, \beta T'\epsilon/T)$-RDP. According to Lemma 3.9 and $\alpha = \log(1/\delta)/\big((1-\beta)\epsilon\big) + 1$, we have that after T' iterations, Algorithm 1 satisfies $(T'\epsilon/T, \delta)$-DP.

6.2 UTILITY GUARANTEE

According to the definition of $\widetilde{\boldsymbol{\theta}}$, we have

$$\begin{aligned}\mathbb{E}\|\nabla F(\widetilde{\boldsymbol{\theta}})\|_2 &= \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(\boldsymbol{\theta}^t)\|_2 \\ &\le \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\|\mathbf{v}_p^t\big\|_2 + \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla F(\boldsymbol{\theta}^t) - \mathbf{v}_p^t\big\|_2,\end{aligned}$$

where the expectation is taken over all the randomness of the algorithm. The key challenge in establishing a tight utility guarantee is to derive tight upper bounds for $\sum_{t=0}^{T-1}\mathbb{E}\big\|\mathbf{v}_p^t\big\|_2/T$ and $\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla F(\boldsymbol{\theta}^t) - \mathbf{v}_p^t\big\|_2/T$ when we have adaptive stepsize η_t and the random noise $\mathbf{u}^t$ in $\mathbf{v}_p^t$.

First of all, by taking into account the adaptive stepsize η_t, we can upper bound the term $\sum_{t=0}^{T-1}\mathbb{E}\big\|\mathbf{v}_p^t\big\|_2/T$ as follows

$$\frac{4n_0 L D_F}{T\zeta} + \frac{1}{T\zeta}\sum_{t=0}^{T-1}\mathbb{E}\big\|\nabla F(\boldsymbol{\theta}^t) - \mathbf{v}_p^t\big\|_2^2 + 2\zeta,$$

where $D_F = F(\boldsymbol{\theta}^0) - F(\boldsymbol{\theta}^*)$. Furthermore, we can obtain the upper bound for $\sum_{t=0}^{T-1}\mathbb{E}\big\|\mathbf{v}_p^t - \nabla F(\boldsymbol{\theta}^t)\big\|_2^2/T$ as follows

$$\frac{2(1-\gamma)^2\zeta^2}{n_0^2\gamma b} + \frac{2\gamma G^2}{b} + \frac{G^2}{T\gamma b_0} + \frac{Td\sigma^2 + d\sigma_0^2}{T\gamma},$$

where the first term is determined by η_t, and the last term is determined by the random noise $\mathbf{u}^t$ in $\mathbf{v}_p^t$. The last term in this bound is dominated by $d\sigma^2/\gamma$, which validates the necessity of using the adaptive stepsize to control the sensitivity of $\mathbf{v}^t$ and thus enable a small σ^2.

Finally, combining these two new bounds and plugging the value of parameters in Theorem 5.4, we can obtain that

$$\mathbb{E}\|\nabla F(\widetilde{\boldsymbol{\theta}})\|_2 \le C_1\zeta + C_2\frac{\sqrt{GLD_F d\log(1/\delta)}}{n\epsilon\sqrt{\zeta}}.$$

By solving the smallest ζ, we can obtain $\zeta = (GLD_F d\log(1/\delta))^{1/3}/(n\epsilon C_1/C_2)^{2/3}$. Thus we have $\mathbb{E}\|\nabla F(\widetilde{\boldsymbol{\theta}})\|_2 \leq C_3\zeta$, where C_1, C_2, C_3 are some constants.

7 EXPERIMENTS

This section presents results from experiments that evaluate our method's performance on different nonconvex ERM problems and different datasets. All experiments are implemented in Pytorch platform version 1.2.0 within Python 3.7.6. on a local machine which comes with Intel Xeon 4214 CPUs and NVIDIA GeForce RTX 2080Ti GPU (11G GPU RAM).

7.1 NONCONVEX LOGISTIC REGRESSION

We first consider the binary logistic regression problem with a nonconvex regularizer [Reddi et al., 2016b]

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n y_i \log\phi(\mathbf{x}_i^\top\boldsymbol{\theta}) + (1-y_i)\log\left[1-\phi(\mathbf{x}_i^\top\boldsymbol{\theta})\right] + \lambda\sum_{i=1}^d \theta_j^2/(1+\theta_j^2),$$

where $\phi(x) = 1/\big(1+\exp(-x)\big)$ is the sigmoid function, θ_j is the j-th coordinate of $\boldsymbol{\theta}$, and $\lambda > 0$ is the regularization parameter. We set $\lambda = 0.001$ in this experiment. Here, we consider two commonly-used binary classification benchmark datasets: *a9a* dataset, which contains 32561 training examples, 16281 test examples, 123 features, and *ijcnn1* dataset with 49990 training examples, 91701 test examples, 22 features. We report the results for the *a9a* dataset in the main paper and defer the results for the *ijcnn1* dataset to Appendix A.

Baseline methods. We compare our method (DP-SRM) with random round private stochastic gradient descent (RRPSGD) proposed by Zhang et al. [2017], differentially private gradient descent (DP-GD) proposed by Wang et al. [2017], and differentially private adaptive gradient descent (DP-AGD) proposed by Lee and Kifer [2018]. We do not compare our method with Private SpiderBoost [Arora et al., 2022] since it is unclear how to practically determine the privacy guarantee-related parameters of their algorithm.

Gradient clipping and privacy tracking. We use the gradient clipping technique of Abadi et al. [2016] to ensure that at t-th iteration of Algorithm 1, $\|\nabla f_i(\theta^t)\|_2$ and $\|\nabla f_i(\theta^t) - \nabla f_i(\theta^{t-1})\|_2$ are upper bounded by some predefined values C_1 and C_2, respectively. This will ensure that the sensitivity of the gradient estimator $\mathbf{v}^t$ is upper bounded by $2\big((1-\gamma)C_1 + \gamma C_2\big)$ (see equation (6.1)), and gives us the desired privacy protection. At each iteration, we add the Gaussian noise with variance σ^2, and keep track of the RDP according to Lemma 3.7 and transfer it to (ϵ, δ)-DP according to Lemma 3.9.

Parameters. For all the algorithms, the step size is tuned around the theoretical values to give the fastest convergence using grid search. For our method, we tune the batch size b by searching the grid $\{50, 100, 200\}$. We set $C_1 = 1, C_2 = 0.01$ and $\gamma = C_2$. We choose $\epsilon \in \{0.2, 0.5\}$ and $\delta = 10^{-5}$.

Results. Due to the randomized nature of all the algorithms, the experimental results are obtained by averaging the results over 30 runs. Figures 1 shows the objective function value and the gradient norm of different algorithms for privacy budgets $\epsilon \in \{0.2, 0.5\}$ on *a9a* datasets. We also report the 95% confidence interval of these results. We can see from the plots that our DP-SRM algorithm outperforms the other three baseline algorithms in terms of the objective loss, gradient norm, and convergence rate by a large margin. Tables 2 summarizes the test error of different algorithms as well as the CPU time (in seconds) of the training process. The results also corroborate the advantages of our method in terms of accuracy and efficiency.

7.2 CONVOLUTIONAL NEURAL NETWORKS

We compare our algorithm with the differentially private stochastic gradient descent (DP-SGD) algorithm proposed by Abadi et al. [2016] on training convolutional neural networks for image classification on both MNIST [LeCun et al., 1998] and CIFAR-10 [Krizhevsky and Hinton, 2009] datasets.

Architecture for MNIST. For MNIST dataset, we consider a 4 layer CNN [2], which can achieve 99% classification accuracy on the test dataset after training with SGD.

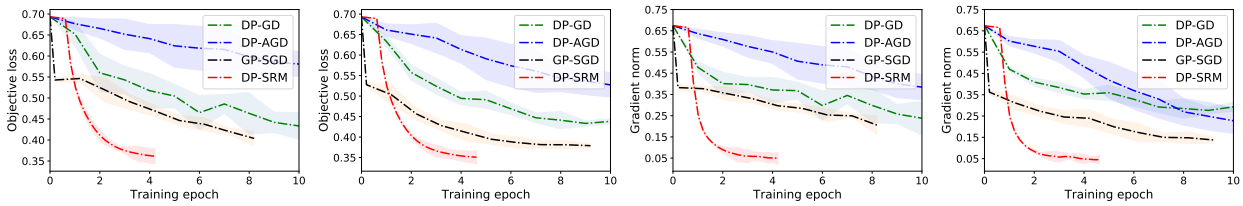
Parameters for MNIST. We choose privacy budgets $\epsilon \in \{1.2, 3.0, 7.0\}$, and set $\delta = 10^{-5}$. To ensure the privacy guarantee (see (6.1)), we set the clipping parameter $C_1 = 1.5$ for the term $\|\nabla f_i(\theta^t)\|_2$. For the term $\|\nabla f_i(\theta^t) - \nabla f_i(\theta^{t-1})\|_2$, we choose the clipping parameter C_2 from the grid $\{0.01, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99\}$. For both DP-SGD and DP-SRM, we tune the batch size b by searching the grid $\{256, 512, 1024\}$ and the step size by $\{0.01, 0.05, 0.1, 0.25, 0.5\}$. For DP-SRM, we tune the batch size b_0 by $\{b, 2b, 4b\}$. In addition, we set the momentum parameter $\gamma = C_2$.

Results for MNIST. Figures 2 illustrates the average test error and the corresponding 95% confidence interval of different methods versus the number of iterations as well as the training time (in seconds) under the privacy budgets $\epsilon = 1.2$ and $\epsilon = 3.0$ over 30 trials. We see similar results under the privacy budget $\epsilon = 7.0$, and thus defer them in Section A in Appendix. The CNN trained by the non-private SGD can achieve 1% test error after 20 epochs. Figure 2(a)

[2] https://github.com/facebookresearch/pytorch-dp.

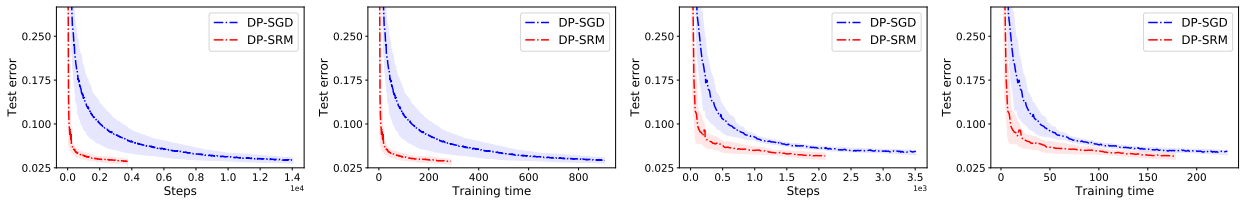
Table 2: Comparison of different algorithms on *a9a* dataset when $\epsilon \in \{0.2, 0.5\}$ and $\delta = 10^{-5}$. We use the STORM algorithm [Cutkosky and Orabona, 2019] as the non-private baseline.

Privacy Budget	Non-private Baseline	Method	Test Error	Data Passes	CPU time (s)	Gradient Norm
$\epsilon = 0.2$	0.3346 (0.007)	DP-GD	0.4155 (0.0107)	20	1.245	0.0953 (0.0212)
		DP-AGD	0.3713 (0.0043)	360	96.21	0.0437 (0.0020)
		RRPSGD	0.4019 (0.0033)	8	39.61	0.2175 (0.0116)
		DP-SRM	**0.3579 (0.0009)**	**4**	**0.6007**	**0.0528 (0.0042)**
$\epsilon = 0.5$	0.3346 (0.007)	DP-GD	0.3859 (0.0057)	20	1.261	0.0866 (0.0129)
		DP-AGD	0.3627 (0.0038)	365	95.45	0.0402 (0.0022)
		RRPSGD	0.3861 (0.0028)	10	52.32	0.1454 (0.0126)
		DP-SRM	**0.3506 (0.0011)**	**5**	**0.7383**	**0.0502 (0.0061)**



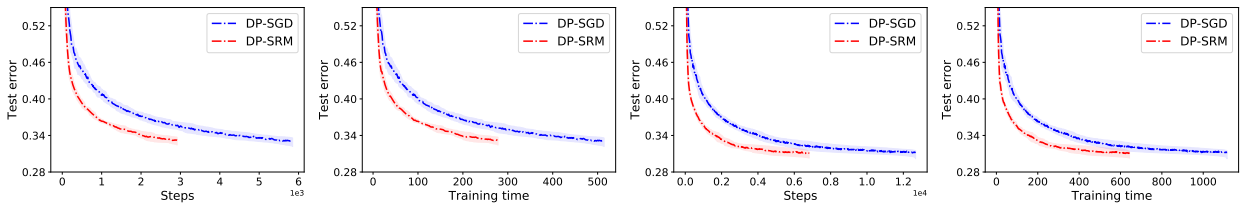
(a) $\epsilon = 0.2$ (b) $\epsilon = 0.5$ (c) $\epsilon = 0.2$ (d) $\epsilon = 0.5$

Figure 1: Results for nonconvex logistic regression on *a9a* dataset. (a), (b) illustrate the objective loss versus the number of epochs. (c), (d) present the gradient norm versus the number of epochs.



(a) $\epsilon = 3.0$ (b) $\epsilon = 3.0$ (c) $\epsilon = 1.2$ (d) $\epsilon = 1.2$

Figure 2: Results on MNIST dataset. (a), (b) depict the test error under the privacy budget $\epsilon = 3.0$. (c), (d) illustrate the test error under the privacy budget $\epsilon = 1.2$.



(a) $\epsilon = 2.0$ (b) $\epsilon = 2.0$ (c) $\epsilon = 4.0$ (d) $\epsilon = 4.0$

Figure 3: Results for CNN6 on CIFAR-10 dataset. (a), (b) depict the test error under the privacy budget $\epsilon = 2.0$. (c), (d) illustrate the test error under the privacy budget $\epsilon = 4.0$

and Figure 2(c) show that our proposed method can achieve 3.62% and 4.49% test errors when $\epsilon = 3.0$ and $\epsilon = 1.2$, which are better than DP-SGD with 3.81% and 5.33% test errors. Besides, our method converges faster than DP-SGD. Figure 2(a) and Figure 2(b) demonstrate that compared with DP-SGD, our method only takes $0.3\times$ iterations and $0.4\times$ training time to achieve comparable performances under the privacy budget $\epsilon = 3.0$.

Architecture for CIFAR-10. We consider two convolutional neural networks for CIFAR-10. The first one is a five layer CNN with two convolutional layers and three fully connected layers, and we call it CNN5 [3]. For CNN5, we train it from the scratch using our DP-SRM method and the DP-SGD method [Abadi et al., 2016] and compare their performances in terms of the model accuracy, iteration numbers and the training time. For the second one, we consider a similar architecture as in Abadi et al. [2016], which has three convolutional layers with 32, 64, 128 filters in each convolution layer and three fully connected layers, and we denote it by CNN6. For CNN6, we follow the same experiment setting as in Abadi et al. [2016]: we use CIFAR-100 dataset as a public dataset, and first train a network with the same architecture on this dataset as the pretrained model. Then, we initialize the convolutional layers of CNN6 using the cnvolutional layers of the pretrained model, and only train the fully connected layers of CNN6 on CIFAR-10 dataset using different private methods.

Parameters for CNN6. We choose three different privacy budgets $\epsilon \in \{2.0, 4.0, 8.0\}$ and $\delta = 10^{-5}$. We set the clipping parameter $C_1 = 2$ for the term $\|\nabla f_i(\theta^t)\|_2$. For the term $\|\nabla f_i(\theta^t) - \nabla f_i(\theta^{t-1})\|_2$, we choose the clipping parameter C_2 by searching the grid $\{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99\}$. For DP-SGD, we tune the step size by searching the grid $\{0.01, 0.02, 0.05, 0.1, 0.15, 0.2\}$ and the batch size by $\{64, 128, 256\}$. For DP-SRM, we tune the batch size b by searching the grid $\{64, 128, 256\}$, step size by $\{0.01, 0.02, 0.05, 0.1, 0.15, 0.2\}$, and b_0 by $\{b, 2b, 4b\}$. In addition, we set the momentum parameter $\gamma = C_2$.

Results for CNN6. Figure 3 presents the average test error and the corresponding 95% confidence interval of different methods versus the number of iterations as well as the training time (in seconds) over 30 trials. The CNN6 trained by the non-private SGD will have 18.5% test error after 150 epochs. The results show that our proposed method can achieve 33.2% and 31.0% test errors given $\epsilon = 2.0$ and $\epsilon = 4.0$, which are comparable to the results of DP-SGD with 33.2% and 31.2% under the same privacy budgets. However, we can see from the plots that our method can significantly reduce the iteration numbers and the training time. For example, when $\epsilon = 4.0$, DP-SGD takes 1.3×10^4 iterations and 1115 seconds to achieve 31.2% test error. In sharp contrast, our method only takes 6.8×10^3 iterations and 643 seconds to achieve 31.0% test error. We can observe similar results for CNN5, which are presented in Section A in Appendix.

[3] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html.

8 CONCLUSIONS

We propose an efficient differentially private algorithm for nonconvex ERM. We prove both privacy and utility guarantees for our method. Both theoretical analyses and experiments demonstrate the advantage of our algorithms compared with the state-of-the-art. It would be very interesting to study our method's performances in super large or even industrial-level neural networks. It would also be very interesting to study the optimization lower bound for the differentially private nonconvex stochastic optimization problem.

ACKNOWLEDGEMENT

We thank the anonymous reviewers for their helpful comments. This work was partially supported by grants from the National Science Foundation (#1717950, #1804603 and #1915813). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, 2016.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning*, 2016.

Raman Arora, Raef Bassily, Tomás González, Cristóbal Guzmán, Michael Menart, and Enayat Ullah. Faster rates of convergence to stationary points in differentially private optimization. *arXiv preprint arXiv:2206.00846*, 2022.

Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Symposium on Foundations of Computer Science*, 2014.

Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Guha Thakurta. Private stochastic convex optimization with optimal rates. In *NeurIPS*, 2019.

Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with Gaussian Differential Privacy. *arXiv preprint arXiv:1911.11607*, 2019.

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and testing unintended memorization in neural networks. In *USENIX Security Symposium*, 2019.

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

Ashok Cutkosky and Francesco Orabona. Momentum-based variance reduction in non-convex sgd. In *NeurIPS*, 2019.

Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian Differential Privacy. *arXiv preprint arXiv:1905.02383*, 2019.

Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Now Publishers, Inc., 2014.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, 2006.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *NeurIPS*, 2018.

Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in linear time. *arXiv preprint arXiv:2005.04763*, 2020.

Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in Pharmacogenetics: An end-to-end case study of personalized Warfarin dosing. In *USENIX Security Symposium*, 2014.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *IEEE Symposium on Security and Privacy*, 2019.

Bargav Jayaraman, Lingxiao Wang, David Evans, and Quanquan Gu. Distributed learning without distress: Privacy-preserving Empirical Risk Minimization. In *NeurIPS*, 2018.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, 2013.

Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, 2012.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

Jaewoo Lee and Daniel Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *NeurIPS*, 2017.

H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

Ilya Mironov. Rényi Differential Privacy. In *IEEE Computer Security Foundations Symposium*, 2017.

Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled Gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Sarah: A novel method for machine learning problems using stochastic recursive gradient. In *34th International Conference on Machine Learning*, 2017.

Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2016.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, 2016a.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for smooth nonconvex optimization. In *IEEE Conference on Decision and Control*, 2016b.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, 2017.

Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *NeurIPS*, 2017.

Di Wang, Changyou Chen, and Jinhui Xu. Differentially private empirical risk minimization with non-convex loss functions. In *International Conference on Machine Learning*, 2019a.

Lingxiao Wang and Quanquan Gu. Differentially private iterative gradient hard thresholding for sparse learning. In *International Joint Conference on Artificial Intelligence*, 2019.

Lingxiao Wang and Quanquan Gu. A knowledge transfer framework for differentially private sparse learning. *AAAI*, 2020.

Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled Renyi differential privacy and analytical moments accountant. In *International Conference on Artificial Intelligence and Statistics*, 2019b.

Zhe Wang, Kaiyi Ji, Yi Zhou, Yingbin Liang, and Vahid Tarokh. Spiderboost and momentum: Faster variance reduction algorithms. *Advances in Neural Information Processing Systems*, 32, 2019c.

Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *ACM International Conference on Management of Data*, 2017.

Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.

Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private ERM for smooth objectives. In *International Joint Conference on Artificial Intelligence*, 2017.

Dongruo Zhou, Pan Xu, and Quanquan Gu. Stochastic nested variance reduction for nonconvex optimization. In *NeurIPS*, 2018.

Yuqing Zhu and Yu-Xiang Wang. Poission subsampled Rényi differential privacy. In *International Conference on Machine Learning*, 2019.