Towards Better Generalization of Adaptive Gradient Methods

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adaptive gradient methods such as AdaGrad, RMSprop and Adam have been optimizers of choice for deep learning due to their fast training speed. However, it was recently observed that their generalization performance is often worse than that of SGD for over-parameterized neural networks. While new algorithms such as AdaBound, SWAT, and Padam were proposed to improve the situation, the provided analyses are only committed to optimization bounds with training, leaving critical generalization capacity unexplored. To close this gap, we propose ***S**table **A**daptive **G**radient **D**escent* (SAGD) for non-convex optimization which leverages differential privacy to boost the generalization performance of adaptive gradient methods. Theoretical analyses show that SAGD has high-probability convergence to a population stationary point. We further conduct experiments on various popular deep learning tasks and models. Experimental results illustrate that SAGD is empirically competitive and often better than baselines.

1 Introduction

We consider in this paper, the following minimization problem:

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) \triangleq \mathbb{E}_{z\sim\mathcal{P}}[\ell(\mathbf{w}, z)] \,, \tag{1}$$

where the *population loss* f is a (possibly) nonconvex objective function (as for most deep learning tasks), $\mathcal{W} \subset \mathbb{R}^d$ is the parameter set and z is the vector of data samples distributed according to an unknown data distribution $\mathcal{P}$. We assume that we have access to an oracle that, given n i.i.d. samples $(\mathbf{z}_1, \dots, \mathbf{z}_n)$, returns the stochastic objectives $(\ell(\mathbf{w}, \mathbf{z}_1), \dots, \ell(\mathbf{w}, \mathbf{z}_n))$. Our goal is to find critical points of the population loss function. Given the unknown data distribution, a natural approach towards solving (1) is empirical risk minimization (ERM) [29], which minimizes the *empirical loss* $\hat{f}(\mathbf{w})$ as follows: $\min_{\mathbf{w}\in\mathcal{W}} \hat{f}(\mathbf{w}) \triangleq \frac{1}{n}\sum_{j=1}^{n} \ell(\mathbf{w}, \mathbf{z}_j)$, when n samples $\mathbf{z}_1, \dots, \mathbf{z}_n$ are observed. Stochastic gradient descent (SGD) [28] which iteratively updates the parameter of a model by descending along the negative gradient computed on a single sample or a mini-batch of samples has been most dominant algorithms for solving the ERM problem, e.g., training deep neural networks. To automatically tune the learning-rate decay in SGD, adaptive gradient methods, such as AdaGrad [6], RMSprop [31], and Adam [16], have emerged leveraging adaptive coordinate-wise learning rates for faster convergence.

However, the generalization ability of these adaptive methods is often worse than that of SGD for over-parameterized neural networks, e.g., convolutional neural network (CNN) for image classification and recurrent neural network (RNN) for language modeling [35]. To mitigate this issue, several recent algorithms were proposed to combine adaptive methods with SGD. For example, AdaBound [21] and SWAT [15] switch from Adam to SGD as the training proceeds, while

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Padam [4, 37] unifies AMSGrad [27] and SGD with a partially adaptive parameter. Despite much efforts on deriving theoretical convergence results of the objective function [36, 34, 39, 5], these newly proposed adaptive gradient methods are often misunderstood regarding their generalization capacity, which is the ultimate goal. On the other hand, current adaptive gradient methods [6, 16, 31, 27, 34] follow a typical stochastic optimization (SO) oracle [28, 12] which uses stochastic gradients to update the parameter. The SO oracle requires *new samples* at every iteration to get the stochastic gradient such that it equals the population gradient in expectation. In practice, however, only finite training samples are available and reused by the optimization oracle for a certain number of times (a.k.a., epochs). Hardt et al. [13] found that the generalization error increases with the number of times the optimization oracle passes the training data. It is thus expected that gradient descent algorithms will be much more well-behaved if we have access to infinite fresh samples. Re-using data samples is therefore a caveat for the generalization of a given algorithm.

In order to tackle the above issues, we propose ***S**table **A**daptive **G**radient **D**escent* (SAGD) which aims at improving the generalization of general adaptive gradient descent algorithms. SAGD behaves similarly to the aforementioned ideal case of infinite fresh samples borrowing ideas from *adaptive data analysis* [8] and *differential privacy* [7]. The main idea of our method is that, at each iteration, SAGD accesses the training set z through a differentially private mechanism and computes an estimated gradient $\nabla\ell(\mathbf{w}, z)$ of the objective function $\nabla f(\mathbf{w})$. It then uses the estimated gradient to perform a descent step using adaptive step size. We prove that the reused data points in SAGD nearly possesses the statistical nature of *fresh samples* yielding to high concentration bounds of the population gradients through the iterations.

Our contributions can be summarized as follows:

- We derive a novel adaptive gradient method, namely SAGD, leveraging ideas of differential privacy and adaptive data analysis aiming at improving the generalization of current baseline methods. A mini-batch variant is also introduced for large-scale learning tasks.
- Our differentially private mechanism, embedded in the SAGD, explores the idea of Laplace Mechanism (adding Laplace noises to gradients) and Thresholdout [7] leading to DPG-Lap and DPG-Sparse methods which potentially saves privacy cost. In particular, we show that differentially private gradients stay close to the population gradients with high probability.
- We establish various theoretical guarantees for our algorithm. We first show that the ℓ_2-norm of the *population gradient*, i.e., $\|\nabla f(\mathbf{w})\|$ obtained by the SAGD converges with high probability. Then, we present a generalization analysis of the proposed algorithms, showing that the norm of the population gradient converges with high probability.
- We conduct several experimental applications based on training neural networks for image classification and language modeling indicating that SAGD outperforms existing adaptive gradient methods in terms of the generalization performance.

The remainder of the paper is organized as follows. Section 2 describes related work and notations. The SAGD algorithm, including the differentially private mechanisms, and its mini-batch variant are described in Section 3. Numerical experiments are presented Section 4. Section 5 concludes our work. Due to space limit, most of the proofs are deferred to the supplementary material.

2 Preliminaries

2.1 Related Work

Adaptive Gradient Methods: In the non-convex setting, existing work on SGD [12] and adaptive gradient methods [36, 34, 39, 5] shows convergence to a stationary point with a rate of $O(1/\sqrt{T})$ where T is the number of stochastic gradient computations. Given n samples, a stochastic oracle can obtain at most n stochastic gradients , which implies convergence to the population stationarity with a rate of $O(1/\sqrt{n})$. In addition, Kuzborskij and Lampert [18], Raginsky et al. [26], Hardt et al. [13], Mou et al. [24], Pensia et al. [25], Chen et al. [5], Li et al. [20] studied the generalization of gradient-based optimization algorithms using the generalization property of algorithm stability [2]. Particularly, Raginsky et al. [26], Mou et al. [24], Li et al. [20], Pensia et al. [25] focus on noisy gradient algorithms, e.g., SGLD, and provide a generalization error (population risk minus empirical risk) bound as $O(\sqrt{T}/n)$. This type of bounds usually has a dependence on the training data and has polynomial dependence on the iteration number T. This work focuses on the first type of bounds, i.e., the ℓ_2-norm of the gradient.

Differential Privacy and Adaptive Data Analysis: Differential privacy [7] was originally studied for preserving the privacy of individual data in the statistical query. Recently, differential privacy has been widely used in the area of optimization. Some pioneering work [3, 1, 33] introduced differential privacy to empirical risk minimization (ERM) to protect sensitive information of the training data. The popular differentially private algorithms includes the gradient perturbation that adds noise to the gradient in gradient descent algorithms [3, 1, 32].

Actually, except for preserving the privacy, differential privacy also has the property of guarantee generalization in adaptive data analysis (ADA) [9, 10, 11]. In ADA, a holdout set is reused for multiple times to test the hypotheses which are generated based previous test result. It has been shown that reusing the holdout set via a differentially private mechanism ensures the validity of the test. In other words, the differentially private reused dataset maintains the statistical nature of fresh samples. Dwork et al. [9, 10, 11] designed a practical method named Thresholdout, which can be used to test a large number of hypotheses. Zhou et al. [38] extended the idea of differential privacy and adaptive data analysis to convex optimization and provides generalization error bound.

2.2 Notations

We use $\mathbf{g}_t$ and $\nabla f(\mathbf{w})$ interchangeably to denote the *population gradient* such that $\mathbf{g}_t = \nabla f(\mathbf{w}_t) = \mathbb{E}_{\mathbf{z}\in\mathcal{P}}[\nabla \ell(\mathbf{w}_t, \mathbf{z})]$. $S = \{\mathbf{z}_1, \ldots, \mathbf{z}_n\}$ denotes the n available training samples. $\hat{\mathbf{g}}_t$ denotes the sample gradient evaluated on S such that $\hat{\mathbf{g}}_t = \nabla \hat{f}(\mathbf{w}) = \frac{1}{n}\sum_{j=1}^{n} \nabla \ell(\mathbf{w}_t, \mathbf{z}_j)$. For a vector $\mathbf{v}$, $\mathbf{v}^2$ represents that $\mathbf{v}$ is element-wise squared. We use $\mathbf{v}^i$ or $[\mathbf{v}]_i$ to denote the i-th coordinate of $\mathbf{v}$ and $\|\mathbf{v}\|_2$ is the ℓ_2-norm of $\mathbf{v}$.

Definition 1. *(Differential Privacy [7]) A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if*

$$\mathbb{P}\{\mathcal{M}(\mathcal{D}) \in \mathcal{Y}\} \leq \exp(\epsilon)\mathbb{P}\{\mathcal{M}(\mathcal{D}\prime) \in \mathcal{Y}\} + \delta.$$

holds for all $\mathcal{Y} \subseteq Range(\mathcal{M})$ and all pairs of adjacent datasets $\mathcal{D}, \mathcal{D}\prime$ that differ on a single data point.

Intuitively, differential privacy means that the outcomes of two nearly identical datasets should be nearly identical such that an analyst will not be able to distinguish any single data point by monitoring the change of the output. In the context of machine learning, this randomized algorithm $\mathcal{M}$ could be a learning algorithm that outputs a classifier, i.e., $\mathcal{M}(D) = f$, where D is the training set. For gradient-based optimization algorithms, $\mathcal{M}$ could be a gradient computing method that outputs an estimated gradient, i.e., $\mathcal{M}(D) = \mathbf{g}$. The general approach for achieving (ϵ, δ)-differential privacy when estimating a deterministic real-valued function $q : \mathcal{Z}^n \to \mathbb{R}^d$ is Laplace Mechanism [7], which adds Laplace noise calibrated to the function q, i.e., $\mathcal{M}(\mathcal{D}) = q(\mathcal{D}) + \mathbf{b}$, where $\mathbf{b}^i, \forall i \in [d]$ is drawn from a Laplace Distribution with variance σ^2 and zero mean.

We make the following assumptions about the objective function throughout the paper. We assume $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable (not necessarily convex), bounded from below by $f^\star$, and has L-Lipschitz gradient, i.e.,

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\| \leq L\|\mathbf{w} - \mathbf{w}'\|, \ \forall\, \mathbf{w}, \mathbf{w}' \in \mathcal{W}.$$

We also assume that the ℓ_1 norm of the individual gradient is bounded: $\|\nabla \ell(\mathbf{w}, z)\|_1 \leq G_1, \ \forall\, \mathbf{w} \in \mathcal{W}, \ \mathbf{z} \in \mathcal{Z}$ and the noisy gradient is bounded: $\|\tilde{\mathbf{g}}_t\|_2 \leq G, \forall t \in [T]$.

3 Stable Adaptive Gradient Descent Algorithm

In this section, we present SAGD with two differentially private methods to compute the estimated gradient, namely DPG-Lap and DPG-Sparse. We present the SAGD algorithm in two parts: adaptive gradient for updating the parameter (Algorithm 1), and **D**ifferential **P**rivate **G**radient (DPG, Algorithm 2) for updating the gradient. Algorithm 1 uses DPG to obtain an estimated gradient (line 4 in Algorithm 1). For DPG, we first provide a basic algorithm named *DPG-Lap* which is based on the *Laplace Mechanism* [7] in Section 3.1. Later on, we provide an advanced version named *DPG-Sparse* which is motivated by sparse vector technique [7] in Section 3.2.

Algorithm 1 SAGD

```
1: Input: Dataset S, certain loss ℓ(·), initial point w_0.
2: Set noise level σ, iteration number T, and step size η_t.
3: for t = 0, ..., T − 1 do
4:     Call DPG(S, ℓ(·), w_t, σ) to compute gradient g̃_t.
5:     m_t = g̃_t and v_t = (1 − β_2) Σ_{i=1}^{t} β_2^{t−i} g̃_i^2.
6:     w_{t+1} = w_t − η_t m_t/(√v_t + ν).
7: end for
```

3.1 SAGD with DPG-LAP

We provide the pseudo code of SAGD in Algorithm 1. Given n training samples S, loss function ℓ, at each iteration $t \in [T]$, instead of computing a stochastic gradient as previous adaptive gradient descent algorithms, Algorithm 1 calls DPG(S, $\ell(\cdot)$, $\mathbf{w}_t$, σ) to access the training set S and obtain an estimated $\tilde{\mathbf{g}}_t$ (line 4), then updates $\mathbf{w}_{t+1}$ based on $\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_t$ using the adaptive step size (line 5, 6): $\mathbf{m}_t = \tilde{\mathbf{g}}_t$, $\mathbf{v}_t = (1-\beta_2)\sum_{i=1}^{t}\beta_2^{t-i}\tilde{\mathbf{g}}_i^2$, and $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{m}_t/(\sqrt{\mathbf{v}_t}+\nu)$. Note that noise variance σ^2, step-size η_t, and iteration number T, β_2, ν are the parameters of Algorithm 1. We analyze the optimal values of them for SAGD in the subsequent sections.

Algorithm 2 DPG-Lap

```
1: Input: Dataset S, certain loss ℓ(·), parameter w_t, noise level σ.
2: Compute full batch gradient on S:
       ĝ_t = (1/n) Σ_{j=1}^{n} ∇ℓ(w_t, z_j).
3: Set g̃_t = ĝ_t + b_t, where b_t^i is drawn i.i.d from Lap(σ), ∀ i ∈ [d].
4: Output: g̃_t.
```

For the DPG, we first consider *DPG-Lap* (Algorithm 2) which adds Laplace noises $\mathbf{b}_t \in \mathbb{R}^d$ to the empirical gradient $\hat{\mathbf{g}}_t = \frac{1}{n}\sum_{j=1}^{n}\nabla\ell(\mathbf{w}_t, \mathbf{z}_j)$ and returns a noisy gradient $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_t + \mathbf{b}_t$ to the optimization oracle Algorithm 1.

To analyze the convergence of SAGD in terms of ℓ_2 norm of the population gradient, we need to show that $\tilde{\mathbf{g}}_t$ approximate the population gradient $\mathbf{g}_t$ with high probability, i.e., the estimation error $\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\|$ is small at every iteration. To make such an analysis, we first present the generalization guarantee of any differentially private algorithm in Lemma 1, then we show that SAGD is differentially private in Lemma 2. It is followed by establishing SAGD's generalization guarantee in Theorem 1, i.e., estimated $\tilde{\mathbf{g}}_t$ approximates the population gradient $\mathbf{g}_t$ with high probability. Last, we prove that SAGD converges to a population stationary point with high probability in Theorem 2.

The general approach for analyzing the estimation error of sample gradient to population gradient is the Hoeffding's bound. Given training set $S \in \mathcal{Z}^n$ and a fixed $\mathbf{w}_0$ chosen to be independent of the dataset S, we have empirical gradient $\hat{\mathbf{g}}_0 = \mathbb{E}_{z\in S}\nabla\ell(\mathbf{w}_0, z)$ and population gradient $\mathbf{g}_0 = \mathbb{E}_{z\sim\mathcal{P}}[\nabla l(\mathbf{w}_0, z)]$. Hoeffding's bound implies generalization of fresh samples, i.e., for every coordinate $i \in [d]$ and $\mu > 0$, empirical gradients are concentrated around population gradients, i.e.,

$$P\{|\hat{\mathbf{g}}_0^i - \mathbf{g}_0^i| \geq \mu\} \leq 2\exp\left(\frac{-2n\mu^2}{4G_\infty^2}\right), \tag{2}$$

where G_∞ is the maximal value of the ℓ_∞-norm of the gradient $\mathbf{g}_0$. Generally, if $\mathbf{w}_1$ is updated using the gradient computed on training set S, i.e., $\mathbf{w}_1 = \mathbf{w}_0 - \eta\hat{\mathbf{g}}_0$, the above concentration inequality *will not* hold for $\hat{\mathbf{g}}_1 = \mathbb{E}_{z\in S}\nabla_i\ell(\mathbf{w}_1, z)$, because $\mathbf{w}_1$ is no longer independent of dataset S. However, Lemma 1 shows that if $\mathbf{w}_t, \forall t \in [T]$ is generated by reusing S under a differentially private mechanism, concentration bounds similar to Eq. (2) will hold for all $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_T$ that are adaptively generated on the same dataset S.

Lemma 1. *Let $\mathcal{A}$ be an (ϵ, δ)-differentially private gradient descent algorithm with access to training set S of size n. Let $\mathbf{w}_t = \mathcal{A}(S)$ be the parameter generated at iteration $t \in [T]$ and $\hat{\mathbf{g}}_t$ the empirical gradient on S. For any $\sigma > 0$, $\beta > 0$, if the privacy cost of $\mathcal{A}$ satisfies $\epsilon \leq \frac{\sigma}{13}$,*

$\delta \leq \frac{\sigma\beta}{26\ln(26/\sigma)}$, *and sample size* $n \geq \frac{2\ln(8/\delta)}{\epsilon^2}$, *we then have*

$$\mathbb{P}\left\{|\hat{\mathbf{g}}_t^i - \mathbf{g}_t^i| \geq \sigma\right\} \leq \beta \quad \textit{for every } i \in [d] \textit{ and every } t \in [T].$$

Lemma 1 is an instance of Theorem 8 from [8] and illustrates, if the privacy cost ϵ is bounded by the estimation error, that differential privacy enables the reused training set to maintain statistical guarantees as a fresh sample. Next, we analyze the privacy cost of SAGD in Lemma 2.

Lemma 2. SAGD *with DPG-Lap is* $(\frac{\sqrt{T\ln(1/\delta)}G_1}{n\sigma}, \delta)$*-differentially private.*

In order to achieve a gradient concentration bound for SAGD with DPG-Lap as described in Lemma 1, we need to set $\frac{\sqrt{T\ln(1/\delta)}G_1}{n\sigma} \leq \frac{\sigma}{13}$, $\delta \leq \frac{\sigma\beta}{26\ln(26/\sigma)}$, and sample size $n \geq \frac{2\ln(8/\delta)}{\epsilon^2}$. We then have the following theorem showing that across all iterations, gradients produced by SAGD with DPG-Lap maintain high probability concentration bounds.

Theorem 1. *Given parameter* $\sigma > 0$*, let* $\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_T$ *be the gradients computed by DPG-Lap in* SAGD *over* T *iterations. Set the total number of iterations* $\frac{2n\sigma^2}{G_1^2} \leq T \leq \frac{n^2\sigma^4}{169\ln(1/(\sigma\beta))G_1^2}$*, then for* $\forall\, t \in [T]$ *any* $\beta > 0$*, and any* $\mu > 0$ *we have:*

$$\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\} \leq d\beta + d\exp(-\mu).$$

Theorem 1 indicates that gradient $\tilde{\mathbf{g}}_t$ produced by DPG-Lap is concentrated around population gradient $\mathbf{g}_t$ with a tight concentration error bound $\sqrt{d}\sigma(1+\mu)$. A higher noise level σ brings a better privacy guarantee and a larger number of iterations T, but meanwhile incurs a larger concentration error $\sqrt{d}\sigma(1+\mu)$. Thus, there is a trade-off between noise and accuracy. β and μ are any positive numbers that illustrate the trade-off between the concentration error and the probability. A larger μ brings a larger concentration error but a smaller probability. For β, if we increase β, we get a larger upper bound on T, which means the concentration bound will hold for more iterations, but we also get a larger probability. Note that although the probability $d\beta + d\exp(-\mu)$ has a dependence on dimension d, we can choose appropriate β and μ to make the probability arbitrarily small. We optimize the choice of β and μ when analyzing the convergence to the population stationary point.

We derive the optimal values of σ and T to optimize the trade-off between statistical rate and optimization rate and obtain the optimal bound in Theorem 2. For brevity, let $\rho_{n,d} \triangleq O(\ln n + \ln d)$.

Theorem 2. *Given training set* S *of size* n*, for* $\nu > 0$*, if* $\eta_t = \eta$ *which are chosen with* $\eta \leq \frac{\nu}{2L}$*,* $\sigma = 1/n^{1/3}$*, and iteration number* $T = n^{2/3}/\left(169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$*, then* SAGD *with DPG-Lap converges to a stationary point of the population risk, i.e.,*

$$\min_{1\leq t\leq T} \|\nabla f(\mathbf{w}_t)\|^2 \leq O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{n^{2/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{n^{2/3}}\right),$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

Theorem 2 shows that, given n samples, SAGD converges to a population stationary point at a rate of $O(1/n^{2/3})$. Particularly, the first term of the bound corresponds to the optimization error $O(1/T)$ with $T = O(n^{2/3})$, while the second is the statistical error depending on available sample size n and dimension d. In terms of computation complexity, SAGD requires $O(n^{5/2})$ stochastic gradient computations for $O(n^{3/2})$ passes over n samples. The current optimization analyses [36, 34, 39, 5] show that adaptive gradient descent algorithms (SO oracle) converges to the population stationary point with a rate of $O(1/\sqrt{T})$ with T stochastic gradient computations. Given n samples, their analyses give a rate of $O(1/\sqrt{n})$. The SAGD achieves a sharper bound compared to the previous analyses. We will consider improving the dependence on dimension d in our future work.

3.2 SAGD with DPG-SPARSE

In this section, we consider the SAGD with an advanced version of DPG named *DPG-Sparse* which is motivated by sparse vector technique [7] aiming to provide a sharper result on the privacy cost ϵ and δ.

Algorithm 3 SAGD with DPG-Sparse

Input: Dataset S, certain loss $\ell(\cdot)$, initial point $\mathbf{w}_0$.
Set noise level σ, iteration number T, and step size η_t.
Split S randomly into S_1 and S_2.
for $t = 0, ..., T-1$ **do**
    Compute full batch gradient on S_1 and S_2:
$$\hat{\mathbf{g}}_{S_1,t} = \frac{1}{|S_1|}\sum_{\mathbf{z}_j \in S_1} \nabla \ell(\mathbf{w}_t, \mathbf{z}_j),$$
$$\hat{\mathbf{g}}_{S_2,t} = \frac{1}{|S_2|}\sum_{\mathbf{z}_j \in S_2} \nabla \ell(\mathbf{w}_t, \mathbf{z}_j).$$
    Sample $\gamma \sim \text{Lap}(2\sigma)$, $\tau \sim \text{Lap}(4\sigma)$.
    if $\|\hat{\mathbf{g}}_{S_1,t} - \hat{\mathbf{g}}_{S_2,t}\| + \gamma > \tau$ **then**
        $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_{S_1,t} + \mathbf{b}_t$, where $\mathbf{b}_t^i$ is drawn i.i.d from $\text{Lap}(\sigma), \forall\, i \in [d]$.
    else
        $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_{S_2,t}$
    end if
    $\mathbf{m}_t = \tilde{\mathbf{g}}_t$ and $\mathbf{v}_t = (1-\beta_2)\sum_{i=1}^{t} \beta_2^{t-i} \tilde{\mathbf{g}}_i^2$.
    $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{m}_t / (\sqrt{\mathbf{v}_t} + \nu)$.
end for
Return: $\tilde{\mathbf{g}}_t$.

Lemma 2 shows that the privacy cost of SAGD with DPG-Lap scales with $O(\sqrt{T})$. In order to guarantee the generalization of SAGD as stated in Theorem 1, we need to control the privacy cost below a certain threshold i.e., $\frac{\sqrt{T \ln(1/\delta)} G_1}{n\sigma} \leq \frac{\sigma}{13}$. However, it limits the iteration number T of SAGD, leading to a compromised optimization term in Theorem 2. To achieve relax the upper bound on the T, we use another differentially private mechanism, i.e., sparse vector technique [8, 10, 11, 7] instead of Laplace Mechanism to reduce the privacy cost. Thus, we propose an alternative to DPG, named SAGD with DPG-Sparse (Algorithm 3).

Given n samples, Algorithm 3 splits the dataset evenly into two parts S_1 and S_2. At every iteration t, Algorithm 3 computes gradients on both datasets: $\hat{\mathbf{g}}_{S_1,t} = \frac{1}{|S_1|}\sum_{\mathbf{z}_j \in S_1} \nabla \ell(\mathbf{w}_t, \mathbf{z}_j)$ and $\hat{\mathbf{g}}_{S_2,t} = \frac{1}{|S_2|}\sum_{\mathbf{z}_j \in S_2} \nabla \ell(\mathbf{w}_t, \mathbf{z}_j)$. It then validates $\hat{\mathbf{g}}_{S_1,t}$ with $\hat{\mathbf{g}}_{S_2,t}$. That is, if the norm of their difference is greater than a random threshold $\tau - \gamma$, it then returns $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_{S_1,t} + \mathbf{b}_t$, otherwise $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_{S_2,t}$. Note that Algorithm 3 is an extension of Thresholdout in Zhou et al. [38]. Inspired by Thresholdout, Zhou et al. [38] proposed stable gradient descent algorithms which use a similar framework as DPG-Sparse to compute an estimated gradient by validating each coordinate of $\hat{\mathbf{g}}_{S_1,t}$ and $\hat{\mathbf{g}}_{S_2,t}$. However, their method is computationally expensive in high-dimensional settings such as deep neural networks.

To analyze the privacy cost of DPG-Sparse, let C_s be the number of times the validation fails, i.e., $\|\hat{\mathbf{g}}_{S_1,t} - \hat{\mathbf{g}}_{S_2,t}\| + \gamma > \tau$ is true, over T iterations in SAGD. The following Lemma presents the privacy cost of SAGD with DPG-Sparse.

Lemma 3. SAGD *with DPG-Sparse (Algorithm 3) is* $(\frac{\sqrt{C_s \ln(2/\delta)} 2G_1}{n\sigma}, \delta)$*-differentially private.*

Lemma 3 shows that the privacy cost of SAGD with DPG-Sparse scales with $O(\sqrt{C_s})$ where $C_s \leq T$. In other words, DPG-Sparse saves the privacy cost of SAGD. In order to achieve the generalization guarantee of SAGD with DPG-Sparse as stated in Lemma 1, by considering the guarantee of Lemma 3, we only need to set $\frac{\sqrt{C_s \ln(1/\delta)} G_1}{n\sigma} \leq \frac{\sigma}{13}$, which potentially improves the upper bound of T. The following theorem shows the generalization guarantee of $\tilde{\mathbf{g}}_t$ generated by SAGD with DPG-Sparse.

Theorem 3. *Given parameter $\sigma > 0$, let $\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_T$ be the gradients computed by DPG-Sparse over T iterations. With a budget $\frac{n\sigma^2}{2G_1^2} \leq C_s \leq \frac{n^2\sigma^4}{676 \ln(1/(\sigma\beta)) G_1^2}$, for $\forall\, t \in [T]$, any $\beta > 0$, and any $\mu > 0$ we have*

$$\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\} \leq d\beta + d\exp(-\mu).$$

In the worst case $C_s = T$, we can recover the upper bound of T as $T \leq \frac{n^2\sigma^4}{676\ln(1/(\sigma\beta))G_1^2}$. DPG-Sparse behaves as DPG-Lap in this worst case. The following theorem displays the *worst case bound* of SAGD with DPG-Sparse.

Theorem 4. *Given training set S of size n, for $\nu > 0$, if $\eta_t = \eta$ which are chosen with $\eta \leq \frac{\nu}{2L}$, noise level $\sigma = 1/n^{1/3}$, and iteration number $T = n^{2/3}/\left(676G_1^2(\ln d + \frac{7}{3}\ln n)\right)$, then* SAGD *with DPG-Sparse guarantees convergence to a stationary point of the population risk:*

$$\min_{1\leq t\leq T} \|\nabla f(\mathbf{w}_t)\|^2 \leq O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{n^{2/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{n^{2/3}}\right),$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

Theorem 4 shows that the worst case of SAGD with DGP-Sparse converges to a population stationary point at a rate of $O(1/n^{2/3})$ which is the same as that of SAGD with DGP-Lap. One could obtain a sharper bound if C_s is much smaller than T. For example, if $C_s = O(\sqrt{T})$, the upper bound of T can be improved from precious $T \leq O(n^2)$ to $T \leq O(n^4)$, beyond trading off between statistical rate and optimization rate. One might consider such an analysis in the future work.

3.3 Mini-batch Stable Adaptive Gradient Descent Algorithm

The mini-batch SAGD is described in Algorithm 4. The training set S is first partitioned into B batches with m samples for each batch. At each iteration t, Algorithm 4 uses DPG to access one batch to obtain a differential private gradient $\tilde{\mathbf{g}}_t$ (line 6) and then update $\mathbf{w}_t$ (line 7-8).

Algorithm 4 Mini-Batch SAGD

1: **Input**: Dataset S, certain loss $\ell(\cdot)$, initial point $\mathbf{w}_0$.
2: Set noise level σ, epoch number T, batch size m, and step size η_t.
3: Split S into $B = n/m$ batches: $\{s_1, ..., s_B\}$.
4: **for** *epoch* $= 1, ..., T$ **do**
5:     **for** $k = 1, ..., B$ **do**
6:         Call DPG(S_k, $\ell(\cdot)$, $\mathbf{w}_t$, σ) to compute $\tilde{\mathbf{g}}_t$.
7:         $\mathbf{m}_t = \tilde{\mathbf{g}}_t$ and $\mathbf{v}_t = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}\tilde{\mathbf{g}}_i^2$.
8:         $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \mathbf{m}_t/(\sqrt{\mathbf{v}_t} + \nu)$.
9:     **end for**
10: **end for**

Theorem 5. *Given training set S of size n, with $\nu > 0$, $\eta_t = \eta \leq \frac{\nu}{2L}$, noise level $\sigma = 1/n^{1/3}$, and epoch $T = m^{4/3}/\left(n169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$, then the mini-batch* SAGD *with DPG-Lap guarantees convergence to a stationary point of the population risk, i.e.,*

$$\min_{t=1,...,T} \|\nabla f(\mathbf{w}_t)\|^2 \leq O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{(mn)^{1/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{(mn)^{1/3}}\right),$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

Theorem 5 describes the convergence rate of the mini-batch SAGD in terms of batch size m and sample size n, i.e., $O(1/(mn)^{1/3})$. When $m = \sqrt{n}$, mini-batch SAGD achieves the convergence of rate $O(1/\sqrt{n})$. When $m = n$, i.e., in the full batch setting, Theorem 5 recovers SAGD's convergence rate $O(1/n^{2/3})$. In terms of computational complexity, the mini-batch SAGD requires $O(m^{7/3}/n)$ stochastic gradient computations for $O(m^{4/3}/n)$ passes over m samples, while SAGD requires $O(n^{5/3})$ stochastic gradient computations. Thus, the mini-batch SAGD has advantages in saving computation complexity, but converges slower than SAGD.

4 Numerical Experiments

In this section, we empirically evaluate the mini-batch SAGD for training various modern deep learning models and compare them with popular optimization methods, including SGD (with momentum), Adam, Padam, AdaGrad, RMSprop, and Adabound. We consider three tasks: the MNIST image classification task [19], the CIFAR-10 image classification task [17], and the language modeling task on Penn Treebank [22]. The setup of each task is given in Table 1. After describing the experimental setup, we discuss the results on three tasks in Section 4.2.

Table 1: Neural network architecture setup.

Dataset	Network Type	Architecture
MNIST	Feedforward	2-Layer with ReLU
MNIST	Feedforward	2-Layer with Sigmoid
CIFAR-10	Deep Convolutional	VGG-19
CIFAR-10	Deep Convolutional	ResNet-18
Penn Treebank	Recurrent	2-Layer LSTM
Penn Treebank	Recurrent	3-Layer LSTM

4.1 Environmental Settings

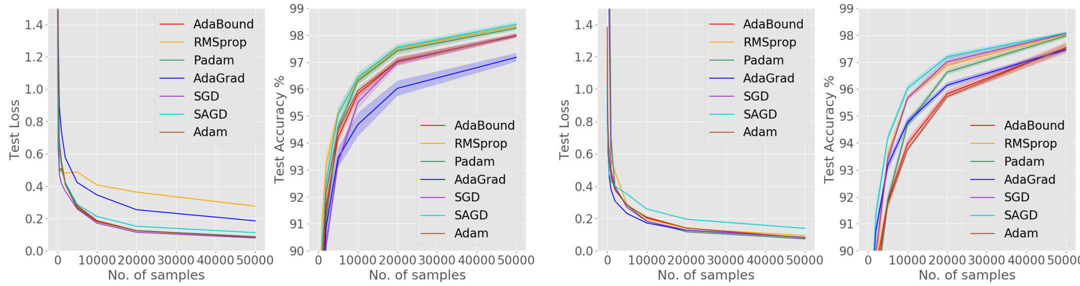
Datasets and Evaluation Metrics: The MNIST dataset has a training set of 60000 examples and a test set of 10000 examples. The CIFAR-10 dataset consists of 50000 training images and 10000 test images. The Penn Treebank dataset contains 929589, 73760, and 82430 tokens for training, validation, and test, respectively. To better understand the generalization ability of each optimization algorithm with an increasing training sample size n, for each task, we construct multiple training sets of different size by sampling from the original training set. For MNIST, training sets of size $n \in \{50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000\}$ are constructed. For CIFAR10, training sets of size $n \in \{200, 500, 1000, 2000, 5000, 10000, 20000, 30000, 50000\}$ are constructed. For each n, we train the model and report the loss and accuracy on the test set. For Penn Treebank, all training samples are used to train the model and we report the training perplexity and the test perplexity across epochs. For training, a fixed budget on the number of epochs is assigned for every task. We choose the settings achieving the lowest final training loss. Cross-entropy is used as our loss function throughout experiments. The mini-batch size is set to be 128 for CIFAR10 and MNIST, 20 for Penn Treebank. We repeat each experiment 5 times and report the mean and standard deviation of the results.

Hyper-parameter setting: Optimization hyper-parameters affect the quality of solutions. Particularly, Wilson et al. [35] found that the initial step size and the scheme of decaying step sizes have a marked impact on the performance. We follow the logarithmically-spaced grid method in Wilson et al. [35] to tune the step size. Specifically, we start with a logarithmically-spaced grid of four step sizes. If the parameter performs best at an extreme end of the grid, a new grid will be tried until the best parameter lies in the middle of the grid. Once the interval of the best step size is located, we change to the linear-spaced grid to further search for the optimal one. In addition, the strategy of decaying step sizes is specified in the subsections of each task.

Noise parameter of SAGD: We set the variance of noise σ^2 for SAGD for each experiment as the value stated in Theorem 5 such that $\sigma^2 = 1/n^{2/3}$, where n is the size of training set. The other parameters, such as ν, β_2, and T follow the default setting as other adaptive gradient descent algorithms such as RMSprop. The step size η of SAGD follows the logarithmically-spaced grid method in Wilson et al. [35].

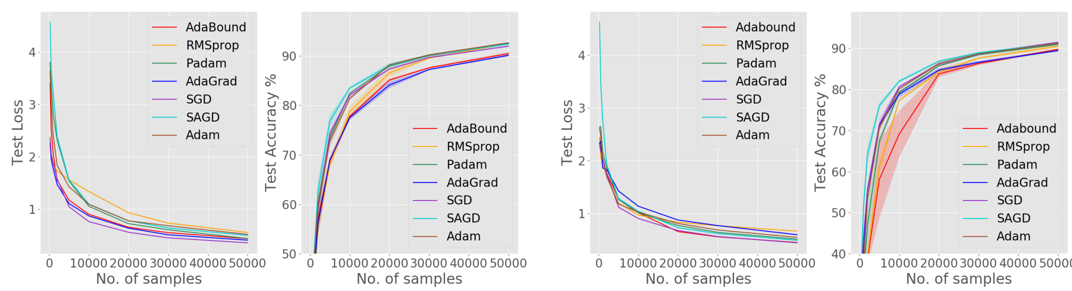
4.2 Numerical results

Feedforward Neural Network. For image classification on MNIST, we focus on two 2-layer fully connected neural networks with ReLU activation and Sigmoid activation, respectively. We run 100 epochs and decay the learning rate by 0.5 every 30 epochs. Figure 1 presents the loss and accuracy on the test set given different training sizes. Since all algorithms attain the 100% training accuracy, the



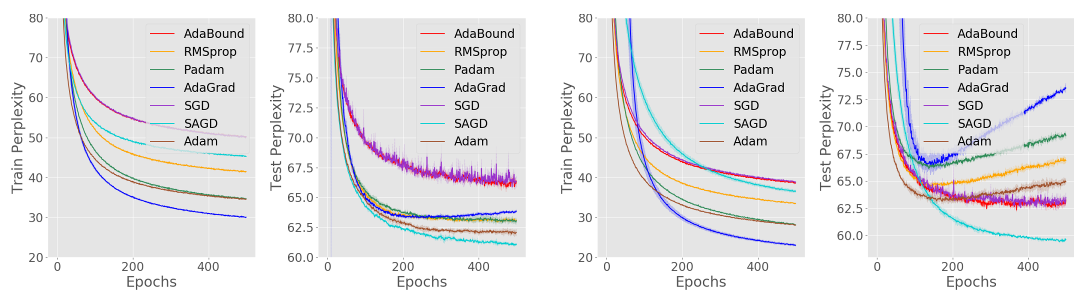
(a) MNIST, 2-layer ReLU (b) MNIST, 2-layer Sigmoid

Figure 1: Test loss and accuracy of ReLU neural network and Sigmoid neural network on MNIST. The X-axis is the number of train samples, and the Y-axis is the loss/accuracy. In both cases, SAGD obtains the best test accuracy among all the methods.



(a) CIFAR10, ResNet-18 (b) CIFAR10, VGG-19

Figure 2: Test loss and accuracy of ResNet-18 andVGG-19 on CIFAR10. The X-axis and the Y-axis refer to Figure 1. For ResNet-18, SAGD achieves the lowest test loss. For VGG-19, SAGD achieves the best test accuracy among all the methods.



(a) Penn Treebank, 2-Layer LSTM (b) Penn Treebank, 3-Layer LSTM

Figure 3: Train and test perplexity of 2-layer LSTM and 3-layer LSTM. The X-axis is the number of epochs, and the Y-axis is the train/test perplexity. Although adaptive methods such as AdGrad, Padam, Adam, and RMsprop achieves better training performance than SAGD, SAGD performs the best in terms of the test perplexity among all the methods.

performance on the training set is omitted. Figure 1 (a) shows that, for ReLU neural network, SAGD performs slightly better than the other algorithms in terms of test accuracy. When $n = 50000$, SAGD gets a test accuracy of $98.38 \pm 0.13\%$. Figure 1 (b) presents the results on Sigmoid neural network. SAGD achieves the best test accuracy among all the algorithms. When $n = 50000$, SAGD reaches the highest test accuracy of $98.14 \pm 0.11\%$, outperforming other adaptive algorithms.

Convolutional Neural Network. We use ResNet-18 [14] and VGG-19 [30] for the CIFAR-10 image classification task. We run 100 epochs and decay the learning rate by 0.1 every 30 epochs. The results are presented in Figure 2. Figure 2 (a) shows that SAGD has higher test accuracy than

Table 2: Test Perplexity of LSTMs on Penn Treebank. Bold number indicates the best result.

	RMSprop	Adam	AdaGrad	Padam	AdaBound	SGD	SAGD
2-layer LSTM	62.87 $\pm$ 0.05	60.58 $\pm$ 0.37	62.20 $\pm$ 0.29	62.85 $\pm$ 0.16	65.82 $\pm$0.08	65.96 $\pm$ 0.23	**61.02 $\pm$ 0.08**
3-layer LSTM	63.97 $\pm$ 018	63.23 $\pm$ 004	66.25 $\pm$ 0.31	66.45 $\pm$ 0.28	62.33 $\pm$0.07	62.51 $\pm$0.11	**59.43 $\pm$ 0.24**

the other algorithms when the sample size is small i.e., $n \leq 20000$. When $n = 50000$, SAGD achieves nearly the same test accuracy as Adam, Padam, and RMSprop. In detail, SAGD has test accuracy $92.48 \pm 0.09\%$. Non-adaptive algorithm SGD performs better than the other algorithms in terms of test loss. Figure 2 (b) reports the results on VGG-19. Although SAGD has a higher test loss than the other algorithms, it achieves the best test accuracy, especially when n is small. Non-adaptive algorithm SGD performs better than the other adaptive gradient algorithms regarding the test accuracy. When $n = 50000$, SGD has the best test accuracy $91.36 \pm 0.04\%$. SAGD achieves accuracy $91.26 \pm 0.05\%$

Recurrent Neural Network. Finally, an experiment on Penn Treebank is conducted for the language modeling task with 2-layer Long Short-Term Memory (LSTM) [23] network and 3-layer LSTM. We train them for a fixed budget of 500 epochs and omit the learning-rate decay. Perplexity is used as the metric to evaluate the performance and learning curves are plotted in Figure 3. Figure 3 (a) shows that for the 2-layer LSTM, AdaGrad, Padam, RMSprop and Adam achieve a lower training perplexity than SAGD. However, SAGD performs the best in terms of the test perplexity. Specifically, SAGD achieves 61.02 ± 0.08 test perplexity. Especially, It is observed that after 200 epochs, the test perplexity of AdaGrad and Adam starts increasing, but the training perplexity continues decreasing (over-fitting occurs). Figure 3 (b) reports the results for the 3-layer LSTM. We can see that the perplexity of AdaGrad, Padam, Adam, and RMSprop start increasing significantly after 150 epochs (*over-fitting*). But the perplexity of SAGD keeps decreasing. SAGD and SGD and AdaBounds perform better than AdaGrad, Padam, Adam, and RMSprop in terms of over-fitting. Table 2 shows the best test perplexity of 2-layer LSTM and 3-layer LSTM for all the algorithms. We can observe that the SAGD achieves the best test perplexity 59.43 ± 0.24 among all the algorithms.

5 Conclusion

In this paper, we focus on the generalization ability of adaptive gradient methods. Concerned with the observation that adaptive gradient methods generalize worse than SGD for over-parameterized neural networks and the theoretical understanding of the generalization of those methods is limited, we propose stable adaptive gradient descent methods (SAGD), which boost the generalization performance in both theory and practice through a novel use of differential privacy. The proposed algorithms generalize well with provable high-probability convergence bounds of the population gradient. Experimental studies demonstrate the proposed algorithms are competitive and often better than baseline algorithms for training deep neural networks. In future work, we will consider improving our analysis in several ways, e.g., improvement of the dependence on dimension and sharper bounds of SAGD with DPG-Sparse.

References

[1] R. Bassily, A. Smith, and A. 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.

[2] O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

[3] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[4] J. Chen and Q. Gu. Closing the generalization gap of adaptive gradient methods in training deep neural networks. *arXiv preprint arXiv:1806.06763*, 2018.

[5] X. Chen, S. Liu, R. Sun, and M. Hong. On the convergence of a class of adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019.

[6] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[7] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[8] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Generalization in adaptive data analysis and holdout reuse. *arXiv preprint arXiv:1506.02629*, 2015.

[9] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, pages 2350–2358, 2015.

[10] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

[11] C. Dwork, V. Feldman, M. Hardt, T. Pitassi, O. Reingold, and A. L. Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126. ACM, 2015.

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

[13] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234, 2016.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] N. S. Keskar and R. Socher. Improving generalization performance by switching from adam to sgd. *arXiv preprint arXiv:1712.07628*, 2017.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *In Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

[17] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[18] I. Kuzborskij and C. Lampert. Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 2820–2829, 2018.

[19] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] J. Li, X. Luo, and M. Qiao. On generalization error bounds of noisy gradient methods for non-convex learning. *arXiv preprint arXiv:1902.00621*, 2019.

[21] L. Luo, Y. Xiong, and Y. Liu. Adaptive gradient methods with dynamic bound of learning rate. In *International Conference on Learning Representations*, 2019.

[22] B. S. Marcus, Mitchell and M. A. Marcinkiewicz. Building a large annotated corpus of english: the penn treebank. *Computational linguistics-Association for Computational Linguistics*, 19 (2):313–330, 1993.

[23] S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing LSTM language models. In *International Conference on Learning Representations*, 2018.

[24] W. Mou, L. Wang, X. Zhai, and K. Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Conference On Learning Theory*, pages 605–638, 2018.

[25] A. Pensia, V. Jog, and P.-L. Loh. Generalization error bounds for noisy, iterative algorithms. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 546–550. IEEE, 2018.

[26] M. Raginsky, A. Rakhlin, and M. Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703, 2017.

[27] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018.

[28] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[29] S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] T. Tieleman and G. Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012.

[32] D. Wang and J. Xu. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1182–1189, 2019.

[33] D. Wang, M. Ye, and J. Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

[34] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: sharp convergence over nonconvex landscapes. In *International Conference on Machine Learning*, pages 6677–6686, 2019.

[35] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, pages 4148–4158, 2017.

[36] M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems*, pages 9793–9803, 2018.

[37] D. Zhou, Y. Tang, Z. Yang, Y. Cao, and Q. Gu. On the convergence of adaptive gradient methods for nonconvex optimization. *arXiv preprint arXiv:1808.05671*, 2018.

[38] Y. Zhou, S. Chen, and A. Banerjee. Stable gradient descent. In *UAI*, pages 766–775, 2018.

[39] F. Zou, L. Shen, Z. Jie, W. Zhang, and W. Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11127–11135, 2019.

A DIFFERENTIAL PRIVACY AND GENERALIZATION ANALYSIS

By applying Theorem 8 from Dwork et al. [9] to gradient computation, we can get the Lemma 1.

Lemma 1. *Let $\mathcal{A}$ be an (ϵ,δ)-differentially private gradient descent algorithm with access to training set S of size n. Let $\mathbf{w}_t = \mathcal{A}(S)$ be the parameter generated at iteration $t \in [T]$ and $\hat{\mathbf{g}}_t$ the empirical gradient on S. For any $\sigma > 0$, $\beta > 0$, if the privacy cost of $\mathcal{A}$ satisfies $\epsilon \leq \frac{\sigma}{13}$, $\delta \leq \frac{\sigma\beta}{26\ln(26/\sigma)}$, and sample size $n \geq \frac{2\ln(8/\delta)}{\epsilon^2}$, we then have*

$$\mathbb{P}\left\{|\hat{\mathbf{g}}_t^i - \mathbf{g}_t^i| \geq \sigma\right\} \leq \beta \quad \textit{for every } i \in [d] \textit{ and every } t \in [T].$$

Proof Theorem 8 in Dwork et al. [9] shows that in order to achieve generalization error τ with probability $1-\rho$ for a (ϵ,δ)-differentially private algorithm (i.e., in order to guarantee for every function ϕ_t, $\forall t \in [T]$, we have $\mathbb{P}\left[|\mathcal{P}[\phi_t] - \mathcal{E}_S[\phi_t]| \geq \tau\right] \leq \rho$), where $\mathcal{P}[\phi_t]$ is the population value, $\mathcal{E}_S[\phi_t]$ is the empirical value evaluated on S and ρ and τ are any positive constant, we can set the $\epsilon \leq \frac{\tau}{13}$ and $\delta \leq \frac{\tau\rho}{26\ln(26/\tau)}$. In our context, $\tau = \sigma$, $\beta = \rho$, ϕ_t is the gradient computation function $\nabla\ell(\mathbf{w}_t, \mathbf{z})$, $\mathcal{P}[\phi_t]$ represents the population gradient $\mathbf{g}_t^i$, $\forall i \in [p]$, and $\mathcal{E}_S[\phi_t]$ represents the sample gradient $\hat{\mathbf{g}}_t^i$, $\forall i \in [p]$. Thus we have $\mathbb{P}\left\{\left|\hat{\mathbf{g}}_t^i - \mathbf{g}_t^i\right| \geq \tau\right\} \leq \rho$ if $\epsilon \leq \frac{\sigma}{13}$, $\delta \leq \frac{\sigma\beta}{26\ln(26/\sigma)}$.

A.1 Proof of Lemma 2

Lemma 2. SAGD *with DPG-Lap is* $(\frac{\sqrt{T\ln(1/\delta)}G_1}{n\sigma}, \delta)$*-differentially private.*

Proof At each iteration t, the algorithm is composed of two sequential parts: DPG to access the training set S and compute $\tilde{\mathbf{g}}_t$, and parameter update based on estimated $\tilde{\mathbf{g}}_t$. We mark the DPG as part $\mathcal{A}$ and the gradient descent as part $\mathcal{B}$. We first show $\mathcal{A}$ preserves $\frac{G_1}{n\sigma}$-differential privacy. Then according to the *post-processing property* of differential privacy (Proposition 2.1 in [7]) we have $\mathcal{B} \circ \mathcal{A}$ is also $\frac{G_1}{n\sigma}$-differentially private.

The part $\mathcal{A}$ (DPG-Lap) uses the basic tool from differential privacy, the "Laplace Mechanism" (Definition 3.3 in [7]). The Laplace Mechanism adds i.i.d. Laplace noise to each coordinate of the output. Adding noise from $Lap(\sigma)$ to a query of G_1/n sensitivity preserves $G_1/n\sigma$-differential privacy by (Theorem 3.6 in [7]). Over T iterations, we have T applications of a DPG-Lap. By the advanced composition theorem (Theorem 3.20 in [7]), T applications of a $\frac{G_1}{n\sigma}$-differentially private algorithm is $(\frac{\sqrt{T\ln(1/\delta)}G_1}{n\sigma}, \delta)$-differentially private. So SAGD with DPG-Lap is $(\frac{\sqrt{T\ln(1/\delta)2G_1}}{n\sigma}, \delta)$-differentially private. □

A.2 Proof of Theorem 1

Theorem 1. *Given parameter $\sigma > 0$, let $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_T$ be the gradients computed by DPG-Lap in* SAGD *over T iterations. Set the total number of iterations $\frac{2n\sigma^2}{G_1^2} \leq T \leq \frac{n^2\sigma^4}{169\ln(1/(\sigma\beta))G_1^2}$, then for $\forall t \in [T]$ any $\beta > 0$, and any $\mu > 0$ we have:*

$$\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\} \leq d\beta + d\exp(-\mu).$$

Proof The concentration bound is decomposed into two parts:

$$\begin{aligned}
&\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\} \\
\leq &\underbrace{\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\| \geq \sqrt{d}\sigma\mu\right\}}_{T_1\text{: empirical error}} + \underbrace{\mathbb{P}\left\{\|\hat{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma\right\}}_{T_2\text{: generalization error}}
\end{aligned}$$

In the above inequality, there are two types of error we need to control. The first type of error, referred to as empirical error T_1, is the deviation between the differentially private estimated gradient $\tilde{\mathbf{g}}_t$ and the empirical gradient $\hat{\mathbf{g}}_t$. The second type of error, referred to as generalization error T_2, is the deviation between the empirical gradient $\hat{\mathbf{g}}_t$ and the population gradient $\mathbf{g}_t$.

The second term T_2 can be bounded thorough the generalization guarantee of differential privacy. Recall that from Lemma 1, under the condition in Theorem 3, we have for all $t \in [T]$, $i \in [d]$:

$$\mathbb{P}\left\{|\hat{\mathbf{g}}_t^i - \mathbf{g}_t^i| \geq \sigma\right\} \leq \beta$$

So that we have

$$\begin{aligned}\mathbb{P}\left\{\|\hat{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma\right\} &\leq \mathbb{P}\left\{\|\hat{\mathbf{g}}_t - \mathbf{g}_t\|_\infty \geq \sigma\right\} \\ &\leq d\mathbb{P}\left\{|\hat{\mathbf{g}}_t^i - \mathbf{g}_t^i| \geq \sigma\right\} \\ &\leq d\beta \end{aligned} \tag{3}$$

Now we bound the second term T_1. Recall that $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_t + \mathbf{b}_t$, where $\mathbf{b}_t$ is a noise vector with each coordinate drawn from Laplace noise $\text{Lap}(\sigma)$. In this case, we have

$$\begin{aligned}\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_t\| \geq \sqrt{d}\sigma\mu\right\} &\leq \mathbb{P}\left\{\|\mathbf{b}_t\| \geq \sqrt{d}\sigma\mu\right\} \\ &\leq \mathbb{P}\left\{\|\mathbf{b}_t\|_\infty \geq \sigma\mu\right\} \\ &\leq d\mathbb{P}\left\{|\mathbf{b}_t^i| \geq \sigma\mu\right\} \\ &= d\exp(-\mu) \end{aligned} \tag{4}$$

The second inequality comes from $\|\mathbf{b}_t\| \leq \sqrt{d}\|\mathbf{b}_t\|_\infty$. The last equality comes from the property of Laplace distribution. Combine (3) and (4), we complete the proof. □

A.3 Proof of Lemma 3

Lemma 3. SAGD *with DPG-Sparse (Algorithm 3) is* $(\frac{\sqrt{C_s \ln(2/\delta)}2G_1}{n\sigma}, \delta)$*-differentially private.*

Proof At each iteration t, the algorithm is composed of two sequential parts: DPG-Sparse (part $\mathcal{A}$) and parameter update based on estimated $\tilde{\mathbf{g}}_t$ (part $\mathcal{B}$). We first show $\mathcal{A}$ preserves $\frac{2G_1}{n\sigma}$-differential privacy. Then according to the *post-processing property* of differential privacy (Proposition 2.1 in [7]) we have $\mathcal{B} \circ \mathcal{A}$ is also $\frac{2G_1}{n\sigma}$-differentially private.

The part $\mathcal{A}$ (DPG-Sparse) is a composition of basic tools from differential privacy, the "Sparse Vector Algorithm" (Algorithm 2 in [7]) and the "Laplace Mechanism" (Definition 3.3 in [7]). In our setting, the sparse vector algorithm takes as input a sequence of T sensitivity G_1/n queries, and for each query, attempts to determine whether the value of the query, evaluated on the private dataset S_1, is above a fixed threshold $\gamma + \tau$ or below it. In our instantiation, the S_1 is the private data set, and each function corresponds to the gradient computation function $\hat{\mathbf{g}}_t$ which is of sensitivity G_1/n. By the privacy guarantee of the sparse vector algorithm, the sparse vector portion of SAGD satisfies $G_1/n\sigma$-differential privacy. The Laplace mechanism portion of SAGD satisfies $G_1/n\sigma$-differential privacy by (Theorem 3.6 in [7]). Finally, the composition of two mechanisms satisfies $\frac{2G_1}{n\sigma}$-differential privacy. For the sparse vector technique, only the query that fails the validation, corresponding to the 'above threshold', release the privacy of private dataset S_1 and pays a $\frac{2G_1}{n\sigma}$ privacy cost. Over all the iterations T, We have C_s queries fail the validation. Thus, by the advanced composition theorem (Theorem 3.20 in [7]), C_s applications of a $\frac{2G}{n\sigma}$-differentially private algorithm is $(\frac{\sqrt{C_s \ln(2/\delta)}2G_1}{n\sigma}, \delta)$-differentially private. So SAGD with DPG-Sparse is $(\frac{\sqrt{C_s \ln(2/\delta)}2G_1}{n\sigma}, \delta)$-differentially private. □

A.4 Proof of Theorem 3:

Theorem 3. *Given parameter $\sigma > 0$, let $\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_T$ be the gradients computed by DPG-Sparse over T iterations. With a budget $\frac{n\sigma^2}{2G_1^2} \leq C_s \leq \frac{n^2\sigma^4}{676\ln(1/(\sigma\beta))G_1^2}$, for $\forall\, t \in [T]$, any $\beta > 0$, and any $\mu > 0$ we have*

$$\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\} \leq d\beta + d\exp(-\mu).$$

Proof The concentration bound can be decomposed into two parts:

$$\begin{aligned}
&\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\} \\
\leq &\underbrace{\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_{s_1,t}\| \geq \sqrt{d}\sigma\mu\right\}}_{T_1\text{: empirical error}} + \underbrace{\mathbb{P}\left\{\|\hat{\mathbf{g}}_{s_1,t} - \mathbf{g}_t\| \geq \sqrt{d}\sigma\right\}}_{T_2\text{: generalization error}}
\end{aligned}$$

So that we have

$$\begin{aligned}
\mathbb{P}\left\{\|\hat{\mathbf{g}}_{s_1,t} - \mathbf{g}_t\| \geq \sqrt{d}\sigma\right\} &\leq \mathbb{P}\left\{\|\hat{\mathbf{g}}_{s_1,t} - \mathbf{g}_t\|_\infty \geq \sigma\right\} \\
&\leq d\mathbb{P}\left\{|\hat{\mathbf{g}}^i_{s_1,t} - \mathbf{g}^i_t| \geq \sigma\right\} \\
&\leq d\beta
\end{aligned} \tag{5}$$

Now we bound the second term T_1 by considering two cases, by depending on whether DPG-3 answers the query $\tilde{\mathbf{g}}_t$ by returning $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_{s_1,t} + \mathbf{v}_t$ or by returning $\tilde{\mathbf{g}}_t = \hat{\mathbf{g}}_{s_2,t}$. In the first case, we have

$$\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_{s_1,t}\| = \|\mathbf{v}_t\|$$

and

$$\begin{aligned}
\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_{s_1,t}\| \geq \sqrt{d}\sigma\mu\right\} &= \mathbb{P}\left\{\|\mathbf{v}_t\| \geq \sqrt{d}\sigma\mu\right\} \\
&\leq d\exp(-\mu)
\end{aligned}$$

The last inequality comes from the $\|\mathbf{v}_t\| \leq \sqrt{d}\|\mathbf{v}_t\|_\infty$ and properties of the Laplace distribution.

In the second case, we have

$$\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_{s_1,t}\| = \|\hat{\mathbf{g}}_{s_2,t} - \hat{\mathbf{g}}_{s_1,t}\| \leq |\gamma| + |\tau|$$

and

$$\begin{aligned}
&\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_{s_1,t}\| \geq \sqrt{d}\sigma\mu\right\} \\
=&\mathbb{P}\left\{|\gamma| + |\tau| \geq \sqrt{d}\sigma\mu\right\} \\
\leq&\mathbb{P}\left\{|\gamma| \geq \frac{2}{6}\sqrt{d}\sigma\mu\right\} + \mathbb{P}\left\{|\tau| \geq \frac{4}{6}\sqrt{d}\sigma\mu\right\} \\
=&2\exp(-\sqrt{d}\mu/6)
\end{aligned}$$

Combining these two cases, we have

$$\begin{aligned}
&\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \hat{\mathbf{g}}_{s_1,t}\| \geq \sqrt{d}\sigma\mu\right\} \\
\leq&\max\left\{\mathbb{P}\left\{\|\mathbf{v}_t\| \geq \sqrt{d}\sigma\mu\right\}, \mathbb{P}\left\{|\gamma| + |\tau| \geq \sqrt{d}\sigma\mu\right\}\right\} \\
\leq&\max\left\{d\exp(-\mu), 2\exp(-\sqrt{d}\mu/6)\right\} \\
=&d\exp(-\mu)
\end{aligned} \tag{6}$$

Combine (5) and (6), we complete the proof.

□

B CONVERGENCE ANALYSIS

In this section, we present the proof of Theorem 2, 4 , 5.

B.1 Proof of Theorem 2 and Theorem 4

Theorem 2. *Given training set S of size n, for $\nu > 0$, if $\eta_t = \eta$ which are chosen with $\eta \le \frac{\nu}{2L}$, $\sigma = 1/n^{1/3}$, and iteration number $T = n^{2/3}/\left(169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$, then* SAGD *with DPG-Lap converges to a stationary point of the population risk, i.e.,*

$$\min_{1\le t\le T} \|\nabla f(\mathbf{w}_t)\|^2 \le O\left(\frac{\rho_{n,d}\,(f(\mathbf{w}_1) - f^\star)}{n^{2/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{n^{2/3}}\right),$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

The proof of Theorem 2 consists of two parts: We first prove that the convergence rate of a gradient-based iterative algorithm is related to the gradient concentration error α and its iteration time T. Then we combine the concentration error α achieved by SAGD with DPG-Lap in Theorem 1 with the first part to complete the proof of Theorem 2.

To simplify the analysis, we first use α and ξ to denote the generalization error $\sqrt{d}\sigma(1+\mu)$ and probability $d\beta + d\exp(-\mu)$ in Theorem 1 in the following analysis. The details are presented in the following theorem.

Theorem 6. *Let $\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_T$ be the noisy gradients generated in Algorithm 1 through DPG oracle over T iterations. Then, for every $t \in [T]$, $\tilde{\mathbf{g}}_t$ satisfies*

$$\mathbb{P}\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \ge \alpha\} \le \xi$$

where the values of α and ξ are given in Section A.

With the guarantee of Theorem 6, we have the following theorem showing the convergence of SAGD.

Theorem 7. *let $\eta_t = \eta$. Further more assume that ν, β and η are chosen such that the following conditions satisfied: $\eta \le \frac{\nu}{2L}$. Under the Assumption A1 and A2, the Algorithm 1 with T iterations, $\phi_t(\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_t) = \tilde{\mathbf{g}}_t$ and $\mathbf{v}_t = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}\tilde{\mathbf{g}}_i^2$ achieves:*

$$\min_{t=1,\ldots,T} \|\nabla f(x_t)\|^2 \le (G+\nu) \times \left(\frac{f(\mathbf{w}_1) - f^\star}{\eta T} + \frac{3\alpha^2}{4\nu}\right) \tag{7}$$

with probability at least $1 - T\xi$.

Now we come to the proof of Theorem 7.

Proof Using the update rule of RMSprop, we have

$$\phi_t(\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_t) = \tilde{\mathbf{g}}_t, \text{ and}$$
$$\psi_t(\tilde{\mathbf{g}}_1, ..., \tilde{\mathbf{g}}_t) = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}\tilde{\mathbf{g}}_i^2.$$

Thus, the update of Algorithm 1 becomes:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\tilde{\mathbf{g}}_t/(\sqrt{\mathbf{v}_t} + \nu) \text{ and}$$
$$\mathbf{v}_t = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}\tilde{\mathbf{g}}_i^2.$$

Let $\Delta_t = \tilde{\mathbf{g}}_t - g_t$, we have

$$
\begin{aligned}
& f(\mathbf{w}_{t+1}) \\
\leq & f(\mathbf{w}_t) + \langle \mathbf{g}_t, \mathbf{w}_{t+1} - \mathbf{w}_t \rangle + \frac{L}{2} \left\| \mathbf{w}_{t+1} - \mathbf{w}_t \right\|^2 \\
= & f(\mathbf{w}_t) - \eta_t \left\langle \mathbf{g}_t, \tilde{\mathbf{g}}_t / (\sqrt{\mathbf{v}_t} + \nu) \right\rangle + \frac{L\eta_t^2}{2} \left\| \frac{\tilde{\mathbf{g}}_t}{(\sqrt{\mathbf{v}_t} + \nu)} \right\|^2 \\
= & f(\mathbf{w}_t) - \eta_t \left\langle \mathbf{g}_t, \frac{\mathbf{g}_t + \Delta_t}{\sqrt{\mathbf{v}_t} + \nu} \right\rangle + \frac{L\eta_t^2}{2} \left\| \frac{\mathbf{g}_t + \Delta_t}{\sqrt{\mathbf{v}_t} + \nu} \right\|^2 \\
\leq & f(\mathbf{w}_t) - \eta_t \left\langle \mathbf{g}_t, \frac{\mathbf{g}_t}{\sqrt{\mathbf{v}_t} + \nu} \right\rangle - \eta_t \left\langle \mathbf{g}_t, \frac{\Delta_t}{\sqrt{\mathbf{v}_t} + \nu} \right\rangle \\
& + L\eta_t^2 \left(\left\| \frac{\mathbf{g}_t}{\sqrt{\mathbf{v}_t} + \nu} \right\|^2 + \left\| \frac{\Delta_t}{\sqrt{\mathbf{v}_t} + \nu} \right\|^2 \right) \\
= & f(\mathbf{w}_t) - \eta_t \sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} - \eta_t \sum_{i=1}^{d} \frac{\mathbf{g}_t^i \Delta_t^i}{\sqrt{\mathbf{v}_t^i} + \nu} \\
& + L\eta_t^2 \left(\sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2}{(\sqrt{\mathbf{v}_t^i} + \nu)^2} + \sum_{i=1}^{d} \frac{[\Delta_t]_i^2}{(\sqrt{\mathbf{v}_t^i} + \nu)^2} \right) \\
\leq & f(\mathbf{w}_t) - \eta_t \sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} + \frac{\eta_t}{2} \sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2 + [\Delta_t]_i^2}{\sqrt{\mathbf{v}_t^i} + + \nu} \\
& + \frac{L\eta_t^2}{\nu} \left(\sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} + \sum_{i=1}^{d} \frac{[\Delta_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} \right) \\
= & f(\mathbf{w}_t) - \left(\eta_t - \frac{\eta_t}{2} - \frac{L\eta_t^2}{\nu} \right) \sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} \\
& + \left(\frac{\eta_t}{2} + \frac{L\eta_t^2}{\nu} \right) \sum_{i=1}^{d} \frac{[\Delta_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu}
\end{aligned}
$$

Given the parameter setting from the theorem, we see the following condition hold:

$$
\frac{L\eta_t}{\nu} \leq \frac{1}{4}.
$$

Then we obtain

$$
\begin{aligned}
f(\mathbf{w}_{t+1}) & \leq f(\mathbf{w}_t) - \frac{\eta}{4} \sum_{i=1}^{d} \frac{[\mathbf{g}_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} + \frac{3\eta}{4} \sum_{i=1}^{d} \frac{[\Delta_t]_i^2}{\sqrt{\mathbf{v}_t^i} + \nu} \\
& \leq f(\mathbf{w}_t) - \frac{\eta}{G + \nu} \|\mathbf{g}_t\|^2 + \frac{3\eta}{4\epsilon} \|\Delta_t\|^2
\end{aligned}
$$

The second inequality follows from the fact that $0 \leq \mathbf{v}_t^i \leq G^2$. Using the telescoping sum and rearranging the inequality, we obtain

$$
\frac{\eta}{G + \nu} \sum_{t=1}^{T} \|\mathbf{g}_t\|^2 \leq f(\mathbf{w}_1) - f^\star + \frac{3\eta}{4\epsilon} \sum_{t=1}^{T} \|\Delta_t\|^2
$$

Multiplying with $\frac{G+\nu}{\eta T}$ on both sides and with the guarantee in Theorem 1 that $\|\Delta_t\| \leq \alpha$ with probability at least $1 - \xi$, we obtain

$$
\min_{t=1,\ldots,T} \|\mathbf{g}_t\|^2 \leq (G + \nu) \times \left(\frac{f(\mathbf{w}_1) - f^\star}{\eta T} + \frac{3\alpha^2}{4\nu} \right)
$$

with probability at least $1 - T\xi$.

$\square$

Proof of Theorem 2:

Proof First consider the gradient concentration bound achieved by SAGD (Theorem 1 and Theorem 3) that if $\frac{2n\sigma^2}{G_1^2} \leq T \leq \frac{n^2\sigma^4}{169\ln(1/(\sigma\beta))G_1^2}$, we have

$$\begin{aligned}&\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\}\\ \leq& d\beta + d\exp(-\mu), \ \ \forall t \in [T].\end{aligned}$$

Then bring the setting in Theorem 2 that $\sigma = 1/n^{1/3}$, let $\mu = \ln(1/\beta)$ and $\beta = 1/(dn^{5/3})$, we have

$$\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\|^2 \leq d(1 + \ln d + \frac{5}{3}\ln n)^2/n^{2/3}$$

with probability at least $1 - 1/n^{5/3}$, when we set $T = n^{2/3}/\left(169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$.

Connect this result with Theorem 7, so that we have $\alpha^2 = d(1+\ln d+\frac{5}{3}\ln n)^2/n^{2/3}$ and $\xi = 1/n^{5/3}$. Bring the value α^2, ξ and $T = n^{2/3}/\left(169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$ into (7), with $\rho_{n,d} = O\left(\ln n + \ln d\right)$, we have

$$\begin{aligned}&\min_{t=1,\ldots,T}\|\nabla f(\mathbf{w}_t)\|^2\\ \leq& O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{n^{2/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{n^{2/3}}\right)\end{aligned}$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

Here we complete the proof.

$\square$

Theorem 4. *Given training set S of size n, for $\nu > 0$, if $\eta_t = \eta$ which are chosen with $\eta \leq \frac{\nu}{2L}$, noise level $\sigma = 1/n^{1/3}$, and iteration number $T = n^{2/3}/\left(676G_1^2(\ln d + \frac{7}{3}\ln n)\right)$, then* SAGD *with DPG-Sparse guarantees convergence to a stationary point of the population risk:*

$$\min_{1\leq t\leq T}\|\nabla f(\mathbf{w}_t)\|^2 \leq O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{n^{2/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{n^{2/3}}\right),$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

Proof The proof of Theorem 4 follows the proof of Theorem 2 by considering the works case $C_s = T$. $\square$

B.2 Proof of Theorem 5

Theorem 5. *Given training set S of size n, with $\nu > 0$, $\eta_t = \eta \leq \frac{\nu}{2L}$, noise level $\sigma = 1/n^{1/3}$, and epoch $T = m^{4/3}/\left(n169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$, then the mini-batch* SAGD *with DPG-Lap guarantees convergence to a stationary point of the population risk, i.e.,*

$$\min_{t=1,\ldots,T}\|\nabla f(\mathbf{w}_t)\|^2 \leq O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{(mn)^{1/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{(mn)^{1/3}}\right),$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$.

Proof When mini-batch SAGD calls **DPG** to access each batch s_k with size m for T times, we have mini-batch SAGD preserves $(\frac{\sqrt{T\ln(1/\delta)}G_1}{m\sigma}, \delta)$-deferential privacy for each batch s_k. Now consider the gradient concentration bound achieved by DPG-Lap (Theorem 1) that if $\frac{2m\sigma^2}{G_1^2} \leq T \leq \frac{m^2\sigma^4}{169\ln(1/(\sigma\beta))G_1^2}$, we have

$$\begin{aligned}&\mathbb{P}\left\{\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\| \geq \sqrt{d}\sigma(1+\mu)\right\}\\ \leq & d\beta + d\exp(-\mu), \ \ \forall t \in [T].\end{aligned}$$

Then bring the setting in Theorem 5 that $\sigma = 1/(nm)^{1/6}$, let $\mu = \ln(1/\beta)$ and $\beta = 1/(dn^{5/3})$, we have

$$\|\tilde{\mathbf{g}}_t - \mathbf{g}_t\|^2 \leq d(1 + \ln d + \frac{5}{3}\ln n)^2/n^{2/3}$$

with probability at least $1 - 1/n^{5/3}$, when we set
$T = (mn)^{1/3}/\left(169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$.

Connect this result with Theorem 7, so that we have $\alpha^2 = d(1 + \ln d + \frac{5}{3}\ln n)^2/(mn)^{1/3}$ and $\xi = 1/n^{5/3}$. Bring the value α^2, ξ and $T = (mn)^{1/3}/\left(169G_1^2(\ln d + \frac{7}{3}\ln n)\right)$ into (7), with $\rho_{n,d} = O\left(\ln n + \ln d\right)$, we have

$$\begin{aligned}&\min_{t=1,\ldots,T}\|\nabla f(\mathbf{w}_t)\|^2\\ \leq & O\left(\frac{\rho_{n,d}\left(f(\mathbf{w}_1) - f^\star\right)}{(mn)^{1/3}}\right) + O\left(\frac{d\rho_{n,d}^2}{(mn)^{1/3}}\right)\end{aligned}$$

with probability at least $1 - O\left(\frac{1}{\rho_{n,d}n}\right)$. Here we complete the proof.

□