When Does Differentially Private Learning Not Suffer in High Dimensions?

Xuechen Li*
Stanford University
lxuechen@cs.stanford.edu

Daogao Liu*
University of Washington
dgliu@uw.edu

Tatsunori Hashimoto
Stanford University
thashim@stanford.edu

Huseyin A. Inan
Microsoft Research
huseyin.inan@microsoft.com

Janardhan Kulkarni
Microsoft Research
jakul@microsoft.com

Yin Tat Lee
University of Washington
Microsoft Research
yintat@uw.edu

Abhradeep Guha Thakurta
Google Research
athakurta@google.com

Abstract

Large pretrained models can be fine-tuned with differential privacy to achieve performance approaching that of non-private models. A common theme in these results is the surprising observation that high-dimensional models can achieve favorable privacy-utility trade-offs. This seemingly contradicts known results on the model-size dependence of differentially private convex learning and raises the following research question: When does the performance of differentially private learning not degrade with increasing model size? We identify that the magnitudes of gradients projected onto subspaces is a key factor that determines performance. To precisely characterize this for private convex learning, we introduce a condition on the objective that we term *restricted Lipschitz continuity* and derive improved bounds for the excess empirical and population risks that are dimension-independent under additional conditions. We empirically show that in private fine-tuning of large language models, gradients obtained during fine-tuning are mostly controlled by a few principal components. This behavior is similar to conditions under which we obtain dimension-independent bounds in convex settings. Our theoretical and empirical results together provide a possible explanation for the recent success of large-scale private fine-tuning. Code to reproduce our results can be found at https://github.com/lxuechen/private-transformers/tree/main/examples/classification/spectral_analysis.

1 Introduction

Recent works have shown that large publicly pretrained models can be differentially privately fine-tuned on small downstream datasets with performance approaching those attained by non-private models. In particular, past works showed that pretrained BERT [DCLT18] and GPT-2 [RNSS18, RWC+19] models can be fine-tuned to perform well for text classification and generation under a privacy budget of $\varepsilon \in [2, 6]$ [LTLH21, YNB+21]. More recently, it was shown that pretrained

*Equal contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

ResNets [HZRS16] and vision-Transformers [DBK+20] can be fine-tuned to perform well for ImageNet classification under single digit privacy budgets [DBH+22, MTKC22].

One key ingredient in these successes has been the use of large pretrained models with millions to billions of parameters. These works generally highlighted the importance of two phenomena: (i) large pretrained models tend to experience good privacy-utility trade-offs when fine-tuned, and (ii) the trade-off improves with the improvement of the quality of the pretrained model (correlated with increase in size). While the power of scale and pretraining have been demonstrated numerous times in non-private deep learning [KMH+20], one common wisdom in private learning had been that large models tend to perform worse. This intuition was based on (a) results in differentially private convex optimization, most of which predicted that errors would scale proportionally with the dimension of the learning problem in the worst case, and (b) empirical observations that the noise injected to ensure privacy tends to greatly exceed the gradient in magnitude for large models [YZCL21a, Kam20].

For instance, consider the problem of differentially private convex *empirical risk minimization* (ERM). Here, we are given a dataset of n examples $\mathcal{D} = \{s_j\}_{j=1}^n \in \mathcal{S}^n$, a convex set $\mathcal{K} \subseteq \mathbb{R}^d$ (not necessarily bounded), and the goal is to perform the optimization

$$\text{minimize}_{x \in \mathcal{K}} F(x; \mathcal{D}) = \frac{1}{n} \sum_{j=1}^{n} f(x; s_j)$$

subject to differential privacy, where $f(\cdot; s)$ is convex over $\mathcal{K}$ for all $s \in \mathcal{S}$. For bounded $\mathcal{K}$, past works presented matching upper and lower bounds that are dimension-dependent under the usual Lipschitz assumption on the objective [BST14, CMS11]. These results seem to suggest that the performance of differentially private ERM algorithms inevitably degrades with increasing problem size in the worst case, and present a seeming discrepancy between recent empirical results on large-scale fine-tuning.[2]

To better understand the relation between problem size and the performance of differentially private learning, we study the following question both theoretically and empirically:

When does the performance of differentially private stochastic gradient descent (DP-SGD) not degrade with increasing problem dimension?

On the theoretical front, we show that DP-SGD can result in dimension-independent error bounds even when gradients span the entire ambient space for unconstrained optimization problems. We identify that the standard dependence on the dimension of the ambient space can be replaced by the magnitudes of gradients projected onto subspaces of varying dimensions. We formalize this in a condition that we call *restricted Lipschitz continuity* and derive refined bounds for the excess empirical and population risks for DP-SGD when loss functions obey this condition. We show that when the restricted Lipschitz coefficients decay rapidly, both the excess empirical and population risks become dimension-independent. This extends a previous work which derived rank-dependent bounds for learning generalized linear models in an unconstrained space [SSTT21].

Our theoretical results shed light on the recent success of large-scale differentially private fine-tuning. We empirically show that gradients of language models during fine-tuning are mostly controlled by a few principal components — a behavior that is similar to conditions under which we obtain dimension-independent bounds for private convex ERM. This provides a possible explanation for the observation that densely fine-tuning with DP-SGD need not necessarily experience much worse performance than sparsely fine-tuning [LTLH21]. Moreover, it suggests that DP-SGD can be adaptive to problems that are effectively low-dimensional (as characterized by restricted Lipschitz continuity) without further algorithmic intervention.

We summarize our contributions below.

(1) We introduce a condition on the objective function that we term restricted Lipschitz continuity. This condition generalizes the usual Lipschitz continuity notion and gives rise to refined analyses when magnitudes of gradients projected onto diminishing subspaces decay rapidly.

(2) Under restricted Lipschitz continuity, we present refined bounds on the excess empirical and population risks for DP-SGD when optimizing convex objectives. These bounds generalize

[2] We judiciously choose to describe the discrepancy as seeming, since the refined analysis presented in the current work suggests that the discrepancy is likely non-existent.

previous dimension-independent results [SSTT21] and are broadly applicable to cases where gradients are full rank but most coordinates only marginally influence the objective.

(3) Our theory sheds light on recent successes of large-scale differentially private fine-tuning of language models. We show that gradients obtained through fine-tuning mostly lie in a subspace spanned by a few principal components — a behavior similar to when optimizing a restricted Lipschitz continuous loss with decaying coefficients. These empirical results provide a possible explanation for the recent success of large-scale private fine-tuning.

2 Preliminaries

We define the notation used throughout this work and state the problems of differentially private empirical risk minimization and differentially private stochastic convex optimization. Finally, we give a brief recap of differentially private stochastic gradient descent, and existing dimension-dependent and dimension-independent results in the literature.

Notation & Terminology. For a positive integer $n \in \mathbb{N}_+$, define the shorthand $[n] = \{1, \ldots, n\}$. For a vector $x \in \mathbb{R}^d$, denote its ℓ_2-norm by $\|x\|_2$. Given a symmetric $M \in \mathbb{R}^{d \times d}$, let $\lambda_1(M) \geq \lambda_2(M) \geq \cdots \geq \lambda_d(M)$ denote its eigenvalues. Given a positive semidefinite matrix A, let $\|x\|_A = (x^\top A x)^{1/2}$ denote the induced Mahalanobis norm. For scalar functions f and g, we write $f \lesssim g$ if there exists a positive constant C such that $f(x) \leq C g(x)$ for all input x in the domain.

2.1 Differentially Private Empirical Risk Minimization and Stochastic Convex Optimization

Before stating the theoretical problem of interest, we recall the basic concepts of Lipschitz continuity, convexity, and approximate differential privacy.

Definition 2.1 (Lipschitz Continuity)**.** The loss function $h : \mathcal{K} \to \mathbb{R}$ is G-Lipschitz with respect to the ℓ_2 norm if for all $x, x' \in \mathcal{K}$, $|f(x) - f(x')| \leq G\|x - x'\|_2$.

Definition 2.2 (Convexity)**.** The loss function $h : \mathcal{K} \to \mathbb{R}$ is convex if $h(\alpha x + (1-\alpha)y) \leq \alpha h(x) + (1-\alpha)h(y)$, for all $\alpha \in [0, 1]$, and x, y in a convex domain $\mathcal{K}$.

Definition 2.3 (Approximate Differential Privacy [DR+14])**.** A randomized algorithm $\mathcal{M}$ is (ε, δ)-differentially private if for all neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ that differ by a single record and all sets $\mathcal{O} \subset \mathrm{range}(\mathcal{M})$, the following expression holds

$$\Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}] + \delta.$$

In this work, we study both *differentially private empirical risk minimization* (DP-ERM) for convex objectives and *differentially private stochastic convex optimization* (DP-SCO).

In DP-ERM for convex objectives, we are given a dataset $\mathcal{D} = \{s_j\}_{j \in [n]} \in \mathcal{S}^n$ of n examples. Each per-example loss $f(\cdot; s_j)$ is convex over the convex body $\mathcal{K} \subseteq \mathbb{R}^d$ and G-Lipschitz. We aim to design an (ε, δ)-DP algorithm that returns a solution x^{priv} which approximately minimizes the empirical risk $F(x; \mathcal{D}) := \frac{1}{n} \sum_{s_j \in \mathcal{D}} f(x; s_j)$. The term $\mathbb{E}_{x^{\text{priv}}} \left[F(x^{\text{priv}}; \mathcal{D}) - \min_{x \in \mathcal{K}} F(x; \mathcal{D}) \right]$ is referred to as the *excess empirical risk*.

In DP-SCO, we assume the per-example loss $f(\cdot; s)$ is convex and G-Lipschitz for all $s \in \mathcal{S}$, and we are given n examples drawn i.i.d. from some (unknown) distribution $\mathcal{P}$. The goal is to design an (ε, δ)-DP algorithm which approximately minimizes the population risk $F(x; \mathcal{P}) := \mathbb{E}_{s \sim \mathcal{P}}[f(x; s)]$. The term $\mathbb{E}_{x^{\text{priv}}} \left[F(x^{\text{priv}}; \mathcal{P}) - \min_{x \in \mathcal{K}} F(x; \mathcal{P}) \right]$ is referred to as the *excess population risk*.

2.2 Differentially Private Stochastic Gradient Descent

Differentially Private Stochastic Gradient Descent (DP-SGD) [ACG+16, SCS13] is a popular algorithm for DP convex optimization. For the theoretical analysis, we study DP-SGD for *unconstrained* optimization. To facilitate analysis, we work with the ℓ_2 regularized objective expressed as

$$F_\alpha(x; \mathcal{D}) = \frac{1}{n} \sum_{j=1}^{n} f(x; s_j) + \frac{\alpha}{2} \|x - x^{(0)}\|_2^2.$$

To optimize this objective, DP-SGD independently samples an example in each iteration and updates the solution by combining the gradient with an isotropic Gaussian whose scale is proportional to G, the Lipschitz constant of f. Algorithm 1 presents the pseudocode.

Algorithm 1: DP-SGD for optimizing regularized finite-sum objectives

Input: Initial iterate $x^{(0)}$, dataset $\mathcal{D} = \{s_j\}_{j\in[n]}$, per-step noise magnitude σ, number of updates T, learning rate η, Lipschitz constant G of f.

for $t = 1, \ldots, T$ **do**

  $j_t \sim \text{Uniform}([n])$

  $x^{(t)} = x^{(t-1)} - \eta\Big(\nabla f(x^{(t-1)}; s_{j_t}) + \alpha(x^{(t-1)} - x^{(0)}) + G \cdot \zeta_t\Big), \quad \zeta_t \sim \mathcal{N}(0, \sigma^2 I_d)$

end

Return: $\overline{x} \stackrel{\text{def}}{=} \frac{1}{T}\sum_{t=1}^{T} x^{(t)}$.

It is straightforward to show that Algorithm 1 satisfies differential privacy. The following lemma quantifies the overall privacy spending and builds on a long line of work on accounting the privacy loss of compositions [ACG+16, BBG18].

Lemma 2.4 ([KLL21])**.** *There exists constants c_1 and c_2 such that for $n \geq 10$, $\varepsilon < c_1 T/n^2$ and $\delta \in (0, \frac{1}{2}]$, DP-SGD (Algorithm 1) is (ε, δ)-DP whenever $\sigma \geq \frac{c_2\sqrt{T\log(1/\delta)}}{\varepsilon n}$.*

2.3 On the Dimension Dependence of Private Learning

Early works on bounding the excess empirical and population risks for privately optimizing convex objectives focused on a constrained optimization setup where algorithms typically project iterates back onto a fixed bounded domain after each noisy gradient update. Results in this setting suggested that risks are inevitably dimension-dependent in the worst case. For instance, it was shown that the excess empirical risk bound $\Theta(G\,\|\mathcal{K}\|_2\sqrt{d\log(1/\delta)}n^{-1}\varepsilon^{-1})$ and excess population risk bound $\Theta(G\,\|\mathcal{K}\|_2\,(n^{-1/2} + \sqrt{d\log(1/\delta)}n^{-1}\varepsilon^{-1}))$ are tight when privately optimizing convex G-Lipschitz objectives in a convex domain of diameter $\|\mathcal{K}\|_2$ [BST14]. Moreover, the lower bound instances in these works imply that such dimension-dependent lower bounds also apply when one considers the class of loss functions whose gradients are low-rank.

The body of work on unconstrained convex optimization is arguably less abundant, with the notable result that differentially private gradient descent need not suffer from a dimension-dependent penalty when learning generalized linear models with low-rank data (equivalently stated, when gradients are low-rank) [SSTT21]. Our main theoretical results (Theorems 3.3 and 3.5) extend this line of work and show that dimension-independence is achievable under weaker conditions.

3 Refined Dimension-Dependence via Restricted Lipschitz Continuity

In this section, we formally introduce the restricted Lipschitz continuity condition and derive improved bounds for the excess empirical and population risks for DP-SGD when optimizing convex objectives.

Definition 3.1 (Restricted Lipschitz Continuity)**.** We say that the loss function $h : \mathcal{K} \to \mathbb{R}$ is restricted Lipschitz continuous with coefficients $G_0, \ldots, G_d$, if for all $k \in [d]$, there exists an orthogonal projection matrix P_k with rank k such that

$$\|(I - P_k)\nabla h(x)\|_2 \leq G_k,$$

for all $x \in \mathcal{K}$ and all subgradients $\nabla h(x) \in \partial h(x)$.

Note that any G-Lipschitz function is also trivially restricted Lipschitz continuous with coefficients $G = G_0 = G_1 = \cdots = G_d$, since orthogonal projections never increase the ℓ_2-norm of a vector (generally, it is easy to see that $G = G_0 \geq G_1 \geq G_2 \geq \cdots \geq G_d = 0$). On the other hand, we expect that a function which exhibits little growth in subspaces of dimension k to have a restricted Lipschitz coefficient G_{d-k} of almost 0.

Our bounds on DP convex optimization will characterize errors through the use of restricted Lipschitz coefficients. We now summarize the main assumptions before presenting the core theoretical results.

Assumption 3.2. *The per-example loss function $f(\cdot; s)$ is convex and G-Lipschitz continuous for all $s \in \mathcal{S}$. The average loss $F(\cdot; \mathcal{D})$ is restricted Lipschitz continuous with coefficients $\{G_k\}_{k=1}^d$.*

3.1 Bounds for Excess Empirical Loss

We present the main theoretical result on DP-ERM for convex objectives. The result consists of two pieces: Equation (1) is a general bound that is applicable to any sequence of restricted Lipschitz coefficients; Equation (2) specializes the previous bound when the sequence of coefficients decays rapidly and demonstrates dimension-independent error scaling.

Theorem 3.3 (Excess Empirical Loss)**.** *Let $\delta \in (0, \frac{1}{2}]$, and $\varepsilon \in (0, 10]$. Under Assumption 3.2, for any positive integer $k \in [d]$, setting $T = \Theta(n^2 + d\log^2 d)$, $\sigma = \Theta\left(\frac{\sqrt{T\log(1/\delta)}}{n\varepsilon}\right)$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2 \cdot k\sigma^2}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$, where $S = \lfloor \log(d/k) \rfloor + 1$, DP-SGD (Algorithm 1) is (ε, δ)-DP, and*

$$\mathbb{E}\left[F(\overline{x}; \mathcal{D}) - \min_x F(x; \mathcal{D})\right] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}, \tag{1}$$

where $\|x^{(0)} - \operatorname{argmin}_x F(x; \mathcal{D})\|_2 \le D$, $\overline{x}$ is the (random) output of DP-SGD (Algorithm 1), and the expectation is over the randomness of $\overline{x}$. Moreover, if for some $c > 1/2$, we have $G_k \le G_0 k^{-c}$ for all $k \in [d]$, and in addition $n \ge \varepsilon^{-1}\sqrt{\log(1/\delta)}$, minimizing the right hand side of (1) with respect to k yields

$$\mathbb{E}\left[F(\overline{x}; \mathcal{D}) - \min_x F(x; \mathcal{D})\right] \lesssim G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}. \tag{2}$$

We include a sketch of the proof techniques in Section 3.3 and defer the full proof to Appendix A.

Remark 3.4. Consider DP-ERM for learning generalized linear models with convex and Lipschitz losses. When the (empirical) data covariance is of rank $r < d$, the span of gradients $\operatorname{span}(\{\nabla_x F(x)\})$ is also of rank r. Thus, the average loss is restricted Lipschitz continuous with coefficients where $G_{r'} = 0$ for all $r' > r$. Setting $k = r$ in (1) yields the excess empirical risk bound of order $O\left(G_0 D\sqrt{r \cdot \log(1/\delta)}\varepsilon^{-1}n^{-1}\right)$. This recovers the previous dimension-independent result [SSTT21].

The restricted Lipschitz continuity condition can be viewed as a generalized notion of rank. The result captured in (2) suggests that the empirical loss achieved by DP-SGD does not depend on the problem dimension if the sequence of restricted Lipschitz coefficients decays rapidly. We leverage these insights to build intuition for understanding private fine-tuning of language model in Section 4.

3.2 Bounds for Excess Population Loss

For DP-SCO, we make use of the *stability* of DP-SGD to bound its generalization error [BE02], following previous works [BFTT19, BFGT20, SSTT21]. The bound on the excess population loss follows from combining the bounds on the excess empirical risk and the generalization error.

Theorem 3.5 (Excess Population Loss)**.** *Let $\delta \in (0, \frac{1}{2}]$, and $\varepsilon \in (0, 10]$. Under Assumption 3.2, for any positive integer $k \in [d]$, by setting $T = \Theta(n^2 + d\log^2 d)$, $\sigma = \Theta\left(\frac{\sqrt{T\log(1/\delta)}}{n\varepsilon}\right)$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2 (T/n + k\sigma^2)}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$, where $S = \lfloor \log(d/k) \rfloor + 1$, DP-SGD (Algorithm 1) is (ε, δ)-DP, and*

$$\mathbb{E}\left[F(\overline{x}; \mathcal{P}) - \min_x F(x; \mathcal{P})\right] \lesssim \frac{G_0 D}{\sqrt{n}} + \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2},$$

where $\|x^{(0)} - \operatorname{argmin}_x F(x;\mathcal{P})\|_2 \le D$, $\overline{x}$ *is the (random) output of DP-SGD (Algorithm 1), and the expectation is over the randomness of* $\overline{x}$.

Moreover, if for some $c > 1/2$, *we have* $G_k \le G_0 k^{-c}$ *for all* $k \in [d]$, *and in addition* $n > \varepsilon^{-1}\sqrt{\log(1/\delta)}$, *minimizing the above bound with respect to* k *yields*

$$\mathbb{E}\left[F(\overline{x};\mathcal{P}) - \min_x F(x;\mathcal{P})\right] \lesssim \frac{G_0 D}{\sqrt{n}} + G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

Remark 3.6. Our result on DP-SCO also recovers the DP-SCO rank-dependent result for learning generalized linear models with convex and Lipschitz losses [SSTT21].

Remark 3.7. When $c > 1/2$, $\varepsilon = \Theta(1)$ and $\delta = 1/\mathrm{poly}(n)$, the population loss matches the (non-private) informational-theoretical lower bound $\Omega(G_0 D/\sqrt{n})$ [AWBR09].

Remark 3.8. Our results on DP-ERM and DP-SCO naturally generalize to (full-batch) DP-GD.

3.3 Overview of Proof Techniques

The privacy guarantees in Theorems 3.3 and 3.5 follow from Lemma 2.4. It suffices to prove the utility guarantees. We give an outline of the main proof technique and defer full proofs to the supplementary. The following is a sketch of the core technique for deriving (2) in Theorem 3.3.

By convexity, the error term of SGD is upper bounded as follows

$$f_j(x^{(t)}) - f_j(x^*) \le \nabla f_j(x^{(t)})^\top (x^{(t)} - x^*), \tag{3}$$

where $j \in [n]$ is the random index sampled at iteration t. By definition of G_k, we know that there is a k dimensional subspace U such that the gradient component orthogonal to U is small when G_k is small. Naïvely, one decomposes the gradient $\nabla f_j(x^{(t)}) = g_1 + g_2$, where $g_1 \in U$ and $g_2 \in U^\perp$, and separately bounds the two terms $g_1^\top(x^{(t)} - x^*)$ and $g_2^\top(x^{(t)} - x^*)$. Since g_1 lies in a k dimensional subspace, one can follow existing arguments on DP-SGD to bound $g_1^\top(x^{(t)} - x^*)$. Unfortunately, this argument does not lead to dimension-independence. Although $\|g_2\|_2 \le G_k$ (which can be small for large k), the term $\|x^{(t)} - x^*\|_2$ can be as large as $\Omega(\sqrt{d})$ with high probability due to the isotropic Gaussian noise injected in DP-SGD. Therefore, the naïve upper bound on $|g_2^\top(x^{(t)} - x)|$ can be as large $O(\sqrt{d}G_k)$.

Our key idea is to partition the whole space $\mathbb{R}^d$ into $\lfloor \log(d/k) \rfloor + 2$ orthogonal subspaces, expressing the error term $\nabla f_j(x^{(t)})^\top (x^{(t)} - x^*)$ as the sum of individual terms each of which corresponds to a projection to a particular subspace. Fix the positive integer $k \le d$, and consider the following subspaces: Let $U_0 = \mathrm{range}(P_k)$, U_s be the subspace orthogonal to all previous subspaces such that $\bigoplus_{i=0}^{s} U_i \supseteq \mathrm{range}(P_{2^s k})$ for $s = 1, 2, \cdots, \lfloor \log(d/k) \rfloor$, and U_S be the subspace such that the orthogonal direct sum of all subspaces $\{U_i\}_{i=0}^{S}$ is $\mathbb{R}^d$ where $S = \lfloor \log(d/k) \rfloor + 1$. Here, P_i is the orthogonal projection matrix with rank i promised by G_i in Assumption 3.2. Let Q_s be the orthogonal projection to the subspace U_s. Observe that $\mathrm{rank}(Q_s) \le 2^s k$ and $\|Q_s \nabla F(x)\|_2 \le G_{2^{s-1}k}$ for all x and all $s \ge 1$. Rewriting the right hand side of (3) with this decomposition yields

$$f_j(x^{(t)}) - f_j(x^*) \le \left(Q_0 \nabla f_j(x^{(t)}) + \sum_s Q_s \nabla f_j(x^{(t)})\right)^\top (x^{(t)} - x^*).$$

On the one hand, if G_k decays rapidly, $\|\mathbb{E}_j[Q_s \nabla f_j]\|_2$ will be small for large s. On the other hand, we expect $\|Q_s(x^{(t)} - x^*)\|_2$ to be small for small s where Q_s is an orthogonal projection onto a small subspace. Thus, for each s, $\nabla f_j(x^{(t)})^\top Q_s(x^{(t)} - x^*)$ is small either due to a small gradient (small $Q_s \nabla f_j$ in expectation over the random index) or small noise (small $Q_s(x^{(t)} - x^*)$), since noise injected in DP-SGD is isotropic. More formally, in Lemma A.1, we show that for any projection matrix Q with rank r, $\|Q(x^{(t)} - x^{(0)})\|_2$ can be upper bounded by a term that depends only on r (rather than d).

4 Numerical Experiments

The aim of this section is twofold. In Section 4.1, we study a synthetic example that matches our theoretical assumptions and show that DP-SGD attains dimension-independent empirical and

population loss when the sequence of restricted Lipschitz coefficients decays rapidly—even when gradients span the entire ambient space. In Section 4.2, we study a stylized example of privately fine-tuning large language models. Building on the previous theory, we provide insights as to why dense fine-tuning can yield good performance.

4.1 Synthetic Example: Estimating the Generalized Geometric Median

We privately estimate the geometric median which minimizes the average Mahalanobis distance. Specifically, let $x_i \in \mathbb{R}^d$ for $i \in [n]$ be feature vectors drawn i.i.d. from some distribution P_x, each of which is treated as an individual record. Denote the entire dataset as $\mathcal{D} = \{x_i\}_{i=1}^n$. Subject to differential privacy, we perform the following optimization

$$\min_{x \in \mathbb{R}^d} F_\alpha(x) = \frac{1}{n}\sum_{i=1}^n f_i(x) + \frac{\alpha}{2}\left\|x - x^{(0)}\right\|_2^2 = \frac{1}{n}\sum_{i=1}^n \|x - x_i\|_A + \frac{\alpha}{2}\left\|x - x^{(0)}\right\|_2^2, \tag{4}$$

where we adopt the shorthand $f_i(x) = f(x; x_i) = \|x - x_i\|_A$. When $A = I_d$ and $\alpha = 0$ (without the regularization term), the problem reduces to estimating the usual geometric median (commonly known as center of mass).

For this example, individual gradients are bounded since $\|\nabla f_i(x)\|_2 = \|A(x - x_i)/\|x - x_i\|_A\|_2 \leq \lambda_1(A^{1/2}) = G_0$. More generally, the restricted Lipschitz coefficients of $F(x)$ are the eigenvalues of $A^{1/2}$, since

$$\|Q_k \nabla F(x)\|_2 = \left\| Q_k A^{1/2} \frac{1}{n}\sum_{i=1}^n \frac{A^{1/2}(x - x_i)}{\|x - x_i\|_A} \right\|_2 \leq \|Q_k A^{1/2}\|_{\text{op}} = \lambda_{k+1}(A^{1/2}) = G_k,$$

where $Q_k = I - P_k$ is chosen to be the rank $(d - k)$ orthogonal projection matrix that projects onto the subspace spanned by the bottom $(d - k)$ eigenvectors of $A^{1/2}$.

To verify our theory, we study the optimization and generalization performance of DP-SGD for minimizing (4) under Mahalanobis distances induced by different A as the problem dimension grows. The optimization performance is measured by the final training error, and the generalization performance is measured by the population quantity $\mathbb{E}_{x \sim P_x, \overline{x}}[\|\overline{x} - x\|_A]$, where $\overline{x}$ denotes the random output of DP-SGD. We study the dimension scaling behavior for A being one of

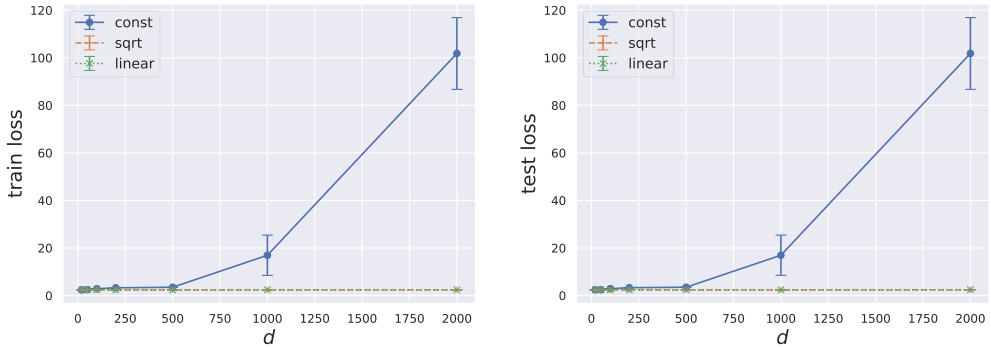
$$A_{\text{const}} = \text{diag}(1, \ldots, 1), \quad A_{\text{sqrt}} = \text{diag}(1, 1/\sqrt{2}, \ldots, 1/\sqrt{d}), \quad A_{\text{linear}} = \text{diag}(1, 1/2, \ldots, 1/d),$$

where diag : $\mathbb{R}^d \to \mathbb{R}^{d \times d}$ maps vectors onto square matrices with inputs on the diagonal. In all cases, the span of gradients span$(\{\nabla F(x)\})$ is the ambient space $\mathbb{R}^d$, since A is of full rank. To ensure the distance from the initial iterate $\beta^{(0)} = 0$ to the optimum is the same for problem instances of different dimensions, we let feature vectors $\{x_i\}_{i=1}^n$ take zero values in any dimension $k > d_{\text{min}}$, where d_{min} is the dimension of the smallest problem in our experiments. Our theoretical bounds suggest that when the sequence of restricted Lipschitz coefficients is constant (when $A = A_{\text{const}}$), the excess empirical loss grows with the problem dimension, whereas when the sequence of kth-Lipschitz constants rapidly decays with k (when $A = A_{\text{sqrt}}$ or $A = A_{\text{linear}}$), the excess empirical loss does not grow beyond a certain problem dimension. Figure 1 empirically captures this phenomenon. We include additional experimental setup details in Appendix C.

4.2 Why Does Dense Fine-Tuning Work Well for Pretrained Language Models?

Stated informally, our bounds in Theorem 3.5 imply that DP-SGD obtains dimension-independent errors if gradients approximately reside in a subspace much smaller than the ambient space. Inspired by these results for the convex case, we now turn to study dense language model fine-tuning [LTLH21] and provide a possible explanation for their recent intriguing success — fine-tuning gigantic parameter vectors frequently results in moderate performance drops compared to non-private learning.

In the following, we present evidence that gradients obtained through fine-tuning mostly lie in a small subspace. We design subsequent experiments to work under a simplified setup. Specifically, we fine-tune DistilRoBERTa [SDCW19, LOG+19] under $\varepsilon = 8$ and $\delta = 1/n^{1.1}$ for sentiment classification on the SST-2 dataset [SPW+13]. We reformulate the label prediction problem as templated text prediction [LTLH21], and fine-tune only the query and value matrices in attention layers.



(a) empirical loss

(b) (estimated) population loss

Figure 1: The empirical and population losses grow with increasing problem dimension when the sequence of restricted Lipschitz coefficients remain constant. On the other hand, these losses remain almost constant when the sequence of restricted Lipschitz coefficients decays rapidly. Error bars represent one standard deviation over five runs of DP-SGD with the same hyperparameters which were tuned on separate validation data. For the same A, the optimal training error $\min_{x \in \mathbb{R}^d} F(x)$ is the same for problem instances with different dimensions (thus errors do not scale if learning was non-private). Each training run was performed with $\varepsilon = 2$, $\delta = 10^{-6}$, and $n = 10000$.

We focus on fine-tuning these specific parameter matrices due to the success of LoRA for non-private learning [HSW+21] which focuses on adapting the attention layers. Unlike LoRA, we fine-tune all parameters in these matrices rather than focusing on low-rank updates. This gives a setup that is lightweight enough to run spectral analyses computationally tractably but retains enough parameters (≈ 7 million) such that a problem of similar scale outside of fine-tuning results in substantial losses in utility.[3] For our setup, DP-SGD obtains a dev set accuracy approximately of 90% and 92%, privately and non-privately, respectively. These numbers are similar to previous results obtained with the same pretrained model [YNB+21, LTLH21]. We include the full experimental protocol and additional results in Appendix D.

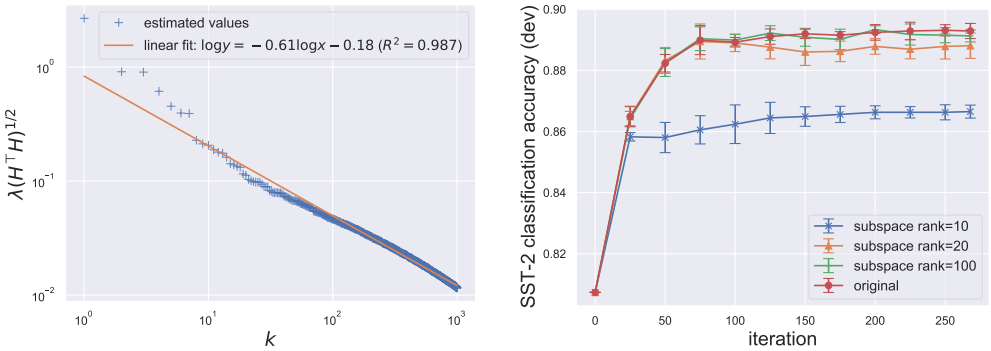
To provide evidence for the small subspace hypothesis, we sample gradients during fine-tuning and study their principal components. Specifically, we "over-train" by privately fine-tuning for $r = 2 \times 10^3$ updates and collect all the non-privatized average clipped gradients along the optimization trajectory. While fine-tuning for 200 and 2k updates have similar final dev set performance under our hyperparameters, the increased number of steps allows us to collect more gradients around the converged solution. This yields a gradient matrix $H \in \mathbb{R}^{r \times p}$, where $p \approx 7 \times 10^6$ is the size of the parameter vector. We perform PCA for H with the orthogonal iteration algorithm [Dem97] and visualize the set of estimated singular values $\sigma_i(H) = \lambda_i(H^\top H)^{1/2}$ in terms of both (i) the density estimate, and (ii) their relation with the rank. Figure 2 (a) shows the top 1000 singular values sorted and plotted against their rank k and the least squares fit on log-transformed inputs and outputs. The plot displays few large singular values which suggests that gradients are controlled through only a few principal directions. The linear fit suggests that singular values decay rapidly (at a rate of approximately $k^{-0.6}$).

To study the effects that different principal components have on fine-tuning performance, we further perform the following re-training experiment. Given the principal components, we privately re-fine-tune with gradients projected onto the top $k \in \{10, 20, 100\}$ components. Note that this projection applies only to the (non-privatized) average clipped gradients and the isotropic DP noise is still applied to all dimensions. Figure 2 (b) shows that the original performance can be attained by optimizing within a subspace of only dimension $k = 100$, suggesting that most of the dimensions of the 7 million parameter vector encode a limited learning signal.

While these empirical results present encouraging insights for the dimension-independent performance of fine-tuning, we acknowledge that this is not a complete validation of the restricted Lipschitz

[3] For instance, an off-the-shelf ResNet image classifier has 10 to 20+ million parameters. A plethora of works report large performance drops when training these models from scratch [YZCL21b, LWAFF21, DBH+22].

continuity condition and fast decay of coefficients (even locally near the optimum). We leave a more thorough analysis with additional model classes and fine-tuning tasks to future work.



(a) singular values decay with rank

(b) retrain in fixed subspace

Figure 2: Gradients obtained through fine-tuning are controlled by a few principal components. *Left:* Singular values decay rapidly with their rank. *Right:* Retraining with gradients projected onto a subspace (but noise is not projected!) is sufficient to recover original performance.

5 Related Work

DP-ERM and DP-SCO are arguably the most well-studied areas of differential privacy [CMS11, KST12, BST14, SCS13, WYX17, FTS17, BFTT19, MRTZ17, ZZMW17, WLK+17, FKT20, INS+19, BFGT20, STT20, LL21, AFKT21, BGN21, GTU22, GLL22]. Tight dependence on the number of model parameters and the number of samples is known for both DP-ERM [BST14] and DP-SCO [BFTT19]. In particular, for the error on general convex losses, an explicit polynomial dependence on the number of optimization variables is necessary. However, it is shown that if gradients lie in a fixed low-rank subspace M, the dependence on dimension d can be replaced by $\mathsf{rank}(M)$ which can be significantly smaller [JT14, STT20]. We extend this line of work to show that under a weaker assumption (restricted Lipschitz continuity with decaying coefficients) one can obtain analogous error guarantees that are independent of d, but do not require the gradients of the loss to strictly lie in any fixed low-rank subspace M. As a consequence, our results provide a plausible explanation for the empirical observation that dense fine-tuning can be effective and that fine-tuning a larger model under DP can generally be more advantageous in terms of utility than fine-tuning a smaller model [LTLH21, YNB+21]. A concurrent work shows that the standard dimension dependence of DP-SGD can be replaced by a dependence on the trace of the Hessian assuming the latter quantity is uniformly bounded [MMZ22].

A complementary line of work designed variants of DP-SGD that either explicitly or implicitly control the subspace in which gradients are allowed to reside [AGM+21, LVS+21, ADF+21, KDRT21, YZCL21b]. They demonstrated improved dependence of the error on the dimension if the true gradients lie in a "near" low-rank subspace. Our results are incomparable to this line of work because of two reasons: (i) Our algorithm is vanilla DP-SGD and does not track the gradient subspace either explicitly or implicitly, and hence does not change the optimization landscape. Our improved dependence on dimensions is an artifact of the analysis. (ii) Our analytical results do not need the existence of any public data to obtain tighter dependence on dimensions. All prior works mentioned above need the existence of public data to demonstrate any improvement.

On the empirical front, past works have observed that for image classification tasks, gradients of ConvNets converge to a small subspace spanned by the top directions of the Hessian. In addition, this span remains stable for long periods of time during training [GARD18]. While insightful, this line of work does not look at language model fine-tuning. Another line of work measures for language model fine-tuning the *intrinsic dimension*—the minimum dimension such that optimizing in a randomly sampled subspace of such dimension approximately recovers the original performance [LFLY18, AZG20]. We note that a small intrinsic dimension likely suggests that gradients are approximately low rank. Yet, this statement should not be interpreted as a strict implication, since the notion of

intrinsic dimension is at best vaguely defined (e.g., there's no explicit failure probability threshold over the randomly sampled subspace in the original statement), and the definition involves not a fixed subspace but rather a randomly sampled one.

6 Conclusion

We made an attempt to reconcile two seemingly conflicting results: (i) in private convex optimization, errors are predicted to scale proportionally with the dimension of the learning problem; while (ii) in empirical works on large-scale private fine-tuning through DP-SGD, privacy-utility trade-offs become better with increasing model size. We introduced the notion of restricted Lipschitz continuity, with which we gave refined analyses of DP-SGD for DP-ERM and DP-SCO. When the magnitudes of gradients projected onto diminishing subspaces decay rapidly, our analysis showed that excess empirical and population losses of DP-SGD are independent of the model dimension. Through preliminary experiments, we gave empirical evidence that gradients of large pretrained language models obtained through fine-tuning mostly lie in the subspace spanned by a few principal components. Our theoretical and empirical results together give a possible explanation for recent successes in large-scale differentially private fine-tuning.

Given our improved upper bounds on the excess empirical and population risks for differentially private convex learning, it is instructive to ask if such bounds are tight in the mini-max sense. We leave answering this inquiry to future work. In addition, while we have presented encouraging empirical evidence that fine-tuning gradients mostly lie in a small subspace, more work is required to study the robustness of this phenomenon with respect to the model class and fine-tuning problem. Overall, we hope that our work leads to more research on understanding conditions under which DP learning does not degrade with increasing problem size, and more generally, how theory can inform and explain the practical successes of differentially private deep learning.

Acknowledgements

XL is supported by a Stanford Graduate Fellowship.

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.

[ADF+21] Hilal Asi, John Duchi, Alireza Fallah, Omid Javidbakht, and Kunal Talwar. Private adaptive gradient methods for convex optimization. In *International Conference on Machine Learning*, pages 383–392. PMLR, 2021.

[AFKT21] Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: Optimal rates in ℓ_1 geometry. *arXiv preprint arXiv:2103.01516*, 2021.

[AGM+21] Ehsan Amid, Arun Ganesh, Rajiv Mathews, Swaroop Ramaswamy, Shuang Song, Thomas Steinke, Vinith M. Suriyakumar, Om Thakkar, and Abhradeep Thakurta. Public data-assisted mirror descent for private model training. *CoRR*, abs/2112.00193, 2021.

[AWBR09] Alekh Agarwal, Martin J Wainwright, Peter Bartlett, and Pradeep Ravikumar. Information-theoretic lower bounds on the oracle complexity of convex optimization. *Advances in Neural Information Processing Systems*, 22, 2009.

[AZG20] Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

[BBG18] Borja Balle, Gilles Barthe, and Marco Gaboardi. Privacy amplification by subsampling: Tight analyses via couplings and divergences. *Advances in Neural Information Processing Systems*, 31, 2018.

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[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*, 33:4381–4391, 2020.

[BFTT19] Raef Bassily, Vitaly Feldman, Kunal Talwar, and Abhradeep Thakurta. Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, pages 11279–11288, 2019.

[BGN21] Raef Bassily, Cristóbal Guzmán, and Anupama Nandi. Non-euclidean differentially private stochastic convex optimization. In *Conference on Learning Theory*, pages 474–499. PMLR, 2021.

[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.

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

[DBH+22] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

[DBK+20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[DCLT18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Dem97] James W Demmel. *Applied numerical linear algebra*. SIAM, 1997.

[DR+14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[FKT20] Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private stochastic convex optimization: optimal rates in linear time. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 439–449, 2020.

[FTS17] Kazuto Fukuchi, Quang Khai Tran, and Jun Sakuma. Differentially private empirical risk minimization with input perturbation. In *International Conference on Discovery Science*, pages 82–90. Springer, 2017.

[GARD18] Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

[GKX19] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

[GLL22] Sivakanth Gopi, Yin Tat Lee, and Daogao Liu. Private convex optimization via exponential mechanism. *arXiv preprint arXiv:2203.00263*, 2022.

[GTU22] Arun Ganesh, Abhradeep Thakurta, and Jalaj Upadhyay. Langevin diffusion: An almost universal algorithm for private euclidean (convex) optimization. *arXiv preprint arXiv:2204.01585*, 2022.

[GWG19] Diego Granziol, Xingchen Wan, and Timur Garipov. Deep curvature suite. *arXiv preprint arXiv:1912.09656*, 2019.

[HSW+21] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 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 *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[JT14] Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484, 2014.

[Kam20] Gautam Kamath. Lecture 14 — Private ML and Stats: Modern ML. http://www.gautamkamath.com/CS860notes/lec14.pdf, 2020.

[KDRT21] Peter Kairouz, Monica Ribero Diaz, Keith Rush, and Abhradeep Thakurta. (nearly) dimension independent private erm with adagrad rates via publicly estimated subspaces. In *COLT*, 2021.

[KLL21] Janardhan Kulkarni, Yin Tat Lee, and Daogao Liu. Private non-smooth empirical risk minimization and stochastic convex optimization in subquadratic steps. *arXiv preprint arXiv:2103.15352*, 2021.

[KMH+20] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[KST12] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1, 2012.

[LFLY18] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[LL21] Daogao Liu and Zhou Lu. Curse of dimensionality in unconstrained private convex erm. *arXiv preprint arXiv:2105.13637*, 2021.

[LOG+19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[LTLH21] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

[LVS+21] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Zhiwei Steven Wu. Leveraging public data for practical private query release. *arXiv preprint arXiv:2102.08598*, 2021.

[LWAFF21] Zelun Luo, Daniel J Wu, Ehsan Adeli, and Li Fei-Fei. Scalable differential privacy with sparse network finetuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5059–5068, 2021.

[MMZ22] Yi-An Ma, Teodor Vanislavov Marinov, and Tong Zhang. Dimension independent generalization of dp-sgd for overparameterized smooth convex optimization. *arXiv preprint arXiv:2206.01836*, 2022.

[MRTZ17] H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

[MTKC22] Harsh Mehta, Abhradeep Thakurta, Alexey Kurakin, and Ashok Cutkosky. Large scale transfer learning for differentially private image classification. *arXiv preprint arXiv:2205.02973*, 2022.

[NDR17] Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. The e2e dataset: New challenges for end-to-end generation. *arXiv preprint arXiv:1706.09254*, 2017.

[RNSS18] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[RWC+19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[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.

[SDCW19] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[SPW+13] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[SSTT21] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics*, pages 2638–2646. PMLR, 2021.

[STT20] Shuang Song, Om Thakkar, and Abhradeep Thakurta. Characterizing private clipped gradient descent on convex generalized linear problems. *arXiv preprint arXiv:2006.06783*, 2020.

[WLK+17] Xi Wu, Fengan Li, Arun Kumar, Kamalika Chaudhuri, Somesh Jha, and Jeffrey F. Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In Semih Salihoglu, Wenchao Zhou, Rada Chirkova, Jun Yang, and Dan Suciu, editors, *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD*, 2017.

[WYX17] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[YNB+21] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

[YZCL21a] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. *arXiv preprint arXiv:2102.12677*, 2021.

[YZCL21b] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[ZZMW17] Jiaqi Zhang, Kai Zheng, Wenlong Mou, and Liwei Wang. Efficient private erm for smooth objectives. In *IJCAI*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work presents theory which characterizes the performance of existing algorithms. This work would not lead to any direct societal impact.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Proof of Theorem 3.3

Before bounding the utility of DP-SGD, we first bound $x^{(t)} - x^{(0)}$ in expectation. For simplicity, we write $f_j(\cdot)$ for $f(\cdot; s_j)$ and $F(\cdot)$ for $F(\cdot; \mathcal{D})$ when there is no ambiguity.

Lemma A.1. *Suppose Assumption 3.2 holds. Let Q be any orthogonal projection matrix with rank r and suppose that $\|Q\nabla f(x; s)\|_2 \leq G_Q$ for all $x \in \mathbb{R}^d$ and $s \in \mathcal{S}$. If we set $\eta \leq \frac{1}{2\alpha}$ in DP-SGD, then for all $t > 0$, we have*

$$\mathbb{E}\,\|Q(x^{(t)} - x^{(0)})\|_2^2 \leq \frac{4G_Q^2}{\alpha^2} + \frac{2\eta G_0^2}{\alpha}(1 + r\sigma^2).$$

Proof of Lemma A.1. By the assumption, we know $\|Q\nabla F(x)\|_2 \leq G_Q$ and $\|\nabla F(x)\|_2 \leq G_0$. Let $z^{(t)} = x^{(t)} - x^{(0)}$. Note that

$$\begin{aligned} z^{(t+1)} &= x^{(t+1)} - x^{(0)} \\ &= x^{(t)} - \eta\Big(\nabla f_j(x^{(t)}) + \alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\Big) - x^{(0)} \\ &= (1 - \alpha\eta)z^{(t)} - \eta(\nabla f_j(x^{(t)}) + G_0 \cdot \zeta), \end{aligned}$$

where $\zeta \sim \mathcal{N}(0, \sigma^2 I_d)$ is the isotropic Gaussian noise drawn in $(t+1)$th step. For simplicity, we use $\tilde{\nabla} f_j(x^{(t)})$ to denote the noisy subgradient $\nabla f_j(x^{(t)}) + G_0 \cdot \zeta$. Hence, we have

$$\|Qz^{(t+1)}\|_2^2 = (1 - \alpha\eta)^2\|Qz^{(t)}\|_2^2 - 2\eta(1 - \alpha\eta)(Qz^{(t)})^\top Q\tilde{\nabla} f_j(x^{(t)}) + \eta^2\|Q\tilde{\nabla} f_j(x^{(t)})\|_2^2.$$

Taking expectation over the random sample f_j and random Gaussian noise ζ, we have

$$\begin{aligned} \mathbb{E}\,\|Qz^{(t+1)}\|_2^2 =& (1 - \alpha\eta)^2\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 - 2\eta(1 - \alpha\eta) \cdot \mathbb{E}\left((Qz^{(t)})^\top(Q\nabla F(x^{(t)}))\right) \\ &+ \eta^2\,\mathbb{E}\,\|Q\tilde{\nabla} f_j(x^{(t)})\|_2^2 \\ \leq& (1 - \alpha\eta)\,\mathbb{E}[\|Qz^{(t)}\|_2^2] + 2\eta G_Q \cdot \mathbb{E}[\|Qz^{(t)}\|_2] + \eta^2 G_0^2(1 + r\sigma^2), \end{aligned}$$

where we used the fact that ζ has zero mean, $\|\nabla f_j(x^{(t)})\|_2 \leq G_0$, $\|Q\nabla F(x^{(t)})\|_2 \leq G_Q$ and $\eta \leq \frac{1}{2\alpha}$. Further simplifying and taking expectation over all iterations, we have

$$\begin{aligned} \mathbb{E}\,\|Qz^{(t+1)}\|_2^2 &\leq (1 - \alpha\eta)\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 + 2\eta(\frac{\alpha}{4}\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 + \frac{1}{\alpha}G_Q^2) + \eta^2 G_0^2(1 + r\sigma^2) \\ &\leq (1 - \frac{\alpha\eta}{2})\,\mathbb{E}\,\|Qz^{(t)}\|_2^2 + \frac{2\eta}{\alpha}G_Q^2 + \eta^2 G_0^2(1 + r\sigma^2). \end{aligned} \tag{5}$$

Using that $z^{(0)} = 0$ we know $\mathbb{E}\,\|Qz^{(0)}\|_2^2 = 0$. Solving the recursion (Equation (5)) gives

$$\mathbb{E}\,\|Qz^{(t)}\|_2^2 \leq \frac{2}{\alpha\eta}(\frac{2\eta}{\alpha}G_Q^2 + \eta^2 G_0^2(1 + r\sigma^2))$$

for all t. This proves the result. □

Now, we are ready to bound the utility. The proof builds off the standard mirror descent proof.

Lemma A.2. *Let $\delta \in (0, \frac{1}{2}]$, and $\varepsilon \in (0, 10]$. Under Assumption 3.2, let $x^{(0)}$ be the initial iterate and $x^* = \operatorname{argmin}_x F(x)$. Suppose $\|x^{(0)} - x^*\|_2 \leq D$. For any positive integer $k \leq d$, setting $T = \Theta(n^2 + d\log^2 d)$, $\sigma = \Theta\left(\frac{\sqrt{T\log(1/\delta)}}{n\varepsilon}\right)$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2 \cdot k\sigma^2}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$, we have*

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2},$$

where $S = \lfloor \log(d/k) \rfloor + 1$, $\overline{x}$ is the output of DP-SGD, and the expectation is under the randomness of DP-SGD.

Moreover, if $G_k \le G_0 k^{-c}$ for each k for some $c > 1/2$, and in addition $n > \varepsilon^{-1}\sqrt{\log(1/\delta)}$, picking the best $k \in [d]$ for the bound above gives

$$\mathbb{E}[F(\overline{x};\mathcal{D}) - F(x^*;\mathcal{D})] \lesssim G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

Proof of Lemma A.2. Fix a positive integer $1 \le k \le d$. Our key idea is to split the whole space $\mathbb{R}^d$ into different subspaces. We define the following set of subspaces:

- $U_0 = \text{range}(P_k)$.
- For $s = 1, 2, \ldots, \lfloor \log(d/k) \rfloor$, let $U_s \subseteq \text{range}(P_{2^s k})$ be a subspace with maximal dimension such that $U_s \perp U_i$ for all $i = 0, \ldots, s-1$.
- For $S = \lfloor \log(d/k) \rfloor + 1$, let $U_S \subseteq \mathbb{R}^d$ be the subspace such that $\bigoplus_{i=0}^{S} U_i = \mathbb{R}^d$, and $U_S \perp U_i$ for all $i = 0, \ldots, S-1$.

Recall P_i is the orthogonal projection matrix with rank i that gives rise to G_i in Assumption 3.2. In the above, we have assumed that the base of $\log$ is 2.

Let Q_s be the orthogonal projection matrix that projects vectors onto the subspace U_s. Note that $\text{rank}(Q_s) \le 2^s k$ since $U_s \subseteq \text{range}(P_{2^s k})$.

Moreover, it's clear that $U_s \perp \text{range}(P_{2^{s-1}k})$ for all $s \in \{1, \ldots, S\}$. This is because by construction $\bigoplus_{i=0}^{s-1} U_i \supseteq \text{range}(P_{2^{s-1}k})$, and that $U_s \perp \bigoplus_{i=0}^{s-1} U_i$.

Thus,

$$\|Q_s \nabla F(x)\|_2 = \|Q_s(I - P_{2^{s-1}k})\nabla F(x)\|_2 \le \|Q_s\|_{\text{op}} \|(I - P_{2^{s-1}k})\nabla F(x)\|_2 \le G_{2^{s-1}k} \quad (6)$$

for all $x \in \mathbb{R}^d$ and all $s \in \{1, \ldots, S\}$.

Let $j \in [n]$ be the (uniformly random) index sampled in iteration t of DP-SGD. By convexity of the individual loss f_j,

$$f_j(x^{(t)}) - f_j(x^*) \le \nabla f_j(x^{(t)})^\top (x^{(t)} - x^*).$$

By construction, $\mathbb{R}^d$ is the orthogonal direct sum of the subspaces $\{U_j\}_{j=0}^{S}$, and thus any vector $v \in \mathbb{R}^d$ can be rewritten as the sum $\sum_{i=0}^{S} Q_i v$. We thus split the right hand side of the above as follows

$$f_j(x^{(t)}) - f_j(x^*) \le \left(Q_0 \nabla f_j(x^{(t)}) + \sum_{s=1}^{S} Q_s \nabla f_j(x^{(t)})\right)^\top (x^{(t)} - x^*). \quad (7)$$

We use different approaches to bound $(Q_0 \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*)$ and $(Q_s \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*)$ when $s \ge 1$, and we discuss them separately in the following.

Term one: Bound $(Q_0 \nabla f_j(x^{(t)}))^\top (x^{(t)} - x^*)$. Recall that

$$x^{(t+1)} = x^{(t)} - \eta\Big(\nabla f_j(x^{(t)}) + \alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\Big)$$

for some Gaussian $\zeta \sim \mathcal{N}(0, \sigma^2 I_d)$. Hence, we have

$$\begin{aligned}
&(\nabla f_j(x^{(t)}))^\top Q_0 (x^{(t)} - x^*) \\
=\ & \left(\frac{1}{\eta}(x^{(t)} - x^{(t+1)}) - \alpha(x^{(t)} - x^{(0)}) - G_0 \cdot \zeta\right)^\top Q_0 (x^{(t)} - x^*) \\
=\ & \left(\frac{1}{\eta} Q_0 (x^{(t)} - x^{(t+1)})\right)^\top Q_0 (x^{(t)} - x^*) - \Big(\alpha(x^{(t)} - x^{(0)}) + G_0 \cdot \zeta\Big)^\top Q_0 (x^{(t)} - x^*)
\end{aligned}$$

$$
\begin{aligned}
=&\frac{1}{2\eta}\left(\|Q_0(x^{(t)}-x^*)\|_2^2-\|Q_0(x^{(t+1)}-x^*)\|_2^2+\|Q_0(x^{(t)}-x^{(t+1)})\|_2^2\right)\\
&-\left(\alpha(x^{(t)}-x^{(0)})+G_0\cdot\zeta\right)^\top Q_0(x^{(t)}-x^*),
\end{aligned}
\tag{8}
$$

where we used the fact that $Q_0^2 v = Q_0 v$ for any $v \in \mathbb{R}^d$ (since Q_0 is a projection matrix), and the last equality follows from that

$$
\begin{aligned}
&2(Q_0(x^{(t)}-x^{(t+1)}))^\top Q_0(x^{(t)}-x^*)\\
=&\|Q_0(x^{(t)}-x^*)\|_2^2-\|Q_0(x^{(t+1)}-x^*)\|_2^2+\|Q_0(x^{(t)}-x^{(t+1)})\|_2^2.
\end{aligned}
$$

Taking expectation on ζ over both sides of Equation (8) and making use of the fact that ζ has mean 0, we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\zeta}(Q_0\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*)\\
=&\frac{1}{2\eta}\left(\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t+1)}-x^*)\|_2^2+\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(t+1)})\|_2^2\right)\\
&-\alpha\mathop{\mathbb{E}}_{\zeta}\left((x^{(t)}-x^{(0)})^\top Q_0(x^{(t)}-x^*)\right).
\end{aligned}
$$

Recalling the definition of Q_0 and that Q_0 has rank at most k, one has

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(t+1)})\|_2^2 &= \eta^2\mathop{\mathbb{E}}_{\zeta}\|Q_0\big(\nabla f_j(x^{(t)})+\alpha(x^{(t)}-x^{(0)})+G_0\cdot\zeta\big)\|_2^2\\
&= \eta^2\mathop{\mathbb{E}}_{\zeta}\|Q_0\big(\nabla f_j(x^{(t)})+\alpha(x^{(t)}-x^{(0)})\big)\|_2^2+\eta^2G_0^2k\sigma^2\\
&\le 2\eta^2G_0^2(1+k\sigma^2)+2\eta^2\alpha^2\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(0)})\|_2^2.
\end{aligned}
$$

Moreover, one has

$$
\begin{aligned}
&-\alpha(x^{(t)}-x^{(0)})^\top Q_0(x^{(t)}-x^*)\\
=&-\alpha(x^{(t)}-x^{(0)})^\top Q_0(x^{(t)}-x^{(0)})-\alpha(x^{(t)}-x^{(0)})^\top Q_0(x^{(0)}-x^*)\\
\le&-\frac{\alpha}{2}\|Q_0(x^{(t)}-x^{(0)})\|_2^2+\frac{\alpha}{2}\|Q_0(x^{(0)}-x^*)\|_2^2.
\end{aligned}
$$

Hence, we have

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\zeta}(Q_0\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*)\\
\le&\frac{1}{2\eta}\left(\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t+1)}-x^*)\|_2^2\right)+\eta G_0^2(1+k\sigma^2)\\
&+\eta\alpha^2\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(0)})\|_2^2-\frac{\alpha}{2}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^{(0)})\|_2^2+\frac{\alpha}{2}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(0)}-x^*)\|_2^2\\
\le&\frac{1}{2\eta}\left(\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t)}-x^*)\|_2^2-\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(t+1)}-x^*)\|_2^2\right)+\eta G_0^2(1+k\sigma^2)+\frac{\alpha}{2}\mathop{\mathbb{E}}_{\zeta}\|Q_0(x^{(0)}-x^*)\|_2^2,
\end{aligned}
$$

where we used $\eta \le \frac{1}{2\alpha}$ at the end.

Term two: Bound $(Q_s\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*)$. We bound the objective above for each s separately. By taking expectation over the random f_j, we have

$$
\begin{aligned}
\mathop{\mathbb{E}}_{f_j}(Q_s\nabla f_j(x^{(t)}))^\top(x^{(t)}-x^*) &= (Q_s\nabla F(x^{(t)}))^\top(x^{(t)}-x^*)\\
&\le \|Q_s\nabla F(x^{(t)})\|_2\cdot\|Q_s(x^{(t)}-x^*)\|_2\\
&\le \frac{1}{\alpha_s}\|Q_s\nabla F(x^{(t)})\|_2^2+\frac{\alpha_s}{4}\|Q_s(x^{(t)}-x^*)\|_2^2\\
&\le \frac{G_{2^{s-1}k}^2}{\alpha_s}+\frac{\alpha_s}{2}\|Q_s(x^{(t)}-x^{(0)})\|_2^2+\frac{\alpha_s}{2}\|Q_s(x^{(0)}-x^*)\|_2^2,
\end{aligned}
$$

where we chose $\alpha_s = \alpha s^{-2} 2^{-s}$ and used the bound (6) and Young's inequality at the end.

Combination: Combining both terms into (7) and taking expectation over all randomness, we have

$$
\begin{aligned}
&\mathbb{E}[F(x^{(t)}) - F(x^*)] \\
\leq\ & \frac{1}{2\eta}(\mathbb{E}\,\|Q_0(x^{(t)} - x^*)\|_2^2 - \mathbb{E}\,\|Q_0(x^{(t+1)} - x^*)\|_2^2) + \eta G_0^2(1 + k\sigma^2) + \frac{\alpha}{2}\,\mathbb{E}\,\|Q_0(x^{(0)} - x^*)\|_2^2 \\
&+ \sum_{s=1}^{S} \frac{G_{2^{s-1}k}^2}{\alpha_s} + \frac{1}{2}\sum_{s=1}^{S} \alpha_s\,\mathbb{E}\,\|Q_s(x^{(t)} - x^{(0)})\|_2^2 + \frac{1}{2}\sum_{s=1}^{S} \alpha_s\,\mathbb{E}\,\|Q_s(x^{(0)} - x^*)\|_2^2 \\
\leq\ & \frac{1}{2\eta}(\mathbb{E}\,\|Q_0(x^{(t)} - x^*)\|_2^2 - \mathbb{E}\,\|Q_0(x^{(t+1)} - x^*)\|_2^2) + \eta G_0^2(1 + k\sigma^2) + \frac{\alpha}{2}\|x^{(0)} - x^*\|_2^2 \\
&+ \sum_{s=1}^{S} \frac{G_{2^{s-1}k}^2}{\alpha_s} + \frac{1}{2}\sum_{s=1}^{S} \alpha_s\,\mathbb{E}\,\|Q_s(x^{(t)} - x^{(0)})\|_2^2.
\end{aligned}
$$

Recall $\alpha \cdot \eta \leq 1/2$. Under the other assumptions, by Lemma A.1, one can show

$$
\begin{aligned}
\mathbb{E}\,\|Q_s(x^{(t)} - x^{(0)})\|^2 &\leq \frac{4G_{2^{s-1}k}^2}{\alpha^2} + \frac{2\eta G_0^2}{\alpha}(1 + 2^s k\sigma^2) \\
&\leq \frac{4G_{2^{s-1}k}^2}{\alpha_s^2} + \frac{2\eta G_0^2}{\alpha_s s^2}(1 + k\sigma^2).
\end{aligned}
$$

Using $\sum_{s=1}^{\infty} s^{-2} \leq 2$, we have

$$
\begin{aligned}
&\mathbb{E}\,F(x^{(t)}) - \mathbb{E}\,F(x^*) \\
\leq\ & \frac{1}{2\eta}(\mathbb{E}\,\|Q_0(x^{(t)} - x^*)\|_2^2 - \mathbb{E}\,\|Q_0(x^{(t+1)} - x^*)\|_2^2) + \frac{\alpha}{2}\|x^{(0)} - x^*\|^2 \\
&+ \eta G_0^2(1 + k\sigma^2) + 3\sum_{s=1}^{S} \frac{G_{2^{s-1}k}^2}{\alpha_s} + 2\eta G_0^2(1 + k\sigma^2) \\
\leq\ & \frac{1}{2\eta}(\mathbb{E}\,\|Q_0(x^{(t)} - x^*)\|_2^2 - \mathbb{E}\,\|Q_0(x^{(t+1)} - x^*)\|_2^2) + \frac{\alpha}{2}\|x^{(0)} - x^*\|_2^2 \\
&+ 3\eta G_0^2(1 + k\sigma^2) + \frac{3}{\alpha}\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2.
\end{aligned}
$$

Summing up over $t = 1, 2, \cdots, T$, by the assumption that $\|x^{(0)} - x^*\|_2 \leq D$ and convexity of the function, we have

$$
\mathbb{E}[F(\overline{x}) - F(x^*)] \leq \frac{D^2}{2\eta T} + 3\eta G_0^2(1 + k\sigma^2) + \frac{\alpha}{2}D^2 + \frac{3}{\alpha}\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2. \tag{9}
$$

Set the parameters $T = c_1(n^2 + d\log^2 d)$, $\sigma = \frac{c_2\sqrt{T\log(1/\delta)}}{n\varepsilon}$, $\eta = \sqrt{\frac{D^2}{T \cdot G_0^2 \cdot k\sigma^2}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$ for some large constants c_1, c_2. Note that this choice of parameters satisfies the condition of

$$
\begin{aligned}
\eta \cdot \alpha &= \sqrt{\frac{D^2}{T \cdot G_0^2 \cdot k\sigma^2}} \cdot \frac{\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}}{D} \\
&\leq \sqrt{\frac{G_0^2(2d)\log^3(2d)}{T \cdot G_0^2 \cdot k\sigma^2}} = \frac{n\varepsilon}{c_2 T}\sqrt{\frac{(2d)\log^3(2d)}{k \cdot \log(1/\delta)}} \\
&\leq \frac{n\varepsilon\sqrt{(2d)\log^3(2d)}}{c_2 T} \leq \frac{1}{2},
\end{aligned}
$$

where we used the fact that $G_k \le G_0$, $s \le S \le \log(2d)$, $T \ge n^2 + d\log^2 d$, and c_2 is large enough. Using the parameters we pick, we have

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$$

Moreover, assuming $G_k \le G_0 k^{-c}$ for some $c > 1/2$, we have $\sqrt{\sum_s s^2 2^s G_{2^{s-1}k}^2} \lesssim G_0/k^c$. Hence,

$$\mathbb{E}[F(\overline{x}) - F(x^*)] \lesssim \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + \frac{G_0 D}{k^c}.$$

Since the above bound holds for all $k \in \{1, \dots, d\}$, we may optimize it with respect to k. Recall by assumption that $n \ge \varepsilon^{-1}\sqrt{\log(1/\delta)}$. Letting

$$k = \min\left\{d, \left\lceil \left(\frac{\varepsilon n}{\sqrt{\log(1/\delta)}}\right)^{\frac{2}{1+2c}} \right\rceil\right\}$$

yields the bound

$$\mathbb{E}[F(\overline{x};\mathcal{D}) - F(x^*;\mathcal{D})] \lesssim G_0 D \cdot \left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

□

Combining the privacy guarantee in Lemma 2.4 and Lemma A.2 directly results in Theorem 3.3.

B Proof of Theorem 3.5

We study the generalization error of DP-SGD. Similar to previous works, we make use of the stability of DP-SGD to bound its generalization error. The bound on the excess population loss follows from combining bounds on the excess empirical loss and the generalization error. Before stating the proof, we first recall two results in the literature.

Lemma B.1 ([BE02, Lemma 7])**.** *Given a learning algorithm $\mathcal{A}$, suppose the dataset $\mathcal{D} = \{s_1, \cdots, s_n\}$ is made up of n i.i.d. samples drawn from the underlying distribution $\mathcal{P}$, and we replace one random sample in $\mathcal{D}$ with a freshly sampled $s' \sim \mathcal{P}$ and get a new neighboring dataset $\mathcal{D}'$. One has*

$$\mathop{\mathbb{E}}_{\mathcal{D},\mathcal{A}}[F(\mathcal{A}(\mathcal{D});\mathcal{P}) - F(\mathcal{A}(\mathcal{D});\mathcal{D})] = \mathop{\mathbb{E}}_{\mathcal{D},s'\sim\mathcal{P},\mathcal{A}}[f(\mathcal{A}(\mathcal{D});s') - f(\mathcal{A}(\mathcal{D}');s')],$$

where $\mathcal{A}(\mathcal{D})$ is the output of $\mathcal{A}$ with input $\mathcal{D}$.

Lemma B.2 ([BFGT20, Theorem 3.3])**.** *Suppose Assumption 3.2 holds, running DP-SGD with step size η on any two neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$ for T steps satisfies that*

$$\mathbb{E}\left[\|\overline{x} - \overline{x}'\|_2\right] \le 4G_0\eta\left(\frac{T}{n} + \sqrt{T}\right),$$

where $\overline{x}$ and $\overline{x}'$ are the outputs of DP-SGD with datasets $\mathcal{D}$ and $\mathcal{D}'$, respectively.

Proof of Theorem 3.5. Let $\overline{x}$ and $\overline{x}'$ be the outputs of DP-SGD when applied to the datasets $\mathcal{D}$ and $\mathcal{D}'$, respectively. $\mathcal{D}'$ is a neighbor of $\mathcal{D}$ with one example replaced by $s' \sim \mathcal{P}$ that is independently sampled. Combining Lemma B.1 and Lemma B.2 yields

$$\begin{aligned}\mathbb{E}[F(\overline{x};\mathcal{P}) - F(\overline{x};\mathcal{D})] &= \mathbb{E}[f(\overline{x};s') - f(\overline{x}';s')] \\ &\le \mathbb{E}[G_0\|\overline{x} - \overline{x}'\|_2]\end{aligned}$$

$$\leq 4G_0^2\eta\left(\frac{T}{n}+\sqrt{T}\right).$$

Similar to the DP-ERM case, by setting $T = c_1(n^2 + d\log^2 d)$, $\sigma = \frac{c_2\sqrt{T\log(1/\delta)}}{n\varepsilon}$, $\eta = \sqrt{\frac{D^2}{T\cdot G_0^2(T/n+k\sigma^2)}}$ and $\alpha = \frac{1}{D}\sqrt{\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2}$ for some large positive constants c_1 and c_2, we conclude that $\eta\cdot\alpha \leq 1/2$. Hence, Equation (9) shows that, for any fixed dataset $\mathcal{D}$ and any x^* such that $\left\|x^{(0)} - x^*\right\|_2 \leq D$, we have

$$\mathbb{E}[F(\overline{x};\mathcal{D}) - F(x^*;\mathcal{D})] \leq \frac{D^2}{2\eta T} + 3\eta G_0^2(1+k\sigma^2) + \frac{\alpha}{2}D^2 + \frac{3}{\alpha}\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2.$$

We can therefore rewrite the population loss as follows

$$\begin{aligned}
&\mathbb{E}[F(\overline{x};\mathcal{P}) - F(x^*;\mathcal{P})] \\
&= \mathbb{E}[F(\overline{x};\mathcal{P}) - F(\overline{x};\mathcal{D})] + \mathop{\mathbb{E}}_{\mathcal{D}}[F(\overline{x};\mathcal{D}) - F(x^*;\mathcal{D})] \\
&\leq 4G_0^2\eta\left(\frac{T}{n}+\sqrt{T}\right) + \frac{D^2}{2\eta T} + 3\eta G_0^2(1+k\sigma^2) + \frac{\alpha}{2}D^2 + \frac{3}{\alpha}\sum_{s=1}^{S} s^2 2^s G_{2^{s-1}k}^2.
\end{aligned}$$

Substituting in the values for parameters T, σ, η, and α yields

$$\mathbb{E}[F(\overline{x};\mathcal{P}) - F(x^*;\mathcal{P})] \lesssim \frac{G_0 D}{\sqrt{n}} + \frac{G_0 D\sqrt{k\log(1/\delta)}}{\varepsilon n} + D\sqrt{\sum_{s=1}^{\infty} s^2 2^s G_{2^{s-1}k}^2}$$

for all positive integers $k \leq d$.

Similarly, if we have $G_k \leq G_0 k^{-c}$ for some $c > 1/2$, and in addition $n > \varepsilon^{-1}\log(1/\delta)$, it immediately follows that

$$\mathbb{E}[F(\overline{x};\mathcal{P}) - \min_x F(x;\mathcal{P})] \lesssim \frac{G_0 D}{\sqrt{n}} + G_0 D\cdot\left(\frac{\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{2c/(1+2c)}.$$

This completes the proof. □

C Protocol for Synthetic Example Experiments in Section 4.1

We detail the construction of the synthetic example in Section 4.1. The training and test sets of this example both have $n_{\text{train}} = n_{\text{test}} = 10000$ instances. Each instance x_i is sampled from a distribution where the first $d_{\text{min}} = 10$ coordinates are all multi-variate normal distributions with mean and standard deviation both being 1. All remaining coordinates are constantly 0. This ensures the optimal non-private training losses for problems of different dimensions are the same.

D Protocol and Additional Fine-Tuning Experiments for Section 4.2

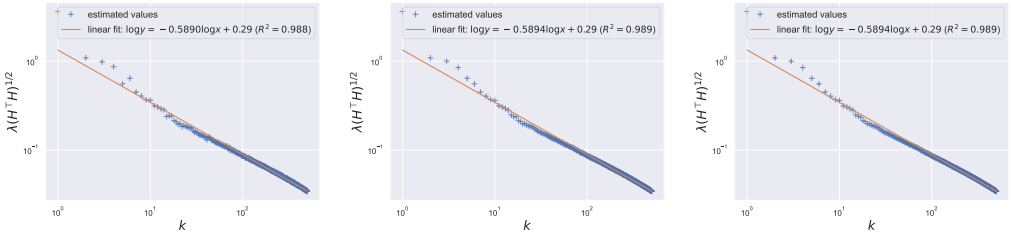
D.1 Experimental Protocol

For experiments in Section 4.2, we fine-tuned the DistilRoBERTa model with $(8, 1/n^{1.1})$-DP on the SST-2 training set with $n \geq 60000$ examples and measured performance on the companion dev set. We used the exact set of hyperparameters presented in [LTLH21] for this task. We repeated our re-training experiments over five independent random seeds. Fine-tuning for 3 epochs on this task takes 10 minutes on an A100-powered machine with sufficient RAM and CPU cores.

Our spectral analysis relies on running the orthogonal iteration algorithm on the set of collected gradients along the private fine-tuning trajectory [Dem97].[4] Unlike other numerical algorithms for estimating eigenvalues of a matrix, orthogonal iteration provably produces the set of eigenvalues that are largest in absolute value (and corresponding eigenvectors, if the underlying matrix is normal) in the limit.[5] Recall that we needed the top eigenvectors for projection in our re-training experiment.[6] By default, we run orthogonal iteration for ten iterations. We show in the following that our results are insensitivity to the number of iterations used in the orthogonal iteration algorithm. Each orthogonal iteration takes 10 minutes on an A100-powered machine with $r = 4000$ gradient samples and $k = 1000$ principal components for the DistilRoBERTa experiments.

D.2 Robustness of Results

Robustness to the number of orthogonal iterations. The orthogonal iteration algorithm is a generalization of the power method that simultaneously produces estimates of multiple eigenvalues. Its convergence is known to be sensitivity to the gap between successive eigenvalues. The algorithm converges slowly if consecutive eigenvalues (with the largest absolute values) are close in absolute value. To confirm that our results aren't sensitivity to the choice of the number of iterations, we visualize the top 500 eigenvalues for the orthogonal iteration algorithm is run for different number of updates. Figure 3 shows that the linear fit to the top 500 eigenvalues remains stable across different number of orthogonal iterations T. Notably, $T = 10$ produces similar results as $T = 100$. These results were obtained with $r = 4000$ gradients.



(a) $T = 10$ (b) $T = 50$ (c) $T = 100$

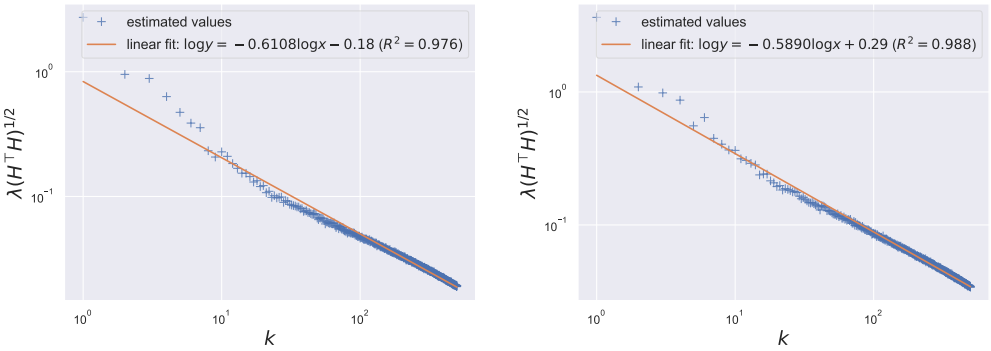
Figure 3: The eigenspectrum remains stable with different number of iterations T.

[4] By gradients, we always mean the average of clipped per-example gradients — before adding noise — in this section.

[5] The orthogonal iteration algorithm is also known as simultaneous iteration or subspace iteration.

[6] Note that one commonly used algorithm in the neural net spectral analysis literature — Lanczos iteration [GKX19] — does not guarantee that the top eigenvalues are produced, even though its spectral estimates are frequently deemed accurate [GWG19].

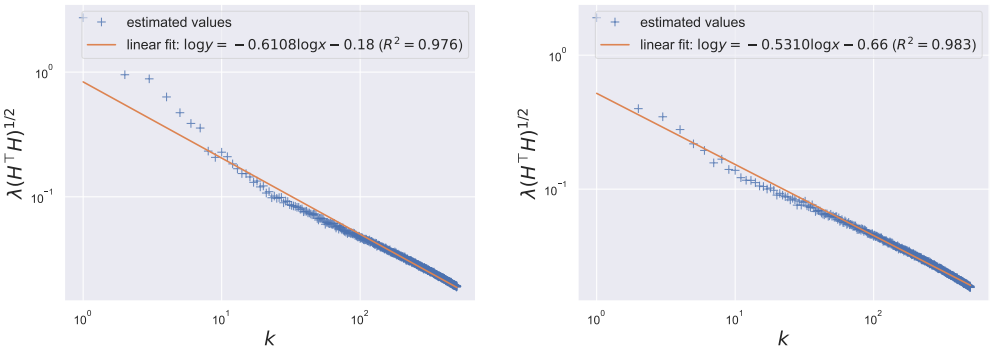
Robustness to the number of gradient samples. We further ran experiments with different numbers of gradient samples r collected along the fine-tuning trajectory, and plot the top 500 eigenvalues. Figure 4 shows that while the slope and intercept of the fitted line in log-log space changes, the change is moderate. Notably, the decaying trend of the top eigenvalues remains stable.



(a) $r = 1000$ (b) $r = 4000$

Figure 4: The fast decaying trend of the eigenspectrum remains stable with different number of samples r used for the spectral analysis.

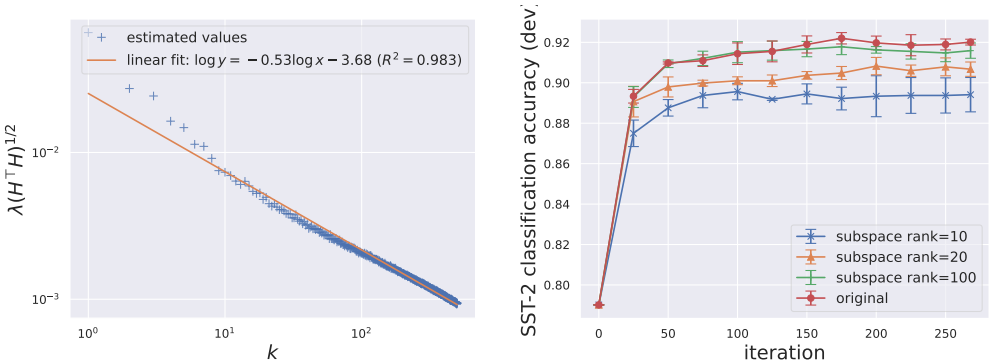
Robustness to the gradient sampling strategy. We observe that gradients at the beginning of fine-tuning tend to be larger in magnitude than gradients collected later on along the optimization trajectory. To eliminate the potential confounder that the top principal components are solely formed by the initial few gradients evaluated during fine-tuning, we re-ran the spectral analysis experiment without the initial gradients. Specifically, we performed PCA for the gradients evaluated from step 300 to step 1300 during private fine-tuning, and compared the distribution of top eigenvalues returned from this setup to when we used the first 1000 gradients. Note the dev set accuracy converged to $\approx 90\%$ by step 200. Figure 5 shows that while the slope and intercept of linear fits are slightly different in the new setup compared to the old setup (when all gradients along the fine-tuning trajectory were used for PCA), that the eigenvalues follow a rapidly decaying trend remains true under both setups.



(a) Gradients at iterations 0 to 1000 (b) Gradients at iterations 300 to 1300

Figure 5: The rapidly decaying trend of the eigenspectrum remains stable with different sampling strategies.

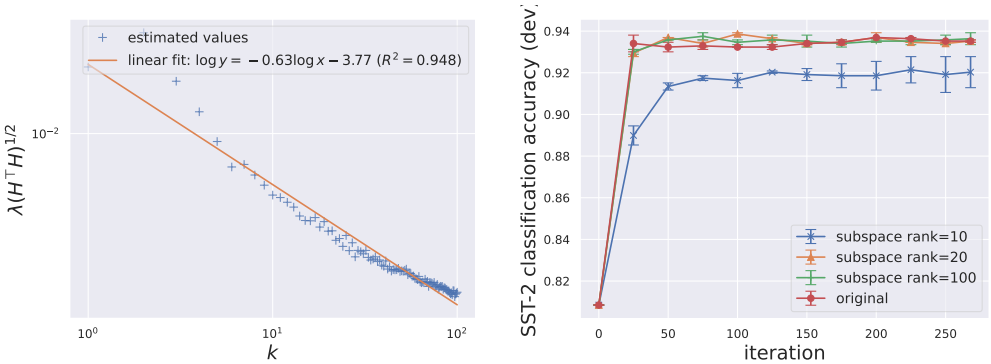
Robustness to model size. In previous experiments, we empirically verified that gradients for fine-tuning DistilRoBERTa are near low rank. Here, we show that similar observations also hold for Roberta-base and Roberta-large when fine-tuning only the attention layers. The former setup has approximately 14 million trainable parameters, while the latter has approximately 50 million. Figures 6 and 7 illustrate these results.



(a) singular values decay with rank

(b) retrain in fixed subspace

Figure 6: Experiments for Roberta-base.



(a) singular values decay with rank

(b) retrain in fixed subspace

Figure 7: Experiments for Roberta-large.

Non-robustness to fine-tuning strategy. Recall our fine-tuning experiments for classification was based on the template-completion formulation detailed in [LTLH21]. As opposed to framing the task as integer label prediction, this formulation requires the model to predict one of K candidate tokens to fill in a templated prompt for a K-way classification problem. While we have also performed the experiment where we re-train in the subspace of top principal components under the usual fine-tuning setup (stack a randomly initialized prediction head on top of the embedding of the `[CLS]` token), we found it difficult to recover the original fine-tuning performance when gradients are projected onto the top eigen-subspace with $d = 100$ dimensions. Retraining performance exhibited high variance and the final dev accuracy was bi-modal over random seeds with near guess accuracy ($\approx 50\%$) and original accuracy ($\approx 90\%$) being the two modes. We suspect this to be caused by the linear prediction head being randomly initialized.

D.3 Additional Fine-Tuning Experiments with DistilGPT-2 for Generation Tasks

Experiments in Section 4.2 demonstrated that gradients from fine-tuning for classification are mostly controlled by a few principal components. In this section, we show that similar observations hold for fine-tuning on a text generation task. We follow the setup and hyperparameters in [LTLH21] for privately fine-tuning DistilGPT-2 on the E2E dataset [NDR17] under $(8, 1/n^{1.1})$-DP. We fine-tune all weight matrices in attention layers that produce the queries, values, and keys. This amounts to fine-tuning approximately 10.6 million parameters of a model with a total parameter count of more than 100 million. We again collected $r = 4000$ gradients evaluated during private fine-tuning, performed PCA, and conducted the eigenspectrum analysis. Figure 8 shows that the top eigenspectrum decays rapidly with a rate similar to what is observed in fine-tuning for the classification problem we studied.

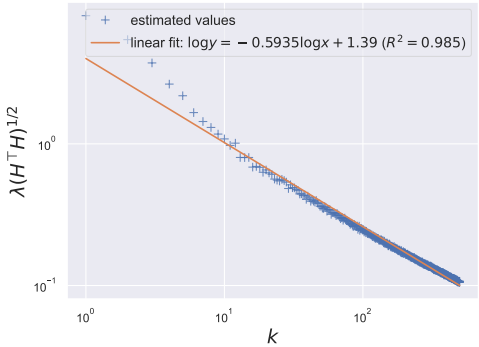


Figure 8: The eigenspectrum of gradients from fine-tuning for text generation rapidly decays.