Public Data-Assisted Mirror Descent for Private Model Training

Ehsan Amid [1] **Arun Ganesh** [2 3] **Rajiv Mathews** [1] **Swaroop Ramaswamy** [1] **Shuang Song** [1] **Thomas Steinke** [1]
Vinith M. Suriyakumar [4 3] **Om Thakkar** [1] **Abhradeep Thakurta** [1]

Abstract

In this paper, we revisit the problem of using *in-distribution* public data to improve the privacy/utility trade-offs for differentially private (DP) model training. (Here, public data refers to auxiliary data sets that have no privacy concerns.) We design a natural variant of DP mirror descent, where the DP gradients of the private/sensitive data act as the linear term, and the loss generated by the public data as the mirror map.

We show that, for linear regression with feature vectors drawn from a non-isotropic sub-Gaussian distribution, our algorithm, PDA-DPMD (a variant of mirror descent), provides population risk guarantees that are asymptotically better than the best known guarantees under DP (without having access to public data), when the number of public data samples is sufficiently large. We further show that our algorithm has natural “noise stability” properties that control the variance due to noise added to ensure DP.

We demonstrate the efficacy of our algorithm by showing privacy/utility trade-offs on four benchmark datasets (StackOverflow, WikiText-2, CIFAR-10, and EMNIST). We show that our algorithm not only significantly improves over traditional DP-SGD, which does not have access to public data, but to our knowledge is the first to improve over DP-SGD on models that have been pre-trained with public data.

1. Introduction

Differentially Private Stochastic Gradient Descent (DP-SGD) (Song et al., 2013; Bassily et al., 2014; Abadi et al., 2016), and its variants (Kairouz et al., 2021b) have become the de facto standard algorithms for training machine learning models with differential privacy (DP) (Dwork et al., 2006). While DP-SGD is known to be optimal in terms of obtaining both optimal excess empirical risk (Bassily et al., 2014), and excess population risk (Bassily et al., 2020b) for convex losses, the obtained error guarantees suffer from an explicit polynomial dependence on the model dimension (p). This polynomial dependence significantly impacts the privacy/utility trade-off when $p \geq n_{\mathsf{priv}}$, where n_{priv} is the number of private training samples. Even empirically, when DP-SGD is used to train large deep learning models, there is a significant drop in accuracy compared to the non-private counterpart (Papernot et al., 2020).

In this paper, we revisit the problem of using public data (i.e., data without privacy concerns) to improve the privacy/utility trade-offs for DP model training. *Specifically, we design central and federated DP variants of mirror descent (Nemirovsky & Yudin, 1983) that use the loss function generated by the public data as the mirror map and DP gradients on the private data as the linear term.* For linear regression, we show that the excess population risk *asymptotically* improves over the best known bounds under DP (without access to public data samples) (Bassily et al., 2014; 2019) when n_{pub} is sufficiently large (i.e., a small polynomial in p), and the public and private feature vectors are drawn from the same non-isotropic sub-Gaussian distribution. Here, n_{pub} is the number of public data samples. Even if n_{pub} is small, our algorithm generalizes DP-SGD, so it never performs worse than DP-SGD.

Furthermore, we show empirically that our DP variant of mirror descent, assisted with public data, can improve the privacy-utility trade-offs by effectively reducing the variance in the noise added to the gradients in DP model training. We show relative improvements up to 5.3% over DP-SGD models, *pre-trained with the same public data*. To our knowledge, this is the first work to demonstrate an increased benefit from public data over just pre-training. Our empirical results are either on simulated linear regression data, or on standard benchmark datasets like StackOverflow, CIFAR-10, EMNIST, and WikiText-2, and only consider 4% of the training data samples (0.03% for StackOverflow) as public.

[1] Google [2] UC Berkeley [3] Part of this work was done while the author was an intern at Google. [4] MIT. Correspondence to: Arun Ganesh <arunganesh@berkeley.edu>, Vinith M. Suriyakumar <vinithms@mit.edu>.

Proceedings of the 39th International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

Learning Geometry with Mirror Maps: Common to most DP model training algorithms, including DP-SGD, DP-FTRL (Kairouz et al., 2021b), and our algorithm, is a DP estimator of the gradient of the loss $\nabla_\theta \mathcal{L}(\theta_t; D_{\mathsf{priv}}) = \sum_{d \in D_{\mathsf{priv}}} \nabla_\theta \ell(\theta_t; d)$ generated by the private dataset D_{priv} at a given model state $\theta_t \in \mathbb{R}^p$. This estimator adds isotropic Gaussian noise $\mathcal{N}(0, \sigma^2 \mathbb{I}_p)$ to $\nabla_\theta \mathcal{L}(\theta_t; D_{\mathsf{priv}})$, where σ depends on the privacy parameters (ε, δ) and the maximum allowable value of $\|\nabla_\theta \ell(\theta_t; d)\|_2$ (a.k.a. the clipping norm (Abadi et al., 2016)).[1] It is well known that for most learning tasks, the set of gradients for $\mathcal{L}(\theta_t; D_{\mathsf{priv}})$ is seldom isotropic (Gur-Ari et al., 2018; Agarwal et al., 2019). Hence, it is natural to wonder if the Gaussian noise in the DP estimator can be made to respect the geometry of the gradients.

Prior works (Zhou et al., 2020; Asi et al., 2021; Kairouz et al., 2021a) have used public data (D_{pub}) to *explicitly* learn this geometry, mostly in the form of preconditioner matrices (Duchi et al., 2011) to be multiplied to the estimated noisy gradients. In this paper, we take an *implicit* approach towards respecting this geometry, by using the loss $\mathcal{L}(\theta; D_{\mathsf{pub}})$ generated by the public data as the mirror map in classical mirror descent. As a first order approximation (formalized in Section 5), one can view it as doing DP-SGD on $\mathcal{L}(\theta; D_{\mathsf{priv}})$ while using $\mathcal{L}(\theta; D_{\mathsf{pub}})$ as a regularizer. This approach has the following advantages: (i) The information of the geometry is "free", i.e., one does not need to learn the preconditioner explicitly from the public data, (ii) Unlike prior works (Zhou et al., 2020; Kairouz et al., 2021a), one does not need to assume that the gradients of $\mathcal{L}(\theta; D_{\mathsf{priv}})$ lie in a low rank subspace, and (iii) It is easier to implement since it does not need to maintain an additional data structure for the preconditioner due to the geometry being implicitly defined. Empirically, our algorithm improves over the state of the art (Asi et al., 2021).

We note that DP mirror descent has been considered before by (Talwar et al., 2014; Wang et al., 2017). Their results are not directly comparable to ours because (i) they do not have access to in-distribution public data, (ii) as shown in (Bassily et al., 2014), without public data, it is impossible to achieve the bounds we achieve, and (iii) in our experiments, we solve unconstrained optimization problems whereas those works choose the mirror map based on the constraint set rather than the dataset. The utility bounds we prove in this paper also apply to a public data-assisted variant of accelerated mirror descent in (Wang et al., 2017).

In-distribution vs. Out-of-distribution Public Data: Prior works have considered settings where the public data comes from the same distribution as the private data (a.k.a. *in-distribution*) (Bassily et al., 2018a; Zhou et al., 2020; Kairouz et al., 2021a; Asi et al., 2021; Wang & Zhou, 2020), and where they can be different (a.k.a. *out-of-distribution*) (Abadi et al., 2016; Papernot et al., 2016; 2018; Li et al., 2021; Liu et al., 2021; Yu et al., 2021).

In the in-distribution setting, it is typical that there are fewer public data samples available than private data samples – i.e., $n_{\mathsf{pub}} \ll n_{\mathsf{priv}}$ – as it is harder to obtain public datasets than ones with privacy constraints attached. In-distribution public data could come from either altruistic *opt-in* users (Merriman, 2014; Avent et al., 2017) or from users who are incentivized to provide such data (e.g., mechanical turks). Out-of-distribution (OOD) public data may be easier to obtain but can have various degrees of freedom; e.g., the domains of private and public data may not be identical, the representation of some classes may vary, the distributions can be mean shifted, etc. It is usually hard to quantify these degrees of freedom to the extent that we can provide precise guarantees. Hence, we leave this aspect for future exploration, and work with the (idealized) assumption that the public data comes from the same distribution as the private data, or, at least, that the differences between these two distributions are not material. It worth emphasizing that although our utility results are for the in-distribution case, our algorithm can be used *as is* in out-of-distribution settings. In a restricted set of experiments, we do compare with one of the SoTA (Asi et al., 2021) for training with OOD public data, and demonstrate improvements in privacy/utility trade-off.

Choice of Empirical Benchmark: Mirror descent as a first step optimizes the mirror map function. In our setting, this corresponds to pre-training on the public loss function $\mathcal{L}(\theta; D_{\mathsf{pub}})$ before running the DP optimization procedure on $\mathcal{L}(\theta; D_{\mathsf{priv}})$. Since pre-training on public data is intuitive and easy, we always compare to DP-SGD (and its variants) that have been pre-trained to convergence with the public loss. We show that our algorithm *outperforms* even pre-trained DP-SGD. To our knowledge, ours is the first empirical work that compares to this strong (but fair) benchmark.

Other Uses of Public Data in DP Learning: The use of in-distribution public data has been extensively explored both theoretically and empirically. On the theoretical side, it has been shown (Alon et al., 2019; Bassily et al., 2020a) that a combination of private and public data samples can yield asymptotically better worst-case PAC learning guarantees than either on their own. Another line of work (Papernot et al., 2016; 2018; Bassily et al., 2018b; Dwork & Feldman, 2018; Nandi & Bassily, 2020) considers public data that is unlabelled, but otherwise comes from the same distribution as the private data; the primary goal is to use the private data to generate labels for the public data, which can then be used arbitrarily. Additionally, (Feldman et al., 2018) showed that for convex ERMs, using $\approx p$

[1] For the ease of presentation, at this point we do not consider the noise due to stochastic mini-batching.

in-distribution public data samples, one can obtain dimension independent population risk guarantees. However, the main tool used to prove DP (i.e., privacy amplification by iteration) heavily relies on convexity. As a result, their algorithm is inapplicable to the deep learning problems we consider in this paper.

So far only two papers have considered out-of-distribution data from a theory standpoint. (Bassily et al., 2020c) assume that whether a data record is public or private depends on its label; e.g., the public data may contain many negative examples, but few positive examples. They show that halfspaces can be learned in this model. (Liu et al., 2021) consider synthetic data generation and provide guarantees that depend on the Rényi divergences between the public and private distributions. (Abadi et al., 2016; Tramer & Boneh, 2020) provided techniques to effectively use out-of-distribution public data for pre-training for DP-SGD. However, they did not consider techniques to improve a pre-trained model using private and public data, which is the focus of our work. A recent work (Yu et al., 2021) uses public data to dynamically adjust the privacy budget and clipping norm. Our technique crucially uses the public data to learn the geometry of the gradients; (Yu et al., 2021) is complementary to ours and can be utilized for potential additional gains from using the public data after pre-training. In Appendix C.5, we discuss the comparison.

1.1. Problem Formulation

Consider the classic DP stochastic convex optimization (DP-SCO) (Chaudhuri et al., 2011; Bassily et al., 2014; 2019; 2020b) setting. Let τ be a distribution over a fixed domain $\mathcal{D}$. Given a dataset $D \in \mathcal{D}^*$ drawn i.i.d. from τ, and a loss function $\ell_{\mathsf{priv}} : \mathbb{R}^p \times \mathcal{D} \to \mathbb{R}$, the objective is to approximately solve $\arg\min\limits_{\theta \in \mathcal{C}} \mathbb{E}_{d\sim\tau}\left[\ell_{\mathsf{priv}}(\theta; d)\right]$, while preserving DP. Here, $\mathcal{C} \subseteq \mathbb{R}^p$ is the constraint set. Usually one solves the SCO problem via empirical risk minimization (ERM), i.e., $\theta^{\mathsf{priv}} \in \arg\min\limits_{\theta\in\mathcal{C}} \mathcal{L}(\theta; D)$, where $\mathcal{L}(\theta; D) = \frac{1}{|D|}\sum\limits_{d\in D} \ell_{\mathsf{priv}}(\theta; d)$, and then uses θ^{priv} as a proxy. Up to a dependence on dimensionality p, in the DP setting, a direct translation from ERM to the SCO setting provides optimal rates (Bassily et al., 2014; 2019; 2020b).

We consider the DP-SCO setting with *heterogeneous data*, where there are two datasets D_{priv} (with n_{priv} samples) and D_{pub} (with n_{pub} samples) drawn i.i.d. from the *same distribution*. The private dataset D_{priv} requires privacy protection, whereas the public dataset D_{pub} does not. Since obtaining such data can be expensive, for our empirical evaluation, we assume $n_{\mathsf{pub}} \ll n_{\mathsf{priv}}$ (e.g., $n_{\mathsf{pub}} \leq \frac{1}{20} n_{\mathsf{priv}}$).

Our algorithm allows the usage of a separate public loss function ℓ_{pub}. As a simple demonstration, we give a theoretical analysis where ℓ_{priv} and ℓ_{pub} both correspond to the linear regression loss $\frac{1}{2}(y - \langle \mathbf{x}, \theta\rangle)^2$. In practice too, one will likely choose $\ell_{\mathsf{priv}} = \ell_{\mathsf{pub}}$, but we may clip the gradients of ℓ_{priv} for privacy. In general, ℓ_{pub} can be arbitrary.

We refer the reader to Appendix A for a reference for the notation used throughout the paper.

1.2. Our Contributions

Algorithm: Our algorithm, Public Data Assisted Differentially Private Mirror Descent (PDA-DPMD), is similar to DP-SGD but utilizes the public data in two ways. First, we can pre-train on the public data to obtain a better starting point for training. Second, we use mirror descent, with the public loss function as the mirror map, to reshape the noisy gradients used in DP-SGD. In doing this, PDA-DPMD effectively takes smaller gradient steps and adds less noise in directions where the public loss grows quickly.

Tighter Excess Population Risk for Linear Regression: We consider the standard setting of linear regression where the loss function is $\ell(\theta; d) = \frac{1}{2}(y - \langle \mathbf{x}, \theta\rangle)^2$, with data sample $d = (\mathbf{x}, y)$. Let τ be the data generating distribution and $\theta^* = \arg\min\limits_{\theta\in\mathcal{C}} \mathbb{E}_{d\sim\tau}[\ell(\theta; d)]$ be the population minimizer. We assume a uniform bound on the feature vectors of the form $\|\mathbf{x}\|_2 \leq 1$, and on the response $|y - \langle \mathbf{x}, \theta^*\rangle| \leq 1$. Suppose the feature vectors are drawn i.i.d. from a distribution with covariance matrix $\bar{H}$. In this setting, DP-SGD obtains an error of roughly $\frac{p}{\lambda_{\min}(\bar{H})\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{1}{\lambda_{\min}(\bar{H}) n_{\mathsf{priv}}}$. If we use PDA-DPMD instead, we can show that given a sufficient number of public samples, the first term depends on the *average* rather than the *minimum* eigenvalue. For example, if $\bar{H}$ has one eigenvalue being $1/p^{1.5}$ and the remaining eigenvalues being $1/p$, then with $n_{\mathsf{pub}} = \widetilde{\Omega}(p^{2.5})$ public samples, PDA-DPMD obtains an error of $\frac{p^2}{\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{p^{1.5}}{n_{\mathsf{priv}}}$, whereas DP-SGD gets $\frac{p^{2.5}}{\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{p^{1.5}}{n_{\mathsf{priv}}}$. Since PDA-DPMD generalizes DP-SGD, unsurprisingly, it still recovers the error bound of DP-SGD in the isotropic case. We provide the formal statement in Theorem 4.2.

Local Noise-stability: We show that in addition to achieving better excess population loss bounds, PDA-DPMD has the following "local noise-stability" property: If in a bounded region around the current model θ_t, the public loss is $\lambda_{\mathbf{v}}$-strongly convex in a direction $\mathbf{v}$, then using noisy gradients instead of the exact gradients shifts θ_{t+1} in the direction $\mathbf{v}$ by an amount proportional to $1/\lambda_{\mathbf{v}}$ (see Theorem 4.3 for the formal statement). That is, PDA-DPMD effectively rescales the amount of noise added in any direction to be inversely proportional to the curvature in that direction. Note that this is in spite of the fact that for privacy, the noise we add to the gradients is usually isotropic. Furthermore, PDA-DPMD can perform this rescaling using only a gradient oracle for the public loss function. In other words, a practitioner implementing the algorithm simply needs to choose an appropriate loss function, and PDA-DPMD will "automatically" rescale the effects of noise to match the loss function's curvature.

Empirical Evaluation: On both synthetic and real-world benchmark datasets, we show that PDA-DPMD outperforms DP-SGD, even when they are pre-trained on the public dataset. We provide two sets of experiments. First, a linear regression on a synthetic dataset which closely matches the utility assumptions in the theoretical analysis. Second, we provide results on deep learning benchmark datasets (StackOverflow, WikiText-2, CIFAR-10, and EMNIST).

In Section 3, we consider using DP-SGD and PDA-DPMD to solve a least squares linear regression problem on a synthetic data set generated via the process $y_i \sim N(\langle \mathbf{x}_i, \theta^* \rangle, \sigma^2)$, where $\theta^* \in \mathbb{R}^p$ is the true model. The feature vectors $\mathbf{x}_i$'s are drawn i.i.d. from some fixed distribution. We fix the number of private data samples, and set the number of public samples to be a fixed constant times the dimension (p). We observe that as expected, public data allows us to substantially improve the error in two ways: (i) **Pre-training**: DP-SGD initialized from a model pre-trained on public data has nearly-constant mean-squared error, whereas DP-SGD from a random initialization has mean-squared error scaling with the dimension, and (ii) **Adapting to geometry using public loss**: While DP-SGD initialized from a pre-trained model already achieves near-constant loss, we also observe that PDA-DPMD outperforms DP-SGD due to its error's dependence on the Gaussian width G_Q rather than the dimension. We note that the observed improvement is "automatic" once we choose the mean-squared error to be the loss function.

For the deep learning experiments, since running PDA-DPMD can be computationally expensive, we derive a first-order approximation that can be viewed as DP-SGD on a convex combination of the private and public losses. This makes the running time of our algorithm comparable to DP-SGD when run on the dataset $D_{\mathsf{priv}} \cup D_{\mathsf{pub}}$.

For user-level DP, we conduct experiments in Section 5.2 for next word prediction (NWP) on the StackOverflow dataset. We consider $\sim$0.03% of the original training users as public and pre-train using them. We find that PDA-DPMD tailored towards a user-level setting (e.g., based on DP Federated Averaging (DP-FedAvg) (McMahan et al., 2017b)) outperforms DP-FedAvg: PDA-DPMD obtains a 0.57% absolute (2.7% relative) increase in accuracy, and a 5.3% drop in perplexity. Note that PDA-DPMD can directly apply to Federated Learning (McMahan et al., 2017a) settings since an orchestrating server does not need to provide public data to any participating private client.

For sample-level DP, we conduct experiments in Section 5.3 on two real world tasks: NWP on WikiText-2, and image classification on CIFAR-10 and EMNIST. We consider 4% of the original training data as public and pre-train on it. On all datasets, we can observe that PDA-DPMD outperforms DP-SGD in terms of test loss. On CIFAR-10, the improvement is more than 5%; on EMNIST, 7%; on WikiText-2, log perplexity is improved by more than 0.3%, which is a notable improvement for perplexity.

1.3. Organization

In Section 2 we provide background details on differential privacy and mirror descent In Section 3 we discuss the main algorithmic contribution, and the corresponding privacy guarantee. In Section 4 we provide the privacy/utility trade-offs for linear regression. In Section 5 we provide a detailed empirical evaluation. Additionally, in Appendix A we provide a table of notions.

2. Background

Differential Privacy: Differential Privacy (DP) (Dwork et al., 2006) is a formal method for quantifying the privacy leakage from the output of a data analysis procedure. A randomized algorithm $M : \mathcal{D}^* \to \mathcal{Y}$ is (ε, δ)-DP if, for all neighbouring dataset pairs $D, D' \in \mathcal{D}^*$ and all measurable sets of outputs $S \subseteq \mathcal{Y}$, we have

$$\mathbb{P}\left[M(D) \in S\right] \leq e^{\varepsilon} \cdot \mathbb{P}\left[M(D') \in S\right] + \delta.$$

We define two datasets to be neighbouring if they differ only by the *addition or removal of one person's record*. We ensure differential privacy by adding Gaussian noise to functions of bounded sensitivity. In particular, if ℓ is L-Lipschitz in its first parameter, then $\|\nabla_\theta \ell(\theta; d)\|_2 \leq L$ for all θ and $d \in \mathcal{D}$. Thus adding noise drawn from $\mathcal{N}(0, \sigma^2 \cdot \mathbb{I}_p)$ to the sum $\sum\limits_i \nabla_\theta \ell(\theta, d_i)$ over people's records satisfies DP, where σ scales with L and the desired privacy parameters. The composition and postprocessing properties of differential privacy ensure that, as long as each step in our iterative algorithm satisfies differential privacy, then so does the overall system. We refer the reader to (Dwork & Roth, 2014) for further details of the standard privacy analysis of algorithms like ours.

Mirror Maps: A mirror map is a differentiable function $\Psi : \mathbb{R}^p \to \mathbb{R}$ that is strictly convex. Since Ψ is strictly convex and differentiable, $\nabla\Psi : \mathbb{R}^p \to \mathbb{R}^p$ provides a bijection from $\mathbb{R}^p$ to itself. One can view θ as lying in a primal space and $\nabla\Psi(\theta)$ as lying in a dual space. In turn, we could now consider optimizing over the value $\nabla\Psi(\theta)$ in the dual space instead of θ primal space. Mirror descent does exactly that, performing gradient descent in the dual space by computing the gradient

$\mathbf{g}_t = \nabla\ell(\theta_t)$ (where θ_t lies in the primal space), taking a step in the opposite direction in the dual space, and then using the inverse of the mirror map to determine θ_{t+1}. Mirror descent is essentially motivated as minimizing a (linearized) loss plus a Bregman divergence (induced by Ψ) as the regularizer (Nemirovsky & Yudin, 1983). More formally, similar to proximal gradient descent, mirror descent is equivalent to taking the gradient $\mathbf{g}_t$ and performing the update $\theta_{t+1} = \arg\min_{\theta\in\mathcal{C}}[\eta\langle\mathbf{g}_t, \theta\rangle + B_\Psi(\theta, \theta_t)]$ where $B_\Psi(\theta_1, \theta_2) = \Psi(\theta_1) - \Psi(\theta_2) - \langle\nabla\Psi(\theta_2), \theta_1 - \theta_2\rangle$ is the Bregman divergence generated by Ψ. Note that, if $\Psi(\theta) = \|\theta\|_2^2$, then the Bregman divergence is simply $B_\Psi(\theta_1, \theta_2) = \|\theta_1 - \theta_2\|_2^2$ and mirror descent is equivalent to the usual gradient descent.

Gaussian Width: Given a bounded set $Q \subset \mathbb{R}^d$, the Gaussian width of Q, G_Q, is a measure of how isotropic the set is. G_Q is defined as $\mathbb{E}_{g\sim N(0,\mathbb{I}_p)} \max_{x\in Q}\langle g, x\rangle$. Although the Gaussian width is well-defined for any bounded set, to gain intuition it suffices to consider defining the Gaussian width of convex sets containing the origin such that $\max_{x\in Q}\|x\|_2 = 1$; rescaling any such set by a constant changes the Gaussian width by the same constant. If Q is just the unit ℓ_2-ball, the "most isotropic" set satisfying this condition, then we have $G_Q = \sqrt{p}$; in particular, since every set Q satisfying $\max_{x\in Q}\|x\|_2 = 1$ is contained in the ℓ_2-ball, this is the maximum Gaussian width of any such set. On the other hand, if Q is just the line from the origin to a single unit vector, we have $G_Q = \Theta(1)$. More generally, for any ellipsoid centered at the origin whose axes have radii $0 \leq r_i \leq 1, 1 \leq i \leq p$, we have that the Gaussian width of this ellipsoid is $\Theta(\sqrt{\sum_{i=1}^p r_i^2})$. As other examples, the Gaussian width of the ℓ_1-ball of radius 1 is roughly $\log p$, and the Gaussian width of the ℓ_∞ ball of radius $1/\sqrt{p}$ is roughly $\sqrt{p}$.

3. Algorithm Description

In this section, we present our main algorithm Public Data-Assisted Differentially Private Mirror Descent (PDA-DPMD). Given in Algorithm 1, it is a variant of mirror descent using noisy gradients, but we also pre-train on public data and use the public loss as our mirror map Ψ.

Algorithm 1 Public Data-Assisted Differentially Private Mirror Descent (PDA-DPMD)

```
Input: Public/private datasets $D_{\mathsf{pub}}, D_{\mathsf{priv}}$ of sizes $n_{\mathsf{pub}}, n_{\mathsf{priv}}$, private/public loss functions $\ell_{\mathsf{priv}}, \ell_{\mathsf{pub}}$, privacy parameters $(\varepsilon, \delta)$, number of iterations $T$, learning rate $\eta : \{0, 1, \ldots, T-1\} \to \mathbb{R}^+$, constraint set: $\mathcal{C}$, clipping norm $L$: an upper bound on $\max_{\theta\in\mathcal{C}} \|\nabla\ell_{\mathsf{priv}}(\theta)\|_2$
1: $\Psi(\theta) := \frac{1}{n_{\mathsf{pub}}} \sum_{d\in D_{\mathsf{pub}}} \ell_{\mathsf{pub}}(\theta; d)$
2: $\theta_0 \leftarrow \arg\min_{\theta\in\mathcal{C}} \Psi(\theta)$, $\sigma^2 \leftarrow \frac{8L^2 T \log(1/\delta)}{(\varepsilon n_{\mathsf{priv}})^2}$
3: for $t = 0, \ldots, T-1$ do
4:     $\mathbf{g}_t \leftarrow \frac{1}{n_{\mathsf{priv}}} \sum_{d\in D_{\mathsf{priv}}} \mathsf{clip}\left(\nabla\ell_{\mathsf{priv}}(\theta_t; d), L\right)$, where $\mathsf{clip}\left(\mathbf{v}, L\right) = \mathbf{v} \cdot \min\left\{1, \frac{L}{\|\mathbf{v}\|_2}\right\}$
5:     $\theta_{t+1} \leftarrow \arg\min_{\theta\in\mathcal{C}}\left[\eta_t\langle\mathbf{g}_t + \mathbf{b}_t, \theta\rangle + B_\Psi(\theta, \theta_t)\right]$, where $\mathbf{b}_t \sim \mathcal{N}(0, \sigma^2 \cdot \mathbb{I}_p)$
6: return $\theta_{\mathsf{priv}} := \frac{1}{T} \sum_{t=1}^{T} \theta_t$
```

Note that Line 5 of PDA-DPMD is equivalent to the following: Choose $\theta_{t+1/2}$ to be the point such that $\nabla\Psi(\theta_{t+1/2}) = \nabla\Psi(\theta_t) - \eta(\mathbf{g}_t + \mathbf{b}_t)$, and then use the Bregman projection $\theta_{t+1} = \arg\min_{\theta\in\mathcal{C}} B_\Psi(\theta, \theta_{t+1/2})$. Intuitively, PDA-DPMD is similar to DP-SGD, with the main difference being we apply the gradient steps to $\nabla\Psi(\theta)$ rather than to θ itself. Note that PDA-DPMD reshapes the gradient and noise *automatically* given ℓ_{pub} and D_{pub}. In contrast, e.g., private AdaGrad implementations (Kairouz et al., 2020; Asi et al., 2021) assume knowledge of the geometry of the loss function has already been learned prior to running their algorithms. Also, for an appropriate choice of Ψ, one can recover an algorithm that projects the private gradients to a low-dimensional subspace as in the algorithms of Zhou et al. (2020) and (Kairouz et al., 2020). From e.g. (Abadi et al., 2016) we have the privacy guarantee for Algorithm 1:

Theorem 3.1. *Algorithm 1 (PDA-DPMD) is (ε, δ)-DP with respect to the private dataset D_{priv}.*

4. Case Study: Linear Regression

In this section, we apply Algorithm 1 (PDA-DPMD) to linear regression – an important example that is still amenable to theoretical analysis. We prove utility guarantees, with supporting simulations; proofs are in Appendix B.

Problem setup: Given a data sample $d_i = (\mathbf{x}_i, y_i)$, the loss of a model θ is defined as $\ell(\theta; d_i) := \frac{1}{2}(y_i - \langle\theta, \mathbf{x}_i\rangle)^2$. Consider two datasets drawn i.i.d. from a fixed distribution τ: i) The public dataset D_{pub} with n_{pub} data samples, and ii) The private dataset D_{priv} with n_{priv} data samples. In this section, we will denote both the public and private loss functions (ℓ_{pub} and ℓ_{priv} respectively in Algorithm 1) by ℓ.

Assumption 4.1. *We assume that we are given an initial constraint set*[2] *$\mathcal{C}_0 = \{\theta : \|\theta\|_2 \leq r\}$ with $r = O(1)$ that contains the population minimizer, i.e., $\theta^* = \arg\min_{\theta\in\mathbb{R}^p} \mathbb{E}_{d\sim\tau}\ell(\theta; d) \in \mathcal{C}_0$. We further assume that for each feature vector $\|\mathbf{x}\|_2 \leq 1$, and for each response $|y - \langle\theta^*, \mathbf{x}\rangle| \leq 1$. Let $\bar{H}$ be the Hessian of the loss function $\mathbb{E}_{d\sim\tau}\left[\ell(\theta; d)\right]$. In terms of data set sizes, we assume that*

[2] The assumption that $\mathcal{C}_0$ is centered at the origin is without loss of generality.

$n_{\mathsf{priv}} \geq n_{\mathsf{pub}}$ *and* $n_{\mathsf{pub}} = \Omega\left(\frac{\log(p/\delta)}{\lambda_{\min}(\bar{H})}\right)$.

Excess population risk guarantees: In Theorem 4.2 we first provide the excess population risk guarantee for Algorithm 1 (PDA-DPMD) under Assumption 4.1. Furtheremore, in some regimes we demonstrate asymptotic improvement over standard privacy/utility trade-offs for algorithms without access to public data samples.

Theorem 4.2. *Consider Assumption 4.1. We run Algorithm 1 (PDA-DPMD) using* $L = O(1)$*, constraint set* $\mathcal{C} = \left\{\theta \in \mathcal{C}_0 : \left\|\nabla\Psi(\theta)\right\|_2 = O\left(\sqrt{\frac{\log(1/\delta)}{n_{\mathsf{pub}}}}\right)\right\}$*, and an appropriate choice of* η_t *and* T*. Let*

$$\chi = \max\left\{\frac{1}{\lambda_{\min}(\bar{H})}, \lambda_{\max}(\bar{H})n_{\mathsf{pub}}\right\}\cdot\sum_i \min\left\{1, \frac{\log(1/\delta)}{\lambda_i(\bar{H})^2 n_{\mathsf{pub}}}\right\}.$$

Then, Algorithm 1 is (ε,δ)*-DP and we have the following guarantee on* $\mathcal{L}(\theta) := \mathbb{E}_{d\sim\tau}\left[\ell(\theta; d)\right]$*:*

$$\mathbb{E}\left[\mathcal{L}(\theta_{\mathsf{priv}}) - \mathcal{L}(\theta^*)\right] \leq \widetilde{O}\left(\frac{\chi\log(1/\delta)}{\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{1}{\lambda_{\min}(\bar{H})n_{\mathsf{priv}}}\right).$$

The expectation is over $D_{\mathsf{pub}}, D_{\mathsf{priv}}$*, and the algorithm.* $\widetilde{O}(\cdot)$ *hides polylog factors in* $n_{\mathsf{priv}}, n_{\mathsf{pub}}$ *and* $\lambda_{\min}(\bar{H})$*.*

Proof sketch. To prove Theorem 4.2, we show that the public sample Hessian, private sample Hessian, and population Hessian are all good approximations of each other. This lets us argue the following: for an ellipsoid that is approximately the same shape as $\mathcal{C}$, we can bound the strong convexity parameter of Ψ with respect to this ellipsoid's Minkowski norm. Then, by the concentration of the public sample Hessian, the strong convexity parameters of the population loss and private sample loss with respect to this ellipsoid's Minkowski norm are within a constant factor of the public loss' strong convexity parameter. This lets us use the framework of (Talwar et al., 2014) to obtain the desired excess empirical loss bound, which gives a population loss bound as well by uniform stability.

We note that the idea of using public data to shrink the constraint set $\mathcal{C}$ is similar to the idea used by (Biswas et al., 2020) for mean estimation, though their result iteratively uses each private mean estimate to shrink the constraint set before re-estimating the mean, as opposed to our one-shot approach to shrinking the constraint set using public data.

To interpret χ in Theorem 4.2, note that a natural setting of parameters to consider would be where the feature vectors (i.e., the $\mathbf{x}$'s) are coming from a mean-zero truncated Gaussian distribution with covariance $\frac{1}{p}\cdot\mathbb{I}$. In this case, all λ_i are $1/p$. If $n_{\mathsf{pub}} = \widetilde{\Omega}(p)$, then χ evaluates to p^2, and so we get a bound of $\widetilde{O}\left(\frac{p^2}{\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{p}{n_{\mathsf{priv}}}\right)$, matching the excess population risk of DP-SGD. Note that one can still recover DP-SGD's loss bound with Algorithm 1 even if $n_{\mathsf{pub}} = O(1)$ by instead setting Ψ to be $\frac{1}{2}\left\|\theta\right\|_2^2$ and $\mathcal{C} = \mathcal{C}_0$.

One can also consider a non-isotropic setting, where $\lambda_{\min}(\bar{H})$ is $1/p^{1.5}$ instead of $1/p$, but all other eigenvalues remain roughly $1/p$. In this setting, DP-SGD would give an error bound of $\widetilde{O}\left(\frac{p^{2.5}}{\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{p^{1.5}}{n_{\mathsf{priv}}}\right)$. If $n_{\mathsf{pub}} = \widetilde{\Omega}(p^{3/2})$, we again match the DP-SGD bound. If instead we have $n_{\mathsf{pub}} = \widetilde{\Omega}(p^c)$ for $2 \leq c \leq 2.5$, then χ in our loss bound becomes $p^{4.5-c}$, and our loss bound becomes $\widetilde{O}\left(\frac{p^{4.5-c}}{\varepsilon^2 n_{\mathsf{priv}}^2} + \frac{p^{1.5}}{n_{\mathsf{priv}}}\right)$. Once $c = 2.5$, the first term becomes $\frac{p^2}{\varepsilon^2 n_{\mathsf{priv}}^2}$, matching the corresponding term for the isotropic setting. In contrast, with $n_{\mathsf{pub}} = O(p^3)$, $\left\|\mathcal{C}\right\|_2 = \Theta(\left\|\mathcal{C}_0\right\|_2)$ in this setting (in expectation). In other words, with less than p^3 samples, pre-training does not let us decrease the constraint set's diameter with this many samples, so the standard error bounds of DP-SGD do not improve without many more samples than PDA-DPMD needs to see improvements. This shows that PDA-DPMD asymptotically improves over DP-SGD under a non-isotropic geometry when given sufficiently many public data samples, with the improvement increasing as the number of public samples increases.

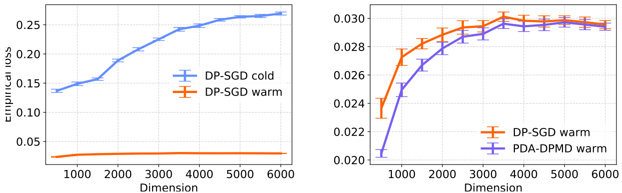
Local Stability Properties: Since in linear regression the public loss function has the same Hessian $\hat{H}_{\mathsf{pub}}$ everywhere, mirror descent effectively is DP-SGD, but applying $\hat{H}_{\mathsf{pub}}^{-1}$ to the noisy gradient. This allows us to readily characterize the effective noise being added, and show that the noise causes each iterate θ_t to be moved by an amount proportional to $1/\lambda_{\mathbf{v}}$ in a direction where the strong convexity parameter is $\lambda_{\mathbf{v}}$:

Theorem 4.3. *Let the Hessian of* Ψ *be* $\hat{H}_{\mathsf{pub}} = \sum_i \lambda_i \mathbf{v}_i\mathbf{v}_i^\top$*, where* $\mathbf{v}_i$ *are the unit eigenvectors of* $\hat{H}_{\mathsf{pub}}$*. Fix an iteration* t *as well as starting point* θ_t *and private gradient* $\mathbf{g}_t$ *in PDA-DPMD. Let* $\bar{\theta}$ *be the value of* θ_{t+1} *after performing the mirror descent update with* $\mathbf{b}_t = \mathbf{0}$ *at iteration* t*, and let where* $\hat{\theta}$ *be the value of the next iterate* θ_{t+1} *if noise is added. Then for any (unit) direction* $\mathbf{v} = \sum_i a_i\mathbf{v}_i$*,*

$$\mathbb{E}\left[\left|\langle\hat{\theta} - \bar{\theta}, \mathbf{v}\rangle\right|\right] = \eta\sigma\sqrt{\frac{2}{\pi}\cdot\sum_i\left(\frac{a_i}{\lambda_i}\right)^2}.$$

In contrast, for DP-SGD, $\mathbb{E}\left[\left|\langle\hat{\theta} - \bar{\theta}, \mathbf{v}\rangle\right|\right] = \eta\sigma\sqrt{\frac{2}{\pi}}$ for all $\mathbf{v}$. The proof follows readily from the observation that the mirror descent update simply applies the inverse of the Hessian to the noisy gradient.

Simulation Results: To corroborate our theoretical results with empirical validation, we run PDA-DPMD on synthetic data for the linear regression problem with mean squared error loss. We vary the dimensionality of the problem p from 500 to 6000. For each p, we generate 10,000 private samples and $1.5p$ public samples. The optimal θ^* is sam-



(a) Cold start DP-SGD vs. warm start DP-SGD.

(b) Warm start DP-SGD vs. PDA-DPMD.

Figure 1: The empirical loss on synthetic linear regression data. The mean and error bars for a 95% confidence interval over 20 runs are plotted. The optimal population loss is 0.01.

pled from $\mathcal{N}(0, \mathbb{I}_p)$. To introduce a non-isotropic geometry, we sample the feature vector $\mathbf{x}_i$ such that 40 of the first $p/5$ features and 80 of the last $4p/5$ features, chosen uniformly at random, are set to 0.05, and the rest of the features are set to 0. In this way, the expected ℓ_2-norm of each feature vector (and in turn each gradient) is $O(1)$, and thus the effects of clipping should not vary with p. The predicted variable y_i is sampled from $\mathcal{N}(\theta^* \cdot \mathbf{x}_i, 0.01)$ so that the population mean squared error loss is always 0.01, i.e. independent of dimension. We set $\varepsilon = 1, \delta = 10^{-5}$.

We consider three algorithms: (i) DP-SGD with a "cold start", i.e. using a random initialization, (ii) DP-SGD with a "warm start" on the model pre-trained with public data, and (iii) PDA-DPMD after pre-training on public data. We perform a grid search over the learning rate, clipping norm, and number of epochs used and report the best empirical loss. We perform 20 trials for each algorithm and dimension. Note that the optimum on the public data can be computed exactly. The mirror descent step can also be solved exactly by applying the inverse of the public Hessian $\mathbf{X}^\top\mathbf{X}$ to the private gradient, since the Hessian is the same everywhere. For numerical stability, we add a small constant times the identity matrix to the Hessian before computing its inverse. We also normalize the Hessian of the loss function so its inverse has maximum eigenvalue of one. This ensures that if the Hessian were a multiple of the identity matrix, DP-SGD and PDA-DPMD would behave exactly the same for the same hyperparameter choice.

Figure 1a shows the empirical loss of cold- and warm-start DP-SGD. Our results show that pre-training with a number of public samples linear in the dimension allows DP-SGD to achieve nearly dimension-independent error. Figure 1b compares warm-start DP-SGD and PDA-DPMD. The loss of PDA-DPMD is never worse than that of warm-start DP-SGD, and can be substantially lower for smaller dimensions. We observed that the ratio of the maximum and minimum eigenvalues of the Hessian $\mathbf{X}^\top\mathbf{X}$ decreases as p increases, which means that the Hessian has poorly-concentrated eigenvalues at small p but gets closer to the identity matrix as p increases. Since PDA-DPMD recovers warm start DP-SGD when the Hessian is the identity, we can expect that PDA-DPMD obtains less of an advantage over DP-SGD as the Hessian gets closer to the identity. The code has been open sourced.[3]

5. Empirical Evaluation

5.1. First-order Approximation to Mirror Descent

In practice, the Mirror Descent (MD) step in Line 5 of Algorithm 1 can be computationally expensive. For settings where (i) the problem is *unconstrained*, i.e., $\mathcal{C} = \mathbb{R}^p$ and (ii) the public loss function $\Psi(\theta)$ may not be strongly convex with respect to the ℓ_2-norm, we can instead use the following more efficient approximation:

$$\theta_{t+1} \leftarrow \theta_t - \eta_t \left(\alpha_t(\mathbf{g}_t + \mathbf{b}_t) + (1-\alpha_t)\nabla\Psi(\theta_t)\right), \quad (1)$$

where η_t is the learning rate, and $\alpha_t \in [0, 1]$ balances the weight of private and public gradient. The derivation of this formula is in Appendix C.1. Notice that $\alpha_t = 1$ corresponds to DP-SGD on private data only. In our experiment, we decrease α_t with a cosine schedule, i.e. $\alpha_t = \cos(\pi t/(2K))$ where K is a hyperparameter that controls how fast α_t decays. In practice, instead of computing $\mathbf{g}_t$ and $\nabla\Psi(\theta_t)$ using all the private and public data, we can estimate them with stochastic gradients.

5.2. Evaluation with User-level Differential Privacy

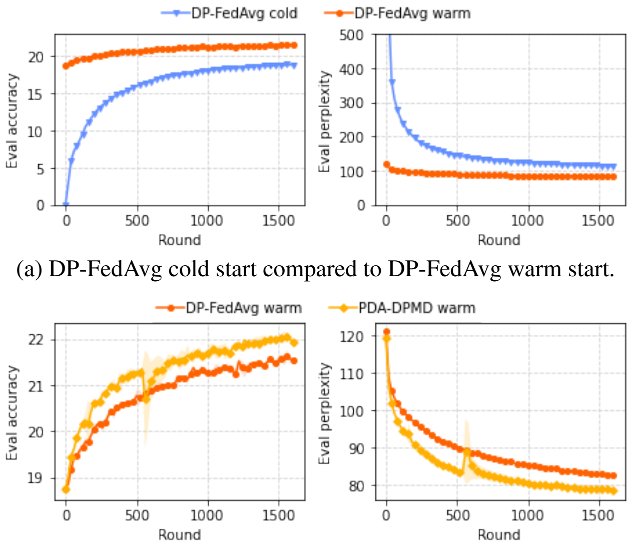
Now, we evaluate our technique (Algorithm 1) with the update step in (1) on settings suited for user-level DP guarantees. In particular, we conduct our evaluation for next word prediction on the benchmark StackOverflow (Overflow, 2018) dataset, which consists of 342.4k users having 135.8M examples in the training set, and 204.1k users having 16.6M examples in the test set. Since StackOverflow is naturally keyed by users, we assume training in a federated learning (McMahan et al., 2017a) setting. For private users, we limit each user to have at most 256 examples. For each user having more than 256 examples in StackOverflow, we create a private user with 256 examples and a new user with at most 256 of the overflowing examples. We randomly assign 100 such newly created users (0.03% of the total number of users) as public users.

We first pre-train on the public users, then use Algorithm 1 with the update step in (1) (with the updates modeled on the DP Federated Averaging optimizer (McMahan et al., 2017b) instead of DP-SGD). The privacy guarantee is thus user-level. We compare with two baselines: "cold start" DP-FedAvg, which uses the private data only, and "warm start" DP-FedAvg, which pre-trains the model with public data and then fine-tunes with the private data using DP-FedAvg. We use the one layer LSTM from (Kairouz et al.,

[3] Code for simulation experiments: `https://github.com/googleinterns/PLPD`

2021b) for our experiments on StackOverflow. See Appendix C.2 for more details.

For pre-processing, we use the 10k most frequently used words, and represent all other words with an unknown token. We set the maximum length of any sequence to 20, and each client has a maximum of 256 sentences. In each training round, we sample 100 clients, and each client uses a batch size of 16 for local training. We train for 1.6k rounds. See Appendix C.2 for more details. We evaluate utility using accuracy on in-vocabulary words and perplexity. The code has been open sourced.[4]



(a) DP-FedAvg cold start compared to DP-FedAvg warm start.

(b) DP-FedAvg warm start and PDA-DPMD warm start.

Figure 2: Stackoverflow. 0.03% public data. Eval accuracy / perplexity vs. training rounds. Averaged over 3 runs.

Algorithm	Accuracy	Perplexity
DP-FedAvg cold	18.52 ± 0.04	116.21 ± 0.08
DP-FedAvg warm	21.2 ± 0.03	85.03 ± 0.06
PDA-DPMD warm	$\mathbf{21.77 \pm 0.02}$	$\mathbf{80.51 \pm 0.2}$

Table 1: Mean and standard deviation of the Test accuracy and perplexity for trained models on StackOverflow.

We compare warm start PDA-DPMD to warm start DP-FedAvg. We fix the noise multiplier to $\sigma = 0.4$ so that the client-level $\varepsilon = 8.32$ for $\delta = 10^{-6}$. We perform a grid search over the server learning rate $\{0.5, 1.0, 3.0\}$, the client learning rate $\{0.1, 0.2, 0.5\}$, and clipping norm $\{0.3, 1.0\}$ for both methods. We perform an additional search for the optimal decay schedule for α for PDA-DPMD using $\cos(\pi t/(iT))$ for $i \in \{2, 3, 4, 5, 8\}$, where $T = 1600$. In Figure 2, we plot the eval accuracy / perplexity across training for $\alpha_t = cos(\pi t/(5T))$. In Table 1, we present the test accuracy and perplexity for the final trained models. We see that PDA-DPMD obtains a 0.57% absolute (2.7% relative) increase in accuracy, and a 5.3% drop in perplexity compared to warm start DP-FedAvg. *We demonstrate an increased benefit from incorporating the public data in training over and above just pre-training*, which to our knowledge, has not been achieved in prior work.

5.3. Evaluation with Sample-level Differential Privacy

Next, we demonstrate the efficacy of our technique with the update step in (1) on two real world tasks across three benchmark datasets: next word prediction on WikiText-2 (Merity et al., 2017), and image classification on CIFAR-10 (Krizhevsky, 2009) and EMNIST (ByMerge split) (Cohen et al., 2017). For each dataset, we randomly assign 4% of the original training data as public, and the rest as private. We do not consider a larger amount of in-distribution public data as that could make the problem trivial.

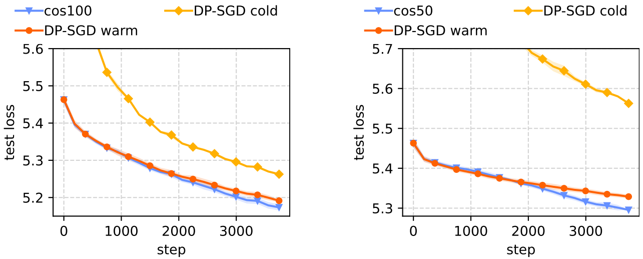
As in Section 5.2, we first pre-train on the public data, then use Algorithm 1 with update rule (1). We compare our algorithm with two baselines (analogous to those in Section 5.2): "cold start" DP-SGD, and "warm start" DP-SGD. In this setting too, we see benefits from the public data above just pre-training. For WikiText-2, we use an LSTM model from (Asi et al., 2021). For CIFAR-10 and EMNIST, we use network architectures considered in prior works (Papernot et al., 2020; Kairouz et al., 2021b). See Appendix C.2 for more details.

Empirical Evaluation on WikiText-2: Our setup mainly follows Asi et al. (2021). As a preprocessing step, we take the top 8k most frequent words and convert the rest into a special token representing unknown word. The dataset is then split into 48.8k length-35 sequences, and we consider sequence-level privacy here. After pre-training, we fine-tune the model with batch size 250 for 20 epochs. We search for optimal K in $\{100, 200, 500\}$. For two different privacy levels, $\varepsilon = 15.7$ and 1.71 at $\delta = 10^{-5}$ (corresponding to $\sigma = 0.5$ and 1.08, respectively), Figure 3 shows the test loss for PDA-DPMD with $K = 100$ and 50 respectively, and for the two baselines. From Table 2, we see that PDA-DPMD obtains the smallest test loss for both the privacy levels. Also, comparing the two DP-SGD baselines, we can see that using public data for pre-training provide trained models with higher utility.

Though our work's focus is on in-distribution public data, we additionally compare with SoTA (Asi et al., 2021) which uses WikiText-103 (Merity et al., 2017) as the public data. In that setting[5], our warm start DP-SGD baseline

[4] Code for user-level experiments: https://github.com/google-research/public-data-in-dpfl

[5] The implementation of Asi et al. (2021) was not made public



(a) $\sigma = 0.5$. $\varepsilon = 15.7$. (b) $\sigma = 1.08$. $\varepsilon = 1.71$.

Figure 3: WikiText-2. 4% public data. Validation and test loss. Averaged over 3 runs. Full plots in Appendix C.3.

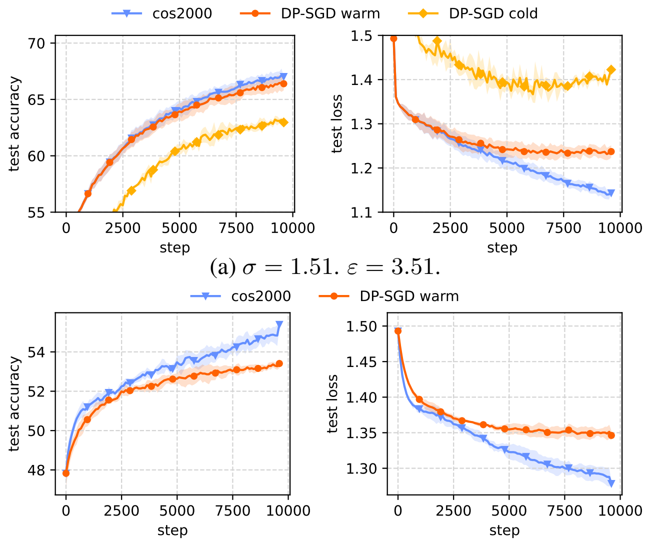
Table 2: Metrics for the final models for each configuration.

Dataset, metrics	Algorithm	Smaller σ	Larger σ
WikiText-2, test loss	DP-SGD cold DP-SGD warm PDA-DPMD	5.2626 5.1914 **5.1736**	5.5627 5.3288 **5.2956**
CIFAR-10, accuracy / test loss	DP-SGD cold DP-SGD warm PDA-DPMD	62.9633 / 1.4225 66.3933 / 1.2371 **67.0300 / 1.1435**	40.6000 / 1.6890 53.4100 / 1.3462 **55.3950 / 1.2785**
EMNIST, accuracy / test loss	DP-SGD cold DP-SGD warm PDA-DPMD	87.5671 / 0.5422 87.8534 / 0.5089 **87.9860 / 0.4706**	84.7270 / 0.6170 86.3352 / 0.5586 **86.7229 / 0.4982**

is better than the proposed SoTA in (Asi et al., 2021) by 1.1% for $\varepsilon = 1.0$, and 6.6% for $\varepsilon = 3.0$ in terms of test perplexity. See Appendix C for more details.

Empirical Evaluation on CIFAR-10: CIFAR-10 consists of 50k training images and 10k test images from 10 classes. After pre-training, we fine-tune the model with batch size 500 for 100 epochs. We search for optimal K in $\{200, 500, 1000, 2000, 5000\}$. In Figure 4, for two different privacy levels, $\varepsilon = 3.51$ and 0.19 at $\delta = 10^{-5}$ (corresponding to $\sigma = 1.51$ and 20.0, respectively), we report the test loss and accuracy for $K = 2000$, and for the two baselines. From Table 2, we see that PDA-DPMD provides the best accuracy (even if by a small margin over the warm started DP-SGD baseline). Moreover, PDA-DPMD also results in significantly lower test loss compared to both the baselines for both privacy levels.

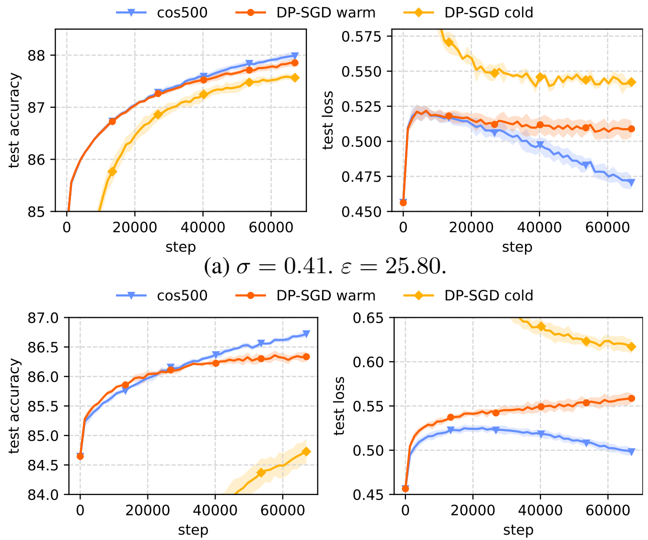
Empirical Evaluation on EMNIST: EMNIST (ByMerge split) consists of 697.9K training images and 116.3k test images from 47 classes. After pre-training, we fine-tune with batch size 500 for 50 epochs. We search for optimal K in $\{200, 500, 1000, 2000, 5000\}$. In Figure 5 and Table 2, for $\sigma = 0.41$ and 1.89, corresponding to privacy $\varepsilon = 25.80$ and 0.48 at $\delta = 10^{-6}$, we report the test loss and accuracy for $K = 500$, and for the two baselines. We see a similar trend as with CIFAR-10.



(a) $\sigma = 1.51$. $\varepsilon = 3.51$.

(b) $\sigma = 20.0$. $\varepsilon = 0.19$. Full plot in Appendix C.3.

Figure 4: CIFAR-10. 4% public data. Test accuracy / loss vs. training steps. Averaged over 3 runs.



(a) $\sigma = 0.41$. $\varepsilon = 25.80$.

(b) $\sigma = 1.89$. $\varepsilon = 0.48$.

Figure 5: EMNIST. 4% public data. Test accuracy / loss vs. training steps. Averaged over 3 runs.

Acknowledgements

We would like to thank Zheng Xu, Ananda Theertha Suresh, Satyen Kale, and Jae Ro for helpful discussions about the user-level experiments.

at the time of this submission, even after contacting the authors. We make our best effort to match their experiment setup. Since the algorithms can be sensitive to hyperparameter choices, for a fair comparison we only use their quoted results.

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.

Agarwal, N., Bullins, B., Chen, X., Hazan, E., Singh, K., Zhang, C., and Zhang, Y. Efficient full-matrix adaptive regularization. In *International Conference on Machine Learning*, pp. 102–110. PMLR, 2019.

Alon, N., Bassily, R., and Moran, S. Limits of private learning with access to public data. *arXiv preprint arXiv:1910.11519*, 2019.

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, 2021.

Avent, B., Korolova, A., Zeber, D., Hovden, T., and Livshits, B. {BLENDER}: Enabling local search with a hybrid differential privacy model. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 2017.

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., Thakkar, O., and Thakurta, A. Model-agnostic private learning. In *NeurIPS*, 2018a.

Bassily, R., Thakurta, A. G., and Thakkar, O. D. Model-agnostic private learning. *Advances in Neural Information Processing Systems*, 2018b.

Bassily, R., Feldman, V., Talwar, K., and Thakurta, A. Private stochastic convex optimization with optimal rates. In *Advances in Neural Information Processing Systems*, 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. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 4381–4391. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper/2020/file/2e2c4bf7ceaa4712a72dd5ee136dc9a8-Paper.pdf`.

Bassily, R., Moran, S., and Nandi, A. Learning from mixtures of private and public populations. *arXiv preprint arXiv:2008.00331*, 2020c.

Biswas, S., Dong, Y., Kamath, G., and Ullman, J. Coinpress: Practical private mean and covariance estimation. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 14475–14485. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/a684eceee76fc522773286a895bc8436-Paper.pdf`.

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

Cohen, G., Afshar, S., Tapson, J., and Schaik, A. V. Emnist: Extending mnist to handwritten letters. *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017. doi: 10.1109/ijcnn.2017.7966217.

Duchi, J., Hazan, E., and Singer, Y. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Dwork, C. and Feldman, V. Privacy-preserving prediction. In *Conference On Learning Theory*, pp. 1693–1702. PMLR, 2018.

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

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., Mironov, I., Talwar, K., and Thakurta, A. Privacy amplification by iteration. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 521–532. IEEE, 2018.

Gur-Ari, G., Roberts, D. A., and Dyer, E. Gradient descent happens in a tiny subspace. *CoRR*, abs/1812.04754, 2018. URL `http://arxiv.org/abs/1812.04754`.

Hayes, T. P. A large-deviation inequality for vector-valued martingales. 2003.

Hazan, E. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Kairouz, P., Ribero, M., Rush, K., and Thakurta, A. Dimension independence in unconstrained private ERM via adaptive preconditioning. *CoRR*, abs/2008.06570, 2020. URL https://arxiv.org/abs/2008.06570.

Kairouz, P., Diaz, M. R., Rush, K., and Thakurta, A. (nearly) dimension independent private erm with adagrad rates
via publicly estimated subspaces. In *COLT*, 2021a.

Kairouz, P., McMahan, B., Song, S., Thakkar, O., Thakurta, A., and Xu, Z. Practical and private (deep) learning without sampling or shuffling. In *ICML*, 2021b.

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009.

Li, X., Tramèr, F., Liang, P., and Hashimoto, T. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

Liu, T., Vietri, G., Steinke, T., Ullman, J., and Wu, Z. S. Leveraging public data for practical private query release. *arXiv preprint arXiv:2102.08598*, 2021.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pp. 1273–1282, 2017a. URL http://proceedings.mlr.press/v54/mcmahan17a.html.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private language models without losing accuracy. *CoRR*, abs/1710.06963, 2017b. URL http://arxiv.org/abs/1710.06963.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Merriman, C. Microsoft reminds privacy-concerned windows 10 beta testers that they're volunteers. *The Inquirer. http://www.theinquirer.net/2374302*, 2014. URL http://www.theinquirer.net/2374302.

Nandi, A. and Bassily, R. Privately answering classification queries in the agnostic pac model. In Kontorovich, A. and Neu, G. (eds.), *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pp. 687–703. PMLR, 08 Feb–11 Feb 2020. URL https://proceedings.mlr.press/v117/nandi20a.html.

Nemirovsky, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. Wiley & Sons, New York, 1983.

Overflow, S. The Stack Overflow Data, 2018. https://www.kaggle.com/stackoverflow/stackoverflow.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

Papernot, N., Thakurta, A., Song, S., Chien, S., and Erlingsson, Ú. Tempered sigmoid activations for deep learning with differential privacy. *arXiv preprint arXiv:2007.14191*, 2020.

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.

Talwar, K., Thakurta, A., and Zhang, L. Private empirical risk minimization beyond the worst case: The effect of the constraint set geometry. *arXiv preprint arXiv:1411.5417*, 2014.

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

Wang, D., Ye, M., and Xu, J. Differentially private empirical risk minimization revisited: Faster and more general. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/f337d999d9ad116a7b4f3d409fcc6480-Paper.pdf.

Wang, J. and Zhou, Z.-H. Differentially private learning with small public data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 2020. doi: 10.1609/aaai.v34i04.6088.

Yu, D., Zhang, H., Chen, W., and Liu, T. 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. URL `https://openreview.net/forum?id=7aogOj_VYO0`.

Zhou, Y., Wu, Z. S., and Banerjee, A. Bypassing the ambient dimension: Private sgd with gradient subspace identification. In *ICLR*, 2020.

A. Notation Reference

We recall here the notation commonly used used throughout the paper:

α	A weighting of public and private gradients
$\mathbf{b}$	Noise added to gradients for privacy
B_Ψ	The Bregman divergence induced by Ψ
$\mathcal{C}$	The constraint set
d	A single sample
$D_{\mathsf{pub}}, D_{\mathsf{priv}}$	The public and private data sets respectively
$\mathcal{D}$	The universe of samples
ε, δ	The privacy parameters
η	The learning rate/step size
$\mathbf{g}$	A (batch) gradient
G_Q	The Gaussian width of Q
$\ell_{\mathsf{pub}}, \ell_{\mathsf{priv}}$	The (per-example) public and private loss functions respectively
L	The Lipschitz constant of ℓ_{priv}
$\mathcal{L}$	A loss on dataset D, or the population loss
λ	An eigenvalue
$n_{\mathsf{pub}}, n_{\mathsf{priv}}$	The number of public and private samples respectively
p	The dimensionality of the optimization problem
Ψ	The empirical public loss, i.e. $\Psi(\theta) = \frac{1}{\lvert D_{\mathsf{pub}}\rvert} \sum_{d \in D_{\mathsf{pub}}} \ell(\theta; d)$; also the mirror map
T	The number of iterations in the optimization algorithm
τ	The distribution from which the training samples are drawn
θ	A solution to the optimization problem

B. Proofs for Section 3

We first show that the set $\mathcal{C}$ contains θ^* with high probability. To do this, we need a bound on the gradients of ℓ at θ^*.

Lemma B.1. *Under Assumption 4.1, for all $d \in supp(\tau)$ we have $\lVert \nabla \ell(\theta^*; d) \rVert_2 \leq 1$.*

Proof. The loss function for the pair $d = (\mathbf{x}, y)$ is $\lVert \mathbf{x} \rVert_2^2$-smooth, and minimized (i.e. has gradient $\mathbf{0}$) at the point $\theta^* + \frac{y - \langle \theta^*, \mathbf{x} \rangle}{\lVert \mathbf{x} \rVert_2^2} \mathbf{x}$. In turn, by smoothness and Assumption 4.1 we have:

$$\lVert \nabla \ell(\theta^*; d) \rVert_2 = \left\lVert \nabla \ell(\theta^*; d) - \nabla \ell(\theta^* + \frac{y - \langle \theta^*, \mathbf{x} \rangle}{\lVert \mathbf{x} \rVert_2^2} \mathbf{x}; d) \right\rVert_2 \leq \lVert \mathbf{x} \rVert_2^2 \cdot \left| \frac{y - \langle \theta^*, \mathbf{x} \rangle}{\lVert \mathbf{x} \rVert_2^2} \right| \cdot \lVert \mathbf{x} \rVert_2 \leq 1.$$

□

We can now show that the gradient of the public loss evaluated at θ^* is bounded with high probability, implying it is in $\mathcal{C}$.

Lemma B.2. *With probability at least $1 - \delta$, for Ψ as defined in Algorithm 1, we have $\lVert \nabla \Psi(\theta^*) \rVert_2 \leq O(\frac{\sqrt{\log(1/\delta)}}{\sqrt{n_{\mathsf{pub}}}})$.*

Proof. Since θ^* is the population minimizer of ℓ in $\mathbb{R}^p$, and $\mathbb{E}_{d \sim \tau}[\ell(\theta; d)]$ is strongly convex, we have $\mathbb{E}_{d \sim \tau}[\nabla \ell(\theta^*; d)] = \mathbf{0}$. The lemma now follows from a vector Azuma inequality (see e.g. (Hayes, 2003)) applied to the vector sum $\nabla \Psi(\theta^*)$, and Lemma B.1, which gives that $\lVert \nabla \ell(\theta^*; d) \rVert_2 \leq 1$ for all $d \in supp(\tau)$. □

We can also use the bound on the gradients $\nabla \ell(\theta^*; d)$ to show every loss function is Lipschitz within the constraint set.

Lemma B.3. *For all d, $\ell(\theta; d)$ is L-Lipschitz within $\mathcal{C}_0$ for $L = O(1)$.*

Proof. Each $\ell(\theta; d)$ is 1-smooth, we have $\theta^* \in \mathcal{C}$. In turn, by smoothness and Lemma B.1, each $\ell(\theta; d)$ is L-Lipschitz for $L = 1 + 2 \lVert \mathcal{C}_0 \rVert_2$, which is $O(1)$ under Assumption 4.1, giving the lemma. □

We now show that the sample Hessian approximates the population Hessian for both D_{priv} and D_{pub}, i.e. the geometry of Ψ matches the population loss' geometry and the private sample loss' geometry.

Lemma B.4. *Let $\hat{H}_{\mathsf{pub}}$ be the Hessian of the public loss function Ψ, and $\hat{H}_{\mathsf{priv}}$ be the Hessian of the private loss function $\frac{1}{n_{\mathsf{priv}}}\sum_{d\in D_{\mathsf{priv}}}\ell(\theta;d)$. Then under Assumption 4.1 with probability $1-\delta$, we have*

$$\frac{1}{2}\bar{H} \preccurlyeq \bar{H} - \frac{\lambda_{\min}(\bar{H})}{2}\mathbb{I} \preccurlyeq \hat{H}_{\mathsf{pub}} \preccurlyeq \bar{H} + \frac{\lambda_{\min}(\bar{H})}{2}\mathbb{I} \preccurlyeq 2\bar{H},$$
$$\frac{1}{2}\bar{H} \preccurlyeq \bar{H} - \frac{\lambda_{\min}(\bar{H})}{2}\mathbb{I} \preccurlyeq \hat{H}_{\mathsf{priv}} \preccurlyeq \bar{H} + \frac{\lambda_{\min}(\bar{H})}{2}\mathbb{I} \preccurlyeq 2\bar{H}.$$

Proof. The outer inequalities $\frac{1}{2}\bar{H} \preccurlyeq \bar{H} - \frac{\lambda_{\min}(\bar{H})}{2}\mathbb{I}$ and $\bar{H} + \frac{\lambda_{\min}(\bar{H})}{2}\mathbb{I} \preccurlyeq 2\bar{H}$ follow from the $\lambda_{\min}(\bar{H})$-strong convexity of the population loss, i.e. $\lambda_{\min}(\bar{H})\mathbb{I} \preccurlyeq \bar{H}$. So it suffices to prove the inner inequalities.

Let H_d be the Hessian of $\ell(\theta;d)$. By 1-smoothness of H and $\lambda_{\min}(\bar{H})$-strong convexity of $\bar{H}$, we have:

$$\mathbf{0} \preccurlyeq H_d \preccurlyeq \mathbb{I} \qquad \forall d,$$
$$\mathbf{0} \preccurlyeq \bar{H} \preccurlyeq \mathbb{I}.$$

And so:

$$-\mathbb{I} \preccurlyeq H_d - \bar{H} \preccurlyeq \mathbb{I} \qquad \forall d.$$

The inner inequalities now follow from a matrix Bernstein inequality, and the sample complexity lower bounds given in Assumption 4.1. □

We can now prove our main result.

Proof of Theorem 4.2. Algorithm 1 is (ε,δ)-DP by Theorem 3.1.

For the utility guarantee, with probability at most 3δ, one of the high probability events described in Lemmas B.2 and B.3 fails to hold. In this case, by e.g., Lemma B.3 we can use a naive bound of $O(\|\mathcal{C}_0\|_2)$ on the loss. Since δ is negligible, the contribution of this case to the expected excess loss is negligible, so we ignore it here. We now wish to follow the analysis of Theorem A.1 in (Talwar et al., 2014). To do so, we need to calculate various parameters in that theorem statement:

- By $\lambda_{\min}(\bar{H})$-strong convexity of Ψ, $\|\mathcal{C}\|_2 = O(\min\{1, \frac{\sqrt{\log(1/\delta)}}{\lambda_{\min}(\bar{H})\sqrt{n_{\mathsf{pub}}}}\})$.
- We can assume without loss of generality that $\|\theta\|_2 \leq r/2$. This is because if we replace r with $2r$ in the definition of $\mathcal{C}_0$, the parameters of the problem do not change asymptotically, but this condition is now enforced. Under this assumption, any line passing through θ^* has an intersection with $\mathcal{C}_0$ of length $\Omega(1)$. Now, by strong convexity and the definition of $\mathcal{C}$, this implies $\mathcal{C}$ is contained within an ellipsoid $\widetilde{Q}$ whose axes are the eigenvectors of $\hat{H}_{\mathsf{pub}}$, and whose axis lengths are $\Theta(\min\{1, \frac{\sqrt{\log(1/\delta)}}{\lambda_i\sqrt{n_{\mathsf{pub}}}}\})$. Furthermore, $\mathcal{C}$ contains $\widetilde{Q}$ rescaled in all dimensions by a constant. This means the symmetric convex hull Q of $\mathcal{C}$ is also contained in $\widetilde{Q}$, and contains $\widetilde{Q}$ rescaled by a constant. So the strong convexity of Ψ with respect to the Q-norm is within a constant factor of the strong convexity of Ψ with respect to the $\widetilde{Q}$-norm.

  Now, let $\|\cdot\|_{\widetilde{Q}}$ be the Minkowski $\widetilde{Q}$-norm $\|\mathbf{x}\|_{\widetilde{Q}} = \min\{a \in \mathbb{R}_{\geq 0} : \mathbf{x} \in a\widetilde{Q}\}$. In the direction of the ith eigenvector, Ψ is $\frac{1}{\lambda_i(\hat{H}_{\mathsf{pub}})}$-strongly convex with respect to the norm $\|\cdot\|_{Q'}$ for the set $Q' = \{\theta \in \mathbb{R}^p : \|\nabla\Psi(\theta)\|_2 \leq 1\}$, so it is $\Theta(\frac{\min\{\lambda_i(\hat{H}_{\mathsf{pub}})^2, \log(1/\delta)/n_{\mathsf{pub}}\}}{\lambda_i(\hat{H}_{\mathsf{pub}})})$-strongly convex with respect to the $\widetilde{Q}$-norm, and thus the Q-norm, in this direction. So Δ, the strong convexity parameter of Ψ with respect to the Q-norm is:

$$\Delta = \Theta\left(\min_i\left\{\frac{\min\{\lambda_i(\hat{H}_{\mathsf{pub}})^2, \log(1/\delta)/n_{\mathsf{pub}}\}}{\lambda_i(\hat{H}_{\mathsf{pub}})}\right\}\right) = \Theta\left(\min\left\{\lambda_{\min}(\hat{H}_{\mathsf{pub}}), \frac{\log(1/\delta)}{\lambda_{\max}(\hat{H}_{\mathsf{pub}})n_{\mathsf{pub}}}\right\}\right)$$

By Lemma B.4, conditioned on the event in that lemma Δ is

$$\Theta\left(\min\left\{\lambda_{\min}(\bar{H}), \frac{\log(1/\delta)}{\lambda_{\max}(\bar{H})n_{\mathsf{pub}}}\right\}\right).$$

- By a similar argument to the previous item, we get that the squared Gaussian width $G_{\mathcal{C}}^2$ is at most $G_{\widetilde{Q}}^2$, which is

$$O\left(\sum_i \min\left\{1, \frac{\log(1/\delta)}{\lambda_i(\bar{H})^2 n_{\mathsf{pub}}}\right\}\right)$$

- By Lemma B.4, conditioned on the event in that lemma, the Hessians of the public sample loss, private sample loss, and population loss are constant-approximations of each other.. From the definition of strong convexity with respect to a function (see Section 2.2 of (Talwar et al., 2014)), any quadratic function is 1-strongly convex with respect to itself, and in turn $\Theta(1)$-strongly convex with respect to another quadratic function whose Hessian is within a constant factor of its own, since this implies the Bregman divergences induced by the two functions are also within a constant factor. So the sample private loss $\frac{1}{n_{\mathsf{priv}}}\sum_{d\in D_{\mathsf{priv}}}\ell(\theta; d)$ is $\Theta(1)$-strongly convex with respect to Ψ.

We will view Algorithm 1 as equivalently using $\Psi' = \frac{1}{\Delta}\Psi$ in place of Ψ, and $\eta_t' = \Delta\eta_t$ in place of η_t. Ψ' is 1-strongly convex with respect to the Q-norm, and the sample private loss is now $\Theta(\Delta)$-strongly convex with respect to Ψ'. Now, following the proof of Theorem A.1 in (Talwar et al., 2014), setting $\eta_t' = 1/\Delta t$ and $T = \frac{\|\mathcal{C}\|_2^2(\varepsilon n_{\mathsf{priv}})^2}{\|\mathcal{C}\|_2^2 + G_{\mathcal{C}}^2}$, conditioned on the high probability events we get an excess empirical loss bound of:

$$\widetilde{O}\left(\frac{\log(1/\delta)\max\{\frac{1}{\lambda_{\min}(\bar{H})}, \lambda_{\max}(\bar{H})n_{\mathsf{pub}}\}\cdot\sum_i \min\left\{1, \frac{\log(1/\delta)}{\lambda_i(\bar{H})^2 n_{\mathsf{pub}}}\right\}}{\varepsilon^2 n_{\mathsf{priv}}^2}\right).$$

For an excess population loss bound, we need to show uniform stability. Note that since the Hessian of Ψ, $\bar{H}_{\mathsf{pub}}$, is fixed everywhere then PDA-DPMD just applies $\bar{H}_{\mathsf{pub}}^{-1} \preccurlyeq O(1/\lambda_{\min}(\bar{H}))\cdot\mathbb{I}_p$ to the noisy gradients. This implies that each step of PDA-DPMD is contractive, and thus that the uniform stability parameter of PDA-DPMD is $O(1/\lambda_{\min}(\bar{H}))$ times that of DP-SGD using the same settings of η_t, T. The uniform stability of DP-SGD on a convex L-Lipschitz loss is $O(\frac{L^2\sum_t \eta_t}{n})$ (see e.g. Appendix A of (Bassily et al., 2019) for a proof). Plugging in the parameters for our setting, this is $O(\log(\varepsilon n_{\mathsf{priv}})n_{\mathsf{priv}})$, so PDA-DPMD has uniform stability parameter $O(\log(\varepsilon n_{\mathsf{priv}})/(\lambda_{\min}(\bar{H})n_{\mathsf{priv}}))$. The expected excess population loss is at most the uniform stability parameter plus the expected excess empirical loss, giving the theorem. □

Proof of Theorem 4.3. Let $\mathbf{b}_t$ be the noise added for privacy. Without noise, mirror descent would set θ^* to be such that:

$$-\eta\mathbf{g}_t = \nabla\Psi(\theta^*) - \nabla\Psi(\theta_t).$$

Similarly, given the noisy gradient $\mathbf{g}_t + \mathbf{b}_t$, mirror descent would set $\hat{\theta}$ to be such that:

$$-\eta(\mathbf{g}_t + \mathbf{b}_t) = \nabla\Psi(\hat{\theta}) - \nabla\Psi(\theta_t).$$

We then have:

$$-\eta\mathbf{b}_t = \nabla\Psi(\hat{\theta}) - \nabla\Psi(\theta^*).$$

In turn, recalling that Ψ has a fixed Hessian we have:

$$\hat{\theta} - \theta^* = -\eta\bar{H}_{\mathsf{pub}}^{-1}\mathbf{b}_t$$

We can now directly prove the theorem:

$$\mathbb{E}\left[|\langle\hat{\theta}-\theta^*,\mathbf{v}\rangle|\right]=\eta\mathbb{E}\left[|\langle\bar{H}_{\mathsf{pub}}^{-1}\mathbf{b}_t,\mathbf{v}\rangle|\right]$$

$$=\eta\mathbb{E}\left[|\langle(\sum_i\frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{b}_t,\mathbf{v}\rangle|\right]=\eta\mathbb{E}\left[|\sum_i\frac{a_i}{\lambda_i}\langle\mathbf{b}_t,\mathbf{v}_i\rangle|\right]$$

$$=\eta\mathbb{E}\left[|\sum_i N(0,(a_i/\lambda_i)^2)|\right]=\eta\mathbb{E}\left[|N(0,\sum_i(a_i/\lambda_i)^2)|\right]=\sqrt{\frac{2}{\pi}}\cdot\eta\sigma\sqrt{\sum_i\left(\frac{a_i}{\lambda_i}\right)^2}.$$

□

C. Additional Details on Real-world Experiments

C.1. First-order Approximation to Mirror Descent

In practice, the Mirror Descent (MD) step in Line 5 of Algorithm 1 is computationally the most challenging. In the following, we provide an approximation of this step in the setting where i) the problem is *unconstrained*, i.e., $\mathcal{C}=\mathbb{R}^p$ and ii). the public loss $\Psi(\theta)$ may not be strongly convex with respect to the ℓ_2-norm. This approximation makes Algorithm 1 efficient in practice.

Consider the following equivalence of Line 5 of Algorithm 1, with $\Psi(\theta)$, the public loss on D_{pub}, replaced by $\widehat{\Psi}(\theta)=\Psi(\theta)+\frac{1}{2}\|\theta\|_2^2$. This follows from Lemma 5.5 of (Hazan, 2019).

$$\theta_{t+1}\leftarrow\arg\min_{\theta\in\mathbb{R}^p}\left\langle\sum_{i=1}^t\eta_i\left(\mathbf{g}_i+\mathbf{b}_i\right),\theta\right\rangle+\widehat{\Psi}(\theta) \tag{2}$$

$$=\arg\min_{\theta\in\mathbb{R}^p}\sum_{i=1}^t\eta_i\left(\langle\mathbf{g}_i+\mathbf{b}_i,\theta\rangle+\frac{1}{t\eta_i}\Psi(\theta)\right)+\frac{1}{2}\|\theta\|_2^2$$

$$\approx\arg\min_{\theta\in\mathbb{R}^p}\sum_{i=1}^t\eta_i\left\langle\mathbf{g}_i+\mathbf{b}_i+\frac{1}{t\eta_i}\nabla\Psi(\theta_i),\theta\right\rangle+\frac{1}{2}\|\theta\|_2^2, \tag{3}$$

where (3) follows from the first-order approximation $\Psi(\theta)\approx\Psi(\theta_i)+\langle\nabla\Psi(\theta_i),\theta-\theta_i\rangle$.

In the experiments, we replace $\mathbf{g}_i+\mathbf{b}_i+\frac{1}{t\cdot\eta_i}\nabla\Psi(\theta_i)$ with $\alpha_i(\mathbf{g}_i+\mathbf{b}_i)+(1-\alpha_i)\nabla\Psi(\theta_i)$, where $\alpha_i\in(0,1]$. This reparamertization helps with more effective hyperparameter tuning while training deep learning models. We therefore have the update rule (1).

C.2. Setup

Network architectures: Table 3 shows model architectures for CIFAR-10, EMNIST, WikiText-2, & StackOverflow.

Hyperparameter Tuning: We keep the clipping norm to be 1. One small difference with the standard DP-SGD update rule is that we enforce an additional clipping step for the privatized gradient for the image classification task, where the clipping norm is the same as the clipping norm of for individual gradient. The reason for this additional step is that the norm of the averaged clipping gradients should still be upper bounded by the clipping norm.

The only hyperparameters that need to be tuned are the learning rate and K that controls the decaying of α_t. For the learning rate, we consider a grid of $\{1,2,5\}\times 10^i$ for different is such that the optimal learning rate does not appear on the boundary. We search for the optimal K in $\{100,200,500\}$ for WikiText-2 and $\{200,500,1000,2000,5000\}$ for the image classification tasks.

C.3. Full Plots for Section 5

In Figure 6, we plot the complete version of Figure 3, and in Figure 7, we show the complete version of Figure 4.

Table 3: Model architectures for real data experiments.

(a) Model architecture for CIFAR-10.

Layer	Parameters
Convolution $\times 2$	32 filters of 3×3, strides 1
Max-Pooling	2×2, stride 2
Convolution $\times 2$	64 filters of 3×3, strides 1
Max-Pooling	2×2, stride 2
Convolution $\times 2$	128 filters of 3×3, strides 1
Max-Pooling	2×2, stride 2
Fully connected	128 units
Softmax	-

-

(b) Model architecture for EMNIST.

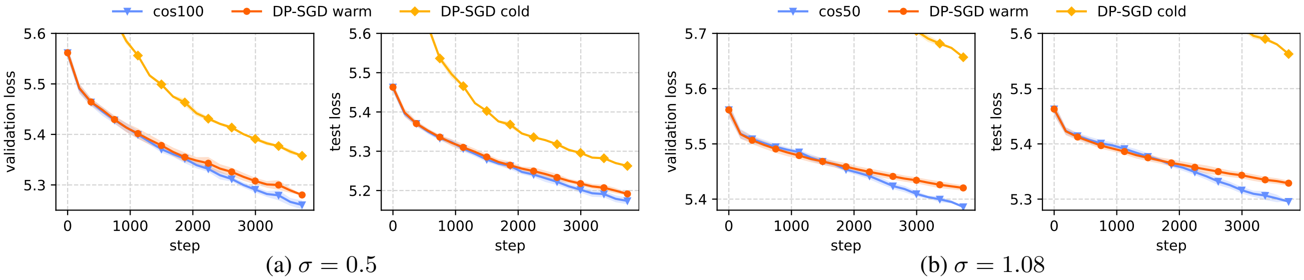
Layer	Parameters
Convolution	16 filters of 8×8, strides 2
Convolution	32 filters of 4×4, strides 2
Fully connected	32 units
Softmax	-

(c) Model architecture for WikiText-2.

Layer	Parameters
Input	8000
Fully connected	120
LSTM $\times 2$	120 hidden units
Fully connected	8000
Softmax	-

(d) Model architecture for StackOverflow.

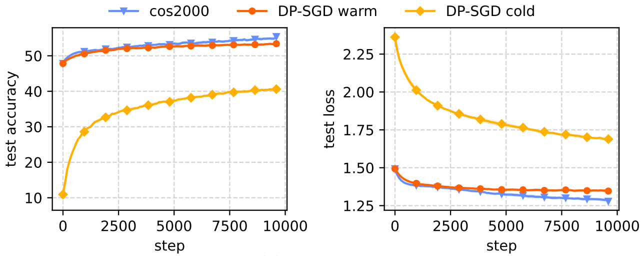
Layer	Parameters
Input	0
embedding	960384
LSTM	2055560
Fully connected	64416
Fully connected	970388
Softmax	-



(a) $\sigma = 0.5$

(b) $\sigma = 1.08$

Figure 6: Full plot for Figure 3. WikiText-2. 4% public data.



(a) $\sigma = 20.0$.

Figure 7: Full plot of Figure 4. CIFAR-10. 4% public data. Test accuracy / loss vs. training steps. Averaged over 3 runs.

C.4. WikiText-2 with WikiText-103 as Public Data

We compare with the SoTA (Asi et al., 2021) which uses WikiText-103 as public data. Specifically, we consider their "LargeAux" setting under $\varepsilon = 1.0$ and 3.0. Since the implementation for Asi et al. (2021) is not public as of writing this work, we make our best effort to match their experiment setup. We note that the data preprocessing and the number of iterations used (thus the noise multiplier for achieving the same ε) might differ.

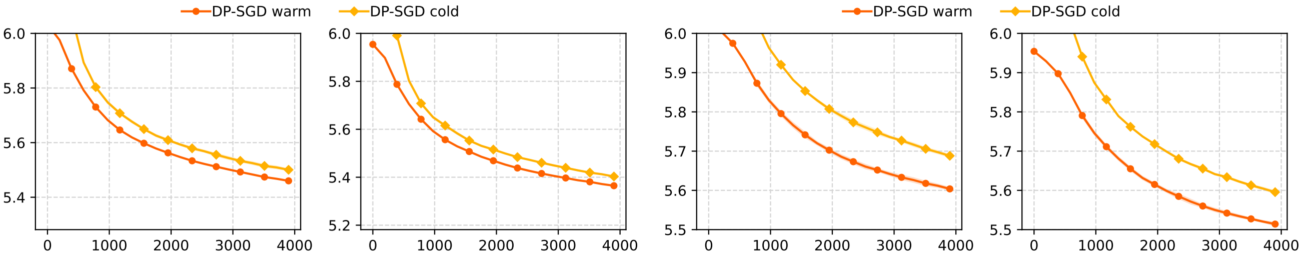
We preprocess WikiText-103 as follows. After processing WikiText-2 as described in Section 5, we convert all words that does not appear in the processed WikiText-2 as the unknown token. Then, we split the sentences into length-35 sequences, and remove all sequences that overlap with WikiText-2. Finally, we randomly sample 48,764 sequences, in order to match the "LargeAux" setting where the public dataset is of the same size as the private training dataset.

Figure 8 shows the results. In our setting, cold start DP-SGD reaches similar log perplexity as those in (Asi et al., 2021), while the warm-start DP-SGD is already better than (Asi et al., 2021) (LargeAux). The final test log perplexities are summarized below, with the results in (Asi et al., 2021) converted from perplexity to log perplexity.

C.5. Comparison with Prior Works

On the caveats of comparing with GEP (Yu et al., 2021): We tried to use the implementation of GEP provided by (Yu et al., 2021) for our experiments on CIFAR-10 with the VGG network we use. However, in spite of using a TeslaV100 GPU, the runs were facing out-of-memory issues. Techniques like GEP and Biased-GEP ((Zhou et al., 2020)) require computing a low-rank subspace for gradient projection; it has been mentioned in (Asi et al., 2021) as well that this can be quite challenging for empirical comparison. Note that the DP-SGD baseline for CIFAR-10 considered in (Yu et al., 2021)

Algorithm	$\varepsilon = 3.0$	$\varepsilon = 1.0$
(Asi et al., 2021) DP-SGD (cold)	5.4819	5.6623
(Asi et al., 2021) (LargeAux)	5.4324	5.5254
Our DP-SGD (cold)	5.4030	5.5956
Our DP-SGD (warm)	5.3646	5.5141



(a) $\sigma = 0.83$, $\varepsilon = 3.0$.

(b) $\sigma = 1.49$, $\varepsilon = 1.0$.

Figure 8: WikiText-2. WikiText-103 as public data.

achieves 52.2% test accuracy at $\varepsilon = 5$ for $\delta = 10^{-5}$, whereas our DP-SGD coldstart baseline achieves 62.96% accuracy at $\varepsilon = 3.51$ at the same δ. Though not directly comparable, under the same public/private data setup on CIFAR-10, best effort results for (Asi et al., 2021) on their network with their method reaches 65.19% accuracy at $\varepsilon = 3.51$. Meanwhile, our algorithm on our network reaches 67.03%.

Comparison with (Asi et al., 2021): We conduct experiments in the setting of (Asi et al., 2021) using PDA-DPMD (Figure 8), and observe that the results are visibly identical to warmstart DP-SGD. The test log perplexity for PDA-DPMD is 5.5146 for $\varepsilon = 1.0$, and 5.3728 for $\varepsilon = 3.0$. This could be attributed to the fact that the public data here (WikiText-103) is not in-distribution with the private data (WikiText-2), and thus the benefits beyond warm-starting might be limited in this case. Note that this is still better than (Asi et al., 2021) (5.5254 for $\varepsilon = 1.0$, and 5.4324 for $\varepsilon = 3.0$).