Why Is Public Pretraining Necessary for Private Model Training?

Arun Ganesh [1] **Mahdi Haghifam** [2] **Milad Nasr** [1] **Sewoong Oh** [1 3] **Thomas Steinke** [1] **Om Thakkar** [1]
Abhradeep Thakurta [1] **Lun Wang** [1]

Abstract

In the privacy-utility tradeoff of a model trained on benchmark language and vision tasks, remarkable improvements have been widely reported when the model is *pretrained* on public data. Some gain is expected as these models inherit the benefits of transfer learning, which is the standard motivation in non-private settings. However, the stark contrast in the gain of pretraining between non-private and private machine learning suggests that the gain in the latter is rooted in a fundamentally different cause. To explain this phenomenon, we hypothesize that the non-convex loss landscape of a model training necessitates the optimization algorithm to go through two phases. In the first, the algorithm needs to select a good "basin" in the loss landscape. In the second, the algorithm solves an easy optimization within that basin. The former is a harder problem to solve with private data, while the latter is harder to solve with public data due to a distribution shift or data scarcity. Guided by this intuition, we provide theoretical constructions that provably demonstrate the separation between private training with and without public pretraining. Further, systematic experiments on CIFAR10 and Librispeech provide supporting evidence for our hypothesis.

1. Introduction

As modern machine learning models are increasingly capable of memorizing the training data, membership inference attacks and data reconstruction attacks have successfully demonstrated the vulnerability of releasing models trained on sensitive data. Differential Privacy (DP), introduced by Dwork et al. (2006), is now a gold standard measure of privacy leakage in training a model with two scalar parameters: $\varepsilon > 0$ and $\delta \in [0, 1]$. By introducing enough randomness in the training, one can ensure that the model does not depend too much on each individual training example. This provides plausible deniability to the participants and evades privacy attacks, achieving strong DP with small values of (ε, δ). We give a formal definition in Definition 1.1.

One of the main challenges in training on private data is that, without pretraining on some publicly available data, utility and privacy trades off unfavorably on standard benchmark tasks. Given a target task, such as table-to-text generation, on a private dataset, say E2E dataset (Novikova et al., 2017), state-of-the-art techniques suffer from significant performance degradation to achieve even an acceptable level of privacy. For example, a weak privacy guarantee of $\varepsilon = 8$ significantly deteriorates the performance of the trained model compared to the one trained without privacy, i.e. $\varepsilon = \infty$ (second row of Table 1). Perhaps surprisingly, there is one simple change to the training algorithm that can significantly reduce this cost of privacy: pretraining the model on some public data (first row of Table 1).

	$\varepsilon = \infty$	$\varepsilon = 8$	cost of privacy
with public pretrain	69.46	63.19	6.27
without public pretrain	65.73	24.25	41.48
gain of public pretraining	3.73	38.94	

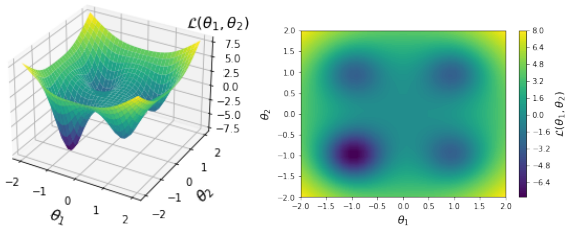
Table 1. BLEU score for generating descriptions of table entries on E2E dataset reported in (Li et al., 2022b, Table 2) with $\delta = 10^{-5}$. The first row is pretrained on GPT-2 (Radford et al., 2019).

Such remarkable gain of public pretraining has been widely observed in standard benchmark vision and language tasks, which we survey in Appendix B. This includes CIFAR-10, MNIST, and Fashion MNIST in (Tramer & Boneh, 2020), CIFAR-100, ImageNet, and Places-365 in (De et al., 2022), text generation with E2E and DART in (Li et al., 2022b), and next word prediction on Reddit dataset (Kerrigan et al., 2020). Note that in all these cases, the public data distribution differs from the target task distribution. Nevertheless, we expect some gain from public pretraining, drawing analogy from its success in non-private training of large models (e.g., first column in Table 1). However, the stark difference

[*]Equal contribution [1]Google. [2]University of Toronto. Part of this work done while the author was an intern at Google. [3]University of Washington.. Correspondence to: Arun Ganesh <arunganesh@google.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

in the gain of pretraining between the non-private case, i.e., $\varepsilon = \infty$, and the weakly private case, say $\varepsilon = 8$, is striking. This suggests that the benefit of public pretraining in differentially private machine learning is a fundamentally different phenomenon from the typical benefits of standard transfer learning (Bozinovski & Fulgosi, 1976; Sharif Razavian et al., 2014; Bommasani et al., 2021). Our goal is to give an insight into when such a phenomenon can be observed by carefully constructing synthetic public and private tasks. Recently, (Li et al., 2022a) formally demonstrates that public data improves on the curse of dimensionality while fine tuning with privacy. To our knowledge, ours is the first work to understand the necessity of public data in private model training.



(a) (b)

Figure 1. An example of a non-convex loss function. While the overall function is non-convex, it consists of many locally convex "basins", some better than others.

In this paper, we provide a theoretical example of a loss function that *requires* pretraining on public data and fine-tuning with private data. Our construction is guided by our hypothesis that the typical population loss landscape of standard machine learning tasks necessitates gradient based algorithms to go through two stages. A conceptual two-dimensional sketch of the landscape we envision is shown in Figure 1. We start from a random initialization close to the origin. In the first stage, the algorithm is directed by the data towards a good basin with small local minima. This is followed by the second stage, where the algorithm solves what is effectively a convex optimization in the selected basin to arrive at the local minima. The key insight is that the first stage of selection should require significantly more samples to solve privately, compared to the number of samples required to solve it without privacy. Concretely, for the example in Figure 1, the gradient at the origin directs to the correct basin with global minima, but the gradient is small. A private gradient descent adds additional noise to the update, increasing the chance of ending up at worse basins. Hence, significantly more private data is needed to overcome the privacy noise. This construction is motivated by the private hypothesis selection problems where a similar fundamental separation in sample complexity is known (Steinke & Ullman, 2017). This intuition would explain the widely observed failure of private training when starting from a random initialization. We turn this hypothesis into concrete constructions in Section 2, where we formally prove the separation in sample complexity.

Main contributions: In Section 2, we construct theoretical tasks to demonstrate the fundamental separation in sample complexity. First, we construct a theoretical loss function and data distribution such that given n_{pub} public samples and n_{priv} private samples from this distribution, $n_{pub} \ll n_{priv}$, pretraining on the public data and fine tuning with the private data achieves a much better loss than any algorithm with access to either alone. Next, we extend our result to a more relevant setting where n_{pub} is large but the public data is out of distribution. This construction exhibits the need to have little to no privacy noise in the first "phase" of non-convex optimization. To the best of our knowledge, this is the first theoretical lower bound demonstrating the need for public pretraining. We wish to emphasize that this lower bound is for a synthetic construction; while our construction is made to capture our two-phase hypothesis, which is made for the deep learning setting (and indeed supported by our experiments), we do not claim that such a lower bound holds for all deep learning problems.

In Section 3, we empirically validate our two-phase hypothesis. First, treating CIFAR-10 as our target private task, we consider a setup where we are allowed T epochs of pre- or post-training on in-distribution public data, out-of-distribution public data, or private data with low noise. In all settings we demonstrate it is best to use all these low- or non-private training epochs on pretraining. This demonstrates that early rounds of training are most sensitive to privacy noise, as conjectured in our two-phase hypothesis. Secondly, we look at a manifold of the loss landscape interpolated between three models trained on Librispeech. We show that a publicly pretrained and privately fine-tuned model ends up in the same basin as a fully publicly trained model. On the other hand, a fully privately trained model ends up in a different basin. This provides evidence that public pretraining's benefits are in part due to selecting a better basin for fine-tuning.

1.1. Other Related Work

Pretraining on public data is now a default choice in large scale private training for NLP tasks (Yu et al., 2022; He et al., 2022; Bu et al.; Ginart et al., 2022), including 175 billion parameter GPT-3 with $\varepsilon = 1$, and vision tasks (Golatkar et al., 2022; Luo et al., 2021; Kurakin et al., 2022; Bu et al., 2022; De et al., 2022). Motivated by pretraining providing good feature representations, Tramer & Boneh (2020) propose using handcrafted features for small scale problems, as opposed to learned features, to improve utility-privacy tradeoff. On the other hand, (Tramèr et al., 2022)

cautions against the indiscriminate use of large-scale public data in DP training, which we discuss in depth in Section 4.

Besides the aforementioned empirical results, public data has been used to show theoretical improvements for problems such as query release (Alon et al., 2019; Bassily et al., 2020a; Liu et al., 2021), mean estimation (Avent et al., 2020; Bie et al., 2022), and optimization (Zhou et al., 2021; Kairouz et al., 2020; Asi et al., 2021b; Amid et al., 2022). In the optimization case, besides pretraining, these papers use public data to learn the geometry of the private loss in various ways and use geometry-aware gradient descent methods, rather than vanilla DP-SGD.

(Steinke & Ullman, 2017) showed that for the problem of selecting from d coins the k that land heads with the highest probability given n sample flips from each coin, any (ε, δ)-DP algorithm with constant error requires $n = \Omega(\sqrt{k} \log d)$. This is in contrast with the non-private case, where $n = O(\log d)$ suffices for any k. Selection and non-convex optimization are tightly connected: (Ganesh et al., 2022) show a reduction from selection to non-convex optimization, by designing a loss with d locally convex basins, each corresponding to a different coin in the selection problem. This gives a different perspective on why the first stage of non-convex optimization may be difficult privately but not with public data: it effectively involves solving a selection problem on the basins in the loss function.

1.2. Background on differential privacy and DP-SCO

Differential privacy is a privacy guarantee for algorithms that can be viewed as random functions of datasets:

Definition 1.1 (Differential Privacy (Dwork et al., 2006))**.** *Let $\mathcal{D}$ be a data domain, and $\mathcal{C}$ be a set of outputs. An algorithm $\mathcal{A} : \mathcal{D}^* \to \mathcal{C}$ is (ε, δ)-differentially private if for any $D, D' \in \mathcal{D}^*$ such that D and D' differ in at most one element and any set of outputs $S \subseteq \mathcal{C}$: $\mathbf{Pr}_{\theta \sim \mathcal{A}(D)} [\theta \in S] \leq e^{\varepsilon} \mathbf{Pr}_{\theta \sim \mathcal{A}(D')} [\theta \in S] + \delta$.*

A well-studied problem in the differential privacy literature is differentially private stochastic (convex) optimization (DP-SCO) (Bassily et al., 2014; 2019; Feldman et al., 2020; Bassily et al., 2020b; Kulkarni et al., 2021; Asi et al., 2021b; Gopi et al., 2022). In DP-SCO, there is a loss function $\ell : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$, and an unknown distribution τ over $\mathcal{D}$. Given n i.i.d. samples from τ, we wish to find $\theta \in \mathcal{C}$ minimizing the population loss $\mathcal{L}(\theta) := \mathbb{E}_{d \sim \tau} [\ell(\theta; d)]$. For any τ we denote the population minimizer by $\theta^*(\tau) := \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta)$. The performance of a DP-SCO algorithm is measured by its *risk*, $\mathbb{E}_{D \sim \tau^n, \theta \sim \mathcal{A}(D)} [\mathcal{L}(\theta)] - \mathcal{L}(\theta^*(\tau))$. DP-SCO captures most machine learning tasks we are interested in. The most widely studied algorithm in the DP-SCO literature is DP-SGD (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016; Bassily et al., 2019; 2020b), which minimizes the empirical loss $\ell(\theta; D) = (1/|D|) \sum_{d \in D} \ell(\theta; d)$ over $\mathcal{C} \subseteq \mathbb{R}^p$ as follows: DP-SGD starts with θ_0, and for t iterations computes $\theta_{t+1} = \theta_t - \eta_t \nabla \ell(\theta_t; D) + \xi_t$, where $\xi_t \sim N(0, \sigma^2 \mathbb{I})$ and σ^2 is chosen to satisfy (ε, δ)-DP.

Perhaps the simplest problem captured by DP-SCO is private mean estimation with identity covariance. The following lemma gives a lower bound on private mean estimation. It follows from Theorem 5.5 of (Bassily et al., 2014) and standard translation of ERM lower bounds to SCO lower bounds (see Appendix C of (Bassily et al., 2019)):

Lemma 1.2. *For $\ell(\theta; d) = (1/2) \|\theta - d\|_2^2$, $\mathcal{C} = \mathbb{R}^p$, and $\mathcal{D} = B_p(0, 1)$ (the p-dimensional ℓ_2-ball of radius 1 centered at the origin), let $\theta^*(\tau) := \arg\min_{\theta \in \mathcal{C}} \mathcal{L}(\theta)$ for a distribution τ over $\mathcal{D}$. For $p \leq \varepsilon^2 n^2$ and $\delta = o(1/n)$, there exists a set of distributions, $\mathcal{T}_1$, over $\mathcal{D}$, such that the following is true. For every (ε, δ)-DP algorithm $\mathcal{A} : \mathcal{D}^n \to \mathcal{C}$, there exists $\tau(\mathcal{A}) \in \mathcal{T}_1$ such that:*

$$\mathbb{E}_{D \sim \tau(\mathcal{A})^n, \theta \sim \mathcal{A}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau(\mathcal{A}))) + \Omega\left(\frac{p}{\varepsilon^2 n^2} + \frac{1}{n}\right).$$

Furthermore, for some $M = \Omega(\frac{\sqrt{p}}{\varepsilon n})$ and all such $\tau \in \mathcal{T}_1$, $|\|\theta^*(\tau)\|_2 - M| \leq 1/n$.

Non-privately, this translates to:

Lemma 1.3. *For $\ell(\theta; d) = \frac{1}{2} \|\theta - d\|_2^2$, $\mathcal{C} = \mathbb{R}^p$, and $\mathcal{D} = B_p(0, 1)$, there exists a set of distributions, $\mathcal{T}_2$, over $\mathcal{D}$ such that the following is true. For every $\mathcal{A} : \mathcal{D}^n \to \mathcal{C}$ there exists $\tau(\mathcal{A}) \in \mathcal{T}_2$ such that:*

$$\mathbb{E}_{D \sim \tau(\mathcal{A})^n, \theta \sim \mathcal{A}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau(\mathcal{A}))) + \Omega\left(\frac{1}{n}\right).$$

These lemmas are the basis of the results in Section 2. Results in (Bassily et al., 2014) and standard translations from empirical loss bounds to population loss bounds via uniform stability (see e.g. (Hardt et al., 2016)) show that DP-SGD achieves upper bounds for mean estimation that match these lower bounds up to polylogarithmic factors.

2. Necessity of public pretraining

A typical scenario in pretraining on public data is when the public dataset is large but is Out-Of-Distribution (OOD); there is a potentially large distribution shift between the public and the private dataset (Yu et al., 2022; He et al., 2022; Bu et al.; Ginart et al., 2022; Golatkar et al., 2022; Luo et al., 2021; Kurakin et al., 2022; Bu et al., 2022; De et al., 2022). In this section, we start with a simpler scenario where a small number of In-Distribution (ID) samples are used in public pretraining. This simplifies the explanation of our construction and also corresponds to realistic scenarios where public data comes from users who consented. The more common OOD case is addressed in Section 2.4.

2.1. Pretraining on in-distribution public data

When a small number of in-distribution samples are publicly available, several techniques have been proposed to improve the accuracy-privacy trade-off. An immediate use is to reduce the sensitivity of a mini-batch gradient by including the public data in the mini-batch. The public data can also be used to compute useful statistics; one can reduce the privacy noise by projecting the gradient onto a low-dimensional subspace computed from public data (Kairouz et al., 2020; Yu et al., 2021; Zhou et al., 2021; Golatkar et al., 2022) and by improving the adaptive clipping method with the geometry of the gradients estimated from public data (Golatkar et al., 2022; Asi et al., 2021a; Nasr et al., 2022). However, by far the most dominant technique in terms of the accuracy gain is pretraining on the in-distribution public data. For example, on CIFAR-10 dataset, one can train a $(\varepsilon = 2, \delta = 10^{-5})$-DP model that achieves 64.9% test accuracy. Treating 4% of the training dataset as public data, the accuracy can be improved by 7.1% (Nasr et al., 2022, Table 1). All the other techniques only give 2.8% extra gain, which includes using public data in fine-tuning, public data assisted adaptive clipping, and averaging past iterates. Such pretraining with in-distribution public data has been successful also in training variational autoencoders (Jiang et al., 2022). We provide systematic study of these gains with numerical experiments on benchmark datasets in Section 3.

Motivated by the practical successes, we first consider the following setup. We are given n_{pub} public examples, D_{pub}, and n_{priv} private examples, D_{priv}, both drawn i.i.d. from the same distribution τ, where $n_{pub} \ll n_{priv}$. We construct τ such that pretraining on small ID public data can significantly improve the performance of a private training. Concretely, we will show that for any integer p, there exists a loss function ℓ, sample sizes n_{pub} and n_{priv}, and a data distribution τ such that (i) any non-private algorithm $\mathcal{A}_{pub}$ given only D_{pub} has worst-case excess population loss lower bounded by $\Omega(1)$; (ii) any (ε, δ)-DP algorithm $\mathcal{A}_{priv}$ given only D_{priv} has worst-case excess population loss lower bounded by $\Omega(1)$; and (iii) a gradient-based algorithm $\mathcal{A}_{mixed}$ that pretrains on D_{pub} and privately fine-tunes on D_{priv} achieves excess population loss upper bounded by $O(1/p)$. In particular, the dimensionality of ℓ, n_{pub}, and n_{priv} are polynomial functions of p. We focus on the unconstrained case where $\mathcal{C} = \mathbb{R}^p$, as it aligns with how differentially private learning models are trained in practice.

2.2. Construction

We first give a high-level overview of a construction for our main theorem and defer details to Appendix C.1. While our construction builds on upper/lower bounds for public/private mean estimation, one can build a similar construction using upper/lower bounds for linear regression instead. This follows via standard reductions from mean estimation to linear regression. We focus here on mean estimation for simplicity of presentation. A reference for notation is in Appendix A.

Our strategy is to concatenate the two known lower bounds for mean estimation with private data in Lemma 1.2 and with public data in Lemma 1.3. Our construction is parameterized by p, the desired multiplicative gap between the excess loss achievable with just public or private data, and the excess loss achievable with both. We consider a distribution τ over a data point $d = (d_1, d_2) \in \mathbb{R}^{p^4} \times \mathbb{R}^p$ whose population mean is $\theta^*(\tau) = (\theta_1^*(\tau), \theta_2^*(\tau)) \in \mathbb{R}^{p^4} \times \mathbb{R}^p$. The first p^4 coordinates are used to construct a hard distribution for private mean estimation with a loss function $\ell_1 : \mathbb{R}^{p^4} \times \mathbb{R}^{p^4} \to \mathbb{R}$, and the following p coordinates are used to construct a hard distribution for public mean estimation with a loss function $\ell_2 : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$. We assume we have n_{pub} public samples and n_{priv} private samples from the same distribution with $n_{pub} \ll n_{priv}$.

We will define an appropriately chosen basin $S \subset \mathbb{R}^{p^4}$ in Eq. (2) such that if θ_1 is far from S, then $\ell((\theta_1, \theta_2)) = \ell_1(\theta_1)$, but inside of S, $\ell((\theta_1, \theta_2)) = \ell_1(\theta_1) + \ell_2(\theta_2)$. In particular, we will choose ℓ_2 that is non-positive everywhere, so that is desirable to be in S with respect to minimizing ℓ.

Starting outside of S, the algorithm first needs to minimize ℓ_1 to reach S. We use ℓ_1 from the private lower bound (Lemma 1.2) such that any private algorithm achieves high excess loss with respect to ℓ_1. On the other hand, an algorithm with a small amount of public data can easily optimize ℓ_1. We will eventually choose S that contains all points close to the optimum of ℓ_1, so any public algorithm will reach S after optimizing ℓ_1, and will not touch θ_2 in doing so. Once inside the basin S, the algorithm needs to also minimize ℓ_2 to reach a small total loss. We use ℓ_2 from the public lower bound (Lemma 1.3) such that a small-size public data alone is not sufficient to (approximately) reach global minima but large-size private data can.

Precisely, we can combine the two loss functions as follows: We first define a parameter $R_2 > 0$ whose value will be specified later. R_2 can be thought of as the width of the "slope" leading into the basin S. Given R_2, we let

$$\ell((\theta_1, \theta_2); (d_1, d_2)) = \ell_1(\theta_1; d_1) + p\, q(\theta_1) \cdot \ell_2(\theta_2; d_2), \quad (1)$$

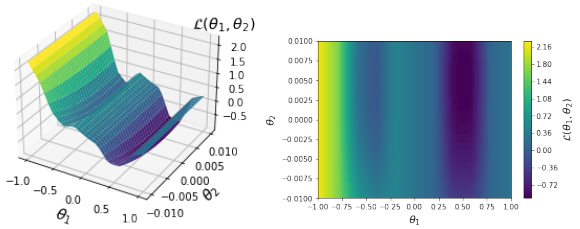
where $\Delta(\theta_1) := \|\theta_1 - \Pi_S(\theta_1)\|_2$ with Π_S being the Euclidean projection into S, and

$$q(\theta_1) := \begin{cases} 0, & \Delta(\theta_1) > R_2 \\ 1 - \frac{\Delta(\theta_1)}{R_2}, & 0 < \Delta(\theta_1) \le R_2 \\ 1 & \Delta(\theta_1) = 0 \end{cases},$$

for some $S \subset \mathbb{R}^{p^4}$. We note that the value of R_2 is largely immaterial to our construction; we could even take $R_2 = 0$, but $R_2 > 0$ gives a continuous ℓ for a cleaner construction.

If θ_1 is far from S, ℓ is just $\ell_1(\theta_1)$. If θ_1 is in S, then ℓ is just $\ell_1(\theta_1) + p \cdot \ell_2(\theta_2)$. In between these two regimes, ℓ interpolates between these two loss functions; this interpolation is technically not necessary for our eventual theorem and proof, but gives a more realistic loss function. Note that ℓ_2 is non-positive, so having larger $q(\theta_1)$ (i.e., being in or close to S) is advantageous with respect to minimizing the term depending on θ_2.

In Figure 2 is an example of our eventual construction. S consists of two basins, centered at -0.5 and 0.5. If θ_1 is near one of these points, then ℓ is a quadratic centered at .005 with respect to θ_2. If θ_1 is far from these points, ℓ is a constant with respect to θ_2. So, if we start at the origin, using gradient-based methods we would first have to optimize θ_1 to get to one of the basins, and then optimize θ_2. With private data choosing the right basin is hard, with public data optimizing θ_2 within a basin is hard.



(a) (b)

Figure 2. (a) A 3-D visualization of the toy example of our construction for ℓ, for one-dimensional θ_1 and θ_2. (b) A heatmap of the same example.

The loss functions: In the initial stage of the algorithm (outside of S), the lower bound for private algorithm follows from the choice of $\ell_1(\theta_1; d_1) := \min\{(1/2)\|\theta_1 - d_1\|_2^2, \frac{9}{2}\}$ defined over the first p^4 coordinates. Note that as long as $\|\theta_1\|_2 \leq 2$ and d_1 is in $\mathcal{D}_1$, this is equivalent to a loss function of $\|\theta_1 - d_1\|_2^2$, i.e. we can still apply Lemma 1.2 to ℓ_1. The minimum is used in our upper bound to keep ℓ_1 bounded in the low-probability event that DP-SGD adds a large amount of noise to θ_1.

For any $\mathcal{A}_{priv}$, we define our basin to include the global minima of ℓ_1 on the distribution $\tau(\mathcal{A})$ in Lemma 1.2. Let $M = \Omega(1)$ be defined as in Lemma 1.2 for the case when dimension is p^4, $\varepsilon = 1$, and $n_{priv} = p^2$, and choose some R_1 (which can be thought of as the radius of our basin) such that $1/n \leq R_1 < M$. Since we know $|\|\theta^*(\tau(\mathcal{A}_{priv}))\| - M| = O(1/n)$, we define our basin S as

$$S := B_{p^4}(0, M + R_1) \setminus B_{p^4}(0, M - R_1) \,. \tag{2}$$

Note that S is the set of all points where ℓ_2-norm of θ_1 is close to M; the basin is a single non-convex set. Our construction can seamlessly generalize to the case where there are numerous disconnected basins to resemble more realistic landscapes. If R_1 is sufficiently large, then Lemma 1.2 guarantees that the population minimizer of ℓ_1 is contained in S and far from the boundary of S for distributions in $\mathcal{T}_1$ as defined in that lemma. Further, by a vector Azuma inequality (Hayes, 2003) the same is true of the empirical minimizer of ℓ_1 over the public data with high probability. We will specify a value of R_1 in Appendix C.1.

In the next stage of the algorithm (inside S), the loss is dominated by $\ell_2(\theta_2, d_2) := \min\{0, \frac{\|\theta_2 - d_2\|_2^2}{2r^2} - \frac{9}{2}\}$ where r is a parameter that scales the domain of ℓ_2. In particular, let $\mathcal{T}_2'$ be the set of p-dimensional data distributions over $\mathcal{D}_2' := B_p(0, r)$, and $\mathcal{T}_2'$ is defined by shrinking the support of each distribution in $\mathcal{T}_2$ (as defined in Lemma 1.3) by a factor of $r < 1$. We will specify the value of r in Appendix C.1; for now, one can think of $r \ll 1$. Since rescaling does not fundamentally change the problem, again Lemma 1.3 (up to a $1/r^2$ rescaling) holds also in $\mathcal{T}_2'$.

Note that as long as $\|\theta_2\|_2 \leq 2r$ and $d_2 \in \mathcal{D}_2'$, minimizing ℓ_2 is equivalent to minimizing $\frac{\|\theta_2 - d_2\|_2^2}{2r^2}$, which is just a rescaling of minimizing $\frac{\|\theta_2 - d_2\|_2^2}{2}$. In other words, we can still apply Lemma 1.3 to ℓ_2. Putting ℓ_1 and ℓ_2 together, our loss is defined in Eq. (1) with a choice of $R_2 < M - R_1$, which implies $q(0) = 0$; the exact value of R_2 is immaterial to our construction and eventual theorem statement.

Toy example of the loss function: In Figure 2 we provide a visualization of our loss $\ell(\cdot, d)$ for a single data point $d = (0.5, 0.005)$ as defined in Eq. (1) for $p = 1$. Here, to simplify the visualization we have chosen $r = 0.01$, $M = 0.5$, $R_1 = 0.1$, $R_2 = 0.2$, which may not correspond to the actual values we choose in our construction. This gives $S = [-0.6, -0.4] \cup [0.4, 0.6]$, and $q(\theta_1) = 0$ if $\theta_1 \in [-\infty, -0.8] \cup [-0.2, 0.2] \cup [0.8, \infty]$. Since $0.5 \in S$ and thus $q(0.5) = 1$, the minimizer is $(0.5, 0.005)$. We can observe the following.

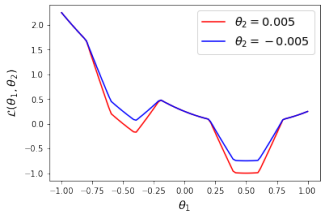


Figure 3. A projection of our two-dimensional toy example loss onto $\theta_2 = 0.005$ and $\theta_2 = -0.005$.

The first stage of the optimization (which corresponds to pretraining) tries to find the right part of the basin S with small $\ell_1(\theta_1)$. For a fixed θ_2, ℓ is a quadratic with respect to θ_1, except for the "wells" centered at $\theta_1 = 0.5$

and -0.5 (Figure 3). In our construction, the population minimizer of θ_1 would always be in one of the basins in $S = [-0.6, -0.4] \cup [0.4, 0.6]$. Note that the two basins are disconnected only because of the choice of $p = 1$.

The second stage of the optimization (which corresponds to fine-tuning) tries to minimize the second loss $\ell_2(\theta_2)$. For a fixed θ_1, ℓ is a quadratic with respect to θ_2. The strong convexity of this quadratic increases with q; when $q(\theta_1) = 0$ (e.g. at $\theta_1 = 0$) then ℓ is a constant with respect to θ_2.

In particular, we can see from Figure 2 that if we are at the origin, we can see that a (non-noisy) gradient step will only optimize over θ_1, but once θ_1 is inside S then gradient steps will optimize both θ_1 and θ_2. Furthermore, if we start at θ_1 in S and run (DP) gradient descent, the scale of θ_2, which is controlled by the choice of r in the definition of ℓ_2, is much smaller than the scale of θ_1. So it should be possible to optimize θ_2 using gradient descent once θ_1 is inside S, without causing θ_1 to move very far. This roughly corresponds to fine-tuning staying within a basin in our hypothesis.

2.3. Analysis

With the above construction, we formally guarantee that for certain sizes of public and private datasets, both datasets are necessary to optimize the loss to a desired level. We defer the proof to Appendix C.1.

Theorem 2.1. *For every integer $p \geq 1$, for some $r > 0$, $\mathcal{C} = \mathbb{R}^{p^4} \times \mathbb{R}^p, \mathcal{D} = B_{p^4}(0,1) \times B_p(0,r)$ there exists ℓ and a set of distributions $\mathcal{T}$ over $\mathcal{D}$ such that:*

(1) For $\delta = o(1/p^2)$, any $(1,\delta)$-DP algorithm $\mathcal{A}_{priv} : \mathcal{D}^{p^2} \to \mathcal{C}$, and any $\mathcal{A}_{pub} : \mathcal{D}^p \to \mathcal{C}$ there exists $\tau \in \mathcal{T}$ such that:

$$\mathbb{E}_{D \sim \tau^{p^2}, \theta \sim \mathcal{A}_{priv}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau)) + \Omega(1),$$

$$\mathbb{E}_{D \sim \tau^{p}, \theta \sim \mathcal{A}_{pub}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau)) + \Omega(1)$$

(2) For any $\delta \geq 2^{-p}$, there exists an algorithm $\mathcal{A}_{mixed} : \mathcal{D}^{p+p^2} \to \mathcal{C}$ which runs gradient descent on the first p examples, followed by $(1,\delta)$-DP-SGD on the last p^2 examples, such that for any $\tau \in \mathcal{T}$:

$$\mathbb{E}_{D \sim \tau^{n}, \theta \sim \mathcal{A}_{mixed}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau)) + O(1/p)$$

This demonstrates that there exist data distributions where a small number of public in-distribution data is necessary to achieve small loss, and pretraining on that public data is sufficient for DP-SGD to achieve the desired level of loss. The first part of the theorem shows that there are data distributions where neither a small-size, $n_{pub} = p$, public data or a large-size, $n_{priv} = p^2$, private data can reach the desired loss. However, on the same data distribution, pretraining on the small-size public data, followed by finetuning on the large-size private data, achieves a desired level, $O(1/p)$, of the excess loss.

Proof Sketch of Theorem 2.1. The high-level idea behind the construction is: Using private data alone cannot achieve risk $o(1)$ on ℓ_1, because ℓ_1 has a high dimension, but using public data can achieve risk $O(1/p)$ because the public mean estimation risk guarantees are dimension-independent. Similarly, using public data alone cannot achieve risk $o(1)$ on ℓ_2, because ℓ_2 has a multiplier of p and the amount of public data we are allowed to use is small. However, using private data can achieve risk $O(1/p)$ on ℓ_2 because ℓ_2 has low dimension, and there is more private data to use.

To prove (1) using these observations, we show that the risk guarantee of $\mathcal{A}$ on ℓ is at least its risk on ℓ_1 or ℓ_2 alone. If $\mathcal{A}_{pub}$ only uses public data, this implies a lower bound on $\mathcal{A}_{pub}$'s risk on ℓ from Lemma 1.3, which holds for some distribution $\tau_2 \in \mathcal{T}_2'$. Similarly, if $\mathcal{A}_{priv}$ only uses private data, this implies a lower bound on $\mathcal{A}_{priv}$'s risk on from Lemma 1.2, for some distribution $\tau_1 \in \mathcal{T}_1$. Then, the product distribution $\tau = \tau_1 \times \tau_2$ gives a simultaneous lower bound on the risk of $\mathcal{A}_{pub}$ and $\mathcal{A}_{priv}$, as desired.

To prove (2), we observe that a single step of (full-batch) gradient descent on the public data takes us the empirical minimizer of ℓ_1, which achieves risk $O(1/p)$ for ℓ_1. If we use an initialization such that $q(\theta_1) = 0$, a single step of gradient descent has no effect on θ_2, since the gradient of ℓ with respect to θ_2 at the initialization is zero. Furthermore, if R_1 is sufficiently large, then with high probability after this single step $\theta_1 \in S$ and is far from the boundary of S, i.e. $q(\theta_1) = 1$ and we have $\ell = \ell_1 + \ell_2$. Then, running DP-SGD with optimal parameters from this point will take θ_2 to a point achieving risk $O(1/p)$ on ℓ_2. However, DP-SGD will also move θ_1, which could worsen our risk on ℓ_1 substantially. We show that if r is sufficiently small, then for DP-SGD with optimal parameters, the amount by which θ_1 moves is $O(1/p)$, and in turn θ_1 remains in S and the risk guarantee on ℓ_1 does not worsen by more than $O(1/p)$. Then, our overall risk guarantee is at most the sum of the risk guarantee on ℓ_1 and ℓ_2 individually, which is $O(1/p)$. □

We note that, based on the above argument, we believe any algorithm has excess loss $\Omega(1/p)$ for some $\tau \in \mathcal{T}$ (i.e., $\mathcal{A}_{mixed}$ is minimax optimal for the set of distributions $\mathcal{T}$), although we do not have a formal proof of this claim.

2.4. Pretraining on out-of-distribution public data

A more common setting in practice is when out-of-distribution large-scale public data is used in pretraining,

as we surveyed in the introduction and at the beginning of Section 2. We modify our previous construction in Theorem 2.1 so that (i) there is a distribution mismatch between the public and private examples and (ii) an arbitrarily large amount, n_{pub}, of public data is available.

Theorem 2.2. *For every integer $p \geq 1$ and $n_{pub} \geq p$, for some $r > 0$, $\mathcal{C} = \mathbb{R}^{p^4} \times \mathbb{R}^p, \mathcal{D} = B_{p^4}(0,1) \times B_p(0,r)$ there exists ℓ and a set $\mathcal{T}$ of pairs of distributions $(\tau_{pub}, \tau_{priv})$ over $\mathcal{D}$ such that:*

(1) For $\delta = o(1/p^2)$, any $(1,\delta)$-DP algorithm $\mathcal{A}_{priv} : \mathcal{D}^{p^2} \to \mathcal{C}$, and any $\mathcal{A}_{pub} : \mathcal{D}^p \to \mathcal{C}$ there exists $(\tau_{pub}, \tau_{priv}) \in \mathcal{T}$ such that:

$$\mathbb{E}_{D \sim \tau_{priv}^{p^2}, \theta \sim \mathcal{A}_{priv}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau_{priv})) + \Omega(1),$$

$$\mathbb{E}_{D \sim \tau_{pub}^{n_{pub}}, \theta \sim \mathcal{A}_{pub}(D)} [\mathcal{L}(\theta)] = \mathcal{L}(\theta^*(\tau_{priv})) + \Omega(1)$$

(2) For any $\delta \geq 2^{-p}$, there exists an algorithm $\mathcal{A}_{mixed} : \mathcal{D}^{n_{pub}+p^2} \to \mathcal{C}$ which runs gradient descent on the first n_{pub} examples, followed by $(1,\delta)$-DP-SGD on the last p^2 examples, such that for any $\tau \in \mathcal{T}$:

$$\mathbb{E}_{D \sim \tau_{pub}^{n_{pub}} \times \tau_{priv}^{p^2}, \theta \sim \mathcal{A}_{mixed}(D)} [\mathcal{L}(\theta)]$$

$$= \mathcal{L}(\theta^*(\tau)) + O(1/p)$$

Here, $\mathcal{L}$ refers to the population loss over τ_{priv}.

This demonstrates that there exist data distributions where out-of-distribution public data is necessary to achieve small test loss on the target private task, and pretraining on the OOD public data is sufficient for DP-SGD to achieve the desired test loss. Note that all three cases are evaluated on the same private population loss, as is the case in real-world scenarios where we care about the performance on the private task.

We prove Theorem 2.2 in Appendix C.1 and give here a proof sketch for what modifications of Theorem 2.1 are needed. In particular, the value of d_2 in the public examples is irrelevant in the upper bound in Theorem 2.1. For example, we could have $d_2 = 0$ in all public examples, and the upper bound is unaffected. With the extra freedom we have in the construction under this distribution mismatch, showing the lower bound on the risk on the private task for algorithms using only public data is easy; the value of d_2 in the public examples encodes no information about the distribution of d_2 in the private examples, so clearly no algorithm with only access to public data can achieve good risk on ℓ_2 alone, regardless of how much public data it has access to.

Data abundance: If we had p^2 ID public examples or p^5 private examples in Theorem 2.1, we could achieve risk $O(1/p)$ in the above construction using only public data or only private data. Of course, if we also have the distribution mismatch in the preceding paragraph, no amount of public data achieves low risk on the private population. In light of this, Theorem 2.1 should not be interpreted as saying that both public and private data are strictly necessary to optimize some loss functions. Instead, a better interpretation might be that a small amount of public data greatly reduces the amount of private data needed to solve an optimization problem. This can be seen as theoretical backing for an empirical observation made in (Tramer & Boneh, 2020; De et al., 2022; Li et al., 2022b; Kerrigan et al., 2020).

Convex losses: Our construction is inherently non-convex, due to the term $q(\theta_1)$ we use to "activate" ℓ_2 only after optimizing over the public data. Surprisingly, in Appendix D we show that Theorem 2.1 can be proven even for (non-isotropic) quadratic losses, at the cost of operating in a constrained setting (i.e. $\mathcal{C}$ is finite). The constrained requirement is necessary since unlike in the construction in this section, we cannot guarantee that gradient descent on the public data does not affect θ_2. However, in the constrained setting we have the guarantee that θ_2 cannot leave the constraint set, so it is okay to take (arbitrarily large) public gradient steps that affect θ_2.

3. Experiments

In this section, we conduct experiments to verify our hypothesis about the two-stage optimization phenomenon.

3.1. CIFAR10 Experiments

Setup: For the ID public data experiment in Figure 4 (left), we train a ConvNet model on CIFAR10 using DP-SGD. We train for 60 epochs with a clipping norm of one, learning rate of 0.001, batch size of 256, and Adam optimizer. Simulating an ID public data setting, we split CIFAR10 (60,000 images) into a public dataset of size 2,000 and a private dataset of size 58,000. We use Adam optimizer with learning rate of 0.002 for the public dataset. For the large-size OOD public data in Figure 4 (right), we used 20,000 images from the training part of the CINIC10 images as the public data.

Results: In Figure 4, we allow a limited number of epochs T_{pub} on the public data. We show test accuracy as a function of t (the x-axis), which is the number of epochs used in public pretraining. The remaining $T_{pub} - t$ epochs are used in public post-training after the private training. For ID public data in the left panel, we choose $T_{pub} = 200$. Using this budget for pretraining has the highest accuracy. This demonstrates that the initial rounds of training are the most sensitive to noise, as is the case in both our hypothesis from

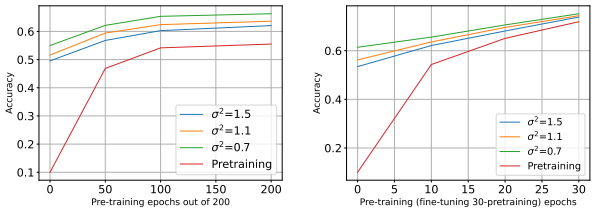


Figure 4. On CIFAR10, pretraining on the public data significantly improves accuracy compared to post-training on the public data for both ID public data (left) and OOD public data (right).

ε	first epoch	last epoch
1	46.7%± 0.3	46.3%± 0.3
3	49.6%± 0.6	48.0%± 0.5
8	54.0%± 0.8	52.0%± 0.9

Table 2. Effect of having higher budget ($\sigma^2 = 0.6$) on the first epoch compared to the last epoch on CIFAR10.

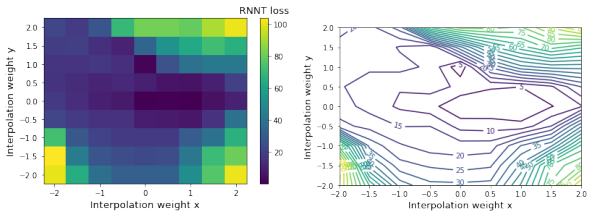
Section 1 and our theoretical construction in Section 2. Note that the benefits of longer pretraining is small after $t = 100$. It is possible that after around 100 epochs, pretraining converges to a good basin and the benefits of public pretraining plateaus afterwards. We see the same trend with OOD public data using CINIC10 dataset with $T_{pub} = 30$, shown in Figure 4 (right). Again, we observe that reducing privacy noise in the earlier rounds of training is more beneficial.

To further demonstrate the importance of the earlier iterations in the training, we designed an experiment where instead of using the same privacy budget for all iterations of private training, we train the first iteration with a lower noise multiplier (using more privacy budget) and compared it a setting where we train the last iteration with the lower noise multiplier. Table 2 compares the results for various choices of the end-to-end ε. Again, we observe that reducing privacy noise in the earlier rounds of training is more beneficial.

3.2. Manifold on Large Speech Model

Setup To better understand the geometry of the loss function for training machine learning models, we evaluate training a ConformerM (Gulati et al., 2020) model on Librispeech (Panayotov et al., 2015) dataset with/without public data pretraining using DP-Adam. Specifically, we train the following three models:

- **Oracle model:** We train a ConformerM model on the complete Librispeech dataset for 100k steps. This is considered as the global minima of the manifold.
- **Private model:** We train a ConformerM model on 90% samples drawn uniformly from the Librispeech dataset using DP-Adam for 20k steps.
- **Private model with public pretraining:** We pretrain a ConformerM model on the 10% of the samples with Adam for 10k steps and then fine-tune on the remaining 90% samples with privacy for 1k steps.



(a) (b)

Figure 5. Projected manifold of ConformerM on Librispeech by interpolating 3 models. (0, 0) is the oracle model, with weights w_1. (1, 0) is the private model with public pretraining, with weights w_2. (0, 1) is the private model, with weights w_3. Any other point (x, y) is the model with weights $(1-x-y)\cdot w_1 + x \cdot w_2 + y \cdot w_3$. We can tell that (0, 0) and (1, 0) are within the same basin while (0, 1) is in a different basin separated by a hill on the manifold. The manifold is constructed by RNNT loss (Graves, 2012) on a 128-sample subset of Librispeech's testother dataset.

Note that the hyper-parameters for the latter two settings are tuned to optimize the test word error rate under the same privacy budget $\varepsilon = 9.8$. We fix the privacy parameter δ to $1e$-6, ensuring that $\delta < n^{-1}$, where n is the number of private samples.

Results As shown in Figure 5, we interpolate the three models above to draw a projected slice of the manifold. From both the heatmap and the contour figures, we can tell that private model with public training falls into the same "basin" as the oracle model, which we refer to as the *global minima basin*. The private model without pretraining falls into a different basin, separated from the global minima basin by a "hill". This is evidence for our hypothesis that public pretraining is useful specifically because it picks a good basin. The ℓ_2-distance between the oracle model and the private model with public pretraining is 671.22, much smaller than the distance between the oracle model and the private model which is 1738.27. This parallels the construction in Section 2, in which private fine-tuning takes place on a smaller scale than pretraining on public data. We also altered the pre-training hyper-parameters of the private model with public pre-training and the results are similar (check Section E for more details).

4. Discussion

In this paper we show that there exists natural learning tasks where public data is necessary and sufficient to achieve a target accuracy under DP model training. However, while empirically we observed this is true for deep learning tasks, theoretically we only showed this for a carefully constructed example. It is an interesting research direction to try to close the gap between our theoretical example for a synthetic construction and our empirical observations on loss landscapes for deep learning models trained on standard benchmarks. Furthermore, we showed the necessity of public data is independent of whether the public data is in-distribution with the private training data or not. (Tramèr et al., 2022) discusses the perils of indiscriminate use of public data in DP training, few of which are: (i) Publicly available data does not necessarily mean that one can use that data set for training models without privacy consideration, as the trained model can release information from the data set verbatim, and (ii) Many existing empirical works on achieving better accuracy for DP training by using public data do not necessarily reflect realistic scenarios for model training. In particular, in real-world settings the available public data can be far out-of-distribution from the private data set. The authors provide prescriptive recommendations on being judicious in the choice of public data for DP training. Our work is complementary to (Tramèr et al., 2022), and we concur with all the concerns in their paper. Given our current impossibility result, and the concerns in (Tramèr et al., 2022), an important research question for future exploration is *given a public data set which may be far out-of distribution from the private training data, what is the best DP training procedure that exploits the public data set to obtain higher accuracy?* To the best of our knowledge, all current works (see Section 1 for reference) on the use of public data in DP training do not provide an answer.

References

Abadi, M., Chu, A., Goodfellow, I. J., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS'16), pp. 308–318, 2016.

Alon, N., Bassily, R., and Moran, S. Limits of private learning with access to public data. Advances in neural information processing systems, 32, 2019.

Amid, E., Ganesh, A., Mathews, R., Ramaswamy, S., Song, S., Steinke, T., Suriyakumar, V. M., Thakkar, O., and Thakurta, A. Public data-assisted mirror descent for private model training. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvári, C., Niu, G., and Sabato, S. (eds.), International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of Proceedings of Machine Learning Research, pp. 517–535. PMLR, 2022. URL https://proceedings.mlr.press/v162/amid22a.html.

Asi, H., Duchi, J., Fallah, A., Javidbakht, O., and Talwar, K. Private adaptive gradient methods for convex optimization. In International Conference on Machine Learning, pp. 383–392. PMLR, 2021a.

Asi, H., Levy, D. A. N., and Duchi, J. Adapting to function difficulty and growth conditions in private optimization. In Advances in Neural Information Processing Systems, 2021b.

Avent, B., Dubey, Y., and Korolova, A. The power of the hybrid model for mean estimation. Proceedings on Privacy Enhancing Technologies, 2020:48–68, 10 2020.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In Proc. of the 2014 IEEE 55th Annual Symp. on Foundations of Computer Science (FOCS), pp. 464–473, 2014.

Bassily, R., Feldman, V., Talwar, K., and Thakurta, A. G. Private stochastic convex optimization with optimal rates. In Advances in Neural Information Processing Systems 32, pp. 11279–11288, 2019.

Bassily, R., Cheu, A., Moran, S., Nikolov, A., Ullman, J., and Wu, S. Private query release assisted by public data. In III, H. D. and Singh, A. (eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of Proceedings of Machine Learning Research, pp. 695–703. PMLR, 13–18 Jul 2020a. URL https://proceedings.mlr.press/v119/bassily20a.html.

Bassily, R., Feldman, V., Guzmán, C., and Talwar, K. Stability of stochastic gradient descent on nonsmooth convex losses. Advances in Neural Information Processing Systems, 33:4381–4391, 2020b.

Bie, A., Kamath, G., and Singhal, V. Private estimation with public data. Advances in neural information processing systems 35, 2022.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258, 2021.

Bozinovski, S. and Fulgosi, A. The influence of pattern similarity and transfer learning upon training of a base perceptron b2. In Proceedings of Symposium Informatica, volume 3, pp. 121–126, 1976.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Differentially private bias-term only fine-tuning of foundation models. In Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022.

Bu, Z., Wang, Y.-X., Zha, S., and Karypis, G. Differentially private optimization on large model at small cost. arXiv preprint arXiv:2210.00038, 2022.

De, S., Berrada, L., Hayes, J., Smith, S. L., and Balle, B. Unlocking high-accuracy differentially private image classification through scale. arXiv preprint arXiv:2204.13650, 2022.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In Proc. of the Third Conf. on Theory of Cryptography (TCC), pp. 265–284, 2006. URL http://dx.doi.org/10.1007/11681878_14.

Feldman, V., Koren, T., and Talwar, K. Private stochastic convex optimization: Optimal rates in linear time. In Proc. of the Fifty-Second ACM Symp. on Theory of Computing (STOC'20), 2020.

Ganesh, A., Thakurta, A., and Upadhyay, J. On the universality of langevin diffusion for private euclidean (convex) optimization, 2022. URL https://arxiv.org/abs/2204.01585. https://openreview.net/forum?id=ZrJPdY5k6sg.

Ginart, A., van der Maaten, L., Zou, J., and Guo, C. Submix: Practical private prediction for large-scale language models. arXiv preprint arXiv:2201.00971, 2022.

Golatkar, A., Achille, A., Wang, Y.-X., Roth, A., Kearns, M., and Soatto, S. Mixed differential privacy in computer vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8376–8386, 2022.

Gopi, S., Lee, Y. T., and Liu, D. Private convex optimization via exponential mechanism. In Conference on Learning Theory, pp. 1948–1989. PMLR, 2022.

Graves, A. Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711, 2012.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. Conformer: Convolution-augmented transformer for speech recognition. Proc. Interspeech 2020, pp. 5036–5040, 2020.

Hardt, M., Recht, B., and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, ICML'16, pp. 1225–1234. JMLR.org, 2016.

Hayes, T. P. A large-deviation inequality for vector-valued martingales. 2003. URL http://agl.cs.unm.edu/~hayes/papers/VectorAzuma/VectorAzuma20030207.pdf.

He, J., Li, X., Yu, D., Zhang, H., Kulkarni, J., Lee, Y. T., Backurs, A., Yu, N., and Bian, J. Exploring the limits of differentially private deep learning with group-wise clipping. arXiv preprint arXiv:2212.01539, 2022.

Jiang, D., Zhang, G., Karami, M., Chen, X., Shao, Y., and Yu, Y. Dp^2-vae: Differentially private pre-trained variational autoencoders. arXiv preprint arXiv:2208.03409, 2022.

Kairouz, P., Ribero, M., Rush, K., and Thakurta, A. Fast dimension independent private adagrad on publicly estimated subspaces. arXiv preprint arXiv:2008.06570, 2020.

Kerrigan, G., Slack, D., and Tuyls, J. Differentially private language models benefit from public pre-training. In Proceedings of the Second Workshop on Privacy in NLP, pp. 39–45, 2020.

Kulkarni, J., Lee, Y. T., and Liu, D. Private non-smooth erm and sco in subquadratic steps. Advances in Neural Information Processing Systems, 34, 2021.

Kurakin, A., Chien, S., Song, S., Geambasu, R., Terzis, A., and Thakurta, A. Toward training at imagenet scale with differential privacy. arXiv preprint arXiv:2201.12328, 2022.

Li, X., Liu, D., Hashimoto, T., Inan, H. A., Kulkarni, J., Lee, Y., and Thakurta, A. G. When does differentially private learning not suffer in high dimensions? In Advances in Neural Information Processing Systems, 2022a.

Li, X., Tramèr, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. In International Conference on Learning Representations, 2022b.

Liu, T., Vietri, G., Steinke, T., Ullman, J., and Wu, S. Leveraging public data for practical private query release. In International Conference on Machine Learning, pp. 6968–6977. PMLR, 2021.

Luo, Z., Wu, D. J., Adeli, E., and Fei-Fei, L. Scalable differential privacy with sparse network finetuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5059–5068, 2021.

Nan, L., Radev, D., Zhang, R., Rau, A., Sivaprasad, A., Hsieh, C., Tang, X., Vyas, A., Verma, N., Krishna, P., et al. Dart: Open-domain structured data record to text generation. arXiv preprint arXiv:2007.02871, 2020.

Nasr, M., Mahloujifar, S., Tang, X., Mittal, P., and Houmansadr, A. Effectively using public data in privacy preserving machine learning, 2022. https://openreview.net/pdf?id=5R96mIU85IW.

Novikova, J., Dušek, O., and Rieser, V. The e2e dataset: New challenges for end-to-end generation. arXiv preprint arXiv:1706.09254, 2017.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: an asr corpus based on public domain audio books. In 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp. 5206–5210. IEEE, 2015.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. OpenAI blog, 1(8):9, 2019.

Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. Cnn features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp. 806–813, 2014.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In 2013 IEEE Global Conference on Signal and Information Processing, pp. 245–248. IEEE, 2013.

Steinke, T. and Ullman, J. Tight lower bounds for differentially private selection. In 2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS), pp. 552–563. IEEE, 2017.

Tramer, F. and Boneh, D. Differentially private learning needs better features (or much more data). In International Conference on Learning Representations, 2020.

Tramèr, F., Kamath, G., and Carlini, N. Considerations for differentially private learning with large-scale public pretraining. arXiv preprint arXiv:2212.06470, 2022.

Yu, D., Zhang, H., Chen, W., and Liu, T.-Y. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In International Conference on Learning Representations, 2021.

Yu, D., Naik, S., Backurs, A., Gopi, S., Inan, H. A., Kamath, G., Kulkarni, J., Lee, Y. T., Manoel, A., Wutschitz, L., et al. Differentially private fine-tuning of language models. In International Conference on Learning Representations, 2022.

Zhou, Y., Wu, S., and Banerjee, A. Bypassing the ambient dimension: Private sgd with gradient subspace identification. In International Conference on Learning Representations, 2021.

A. Notation Reference

Notation	Meaning
$\mathcal{A}$	algorithm, i.e. a randomized map from datasets to outputs
$B_p(c, r)$	p-dimensional ball of radius r centered at c
$\mathcal{C}$	constraint set
D	data set ($\in \mathcal{D}^*$)
d	(singular) data point ($\in \mathcal{D}$)
$\mathcal{D}$	data domain
ε, δ	privacy parameters
η	step size in gradient descent
ℓ	(per-example) loss function
$\mathcal{L}$	population loss function
n	dataset size
$N(\mu, \Sigma)$	normal distribution with mean μ and covariance matrix Σ
p	dimension
Π	projection operator
q	the "activation function" in Section 2
R, r	radii of various sets in our construction
τ	data (population) distribution
$\mathcal{T}$	set of data distributions
θ	model
$\theta^*(\tau)$	population minimizer on the distribution τ

Table 3. Summary of notation

In Table 3, we give a summary of the notation used throughout the paper.

B. Survey of the gain of pretraining

The stark difference in the gain of public pretraining between private and non-private model training has been widely observed in several tasks both in natural language and vision.

Table-To-Text Generation. The experiment details can be found in (Li et al., 2022b).

	BLEU			ROUGE-L		
ε	∞	8	3	∞	8	3
with public pretrain	69.46	63.19	61.52	71.36	66.43	65.67
without public pretrain	65.73	24.25	15.46	68.75	39.95	35.24
gain of public pretraining	3.73	38.94	46.06	2.61	26.48	30.43

Table 4. BLEU and ROUGE-L scores for generating natural language descriptions of table entries on E2E dataset (Novikova et al., 2017) reported in (Li et al., 2022b, Table 2) with $\delta = 10^{-5}$. The pretrained model in the first row is GPT-2.

Image classification. The experimental details can be found in (De et al., 2022).

C. Missing Details from Section 2

Before turning to the proof, we fill in the details of the construction given in Section 2: We choose $R_1 = 1/p^2 + \kappa \log(p)/\sqrt{p}$, where κ is a sufficiently large constant. Any $R_2 < M - R_1$ suffices for our proof. We choose $r = O(\frac{1}{p^{5/2}\sqrt{\log(1/\delta)}})$. We formally define the range of data distributions we use in our construction as a set of products of two distributions: $\mathcal{T} := \{\tau_1 \times \tau_2 | \tau_1 \in \mathcal{T}_1, \tau_2 \in \mathcal{T}_2'\}$, where $\mathcal{T}_1$ is defined as in Lemma 1.2, and $\mathcal{T}_2$ is defined as in Section 2.

	BLEU			ROUGE-L		
ε	∞	8	3	∞	8	3
with public pretrain	42.78	35.06	31.03	56.72	54.58	52.06
without public pretrain	26.79	7.77	3.00	37.86	21.68	17.14
gain of public pretraining	15.99	27.29	28.03	18.86	32.90	34.92

Table 5. BLEU and ROUGE-L scores for generating natural language descriptions of table entries on DART dataset (Nan et al., 2020) reported in (Li et al., 2022b, Table 8) with $\delta = 10^{-5}$. The pretrained model in the first row is GPT-2.

	CIFAR-10				ImageNet			
ε	8	4	2	1	8	4	2	1
with public pretrain	96.7	96.1	95.4	94.7	81.8	79.2	74.7	70.3
without public pretrain	81.4	73.5	65.9	56.8	32.4	–	–	–
gain of public pretraining	15.3	22.6	29.5	37.9	49.4	–	–	–

Table 6. Test accuracy for image classification on CIFAR-10 and ImageNet datasets reported in (De et al., 2022, Table 1) with $\delta = 10^{-5}$ and $8 \cdot 10^{-7}$, respectively. The pretraining public data for CIFAR-10 is ImageNet and for ImageNet is JFT-4B. Without pretraining, private training on ImageNet failed to converge, indicated by $-$.

C.1. Proof of Theorem 2.1

In our proof we will use DP-SGD as instantiated in (Bassily et al., 2014). Combined with results on uniform stability of gradient descent on strongly convex losses (see e.g. (Hardt et al., 2016)), Theorem 2.4 of (Bassily et al., 2014) and its proof implies the following:

Theorem C.1. *Suppose ℓ has Hessian $m\mathbb{I}_p$ and for any d, d', $\|\nabla\ell(\theta; d) - \nabla\ell(\theta; d')\|_2 \leq L$. Then for $T = n^2$, $\eta_t = \frac{1}{mt}$, $\sigma^2 = O(\frac{L^2 \log(1/\delta)}{\varepsilon^2 n^2})$, if $\|\theta_0 - \theta^*\|_2 \leq O(L/m)$ running T steps of DP-SGD with step size η_t in iteration t and variance σ^2 is (ε, δ)-DP and achieves population loss:*

$$O\left(\frac{L^2 p \log(n)\log(1/\delta)}{m\varepsilon^2 n^2} + \frac{L^2}{mn}\right).$$

We also have the following lemma, which effectively says that unconstrained and DP-SGD stays within a ball with high probability.

Lemma C.2. *With probability $1 - T2^{-\Omega(p)}$ over (unconstrained) DP-SGD using the parameters in Theorem C.1, for all $0 \leq t \leq T$, we have $\|\theta_t - \theta^*\|_2 \leq \max\{2\sqrt{p}\sigma/m, \|\theta_0 - \theta^*\|_2\}$.*

Proof. By a multivariate Gaussian tail bound, w.p. $1 - T2^{-\Omega(p)}$ in each iteration of DP-SGD the noise we add has ℓ_2-norm at most $2\sqrt{p}\eta_t\sigma$. Conditioned on this event, since we have an identity quadratic loss and $\eta_t \leq 1/m$ for all t, each step of gradient descent is $(1 - m\eta_t)$ contractive. So we have:

$$\forall t : \|\theta_t - \theta^*\|_2 \leq (1 - m\eta_t)\|\theta_{t-1} - \theta^*\|_2 + 2\sqrt{p}\eta_t\sigma.$$

The lemma follows by induction. □

Proof of Theorem 2.1. We use S, ℓ as defined in Section 2, (and the associated definitions of $\mathcal{D}, \mathcal{T}, q$, etc.).

We will prove (1) in two parts. First, we will show that for any $\mathcal{A}_{priv}$, there exists $\tau_1 \in \mathcal{T}_1$ such that the desired lower bound holds for $\tau_1 \times \tau_2$ for all $\tau_2 \in \mathcal{T}_2'$. Second, we show that for any $\mathcal{A}_{pub}$ there exists $\tau_2 \in \mathcal{T}_2'$ such that the desired lower bound holds for $\tau_1 \times \tau_2$ for all $\tau_1 \in \mathcal{T}_1$. Then taking τ_1 from the first statement and τ_2 from the second statement, both lower bounds hold for $\tau_1 \times \tau_2$ as desired.

Proof of (1) for $\mathcal{A}_{priv}$: Let τ_2 be an arbitrary, fixed member of $\mathcal{T}_2'$. Fix any $\mathcal{A}_{priv}$ and take any distribution $\tau_1 \in \mathcal{T}_1$. Let $\tau(\tau_1) \in \mathcal{T}$ be the distribution over (d_1, d_2) given by sampling $d_1 \sim \tau_1$ and $d_2 \sim \tau_2$. Let $\mathcal{L}$ refer to the population loss

over ℓ, and let $\mathcal{L}_1$ refer to the population loss over $\ell_1(\theta_1, d_1)$ for $d_1 \sim \tau_1$. Consider the following algorithm $\mathcal{A}'_{priv}$ for minimizing ℓ_1 given p^2 samples from τ_1: For each of these samples d_1, $\mathcal{A}'_{priv}$ draws an i.i.d. sample d_2 from τ_2 and pads d_1 with d_2, giving p^2 i.i.d samples from $\tau(\tau_1)$. $\mathcal{A}'_{priv}$ then runs $\mathcal{A}_{priv}$ on these samples, and takes θ_1 from the output of $\mathcal{A}_{priv}$. Note that $\mathcal{A}'_{priv}$ is allowed to know the distribution τ_2 since it is fixed and independent of the data $\mathcal{A}_{priv}$ receives. We observe a few facts about $\mathcal{L}$. First:

$$\mathcal{L}((\theta_1, \theta_2)) - \mathcal{L}((\theta_1, \theta'_2)) = q(\theta_1)(\mathcal{L}_2(\theta_2) - \mathcal{L}_2(\theta'_2)). \tag{3}$$

Since q is non-negative, this gives:

$$\mathcal{L}_2(\theta_2) \leq \mathcal{L}_2(\theta'_2) \leftrightarrow \mathcal{L}((\theta_1, \theta_2)) \leq \mathcal{L}((\theta_1, \theta'_2)) \tag{4}$$

This implies that replacing θ_2 with the population minimizer of $\mathcal{L}_2$ can only improve our risk on ℓ. Next, note that for any $\tau_1 \in \mathcal{T}_1$, since its population minimizer is in S, $q(\theta^*(\tau_1)) = 1$. In turn, $\theta^*(\tau_1 \times \tau_2) = (\theta^*(\tau_1), \theta^*(\tau_2))$ for all $\tau_1 \in \mathcal{T}_1$. Finally, since q is in $[0, 1]$ and $\mathcal{L}_2$ is non-positive this gives:

$$\mathcal{L}((\theta_1, \theta^*(\tau_2))) - \mathcal{L}(\theta^*(\tau(\tau_1))) = \mathcal{L}_1(\theta_1) - \mathcal{L}_1(\theta^*(\tau_1)) - (1 - q(\theta_1)) \cdot \mathcal{L}_2(\theta^*(\tau_2)) \geq \mathcal{L}_1(\theta_1) - \mathcal{L}_1(\theta^*(\tau_1)). \tag{5}$$

In other words, if we choose θ_2 to be the minimizer of $\mathcal{L}_2$, then our risk on $\mathcal{L}$ is at least our risk on $\mathcal{L}_1$ alone. Putting it all together:

$$\begin{aligned}
&\mathbb{E}_{D \sim \tau(\tau_1)^{p^2}} \left[\mathbb{E}_{\theta \sim \mathcal{A}_{priv}(D)} \left[\mathcal{L}(\theta) \right] - \mathcal{L}(\theta^*(\tau(\tau_1))) \right] \\
&\overset{(4)}{\geq} \mathbb{E}_{D \sim \tau(\tau_1)^{p^2}} \left[\mathbb{E}_{(\theta_1, \theta_2) \sim \mathcal{A}_{priv}(D)} \left[\mathcal{L}((\theta_1, \theta^*(\tau_2))) \right] - \mathcal{L}(\theta^*(\tau(\tau_1))) \right] \\
&\overset{(5)}{\geq} \mathbb{E}_{D \sim \tau_1^{p^2}} \left[\mathbb{E}_{\theta_1 \sim \mathcal{A}'_{priv}(D)} \left[\mathcal{L}_1(\theta_1) \right] - \mathcal{L}_1(\theta^*(\tau_1)) \right].
\end{aligned}$$

Using Lemma 1.2, since we are solving a p^4-dimensional mean estimation problem with p^2 samples and $(1, o(1/n)$-DP, we know that the final expression (the risk of $\mathcal{A}'_{priv}$) is $\Omega(1)$ for some $\tau_1 \in \mathcal{T}_1$, which implies the same lower bound on the risk of $\mathcal{A}_{priv}$ for $\tau_1 \times \tau_2$.

Proof of (1) for $\mathcal{A}_{pub}$: Fix an arbitrary $\tau_1 \in \mathcal{T}_1$. Let $\tau(\tau_2)$ denote $\tau_1 \times \tau_2$. Given any $\mathcal{A}_{pub}$, consider $\mathcal{A}'_{pub}$ that takes p samples from τ_2, pads them with i.i.d. samples from τ_1 to get p samples from $\tau(\tau_2)$. It then runs $\mathcal{A}_{pub}$ on these samples, clips the norm of the θ_2 in $\mathcal{A}_{pub}$'s output to be at most r, and uses this as its output.

We again make some observations on ℓ. First, since the population minimizer $\theta^*(\tau_1)$ is always in S by definition, we have $q(\theta^*(\tau_1)) = 1$. Then, since ℓ_2 is non-positive:

$$\mathcal{L}(\theta_1, \theta_2) \geq \mathcal{L}(\theta^*(\tau_1), \theta_2) \tag{6}$$

Next, by definition of ℓ_2 and non-expansiveness of Euclidean projection, clipping θ_2 to a ball of radius r can only decrease the loss, i.e., if we define $\textsc{clip}(\theta_2, r) := \frac{\theta_2}{\max\{1, \|\theta_2\|_2 / r\}}$:

$$\mathcal{L}_2\left(\textsc{clip}(\theta_2, r)\right) \leq \mathcal{L}_2(\theta_2). \tag{7}$$

Then we have:

$$\begin{aligned}
&\mathbb{E}_{D \sim \tau(\tau_2)^p} \left[\mathbb{E}_{\theta \sim \mathcal{A}_{pub}(D)} \left[\mathcal{L}(\theta) \right] - \mathcal{L}(\theta^*(\tau(\tau_2))) \right] \\
&\overset{(6)}{\geq} \mathbb{E}_{D \sim \tau(\tau_2)^p} \left[\mathbb{E}_{(\theta_1, \theta_2) \sim \mathcal{A}_{pub}(D)} \left[\mathcal{L}((\theta^*(\tau_1), \theta_2)) \right] - \mathcal{L}(\theta^*(\tau(\tau_2))) \right] \\
&\overset{(3)}{=} \mathbb{E}_{D \sim \tau(\tau_2)^p} \left[\mathbb{E}_{(\theta_1, \theta_2) \sim \mathcal{A}_{pub}(D)} \left[\mathcal{L}_2(\theta_2) \right] - \mathcal{L}_2(\theta^*(\tau_2)) \right]
\end{aligned}$$

$$\overset{(7)}{\geq} \mathbb{E}_{D\sim\tau_2^p}\left[\mathbb{E}_{\theta_2\sim\mathcal{A}'_{pub}(D)}\left[\mathcal{L}_2(\theta_2)\right] - \mathcal{L}_2(\theta^*(\tau_2))\right].$$

Since θ_2 in the last line has norm at most r, the last expression (the risk of $\mathcal{A}'_{pub}$ on ℓ_2) is equal to p times the risk of $\mathcal{A}'_{pub}$ on a rescaling of the mean-estimation problem in Lemma 1.3, which is $\Omega(1)$ for some $\tau_2 \in \mathcal{T}'_2$. This implies the same lower bound on the risk of $\mathcal{A}_{pub}$ for $\tau_1 \times \tau_2$.

Proof of (2): We initialize $\theta_1 = \theta_2 = 0$ for simplicity[1]. Then, note that the term $p \cdot q(\theta_1) \cdot \ell_2(\theta_2)$ is a constant in the region where $q(\theta_1) = 0$, which includes the origin. So, the gradient of this term is 0 at the origin, and a single step of gradient descent on the public data sets θ_1 to the empirical minimizer of ℓ_1 (which achieves risk $O(1/p)$ on ℓ_1 alone in expectation), and does not affect θ_2. We will then run DP-SGD from this point.

By a vector Azuma inequality, with probability at least $1 - p^{-\Omega(\kappa)}$, $\theta_1 \in S$ and is distance at least $\Omega(\frac{\log p}{\sqrt{p}})$ from the boundary of S. In the $p^{-\Omega(\kappa)}$ probability event this does not happen, since both ℓ_1 and ℓ_2 take on values in an interval of length $O(p)$, our risk is at most $O(p)$, and so for sufficiently large constant κ the contribution of this event to our expected risk is negligible. So we just need to show our overall risk is $O(1/p)$ in expectation when after a single step of gradient descent on the public data, $\theta_1 \in S$ and is distance $\Omega(\frac{\log p}{\sqrt{p}})$ from the boundary.

Note that as long as $\theta_1 \in S$, $q(\theta_1) = 1$ and thus the Lipschitzness of ℓ with respect to θ_1 is $O(1)$. We will argue that if r is sufficiently small, then θ_1 does not move by more than $O(1/p)$ with probability $1 - p^{-\Omega(1)}$ (and as before, if this high probability event does not occur, the contribution to the risk is negligible). As long as θ_1 does not move by more than $O(1/p)$ while we run DP-SGD, it will remain in S since we assume at the start of DP-SGD, θ_1 is distance $\Omega(\frac{\log p}{\sqrt{p}})$ from the boundary of S. Putting it all together, this implies that (i) the excess loss on ℓ_1 does not increase by more than $O(1/p)$, since ℓ_1 is $O(1)$-Lipschitz with respect to θ_1, and (ii) since θ_1 stays within S and thus $q(\theta_1) = 1$, the change in θ_2 is the same as the change if we ran DP-SGD on $\ell_2(\theta_2)$ alone. Lemma C.2 implies that θ_2 stays within $B_p(0, 2r)$, and thus DP-SGD on $\ell(\theta_2)$ is the same as running DP-SGD on the purely quadratic loss $\frac{p}{2r^2}\|\theta_2 - d_2\|_2^2$, with high probability. So by Theorem C.1, DP-SGD with optimal parameters will give θ_2 achieving risk $\widetilde{O}(1/p^2)$ on ℓ_2 alone. This gives an overall risk bound of $O(1/p)$, completing the proof.

Now, the idea is that in DP-SGD as in Theorem C.1, the total movement of θ_1 due to both gradient steps and noise is an increasing function of r, so we can set r to be sufficiently small to guarantee θ_1 does not move by more than $O(1/p)$. Specifically, as long as $\theta_1 \in S$ we have $\ell = \ell_1 + \ell_2$, and so the loss ℓ satisfies $\|\nabla\ell(\theta; d) - \nabla\ell(\theta; d')\|_2 = O(p/r)$ for all d, d' (the gradient difference bound on ℓ_2) as long as $\theta_1 \in S$. We use the optimal setting of parameters in DP-SGD corresponding to $L = O(\frac{p}{r}), m = \frac{p}{r^2}$, noting that the initialization condition in Theorem C.1 is satisfied by $\theta_2 = 0$. If we plug these into the parameter settings in Theorem C.1, we get that we should use $T = \Theta(p^2)$ iterations with step-size $\eta_t = \frac{r^2}{pt}$ and per-iteration variance $\sigma = \Theta(\frac{\sqrt{\log(1/\delta)}}{r^2})$, and achieves risk $\widetilde{O}(1/p)$ on ℓ_2 alone. Then, the movement of θ_1 due to unnoised gradients in DP-SGD is at most $O(1) \cdot \sum_t \eta_t = \Theta(r^2 p \log p)$ (here we use the fact that the Lipschitz constant with respect to θ_1 is $O(1)$ within S, not $O(p/r)$), and with high probability the movement due to the noise is $O(\sqrt{p^4}\sqrt{\sum_t \eta_t^2}\sigma) = O(p^{3/2} r\sqrt{\log(1/\delta)})$. So if $r = O(\frac{1}{p^{5/2}\sqrt{\log(1/\delta)}})$, we get the desired upper bound on the movement of θ_1 during DP-SGD. □

Proof of Theorem 2.2. The proof is almost exactly the same as Theorem 2.1, so we only highlight the changes to that proof.

We define $\mathcal{T}$ similarly to Theorem 2.1: For each $\tau = \tau_1 \times \tau_2$ in $\mathcal{T}$ as defined in Theorem 2.1, we replace it with $(\tau_{pub} = \tau_1 \times Z, \tau_{pub} = \tau_1 \times \tau_2)$ where Z is a point distribution on the origin.

Now, the lower bound in on $\mathcal{A}_{priv}$ in (1) can be proven exactly the same as in Theorem 2.1 since we're using the same set of private distributions. The lower bound on $\mathcal{A}_{pub}$ follows similarly to Theorem 2.1, since we proved it holds for $\tau_1 \times \tau_2$ where τ_1 can be arbitrary. Alternatively, one can note that $\mathcal{A}_{pub}$ learns no information about τ_2 from the public data, so it cannot do better on ℓ_2 than outputting a fixed point, which has risk $\Omega(1)$

The upper bound in (2) follows since the algorithm only evaluates gradients on the public data where $q(\theta_1) = 0$, i.e. it never uses the coordinates in the public data that are changed between this theorem and Theorem 2.1. □

[1] As long as $q(\theta_1) = 0$ and $\|\theta_2\|_2 \leq r$ initially with high probability, the proof still goes through.

D. Quadratic Example

In this section, we show that our construction holds even if the loss function is quadratic, as long as we are okay with using a constrained optimization problem.

Theorem D.1. *For every integer $p \geq 1$, for $\mathcal{C} = \mathcal{D} = B_{p^4}(0,1) \times B_p(0,1)$ there exists ℓ such that:*

(1) For $\delta = o(1/p^2)$, any (non-private) algorithm $\mathcal{A}_{pub} : \mathcal{D}^p \to \mathcal{C}$, and any $(1,\delta)$-DP algorithm $\mathcal{A}_{priv} : \mathcal{D}^{p^2} \to \mathcal{C}$ there exists τ such that:

$$\mathbb{E}_{D_{priv} \sim \tau^{p^2}} \left[\mathbb{E}_{\theta \sim \mathcal{A}_{priv}(D_{priv})} \left[\mathcal{L}(\theta) \right] - \mathcal{L}(\theta^*(\tau(\mathcal{A}_{priv}))) \right] = \Omega(1)$$

$$\mathbb{E}_{D_{pub} \sim \tau^{p}} \left[\mathbb{E}_{\theta \sim \mathcal{A}_{pub}(D_{pub})} \left[\mathcal{L}(\theta) \right] - \mathcal{L}(\theta^*(\tau(\mathcal{A}_{pub}))) \right] = \Omega(1)$$

(2) For any $\delta \geq 2^{-p}$, there exists an algorithm $\mathcal{A}_{mixed} : \mathcal{D}^{p+p^2} \to \mathcal{C}$ which runs projected gradient descent on the first p examples, followed by $(1,\delta)$-DP-SGD on the last p^2 examples, such that for any τ:

$$\mathbb{E}_{D \sim \tau^{p+p^2}} \left[\mathbb{E}_{\theta \sim \mathcal{A}_{mixed}(D)} \left[\mathcal{L}(\theta) \right] - \mathcal{L}(\theta^*(\tau)) \right] = O(1/p)$$

Proof. We will first state ℓ and then prove each item in the theorem statement. We use the following construction: $\mathcal{C} = \mathcal{D} = B_{p^4}(0,1) \times B_p(0,r)$. Let (θ_1, θ_2) denote an element of $\mathcal{C}$, with $\theta_1 \in B_{p^4}(0,1)$ and $\theta_2 \in B_p(0,r)$ for $r = O(\frac{1}{p^{5/2}\sqrt{\log(1/\delta)}})$, and similarly with (d_1, d_2) and $\mathcal{D}$. We let $\ell((\theta_1,\theta_2),(d_1,d_2)) = \frac{1}{2}\left\|\theta_1 - d_1\right\|_2^2 + \frac{p}{2r^2}\left\|\theta_2 - d_2\right\|_2^2$.

As in the proof of Theorem 2.1, we will show (1) in two parts: for any $\mathcal{A}_{priv}$, there exists $\tau_1 \in \mathcal{T}_1$ such that the desired lower bound holds for $\tau_1 \times \tau_2$ for all $\tau_2 \in \mathcal{T}_2'$, and that for any $\mathcal{A}_{pub}$ there exists $\tau_2 \in \mathcal{T}_2'$ such that the desired lower bound holds for $\tau_1 \times \tau_2$ for all $\tau_1 \in \mathcal{T}_1$.

Proof of (1) for $\mathcal{A}_{priv}$: Fix $\mathcal{A}_{priv}$ and take any distribution τ_1 over $B_{p^4}(0,1)$. Let $\tau(\tau_1)$ be the distribution over (d_1, d_2) given by sampling $d_1 \sim \tau_1$ and letting d_2 be the origin with probability 1. Let $\mathcal{L}$ refer to the population loss over ℓ, and let $\mathcal{L}_1$ refer to the population loss over $\ell_1(\theta_1) := \left\|\theta_1 - d_1\right\|_2^2, d_1 \sim \tau_1$.

Now, consider an algorithm $\mathcal{A}'_{priv}$ that takes p^2 samples from τ_1, pads them with the origin to get p^2 samples from $\tau(\tau_1)$ in the preceding paragraph, runs $\mathcal{A}_{priv}$ on these samples, and then takes θ_1 from the output of $\mathcal{A}_{priv}$. Notice that:

$$\begin{aligned}
&\mathbb{E}_{D \sim \tau(\tau_1)^{p^2}} \left[\mathbb{E}_{\theta \sim \mathcal{A}_{priv}(D)} \left[\mathcal{L}(\theta) \right] - \mathcal{L}(\theta^*(\tau(\tau_1))) \right] \\
&\geq \mathbb{E}_{D \sim \tau(\tau_1)^{p^2}} \left[\mathbb{E}_{(\theta_1,\theta_2) \sim \mathcal{A}_{priv}(D)} \left[\mathcal{L}((\theta_1, 0)) \right] - \mathcal{L}(\theta^*(\tau(\tau_1))) \right] \\
&= \mathbb{E}_{D \sim \tau_1^{p^2}} \left[\mathbb{E}_{(\theta_1,\theta_2) \sim \mathcal{A}_{priv}(D), d_1 \sim \tau_1} \left[\frac{1}{2}\left\|\theta_1 - d_1\right\|_2^2 \right] - \mathcal{L}_1(\theta^*(\tau_1)) \right] \\
&= \mathbb{E}_{D \sim \tau_1^{p^2}} \left[\mathbb{E}_{\theta_1 \sim \mathcal{A}'_{priv}(D)} \left[\mathcal{L}_1(\theta_1) \right] - \mathcal{L}_1(\theta^*(\tau_1)) \right].
\end{aligned}$$

By Lemma 1.2, the final expression is $\Omega(1)$ for some distribution $\tau_1(\mathcal{A}_{priv})$. In turn, for the corresponding $\tau(\tau_1(\mathcal{A}_{priv}))$, $\mathcal{A}_{priv}$ has excess population loss $\Omega(1)$ in expectation as desired.

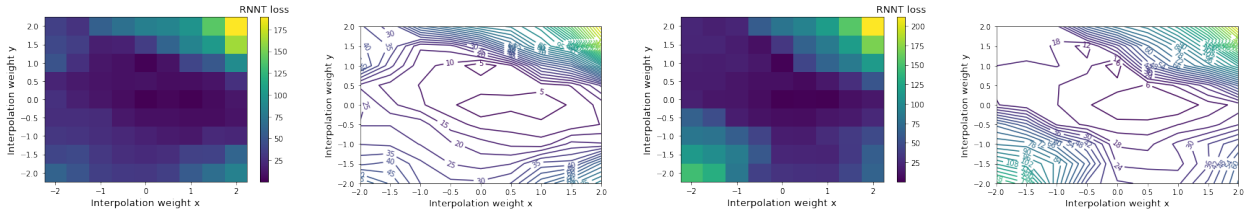
Proof of (1) for $\mathcal{A}_{pub}$: This follows by an argument symmetric to the previous part, except we use Lemma 1.3 instead of Lemma 1.2, and the observation that minimizing $\frac{p}{2r^2}\left\|\theta_2 - d_2\right\|_2^2$ is equivalent to minimizing $\frac{p}{2}\left\|\theta_2 - d_2\right\|_2^2$ over $B_p(0,1)$. In particular, the lower bound on just $\left\|\theta_2 - d_2\right\|_2^2$ given by Lemma 1.2 is $\Omega(1/p)$, and the lower bound of $\Omega(1)$ on $\mathcal{L}$ follows after using the same reduction as in the proof of (1) and taking into account the multiplier $\frac{p}{2}$.

Proof of (2): This follows similarly to Theorem 2.1, so we only highlight the high-level proof and major changes here. A single step of projected gradient descent on the public data gets us to the empirical minimizer of θ_1, which achieves excess risk $O(1/p)$ on $\frac{1}{2}\left\|\theta_1 - d_1\right\|_2^2$. Then, since we are using projected gradient descent, we know θ_2 is distance $O(r)$ from the population minimizer of θ_2, so projected DP-SGD on the private data gets to a point which achieves risk $O(1/p)$ on $\frac{p}{2r^2}\left\|\theta_2 - d_2^2\right\|_2$. By a similar argument to Theorem 2.1, projected DP-SGD does not cause θ_1 to move by more than $O(1/p)$ with high probability if $r = O(\frac{1}{p^{5/2}\sqrt{\log(1/\delta)}})$. □

If we want to take this same example and make it unconstrained, an issue arises: A single step of gradient step with step size 1 will cause θ_2 to move by $1/r^2$, which is far larger than the radius of the ball that θ_2 was restricted to in the constrained setting. In turn, the DP-SGD guarantees worsened. We can remedy this by taking smaller step sizes on the public data so that each step is non-expansive, i.e. θ_2 does not leave the ball and the DP-SGD guarantees still hold. However, in order to do so we need to use step sizes where $\eta = O(r^2)$, which means we will need to take $\Omega(1/r^2)$ steps in order to reduce our distance to the minimizing θ_1 by a constant. Since r is being set to a small value, this is a large number of steps. In other words, it is possible to take this example and make it unconstrained, while still satisfying that public-then-private gradient descent achieves the desired excess loss, but the algorithm will not be efficient.

E. Additional Experiments for Section 3.2

To validate whether Figure 5 is a persistent phenomenon, we altered the public pre-training hyper-parameters and conducted two more sets of experiments. After changing the pre-training sample size and pre-training steps, we observe similar results as shown in Figure 6.



(a) (b) (c) (d)

Figure 6. Projected manifold of ConformerM on Librispeech by interpolating 3 models. In 6(a) and 6(b), the fine-tuned model is pre-trained on 1% of the training samples. In 6(c) and 6(d), the fine-tuned model is pre-trained for only 5k steps. Other hyper-parameters are exactly the same as Figure 5.