ADAPTIVE DIFFERENTIALLY PRIVATE EMPIRICAL RISK MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose an adaptive (stochastic) gradient perturbation method for differentially private empirical risk minimization. At each iteration, the random noise added to the gradient is optimally adapted to the stepsize; we name this process adaptive differentially private (ADP) learning. Given the same privacy budget, we prove that the ADP method considerably improves the utility guarantee compared to the standard differentially private method in which vanilla random noise is added. Our method is particularly useful for gradient-based algorithms with non-constant learning rate, including variants of AdaGrad (Duchi et al., 2011). We provide extensive numerical experiments to demonstrate the effectiveness of the proposed adaptive differentially private algorithm.

1 INTRODUCTION

Publishing deep neural networks such as ResNets (HZRS16) and Transformers (VSP+17) (with billions of parameters) trained on private datasets has become a major concern in the machine learning community; these models can memorize the private training data and can thus leak personal information, such as social security numbers (CTW+20). Moreover, these models are vulnerable to privacy attacks, such as membership inference (SSSS17; GSL+21) and reconstruction (FJR15; NHN+20). Therefore, over the past few years, a considerable number of methods have been proposed to address the privacy concerns described above. One main approach to preserve data privacy is to apply differentially private (DP) algorithms (DKM+06; DR+14; ACG+16; JWK+20) to train these models on private datasets. Differentially private stochastic gradient descent (DP-SGD) is a common privacy-preserving algorithm used for training a model via gradient-based optimization; DP-SGD adds random noise to the gradients during the optimization process (BST14; SCS13; BFGT20).

To be concrete, consider the empirical risk minimization (ERM) on a dataset $\mathcal{D} = \{x_i\}_{i=1}^n$, where each data point $x_i \in \mathcal{X}$. We aim to obtain a private high dimensional parameter $\theta \in \mathcal{R}^d$ by solving

$$\min_{\theta \in \mathbb{R}^d} F(\theta) := \frac{1}{n} \sum_{i=1}^{n} f(\theta; x_i), \tag{1}$$

where the loss function $f(\cdot) : \mathbb{R}^d \times \mathcal{X} \to \mathbb{R}$ is non-convex and smooth at each data point. To measure the performance of gradient-based algorithms for ERM, which enjoys privacy guarantees, we define the *utility* by using the expected ℓ_2-norm of gradient, i.e., $\mathbb{E}[\|\nabla F(\theta)\|]$, where the expectation is taken over the randomness of the algorithm (WYX17; ZZMW17; WJEG19; ZCH+20).[1] The DP-SGD with a Gaussian mechanism solves ERM in (1) by performing the following update at the t-th iteration, for $t \geq 0$ and $\theta_0 \in \mathbb{R}^d$

$$\text{(DP-SGD)} \quad \theta_{t+1} = \theta_t - \eta_t g_t \text{ with the released gradient } g_t = \nabla f(\theta_t; x_{\xi_t}) + Z, \tag{2}$$

where $Z \sim \mathcal{N}(0, \sigma^2 I)$, $\xi_t \sim \text{Uniform}(\{1, 2, \ldots, n\})$, and $\eta_t > 0$ is the stepsize or learning rate. Choosing the appropriate stepsize η_t is challenging in practice, as η_t depends on the unknown Lipschitz parameter of the gradient $\nabla f(\theta; x_i)$ (GL13). Recent popular techniques for tuning η_t include adaptive gradient methods (DHS11) and decaying stepsize schedules (GDG+17). When applying

[1] We examine convergence through the lens of utility guarantees; one may interchangeably use the two words "utility" or "convergence".

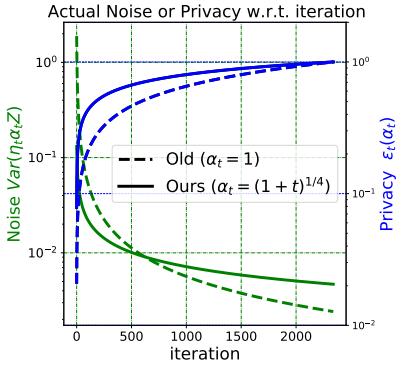

Figure 1: **Comparison between $\alpha_t = 1$ and $\alpha_t = 1/\sqrt{\eta_t}$ in (3).** Set the stepsize $\eta_t = 1/\sqrt{1+t}$ and the same privacy budget at final iteration. The blue curves corresponding to the right vertical axis show the overall privacy for $\alpha_t = 1$ (DP-SGD), represented by the dashed line, and $\alpha_t = 1/\sqrt{\eta_t}$ (ADP-SGD), represented by the solid line. The green curves corresponding to the left vertical axis show the actual Gaussian noise (i.e., $\eta_t\alpha_t Z$) added to the parameter θ_t for $\alpha_t = 1$ (dash line) and $\alpha_t = 1/\sqrt{\eta_t}$ (solid line). The variance of the perturbation using our proposed ADP-SGD decreases much slower than that using DP-SGD. Note that the privacy value we used here is based on the theoretical upper bound.

non-constant stepsizes, most of the existing differentially private algorithms directly follow the standard DP-SGD strategy by adding a simple perturbation (i.e, $Z \sim \mathcal{N}(0, \sigma^2 I)$) to each gradient over the entire sequence of iterations (ZCH+20). This results in a uniformly-distributed privacy budget for each iteration (BST14).

Several theoretical, as well as experimental results, corroborate the validity of the DP-SGD method with a uniformly-distributed privacy budget (BDLS20; ZKY+20; ZCH+20). Indeed, using a constant perturbation intuitively makes sense after noticing that the update in (2) is equivalent to $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t; x_{\xi_t}) - \eta_t Z$. This implies that the size of the true perturbation (i.e., $\eta_t Z$) added to the updated parameters is controlled by η_t. The decaying learning rate η_t thus diminishes the true perturbation added to θ_t. Although the DP-SGD algorithm with decaying noise $\eta_t Z$ is reasonable, it remains to be seen whether or not it is the optimal strategy using the utility measure.

To study the above question, we propose adding a hyperparameter $\alpha_t > 0$ to the private mechanism:

$$\text{(ADP-SGD)} \quad \theta_{t+1} = \theta_t - \eta_t g_t \text{ with the released gradient } g_t = \nabla f(\theta_t; x_{\xi_t}) + \eta_t \alpha_t Z. \tag{3}$$

The role of the hyperparameter α_t is to adjust the variance of the added random noise given the stepsize η_t. It is thus natural to add "adaptive" in front of the name DP-SGD and call our proposed algorithm ADP-SGD. To establish the privacy and utility guarantees of this new method, we first extend the advanced composition theorem (DR+14) so that it treats the case of a non-uniformly distributed privacy budget. We then show that our method achieves an improved utility guarantee when choosing $\alpha_t = 1/\sqrt{\eta_t}$, compared to the standard method using uniformly-distributed privacy budget, which corresponds to $\alpha_t = 1$.

This relationship between α_t and η_t is surprising. Given the same privacy budget and the decaying stepsize $\eta_t < 1$, the best choice – $\alpha_t = 1/\sqrt{\eta_t}$ – results in $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t; x_{\xi_t}) - \sqrt{\eta_t} Z$. This implies that the actual Gaussian noise $\sqrt{\eta_t} Z$ of ADP-SGD decreases *more slowly* than that of the conventional DP-SGD (i.e., $\eta_t Z$). To some extent, this is counter-intuitive in terms of convergence: one may anticipate that a more accurate gradient or smaller perturbation will be necessary as the parameter θ_t reaches a stationary point (i.e., as $\|\nabla F(\theta_t)\| \to 0$) (LK18). See Figure 1 for an illustration. We will explain this interesting finding in Section 4.

Contribution. We summarize our contributions below:

- We propose an adaptive (stochastic) gradient perturbation method – "Adaptive Differentially Private Stochastic Gradient Descent" (ADP-SGD) (Algorithm 1 or (3)) – and show how it can be used to perform differentially private empirical risk minimization. We show that APD-SGD provides a solution to the core question of this paper: *given the same overall privacy budget and iteration complexity, how should we select the gradient perturbation adaptively - across the entire SGD optimization process - to achieve better utility guarantees?* To answer this, we establish the privacy guarantee of ADP-SGD (Theorem 4.1) and find that the best choice of α_t follows an interesting dynamics: $\alpha_t = 1/\sqrt{\eta_t}$ (Theorem 4.2). Compared to the conventional DP-SGD, ADP-SGD with $\alpha_t = 1/\sqrt{\eta_t}$ results in a better utility given the same privacy budget.
- As the ADP-SGD method can be applied using any generic η_t, we discuss the two widely-used stepsize schedules: (1) the polynomially decaying stepsize of the form $\eta_t = 1/\sqrt{1+t}$, and (2) η_t updated by the gradients (DHS11). When using $\eta_t = 1/\sqrt{1+t}$, given the same privacy budgets ε, we obtain a stochastic sequence $\{\theta_t^{\text{ADP}}\}$ for ADP-SGD with $\alpha_t = 1/\sqrt{\eta_t}$, and $\{\theta_t^{\text{DP}}\}$ for

standard DP-SGD. We have the utility guarantees of the two methods, respectively[2]

$$\mathbb{E}[\|\nabla F(\theta_\tau^{\mathrm{ADP}})\|^2] = \widetilde{\mathcal{O}}\left(\frac{\log(T)}{\sqrt{T}} + \frac{d\sqrt{T}}{n^2\varepsilon^2}\right); \quad \mathbb{E}[\|\nabla F(\theta_\tau^{\mathrm{DP}})\|^2] = \widetilde{\mathcal{O}}\left(\frac{\log(T)}{\sqrt{T}} + \frac{d\log(T)\sqrt{T}}{n^2\varepsilon^2}\right),$$

where $\tau := \arg\min_{k\in[T-1]} \mathbb{E}[\|\nabla F(\theta_k)\|^2]$. Compared to the standard DP-SGD, ADP-SGD with $\alpha_t = 1/\sqrt{\eta_t}$ improves the bound by a factor of $\mathcal{O}(\log(T))$ when T and d are large (i.e. high-dimensional settings). When η_t is updated by the gradients (DHS11), the same result holds. See Section 5 for the detailed discussion.

- Finally, we perform numerical experiments to systematically compare the two algorithms: ADP-SGD ($\alpha_t = 1/\sqrt{\eta_t}$) and DP-SGD. In particular, we verify that ADP-SGD with $\alpha_t = 1/\sqrt{\eta_t}$ consistently outperforms DP-SGD when d and T are large. Based on these theoretical bounds and supporting numerical evidence, we believe ADP-SGD has important advantages over past work on differentially private empirical risk minimization.

Notation. In the paper, $[N] := \{0, 1, 2, \ldots, N\}$ and $\{\cdot\} := \{\cdot\}_{t=1}^T$. We write $\|\cdot\|$ for the ℓ_2-norm. F^* is a global minimum of F assuming $F^* > 0$. In addition, we use $D_F := F(\theta_0) - F^*$ and set stepsize $\eta_t = \eta/b_t$ and denote the d-dimensional identity matrix by I_d.

2 Preliminaries

We first make the following assumptions for the objective loss function in (1).

Assumption 2.1. *Each component function $f(\cdot)$ in (1) has L-Lipschitz gradient, i.e.,*

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall x, y \in \mathbb{R}^d. \tag{4}$$

Assumption 2.2. *Each component function $f(\cdot)$ in (1) has bounded gradient, i.e.,*

$$\|\nabla f(x)\| \le G, \quad \forall x \in \mathbb{R}^d. \tag{5}$$

The bounded gradient assumption is a common assumption for analysis of DP-SGD algorithms (WYX17; ZCH+20; ZKY+20) and also common for general adaptive gradient methods such as Adam (RCZ+21; CLSH18; RKK18). One recent popular approach to relax this assumption is through gradient clipping method (CWH20; ATMR19; PSY+19) that we will discuss more in Section 6 as well as in Appendix A. Nevertheless, this assumption would serve a good starting to analyze our proposed method. Next, we introduce differential privacy (DMNS06).

Definition 2.1 (**(ε, δ)-DP**). *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ is (ε, δ)-differentially private if for any two adjacent datasets $\mathcal{D}, \mathcal{D}'$ differing in one sample, and for any subset of outputs $S \subseteq \mathcal{R}$, we have*

$$Pr[\mathcal{M}(D) \in S] \le e^\varepsilon Pr[\mathcal{M}(D') \in S] + \delta.$$

Lemma 2.1 (**Gaussian Mechanism**). *For a given function $h : \mathcal{D} \to \mathbb{R}^d$, the Gaussian mechanism $\mathcal{M}(\mathcal{D}) = h(\mathcal{D}) + Z$ with $Z \sim \mathcal{N}(0, \sigma^2 I_d)$ satisfies $(\sqrt{2\log(1.25/\delta)}\Delta/\sigma, \delta)$-DP, where $\Delta = \sup_{\mathcal{D},\mathcal{D}'} \|h(\mathcal{D}) - h(\mathcal{D}')\|$, $\mathcal{D}, \mathcal{D}'$ are two adjacent datasets, and $\varepsilon, \delta > 0$.*

To achieve differential privacy, we can use the above Gaussian mechanism (DR+14). In our paper, we consider iterative differentially private algorithms, which prompts us to use privacy composition results to establish the algorithms' privacy guarantees after the completion of the final iteration. To this end, we extend the advanced composition theorem (DR+14) to the case in which each mechanism $\mathcal{M}_i$ has its own specific ε_i and δ_i parameters.

Lemma 2.2 (**Extended Advanced Composition**). *Consider two sequences $\{\varepsilon_i\}_{i=1}^k, \{\delta_i\}_{i=1}^k$ of positive numbers satisfying $\varepsilon_i \in (0, 1)$ and $\delta_i \in (0, 1)$. Let $\mathcal{M}_i$ be $(\varepsilon_i, \delta_i)$-differentially private for all $i \in \{1, 2, \ldots, k\}$. Then $\mathcal{M} = (\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is $(\tilde{\varepsilon}, \tilde{\delta})$-differentially private for $\delta' \in (0, 1)$ and*

$$\tilde{\varepsilon} = \sqrt{\sum_{i=1}^k 2\varepsilon_i^2 \log\left(\frac{1}{\delta'}\right)} + \sum_{i=1}^k \frac{\varepsilon_i(e^{\varepsilon_i} - 1)}{(e^{\varepsilon_i} + 1)}, \qquad \tilde{\delta} = 1 - (1 - \delta_1)(1 - \delta_2)\ldots(1 - \delta_k) + \delta'.$$

When $\varepsilon_i = \varepsilon_0$ and $\delta_i = \delta_0$ for all i, Lemma 2.2 reduces to the classical advanced composition theorem (DR+14) restated in Lemma A.2 in the Appendix.

[2]This is an informal statement of Proposition 5.1; the order $\widetilde{\mathcal{O}}$ hides $\log(1/\delta)$, LG^2 and $F(\theta_0) - F^*$ terms. We keep the iteration number T in our results since the theoretical best value of T depends on some unknown parameters such as the Lipschitz parameter of the gradient, which we try to tackle using non-constant stepsizs.

3 AN ADAPTIVE DIFFERENTIALLY PRIVATE ALGORITHM

In this section, we present our proposed algorithm: *adaptive differentially private stochastic gradient descent* (ADP-SGD, Algorithm 1). The "adaptive" part of the algorithm is tightly connected with the choice of the hyper-parameter α_t (see line 5 of Algorithm 1). For $\alpha_t = 1$, ADP-SGD reduces to DP-SGD. As mentioned before, we aim to investigate whether an uneven allocation of the privacy budget for each iteration (via ADP-SGD) will provide a better utility guarantee than the default DP-SGD given the same privacy budget. To achieve this, our proposed ADP-SGD with hyper-parameter α_t adjusts the privacy budget consumed at the t-th iteration according to the current learning rate η/b_{t+1} (see line 6 of Algorithm 1). Moreover, we will update α_t dynamically (see line 5 of Algorithm 1) and show how to choose α_t in Section 4. Before proceeding to analyze Algorithm 1, we state Definition 3.1 to clearly explain the adaptive privacy mechanism for the algorithm.

Definition 3.1 (Adaptive Gaussian Mechanism). *At iteration t in Algorithm 1, the privacy mechanism $\mathcal{M}_t : \mathbb{R}^d \to \mathbb{R}^d$ is:*

$$\mathcal{M}_t(X) = \nabla f(\theta_t; x_{\xi_t}) + \alpha_{t+1} c_t.$$

The hyper-parameter α_{t+1} is adaptive *to the DP-SGD algorithm – specifically, to the stepsize $\eta_{t+1} := \eta/b_{t+1}$.*

Algorithm 1 ADP-SGD (Reduces to **DP-SGD** if $\alpha_t = 1$)

1: Input: θ_0, b_0, α_0 and $\eta > 0$
2: **for** $t = 0, 1, \ldots, T-1$ **do**
3: sample $\xi_t \sim$ Uniform$(1, ..., n)$ and $c_t \sim \mathcal{N}(0, \sigma^2 I)$
4: update $b_{t+1} = \phi_1(b_t, \nabla f(\theta_t; x_{\xi_t}))$
5: update $\alpha_{t+1} = \phi_2(\alpha_t, b_{t+1})$
6: **release** $g_t^b = \frac{\eta}{b_{t+1}}(\nabla f(\theta_t; x_{\xi_t}) + \alpha_{t+1} c_t)$
7: update $\theta_{t+1} = \theta_t - g_t^b$
8: **end for**

Algorithm 1 is a general framework that can cover many variants of stepsize update schedules, including the adaptive gradient algorithms (DHS11; KB14). In particular, we use functions $\phi_1 : \mathbb{R}^2 \to \mathbb{R}$ and $\phi_2 : \mathbb{R}^2 \to \mathbb{R}$ to denote the updating rules for parameters b_t and α_t, respectively. For example, when ϕ_1 is $1/\sqrt{a + ct}$, ϕ_2 is the constant 1 for all t and $a, c > 0$, ADP-SGD reduces to DP-SGD with polynomial decaying stepsizes (BST14). When ϕ_1 is $b_{t+1} = \sqrt{b_t^2 + \|\nabla f(\theta_t; x_{\xi_t})\|^2}$ and ϕ_2 is the constant 1, the algorithm reduces to DP-SGD with a variant of adaptive stepsizes (DHS11). In particular, if we choose ϕ_2 to be 0, the algorithm reduces to the standard SGD.

Similar to classical works studying the convergence of the SGD algorithm (BFGT20; BCN18; WWB19), we will use Assumption 3.1 in addition to Assumption 2.2 and Assumption 2.1.

Assumption 3.1. *$\nabla f(\theta_t; x_{\xi_t})$ is an unbiased estimator of $\nabla F(\theta_k)$. The random indices ξ_t, $t = 0, 1, 2, \ldots,$ are independent of each other and also independent of θ_t and $c_1, \ldots, c_{t-1}$.*

Having defined the ADP-SGD algorithm and established our assumptions, in what follows, we will be answering the paper's central question: *Given the same privacy budget ε, how should one design the gradient perturbation parameters α_t adaptively for each iteration t to achieve a better utility guarantee?* Solving this question is of paramount importance as one can only run these algorithms for a finite number of iterations, therefore, given these constraints, a clear and efficient strategy for improving the constants of the utility bound is necessary.

4 THEORETICAL RESULTS FOR ADP-SGD

In this section, we provide the main results for our method – the privacy and utility guarantees.

Theorem 4.1 (Privacy Guarantee). *Suppose the sequence $\{\alpha_t\}_{t=1}^T$ is known in advance and that Assumption 2.2 holds. Algorithm 1 satisfies (ε, δ)-DP if the random noise c_t has variance*

$$\sigma^2 = \frac{(16G)^2 B_\delta}{n^2 \varepsilon^2} \sum_{t=0}^{T-1} \frac{1}{\alpha_{t+1}^2} \quad \text{with} \quad B_\delta = \log(16T/n\delta)) \log(1.25/\delta). \tag{6}$$

The theorem is proved by using Lemma 2.2 and Definition 3.1 (see Appendix C.1 for details). Note that the term B_δ could be improved by using the moments accountant method (MTZ19), to $\mathcal{O}(\log(1.25/\delta))$ independent of T but with some additional constraints (ACG+16). We keep this format of B_δ as in (6) in order to compare directly with (BST14). Theorem 4.1 shows that σ^2 must scale with $\sum_{t=1}^T 1/\alpha_t^2$. When the complexity T increases, the variance σ^2, regarded as a function of T, could be either large or small, depending on the sequence $\{\alpha_t\}$. If α_t is monotone with rate $\alpha_t^2 \propto t^p$, where $p \in [0, 1]$, then $\sigma^2 \propto T^{1-p}$ for $0 < p < 1$; $\sigma^2 \propto \log(T)$ for $p = 1$; and $\sigma^2 \propto T$ for

$p = 0$ (the default DP-SGD). From a convergence view, $\theta_{t+1} = \theta_t - \eta_t \nabla f(\theta_t; x_{\xi_t}) - \eta_t \alpha_t Z$ implies that the actual Gaussian noise added to the updated parameter θ_t has variance $\eta_t^2 \alpha_t^2 \sigma^2$. Therefore, it is subtle to determine what p would be the best choice for ensuring convergence. In Theorem 4.2, we will see that the optimal choice of the sequence $\{\alpha_t\}_{t=1}^T$ is closely related to the stepsize.

Theorem 4.2 (**Convergence for ADP-SGD**). *Suppose we choose σ^2 - the variance of the random noise in Algorithm 1 - according to (6) in Theorem 4.1 and that Assumption 2.1, 2.2 and 3.1 hold. Furthermore, suppose α_t, b_t are deterministic. The utility guarantee of Algorithm 1 with $\tau \triangleq arg\min_{k\in[T-1]} \mathbb{E}[\|\nabla F(\theta_k)\|^2]$ and $B_\delta = \log(16T/n\delta))\log(1.25/\delta)$ is*

$$\mathbb{E}\|\nabla F(\theta_\tau)\|^2 \leq \frac{1}{\sum_{t=0}^{T-1} \frac{1}{b_{t+1}}} \left(\frac{D_F}{\eta} + \frac{\eta L}{2} \sum_{t=0}^{T-1} \frac{\mathbb{E}\left[\|\nabla f(\theta_t, \xi_t)\|^2\right]}{b_{t+1}^2} + \frac{d(16G)^2 B_\delta}{2n^2\varepsilon^2} M(\{\alpha_t\}, \{b_t\}) \right) \quad (7)$$

where $M(\{\alpha_t\}, \{b_t\}) \triangleq \sum_{t=1}^T (\alpha_t/b_t)^2 \sum_{t=1}^T 1/\alpha_t^2$.

Although the theorem assumes independence between b_{t+1} and the stochastic gradient $\nabla f(\theta_t; x_{\xi_t})$, we shall see in Section 5.2 that a similar bound holds for correlated b_t and $\nabla f(\theta_t; x_{\xi_t})$.

Remark 4.1 (**An optimal relationship between α_t and b_t**). *According to (7), the utility guarantee of Algorithm 1 consists of three terms. The first two terms correspond to the optimization error and the last term is introduced by the privacy mechanism, which is also the dominating term. Note that if we fix $\{b_t\}$ and minimize M with respect to $\{\alpha_t\}$, the minimal value denoted by M_{adp} express as*

$$\min_{\{\alpha_t\}} M(\{\alpha_t\}, \{b_t\}) = M_{\text{adp}} \triangleq \Big(\sum_{t=0}^{T-1} 1/b_{t+1} \Big)^2. \quad (8)$$

Furthermore, $M(\{\alpha_t\}, \{b_t\}) = M_{\text{adp}}$ if $\alpha_t^2 = b_t$. Therefore, if we choose α_t, b_t such that the relationship of $\alpha_t^2 = b_t$ holds, we can achieve the minimum utility guarantee for Algorithm 1.

Suppose the third term in (7) is much larger than the first two terms. We can compare the utility bound for some arbitrary setting of $\{\alpha_t\}$ with the utility bound associated with the optimal setting $\alpha_t^2 = b_t$ by examining the ratio $M(\{\alpha_t\}, \{b_t\})/M_{\text{adp}}$; a large value of this ratio implies a significant reduction in the utility bound is achieved by using Algorithm 1 with $\alpha_t = \sqrt{b_t}$. For example, for the standard DP-SGD method, the function M reduces to $M_{\text{dp}} \triangleq T \sum_{t=0}^{T-1} 1/b_{t+1}^2$. Thus, our proposed method - involving $\alpha_t = \sqrt{b_t}$ - admits a bound improved by a factor of

$$\frac{M_{\text{dp}}}{M_{\text{adp}}} = T \frac{\sum_{t=0}^{T-1} 1/b_{t+1}^2}{(\sum_{t=0}^{T-1} 1/b_{t+1})^2},$$

which is bounded below by one using the Cauchy-Schwarz inequality; thus, ADP-SGD is not worse than DP-SGD for any choice of $\{b_t\}$. In the following section, we will analyze this factor of $M_{\text{dp}}/M_{\text{adp}}$ for two widely-used stepsize schedules: (a) the polynomially decaying stepsize given by $\eta_t = 1/\sqrt{1+t}$; and (b) a variant of adaptive gradient methods (DHS11).

Note that, in addition to $\alpha_t^2 = b_t$, there are other relationships between the sequences $\{(\alpha_t/b_t)^2\}$ and $\{\alpha_t^2\}$ that could lead to the same M_{adp}. For instance, setting $\alpha_t \alpha_{T-(t-1)} = b_t$ is another possibility. Nevertheless, in this paper, we will focus on the $\alpha_t^2 = b_t$ relation, and leave the investigation of other appropriate choices to future work. We emphasize that the bound in Theorem 4.2 only assumes f to have Lipschitz smooth gradients and be bounded. Thus, the theorem applies to both convex or non-convex functions. Since our focus is on the improvement factor $M_{\text{dp}}/M_{\text{tadp}}$, we will assume our functions are non-convex, but the results will also hold for convex functions.

5 Examples for ADP-SGD

We now analyze the convergence bound given in Theorem 4.2 and obtain an explicit form for M in terms of T by setting the stepsize to be $1/b_{t+1} \propto 1/\sqrt{t}$, which is closely related to the polynomially decreasing rate of adaptive gradient methods (DHS11; WWB19) studied in Section 5.2.

Constant stepsize v.s. time-varying stepsize. If the constant step size is used, then there is no need to use the adaptive DP mechanism proposed in this paper as we verify that constant perturbation to the gradient is optimal in terms of convergence. However, as we have explained in introduction, to ease the difficulty of stepsize tuning, time varying stepsize is widely used in many practical applications of deep learning. We will discuss two examples below. In these cases, the standard DP mechanism (i.e. constant perturbation to the gradient) is not the most suitable technique, and our proposed adaptive DP mechanism can give better utility results.

Achieve $\log T$ improvement. We present Proposition 5.1 and Proposition 5.2 to show that our method achieve $\log(T)$ improvement over the vanilla DP-SGD. Although this $\log(T)$ improvement can also be achieved by using the moments accountant method (MAM) (MTZ19), *we emphasize that our proposed method is orthogonal and complementary to MAM* in that the $\log(T)$ improvement is over B_δ using MAM (See discussion after Theorem 4.1) while ours is during the optimization process depending on stepsizes. Nevertheless, since the two techniques are complementary to each other, we can apply them simultaneously, and achieve a $\log^2(T)$ improvement over DP-SGD using advanced composition for $O(1/\sqrt{t})$ stepsizes, compared to a $\log(T)$ improvement using either of them. Thus, an adaptive DP mechanism for algorithms with time-varying stepsizes is very useful.

5.1 Example 1: ADP-SGD with polynomially decaying stepsizes

The first case we consider is the stochastic gradient descent with polynomially decaying stepsizes. More specifically, we let $b_t = (a+ct)^{1/2}$, where $a > 0, c > 0$.

Proposition 5.1 (ADP-SGD v.s. DP-SGD with a polynomially decaying stepsize schedule). *Under the conditions of Theorem 4.2 on f and σ^2, let $b_t = (a+ct)^{1/2}$ with $a > 0, c > 0$ in Algorithm 1. Denote $\tau = \arg\min_{t\in[T-1]} \mathbb{E}[\|\nabla F(\theta_t)\|^2]$, and $B_\delta = \log(16T/n\delta))\log(1.25/\delta)$. If we choose $T \geq 5+4a/c$, we have the following utility guarantee for ADP-SGD ($\alpha_t^2 = b_t$) and DP-SGD ($\alpha_t^2 = 1$) respectively,*

$$\textbf{\textit{(ADP-SGD)}}\ \mathbb{E}[\|\nabla F(\theta_\tau^{\mathrm{ADP}})\|^2] \leq \frac{\sqrt{c}\left(\frac{D_F}{\eta} + \frac{\eta G^2 L B_T}{2c}\right)}{\sqrt{T-1}} + \frac{\eta d L (16G)^2 B_\delta \sqrt{T}}{2n^2\varepsilon^2\sqrt{c}}; \tag{9}$$

$$\textbf{\textit{(DP-SGD)}}\ \mathbb{E}[\|\nabla F(\theta_\tau^{\mathrm{DP}})\|^2] \leq \frac{\sqrt{c}\left(\frac{D_F}{\eta} + \frac{\eta G^2 L B_T}{2c}\right)}{\sqrt{T-1}} + \frac{\eta d L (16G)^2 B_\delta \sqrt{T}\log\left(1+T\frac{c}{a}\right)}{n^2\varepsilon^2\sqrt{c}}. \tag{10}$$

The proof of Proposition 5.1 is given in Appendix D.1 and Appendix D.2. Proposition 5.1 implies $M_{\mathrm{dp}}/M_{\mathrm{adp}} = \mathcal{O}(\log T)$ – that is, ADP-SGD has an improved utility guarantee compared to DP-SGD. Such an improvement can be significant when d is large and LG^2 is large.

5.2 Example 2: ADP-SGD with adaptive stepsizes

We now examine another choice of the term b_t, which relies on a variant of adaptive gradient methods (DHS11). To be precise, we assume b_t is updated according to the norm of the gradient, i.e., $b_{t+1}^2 = b_t^2 + \max\{\|\nabla f(\theta_t; x_{\xi_t})\|^2, \nu\}$, where $\nu > 0$ is a small value to prevent the extreme case in which $1/b_{t+1}$ goes to infinity. When $b_0^2 = \|\nabla f(\theta_t; x_{\xi_t})\|^2 \to 0$, then $\eta/b_1 \to \infty$. We choose this precise equation formula because it is simple and it also represents the core of adaptive gradient methods - updating the stepsize on-the-fly by the gradients (LYC18; WWB19). The conclusions for this variant may transfer to other versions of adaptive stepsizes, and we defer this to future work.

Observe that $b_t \propto t^{1/2}$ since $b_t^2 \in [b_0^2 + tv, b_0^2 + tG]$, which at a first glance indicates that the bound for this adaptive stepsize could be derived via a straightforward application of Proposition 5.1. However, since b_t is now a random variable correlated to the stochastic gradient $\nabla f(\theta_t; x_{\xi_t})$, we cannot directly apply Theorem 4.2 to study b_t. To tackle this, we adapt the proof technique from (WWB19) and obtain Theorem D.1, which we deferred to Appendix D.3 due to space limitation.

As we see, b_t is updated on the fly during the optimization process. Applying our propsoed method with $\alpha_t^2 = b_t$ for this adaptive stepsize is not possible since α_t has to be set beforehand according to Equation (6) in Theorem 4.1. To address this, we note $b_t^2 \in [b_0^2 + tv, b_0^2 + tG]$. Thus, we propose to set $\alpha_t^2 = \sqrt{b_0^2 + tC}$ for some $C \in [\nu, G^2]$ and obtain Proposition 5.2 based on Theorem D.1.

Proposition 5.2 (ADP v.s. DP with an adaptive stepsize schedule). *Under the same conditions of Theorem D.1 on f, σ^2, and b_t, if $\alpha_t = (b_0^2 + tC)^{1/4}$ for some $C \in [\nu, G^2]$, then*

$$\textbf{\textit{(ADP-SGD)}}\quad \mathbb{E}\|\nabla F(\theta_\tau^{\mathrm{ADP}})\|^2 \leq \frac{2GB_{\mathrm{sgd}}}{\sqrt{T-1}} + \frac{128G^3\eta d L B_\delta\sqrt{T}}{n^2\varepsilon^2\nu}. \tag{11}$$

$$\textbf{\textit{(DP-SGD)}}\quad \mathbb{E}\|\nabla F(\theta_\tau^{\mathrm{DP}})\|^2 \leq \frac{2GB_{\mathrm{sgd}}}{\sqrt{T-1}} + \frac{32G^3\eta d L B_\delta\sqrt{T}\log\left(1+T\frac{\nu}{b_0^2}\right)}{n^2\varepsilon^2\nu}. \tag{12}$$

See the proof in Appendix D.4. Similar to the comparison in Proposition 5.1, the key difference between two bounds in (11) and (12) is the last term; using ADP-SGD gives us a tighter utility guarantee than the one provided by DP-SGD by a factor of $\mathcal{O}(\log(T))$. This improvement is significant

when the dimension d is very high, or when either L, G, or T are sufficiently large. Note that the bound in Proposition 5.2 does not reflect the effect of the different choice of C, as the bound corresponds to the worse case scenarios. We will perform experiments testing a wide range of C values and this will allow us to thoroughly examine the properties of ADP-SGD for adaptive stepsizes.

6 EXPERIMENTS

In this section, we present numerical results to support the theoretical findings of our proposed methods. We perform two sets of experiments: (1) when $b_t = \sqrt{20+t}$, we compare ADP-SGD ($\alpha_t^2 = b_t$) with DP-SGD (setting $\alpha_t = 1$ in Algorithm 1); and (2) when b_t is updated by the norm of the gradients, we compare ADP-SGD ($\alpha_t^2 = \sqrt{b_0^2 + tC}$) with DP-SGD. The first set of experiments is meant to examine Proposition 5.1, while the second concerns Proposition 5.2. In addition to the experiments above, in the supplementary material (Appendix F.3), we present strong empirical evidence in support of the claim that using a decaying stepsize schedule yields better results than simply employing a constant stepsize schedule.

Datasets and models. We perform our experiments on CIFAR-10 (KH+09) and MNIST (LBBH98), using a convolution neural network (CNN) for the former and the logistic regression model for the latter. See our CNN design in the appendix. Notably, following previous work (ACG+16), the CNN model is pre-trained on CIFAR-100 and fined-tuned on CIFAR-10. The mini-batch size is 256, and each independent experiment runs on one GPU. We set $\eta = 1$ in Algorithm 1 (line 6) and use the gradient clipping with $C_G \in \{0.5, 1, 2.5, 5\}$ (CWH20; ATMR19; PSY+19). Note that one might need to think about C_G as being approximately closer to the bounded gradient parameter G. We provide a more detailed discussion in Appendix F.1. The privacy budget is set to be $\bar{\varepsilon} = \varepsilon / C_\varepsilon \in \{0.8, 1.2, 1.6, 3.2, 6.4\}$ and we choose $\delta = 10^{-5}$.[3] Given these privacy budgets, we calculate the corresponding variance using Theorem 4.1 (See Appendix G for the code to obtain σ).

6.1 ADP-SGD V.S. DP-SGD WITH POLYNOMIALLY DECAYING STEPSIZES

We focus on understanding the optimality of the theoretical guarantees of Proposition 5.1; the experiments help us further understand how this optimality reflects in generalization. We consider training with $T = 11700, 23400, 39000$ iterations corresponding to $60, 120, 200$ training epochs, which represents the practical scenarios of inadequately limited, considerably standard and sufficiently large time budgets. We repeat the experiments five times, and report the average test accuracy and standard deviation in Table 1 for gradient clipping values $C_G \in \{1, 2.5\}$. We include plots in Figure 2 to provide detailed comparisons between ADP-SGD and DP-SGD. In Appendix F.1 of the supplemental material, we present additional experiments for $C_G \in \{0.5, 5\}$. In addition to the learning rate $\eta_t = 1/\sqrt{20+t}$, we also consider in Appendix F.1 an alternative decaying schedule.

The results in Table 1 and Figure 2 show that the overall performance of our method (ADP-SGD) is better than DP-SGD given a fixed privacy budget and the same complexity T. Particularly, the increasing T tends to enlarge the gap between ADP-SGD and DP-SGD, especially for smaller privacy; for $\bar{\varepsilon} = 0.8$ with $C_G = 1$, we have improvements of 0.8% at epoch 60, 1.48% at epoch 120, and 7.03% at epoch 200. This result is reasonable since, as explained in Proposition 5.1, ADP-SGD improves over DP-SGD by a factor $\log(T)$.

Furthermore, our method is more robust to the predefined complexity T and thus provides an advantage when using longer iterations. For example, for $\bar{\varepsilon} = 3.2$ with $C_G = 2.5$, our method increases from 65.34% to 66.41% accuracy when the iteration complexity of 60 epochs is doubled; it maintains the accuracy 65.74% at the longer epoch 200. In contrast, under the same privacy budget and gradient clipping, DP-SGD suffers the degradation from 66.08% (epoch 60) to 65.17% (epoch 200).

6.2 ADP-SGD V.S. DP-SGD WITH ADAPTIVE STEPSIZES

In this section, we focus on understanding the optimality of the theoretical guarantees of Proposition 5.2; we study the numerical performance of ADP-SGD with stepsizes updated by the gradients. We notice that, at the beginning of the training, the gradient norm in our model lies between

[3] The constant $C_\varepsilon = 16$ in (6). Although $\varepsilon = 16\bar{\varepsilon}$ is large for $\bar{\varepsilon} \in \{0.8, 1.2, 1.6, 3.2, 6.4\}$, they match the numerical privacy $\{0.29, 0.43, 0.57, 1.23, 3.24\}$ calculated by the moments accountant with the noise determined by $T = 11700$ (60 epochs) and the gradient clipping $C_G = 1.0$.

$\bar{\varepsilon}$	Alg	Gradient clipping $C_G = 1.0$ epoch=60	epoch= 120	epoch= 200	Gradient clipping $C_G = 2.5$ epoch= 60	epoch= 120	epoch= 200
0.8	ADP	$\mathbf{56.38} \pm 0.092$	54.20 ± 0.730	51.71 ± 1.092	$\mathbf{48.61} \pm 1.003$	44.11 ± 1.097	39.92 ± 0.284
	DP	$\mathbf{56.13} \pm 0.909$	52.72 ± 0.938	44.68 ± 0.576	$\mathbf{38.06} \pm 1.029$	23.64 ± 0.796	17.75 ± 1.068
	Gap	0.25	1.48	7.03	10.55	20.47	22.17
1.2	ADP	$\mathbf{60.26} \pm 0.319$	60.24 ± 0.365	58.68 ± 0.505	$\mathbf{56.63} \pm 0.308$	52.26 ± 0.328	50.7 ± 1.038
	DP	$\mathbf{60.09} \pm 0.450$	60.02 ± 0.204	57.56 ± 0.514	$\mathbf{55.71} \pm 0.418$	43.16 ± 0.604	32.00 ± 2.281
	Gap	0.17	0.22	1.12	0.92	9.1	18.7
1.6	ADP	61.30 ± 0.219	$\mathbf{61.98} \pm 0.420$	61.88 ± 0.507	$\mathbf{61.52} \pm 0.313$	58.60 ± 0.352	56.07 ± 0.046
	DP	61.18 ± 0.195	$\mathbf{61.89} \pm 0.317$	61.46 ± 0.490	$\mathbf{61.76} \pm 0.454$	55.68 ± 0.243	46.74 ± 0.428
	Gap	0.12	0.09	0.42	-0.24	2.92	9.33
3.2	ADP	61.76 ± 0.490	64.27 ± 0.257	$\mathbf{65.54} \pm 0.066$	65.34 ± 0.12	$\mathbf{66.41} \pm 0.054$	65.74 ± 0.106
	DP	62.02 ± 0.248	63.88 ± 0.275	$\mathbf{65.11} \pm 0.359$	$\mathbf{66.08} \pm 0.130$	65.73 ± 0.353	65.17 ± 0.115
	Gap	-0.26	0.39	0.43	-0.44	0.68	0.57
6.4	ADP	62.2 ± 0.270	64.57 ± 0.515	$\mathbf{65.74} \pm 0.270$	67.35 ± 0.057	68.72 ± 0.045	$\mathbf{69.51} \pm 0.179$
	DP	62.06 ± 0.244	64.61 ± 0.180	$\mathbf{65.84} \pm 0.206$	67.06 ± 0.244	68.46 ± 0.321	$\mathbf{69.28} \pm 0.147$
	Gap	0.14	-0.04	-0.1	0.29	0.26	0.23

Table 1: **Mean accuracy of ADP-SGD/DP-SGD with polynomially decaying stepsizes.** This table reports *Accuracy* with the mean and the corresponding standard deviation over five independent runs given a pair of $(\bar{\varepsilon}, C_G, T, \text{Alg})$. The difference ("Gap") between DP and ADP is provided for visualization purpose. The results suggest that the more iterations or epochs we use, the more improvements ADP-SGD can potentially gain over DP-SGD. The results are reported in percentage (%). The highlight number is the best accuracy in a row among epoch 60, 120 and 200 for the same gradient clipping C_G.

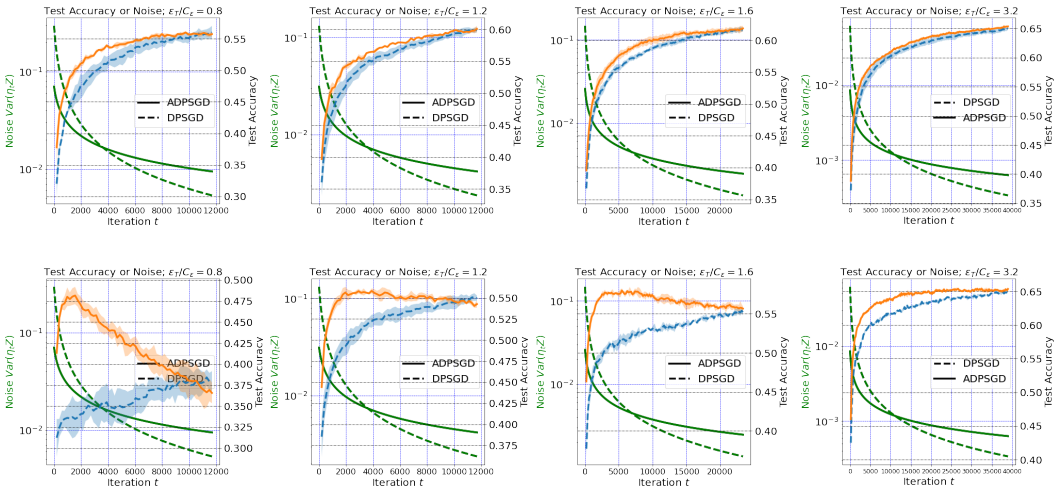

Figure 2: **Detailed performance ADP-SGD/DP-SGD with polynomially decaying stepsizes.** The top row is for gradient clipping $C_G = 1.0$ and bottom for $C_G = 2.5$. Each plot corresponds to a fixed T (see x-axis) and a privacy budget ε (see title). The solid orange and light-blue curves, which correspond to the right vertical y-axis, show the averaged test accuracy for ADP-SGD (solid line) and DP-SGD (dash line). The shaded region is one standard deviation. Same as Figure 1, the monotone green curves, which correspond to the left vertical y-axis, show the actual noise for $\alpha_t = 1/\sqrt{\eta_t}$ (ADP-SGD, the solid line) and $\alpha_t = 1$ (DP-SGD, the dashed line). The top/bottom rows from 1st to 4th column correspond to the privacy budgets from 0.8 (epoch 60), 1.2 (epoch 60), 1.6 (epoch 120), and 3.2 (epoch 200).

0.0001 and 0.001 when $C_G = 1.0$. To remedy this small gradient issue, we let b_t follow a more general form: $b_{t+1}^2 = b_t^2 + \max\{\beta_t \|\nabla f(\theta_t; x_{\xi_t})\|^2, 10^{-5}\}$ with $\beta_t > 1$. Specifically, we set $\beta_t = \max\{\beta/((t \mod 195) + 1)), 1\}$ with β searching in a set $\{1, 512, 1024, 2048, 4096, 8192\}$.[4] See Appendix F.2 for a detailed description. As mentioned in Section 5.2, we set $\alpha_t^2 = \sqrt{b_0^2 + tC}$ in advance with $b_0^2 = 20$, and choose $C \in \{10^{-5}, 10^{-4}, 0.001, 0.01, 0.1, 1\}$. We consider the number of iterations to be $T = 11700$ with the gradient clipping 1.0 and 2.5. Table 2 summarizes the results of DP-SGD and ADP-SGD with the best hyper-parameters averaged over five experimental trials.

[4]This set for β is due to the values of gradient norm as mentioned in the main text. These elements cover a wide range of values that the best test errors are doing as good as or better than the ones given in Table 1.

C_G	Alg	$\bar{\varepsilon} = 0.8$	$\bar{\varepsilon} = 1.6$	$\bar{\varepsilon} = 3.2$	$\bar{\varepsilon} = 6.4$
1.0	ADP	56.68 ± 0.646 (57.65)	62.09 ± 0.346 (62.57)	64.51 ± 0.100 (64.61)	67.75 ± 0.171 (67.91)
	DP	56.24 ± 0.535 (57.02)	62.02 ± 0.264 (62.33)	64.33 ± 0.329 (65.03)	67.42 ± 0.141 (67.7)
2.5	ADP	56.27 ± 0.174 (56.46)	62.38 ± 0.428 (62.86)	64.29 ± 0.408 (64.85)	67.55 ± 0.156 (67.77)
	DP	55.65 ± 0.448 (55.98)	62.23 ± 0.238 (62.62)	64.26 ± 0.140 (64.39)	66.23 ± 0.367 (66.62)

Table 2: **Errors of ADP-SGD vs. DP-SGD with adaptive stepsizes.** This table reports *accuracy* with the mean and the corresponding standard deviation over five independent runs. The value inside the bracket is the highest accuracy over the five runs. Each entry is the best value over 36 pairs of (β, C) for ADP-SGD and 6 values of β for DP-SGD. See the corresponding (β, C) in Table 4. The results indicate that when using adaptive stepsizes, ADP-SGD with various C performs better than DP-SGD.

7 Related work

Differentially private empirical risk minimization. Differentially Private Empirical Risk Minimization (DP-ERM) has been widely studied over the past decade. Many algorithms have been proposed to solve DP-ERM including objective perturbation (CMS11; KST12; INS$^+$19), output perturbation (WLK$^+$17; ZZMW17), and gradient perturbation (BST14; WYX17; JW18). While most of them focus on convex functions, we study DP-ERM with nonconvex loss functions. As most existing algorithms achieving differential privacy in nonconvex ERM are based on the gradient perturbation (BST14; WYX17; WJEG19; ZCH$^+$20), we will also focus on gradient perturbation.

Non-constant stepsizes for SGD and DP-SGD. To ease the difficulty of stepsize tuning, we could apply polynomially decaying stepsize schedules (GKKN19) or adaptive gradient methods that update the stepsize using the gradient information (DHS11; MS10). We called them adaptive stepsizes to distinguish our adaptive deferentially private methods. These non-private algorithms update the stepsize according to the noisy gradients, and achieve favorable convergence behavior (LYC18; LO19; WWB19; RCZ$^+$21).

Empirical evidence suggests that differential privacy with adaptive stepsizes could perform almost as well as – and sometimes better than – DP-SGD with well-tuned stepsizes. This results in a significant reduction in stepsize tuning efforts and also avoids the extra privacy cost (BDLS20; ZKY$^+$20; ZCH$^+$20). Several works (LK18; KH20) also studied the nonuniform allocation of the privacy budget for each iteration. However, (LK18) only proposes a heuristic method and the purpose of (KH20) is to avoid the need for a validation set used to tune stepsizes. In this work, we emphasize on the optimal relationship between the stepsize and the variance of the random noise, and aim to improve the utility guarantee of our proposed method.

8 Conclusion and Future work

In this paper, we proposed an adaptive differentially private stochastic gradient descent method in which the privacy mechanisms can be optimally adapted to the choice of stepsizes at each round, and thus obtain improved utility guarantees over prior work. Our proposed method has not only strong theoretical guarantees but also superior empirical performance. Given high-dimensional settings with only a fixed privacy budget available, our approach with a decaying stepsize schedule shows an improvement in convergence by a magnitude $\mathcal{O}(d\log(T)\sqrt{T}/n^2)$ or a factor with $\mathcal{O}(\log(T))$ relative to DP-SGD.

Note that the sequence $\{\alpha_t\}$ has to be fixed before the optimization process begins, as our method require that the variance σ^2 for some privacy budget ε depends on the $\{\alpha_t\}$ (Theorem 4.1). However, our theorem suggests that the optimal choice of α_t depends on the stepsize (Theorem 4.2), meaning that we have to know the stepsizes a priori; this is not possible for those stepsizes updated on the fly, such as AdaGrad (DHS11) and Adam (KB14). Thus, one potential avenue of future work is to see whether $\{\alpha_t\}$ can be updated on the fly in line with AdaGrad and Adam while maintaining a predefined privacy budget ε. Other future directions can be related to examining more choices of α_t given b_t. As mentioned in the main text, the relation $\alpha_t^2 = b_t$ is not the unique setting to achieve the improved utility guarantees. A thorough investigation on α_t and b_t with various gradient clipping values would therefore be an interesting extension. Finally, our adaptive differential privacy is applied only to a simple first-order optimization; generalizing the analysis to variance-reduced or momentum-based methods could be another interesting direction.

REFERENCES

[ACG+16] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[ATMR19] Galen Andrew, Om Thakkar, H Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *arXiv preprint arXiv:1905.03871*, 2019.

[BCN18] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.

[BDLS20] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. Deep learning with gaussian differential privacy. *Harvard data science review*, 2020(23), 2020.

[BFGT20] Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33, 2020.

[BST14] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[CLSH18] Xiangyi Chen, Sijia Liu, Ruoyu Sun, and Mingyi Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2018.

[CMS11] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[CTW+20] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*, 2020.

[CWH20] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: A geometric perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[DKM+06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[DRV10] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[FJR15] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[FKT20] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *The ACM Symposium on Theory of Computing (STOC)*, 2020.

[GDG+17] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[GKKN19] Rong Ge, Sham M Kakade, Rahul Kidambi, and Praneeth Netrapalli. The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[GL13] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[GSL+21] Umang Gupta, Dimitris Stripelis, Pradeep K. Lam, Paul Thompson, Jose Luis Ambite, and Greg Ver Steeg. Membership inference attacks on deep regression models for neuroimaging. In *Medical Imaging with Deep Learning*, 2021.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[INS+19] 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.

[JD20] Matt Jordan and Alexandros G Dimakis. Exactly computing the local lipschitz constant of relu networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7344–7353. Curran Associates, Inc., 2020.

[JW18] Bargav Jayaraman and Lingxiao Wang. Distributed learning without distress: Privacy-preserving empirical risk minimization. *Advances in Neural Information Processing Systems*, 2018.

[JWK+20] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

[Kam20] Gautam Kamath. Lecture 5: Approximate differential privacy. *Lecture Note*, 2020.

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KH+09] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[KH20] Antti Koskela and Antti Honkela. Learning rate adaptation for differentially private learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2465–2475. PMLR, 26–28 Aug 2020.

[KLN+11] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[KOV17] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017.

[KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, 2012.

[LBBH98] 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.

[LK18] Jaewoo Lee and Daniel Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1656–1665, 2018.

[LO19] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 983–992. PMLR, 2019.

[LYC18] Kfir Y Levy, Alp Yurtsever, and Volkan Cevher. Online adaptive methods, universality and acceleration. *Advances in Neural Information Processing Systems*, 31:6500–6509, 2018.

[MS10] B. McMahan and M. Streeter. Adaptive bound optimization for online convex optimization. *Conference on Learning Theory*, page 244, 2010.

[MTZ19] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *ArXiv*, abs/1908.10530, 2019.

[NHN+20] Yuta Nakamura, Shouhei Hanaoka, Yukihiro Nomura, Naoto Hayashi, Osamu Abe, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. Kart: Privacy leakage framework of language models pre-trained with clinical records. *arXiv preprint arXiv:2101.00036*, 2020.

[PSY+19] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Adaclip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.

[RCZ+21] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konecny, Sanjiv Kumar, and Hugh Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations*, 2021.

[RKK18] Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[SSSS17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[SV18] Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 3839–3848, 2018.

[VSP+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[WGB+20] Bao Wang, Quanquan Gu, March Boedihardjo, Lingxiao Wang, Farzin Barekat, and Stanley J. Osher. DP-LSSGD: A stochastic optimization method to lift the utility in privacy-preserving ERM. In Jianfeng Lu and Rachel Ward, editors, *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, volume 107 of *Proceedings of Machine Learning Research*, pages 328–351, Princeton University, Princeton, NJ, USA, 20–24 Jul 2020. PMLR.

[WJEG19] Lingxiao Wang, Bargav Jayaraman, David Evans, and Quanquan Gu. Efficient privacy-preserving nonconvex optimization. *arXiv e-prints*, pages arXiv–1910, 2019.

[WLK+17] 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.

[WWB19] Rachel Ward, Xiaoxia Wu, and Leon Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686. PMLR, 2019.

[WYX17] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: faster and more general. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2719–2728, 2017.

[ZCH+20] Yingxue Zhou, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Arindam Banerjee. Private stochastic non-convex optimization: Adaptive algorithms and tighter generalization bounds. *arXiv preprint arXiv:2006.13501*, 2020.

[ZKY+20] Yingxue Zhou, Belhal Karimi, Jinxing Yu, Zhiqiang Xu, and Ping Li. Towards better generalization of adaptive gradient methods. *Advances in Neural Information Processing Systems*, 33, 2020.

[ZZMW17] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private ERM for smooth objectives. In *International Joint Conference on Artificial Intelligence*, 2017.