A. Supplementary Material

In the supplementary material we perform, (1) Ablation studies analyzing different components of our algorithm (Appendix A.1), (2) tuning the adaptive clipping threshold can further improve AdaMix (A.2), (3) show that using a larger σ (training for longer) is better (A.3), (4) provide additional experimental results (A.4), (5) provide details experimental details (A.5, A.6), and (6) proofs for all the theoretical results presented in the paper (Appendix B, C, D).

A.1. Ablation Studies

In Fig. 5, we ablate the two components of our algorithm, namely, Subspace Projection and Adaptive Clipping. We consider the Noisy-GD (NGD) as the baseline and analyze the improvement provided by the two components separately and finally in combination (AdaMix). We observe that adaptive clipping is the key component of the AdaMix algorithm. We show that using the two components together performs the best across multiple datasets. In the next section we show that further tuning the adaptive clipping threshold can improve the performance even more on some datasets. In Fig. 6, we plot the relative reconstruction error for the private gradients on the public gradient subspace and a random subspace. We observe that the reconstruction error on the public gradient subspace is at least half the random subspace throughout training, which suggests the importance of the public gradients for AdaMix.

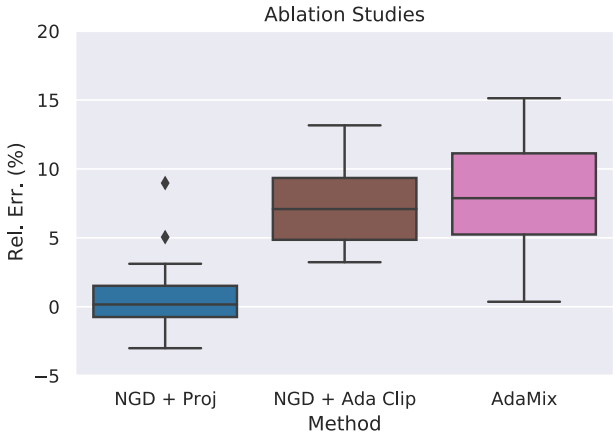

Figure 5. **Ablation Studies** Box plot showing the relative decrease in the test error across multiple datasets when we add Subspace Projection, Adaptive Clipping and Subspace Projection + Adaptive Clipping (AdaMix) to NGD. We observe that adaptive clipping provides more improvement compared to subspace projection, and the combination of the two works the best across 6 datasets.

A.2. Effect of AdaMix clip threshold

We plot the effect of adaptive clipping threshold (percentile) on the test error for AdaMix (we use 90 percentile in all of our experiments) in Fig. 7. However for CUB-200 and Caltech-256 tuning it to 75 percentile works better indicating that AdaMix has more scope for improvement. Note, however that using 90 percentile for CUB-200 and Caltech-256 still outperforms all the baselines.

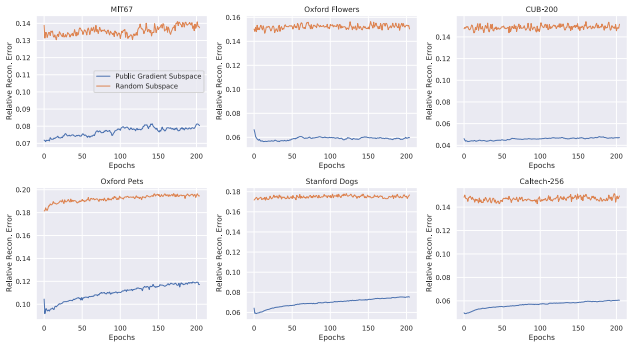

Figure 6. **Random Subspace Projection** We plot the relative reconstruction error of the private gradients when projecting on the subspace spanned by the public gradients or on a random subspace during training (using AdaMix and $\epsilon = 3$). The reconstruction error when projecting on the random subspace is generally more than twice the error obtained projecting on the subspace of public gradients, highlighting the importance of using the latter.

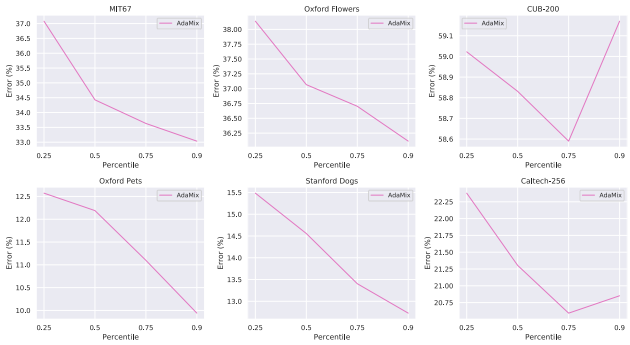

Figure 7. **Effect of adaptive clipping threshold** We plot the test accuracy of AdaMix using different values of adaptive clipping threshold percentile.

A.3. Longer training with more noise

Higher values of σ requires more training steps, however, Theorem 3 shows that it leads to better convergence, which leads to better generalization. In Fig. 8 we see that indeed, at the same level of privacy, using a large of σ significantly reduces the test error compared to using a smaller σ.

A.4. Additional Experiments

In the main paper we presented detailed experimental results on MIT-67, here we present detailed results on all the remaining datasets.

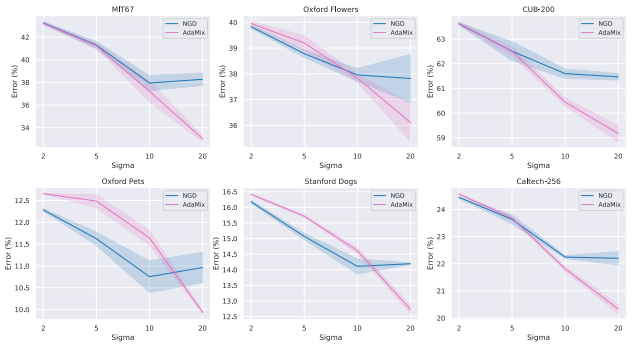

Figure 8. **Larger values of σ performs better.** We plot the test accuracy of NGD and AdaMix using different values of σ for $\epsilon = 3$. A larger σ performs better but needs more training steps thus verifying the claim empirically across multiple datasets.

A.4.1 Test Error vs Privacy

We show the test error obtained by different methods for different levels of privacy ϵ and the robustness to membership attack, similar to Fig. 1 for different datasets in Figs. 9 to 13

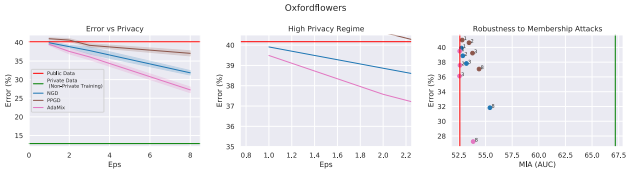

Figure 9. Test error vs Privacy and Robustness to Membership Attacks on Oxford-Flowers

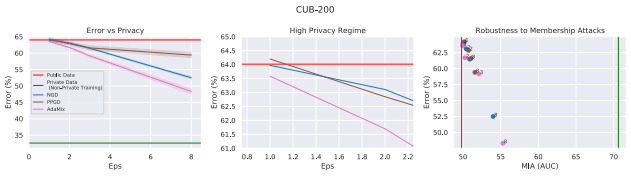

Figure 10. Test error vs Privacy and Robustness to Membership Attacks on CUB-200

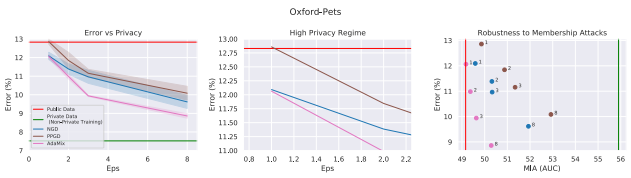

Figure 11. Test error vs Privacy and Robustness to Membership Attacks on Oxford-Pets

A.4.2 Per-Instance Privacy

We show the pDP loss for NGD and AdaMix for different datasets in Figs. 14 to 18, similar to Fig. 3 (we use $\epsilon = 3$).

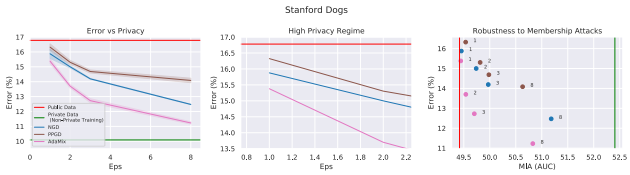

Figure 12. Test error vs Privacy and Robustness to Membership Attacks on Stanford Dogs

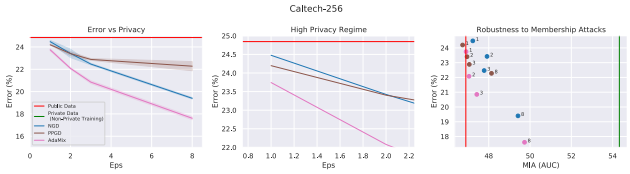

Figure 13. Test error vs Privacy and Robustness to Membership Attacks on Caltech-256

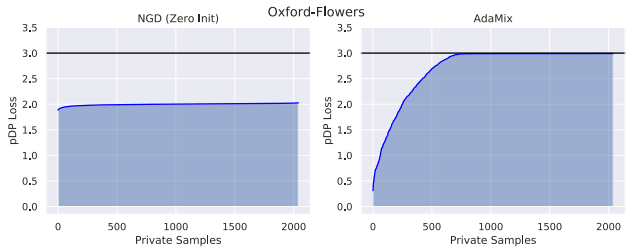

Figure 14. Effect of public data on per-instance DP for Oxford Flowers

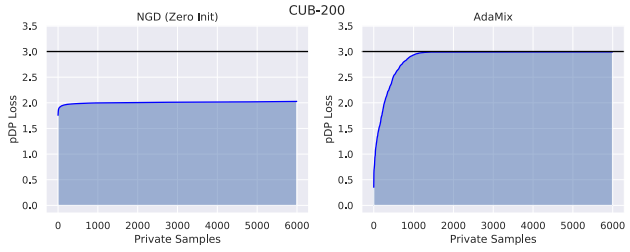

Figure 15. Effect of public data on per-instance DP for CUB-200

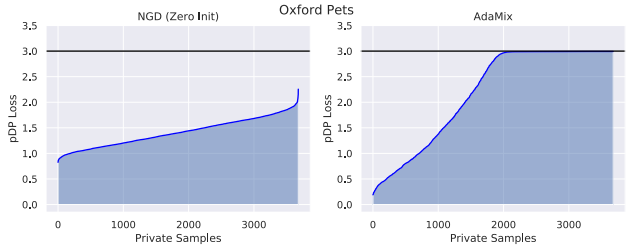

Figure 16. Effect of public data on per-instance DP for Oxford Pets

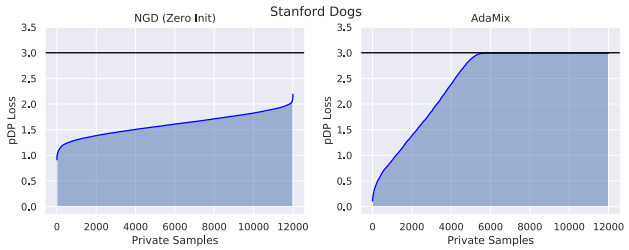

Figure 17. Effect of public data on per-instance DP for Stanford-Dogs

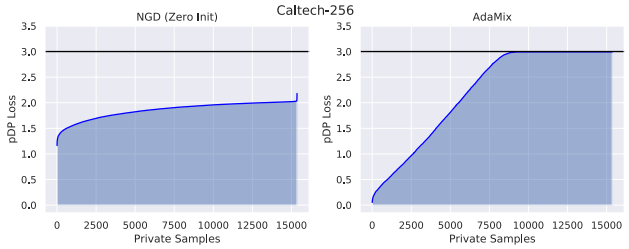

Figure 18. Effect of public data on per-instance DP for Caltech-256

A.5. Multi-modal Initialization

We show the effect of using different initializations and CLIP features for different datasets in Figs. 19 to 22, similar to Fig. 4 (we use $\epsilon = 3$). We show that for Stanford Dogs and Oxford Pets datasets (Fig. 19 and Fig. 20) which have images which are very similar to images in ImageNet, ResNet-50 trained on ImageNet performs better than CLIP features. However, the trend for the effect of initialization remains the same across all the datasets.

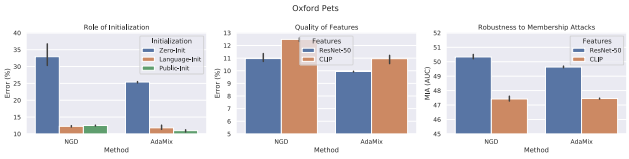

Figure 19. Multi-model initialization and models for Oxford Pets

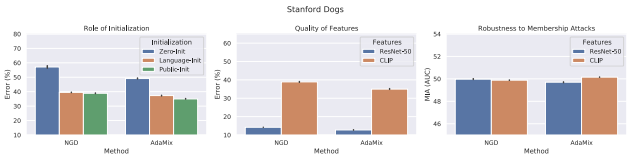

Figure 20. Multi-model initialization and models for Stanford Dogs

A.6. Experimental Details

We use Auto-DP library for all of our experiments. For ResNet-50 features we use the `torchvision` version of the ResNet-50 model and for CLIP features we use the model provided.[3]

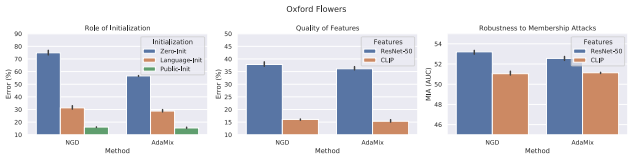

Figure 21. Multi-model initialization and models for Oxford Flowers

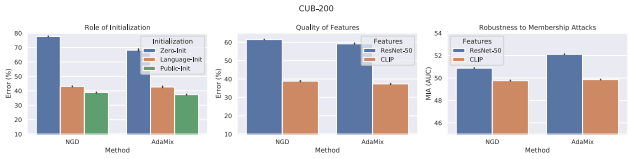

Figure 22. Multi-model initialization and models for CUB-200

To create the public and private datasets, we take 2 samples (for Oxford-Flowers and CUB-200) and 5 samples (for MIT-67, Stanford Dogs, Oxford Pets, Caltech-256) from each class as the public dataset and the remaining samples as the private dataset. In this way, the public set is less than 10% of the private set. We repeat all the experiments with 3 random seeds and report its mean and std.

For all the experiments we try multiple values for the learning rate: $\{1e-3, 2.5e-3, 5e-3\}$ for ResNet-50 experiments and $\{1e-6, 5e-7, 1e-7\}$ for CLIP experiments, and report the best values across 3 random seeds. We use $1e-2$ as the L_2 regularization coefficient across all the experiments. For subspace projection we project the gradients on a 2000-dimensional subspace for experiments with ResNet-50 features and 500-dimensional subspace for experiments with CLIP features. For the clipping threshold percentile we use a constant value of 90 percentile across all the experiments in the paper. In Fig. 7, we show that further tuning the clipping threshold percentile can improve performance on certain datasets, however, even with a pre-decided value of 90 percentile it still out-performs all the other methods.

We train the logistic model on the public data (few shot data) for 200 epochs (sames as iterations since we use gradient descent) for multiple values of learning rate: $\{1e-1, 5e-2, 1e-2\}$ for ResNet-50 features and $\{1e-4, 5e-5, 1e-5\}$ for CLIP features and choose the best performing model. For the private training, we use $\sigma = 20$ for all the experiments and calculate the number of iterations required for private training using methods provided in Auto-DP. In the experiments we observe that choosing a higher value of σ (thus training for more iterations) generally results in better performance.

[3] https://github.com/openai/CLIP

B. Differential privacy basics

In this section, we describes tools we need from differential privacy and use them to prove that Algorithm 2 and Algorithm 3 satisfies a family of differential privacy parameters.

Lemma 6 (Analytical Gaussian mechanism [5])**.** *For a numeric query $f : \mathcal{X}^* \to \mathbb{R}^d$ over a dataset $\mathcal{D}$, the randomized algorithm that outputs $f(\mathcal{D}) + Z$ where $Z \sim \mathcal{N}(0, \sigma^2 I_d)$ satisfies $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$ and $\delta(\epsilon) = \Phi(\frac{\mu}{2} - \frac{\epsilon}{\mu}) - e^{\epsilon}\Phi(-\frac{\mu}{2} - \frac{\epsilon}{\mu})$ with parameter $\mu := \Delta/\sigma$, where $\Delta := \Delta_2^{(f)} = \max_{\mathcal{D}\sim\mathcal{D}'} \|f(\mathcal{D}) - f(\mathcal{D}')\|_2$ is the global L2 sensitivity of f and Φ is the CDF function of $\mathcal{N}(0, 1)$.*

The above lemma tightly characterizes the (ϵ, δ)-DP guarantee of a single invocation of the Gaussian mechanism, and the following lemma shows that we can use the same result for an adaptive composition of a sequence of Gaussian mechanisms.

Definition 7 (Gaussian Differential privacy [14])**.** *We say a mechanism $\mathcal{M}$ satisfies μ-Gaussian differential privacy (GDP), if it satisfies $(\epsilon, \delta(\epsilon))$-DP for all $\epsilon \geq 0$ and $\delta(\epsilon)$ being that of a single Gaussian mechanism (in Lemma 6) with parameter μ.*

Lemma 8 (Composition of Gaussian mechanisms [14])**.** *The adaptive composition of a sequence of Gaussian mechanism with noise level $\sigma_1, \sigma_2, \ldots$ and global L2 sensitivity $\Delta_1, \Delta_2, \ldots$ satisfies μ-GDP with parameter $\mu = \left(\sum_i (\Delta_i/\sigma_i)^2\right)^{1/2}$.*

Specifically, the noisy gradient descent (NoisyGD) algorithm (Algorithm 1) we use is a composition of T Gaussian mechanisms for releasing the gradients and its privacy guarantee is equivalent to that of a single Gaussian mechanism.

Proposition 9. *Let $\nabla f(w)$ be a function of the private dataset with global L2 sensitivity smaller than L for any $w \in \mathcal{W}$, then Algorithm 1 with parameter T, σ^2 such that $\rho := \frac{T^2 L^2}{2\sigma^2}$ satisfies $\sqrt{2\rho}$-Gaussian differential privacy.*

Proof. The proof follows from Lemma 8 as Algorithm 1 is an adaptive composition of T Gaussian mechanisms with global sensitivity L. □

C. Per-instance differential privacy

In this section, we provide details on the per-instance differential privacy [55] that we used to generate Figure 3. To cleanly define per-instance differential privacy, we first define indistinguishability.

Definition 10 ((ϵ, δ)-indistinguishability)**.** *We say two distributions P, Q are (ϵ, δ)-indistinguishable if for any measurable set S*

$$\mathsf{Pr}_P[S] \leq e^{\epsilon} \cdot \mathsf{Pr}_Q[S] + \delta \text{ and } \mathsf{Pr}_Q[S] \leq e^{\epsilon} \cdot \mathsf{Pr}_P[S] + \delta.$$

Definition 11 ([55])**.** *We say a randomized algorithm $\mathcal{M}$ is $(\epsilon(\cdot), \delta)$-per-instance differentially private (pDP) for scalar $\delta \geq 0$ and function $\epsilon : \mathcal{Z}^* \times \mathcal{Z} \to \mathbb{R}_+$, such that for any dataset $D \in \mathcal{Z}^*$, individual $z \in \mathcal{Z}$, $\mathcal{M}(D)$ and $\mathcal{M}(D \cup \{z\})$ are $(\epsilon(D, z), \delta)$-indistinguishable.*

pDP loss ϵ is a function of one particular pair of neighboring datasets. It describes the formal privacy loss incurred to the particular individual z that is added (if z is not part of the input dataset) or removed (if z is part of the input dataset). pDP is a strict generalization of DP, as we can recover DP from pDP by maximizing $\epsilon(\cdot)$ over D, z.

Lemma 12. *If $\mathcal{M}$ is $(\epsilon(\cdot), \delta)$-pDP, then $\mathcal{M}$ is also $(\sup_{D\in\mathcal{Z}^*, z\in\mathcal{Z}} \epsilon(D, z), \delta)$-DP.*

We emphasize that the pDP loss $\epsilon(\cdot)$ itself is data-independent, but specific evaluations of the pDP losses (e.g., $\epsilon(D_{-z}, z)$ or $\epsilon(D, z)$) depends on the private dataset D, thus should not be revealed unless additional care is taken to privately release these numbers.

For our purpose, we are interested in the distribution of pDP losses of individuals in the dataset induced by different DP algorithms. This is used to provide a theoretically-sound alternative to the prevalent practices of using specific attacks (such as membership inference attacks) for evaluating the data-dependent privacy losses. Before we state the pDP bounds of our algorithms, we extend the standard $(\epsilon(\cdot), \delta)$-pDP definition to a per-instance version of the Gaussian DP.

Definition 13. *We say a mechanism is $\mu(\cdot)$-per-instance Gaussian Differentially Private (pGDP), if $(D, D \cup z)$ (and $(D \cup z, D)$) are (ϵ, δ)-indistinguishable for all ϵ, δ parameters described by $\mu(D, z)$-GDP.*

Algorithm 3: (Theoretical version of) AdaMix training algorithm (no clipping, no adaptive projection, slightly different pretraining, theoretically chosen learning rate).

Data: Public dataset D_{pub}, private dataset D_{pri}, privacy parameter (ϵ, δ), noise variance σ, Lipschitz constant L[4], population-level strong convex parameter c, regularization parameter λ and a constraint set $\mathcal{W}$.

Result: $\bar{w}$

// Public Pretraining Phase (OnePassSGD on D_{pub}):

$w_1 = 0$. **for** $t = 1, \ldots, N_{pub}$ **do**

  // In a shuffled order

  $\eta_t = \frac{2}{c(t+1)}$;

  $w_{t+1} \leftarrow \Pi_{\mathcal{W}}\big(w_t - \eta_t \nabla \ell(w_t, (\tilde{x}_t, \tilde{y}_t))\big)$;

end

$w_{\text{ref}} \leftarrow w_{N_{\text{pub}}+1}$.

// Mix Training Phase (NoisyGD on $D_{\text{pub}} \cup D_{\text{pri}}$):

$T \leftarrow \text{Calibrate}(\epsilon, \delta, \sigma)$ // (i.e., $\frac{T\tau^2}{2\sigma^2} =: \rho$)

// NoisyGD on objective function $\mathcal{L}(w) + \frac{\lambda}{2}\|w - w_{\text{ref}}\|^2$

$w_1 = w_{\text{ref}}$;

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

  $\eta_t \leftarrow \frac{2}{\lambda(t+1)}$;

  $n_t \sim N(0, \sigma^2 I)$;

  $w_{t+1} \leftarrow \Pi_{\mathcal{W}}\big(w_t - \eta_t(\sum_{i=1}^{N_{\text{pri}}} \nabla \ell_i(w_t) + \sum_{j=1}^{N_{\text{pub}}} \nabla \tilde{\ell}_j(w_t) + n_t)\big)$;

end

// The following averaging can be implemented incrementally without saving w_t

$\bar{w} \leftarrow \sum_{t=1}^{T} \frac{2t}{T(T+1)} w_t$

This allows us to obtain precise pDP bounds under composition.

Proposition 14 (pDP analysis of AdaMix)**.** *Let $z_1, ..., z_n$ be the data points of the private dataset. Algorithm 3 satisfies $\mu(\cdot)$-pGDP with*

$$\mu(D_{-i}, z_i) = \sqrt{\sum_{t=1}^{T} \frac{\min\{\|\nabla \ell_i(w_t)\|, \tau\}^2}{\sigma^2}}.$$

Similarly, Algorithm 2 satisfies $\mu(\cdot)$-pGDP with

$$\mu(D_{-i}, z_i) = \sqrt{\sum_{t=1}^{T} \frac{\|U^T \tilde{g}_i^{pri}(w_t)\|_2^2}{\tau_t^2 \sigma^2}}.$$

Proof. Both algorithms are the composition of T-Gaussian mechanisms. Thus the results follow by the composition of pGDP. The composition of pGDP is implied by the composition theorem of GDP (Lemma 8) by choosing the space of datasets to be just $\{D, D \cup \{z\}\}$. □

Fixing any δ, we can then compute the corresponding $\epsilon(\cdot)$ for $(\epsilon(\cdot), \delta)$-pDP using the formula of Gaussian mechanism with μ taken to be $\mu(\cdot)$ pointwise.

D. Proofs of the technical results

We will be using the following $O(1/t)$ convergence bound due to Lacoste-Julien, Schmidt and Bach [28], which uses a decaying learning rate and a non-uniform averaging scheme.

[4] As we discussed earlier, per-example gradient clipping can be viewed as Huberizing the loss function in GLMs [48], all our results apply to the updated loss function. The adaptive clipping approach we took, can be viewed as an heuristic that automatically identifies an appropriate level of Huberization.

Theorem 15 (Convergence of SGD for Strongly Convex Objectives [28]). *Let f be a m-strongly convex and defined on a convex set $\mathcal{W}$. Assume stochastic gradient oracle g_t satisfies that $\mathbb{E}[g_t|w_t] \in \partial f(w_t)$ and $\mathbb{E}[\|g_t\|^2] \leq G^2$ for all $t = 1, ..., T$. Then the (projected) stochastic gradient descent with learning rate $\eta_t = \frac{2}{m(t+1)}$ satisfies*

$$\mathbb{E}[f(\sum_{t=1}^{T} \frac{2t}{T(T+1)} w_t)] - f(w^*) \leq \frac{2G^2}{m(T+1)} \tag{1}$$

$$\text{and } \mathbb{E}[\|w_{T+1} - w^*\|^2] \leq \frac{4G^2}{m^2(T+1)}. \tag{2}$$

Corollary 16 (NoisyGD for Strongly Convex Objectives). *Let $J(w) = \mathcal{L}(w) + \frac{\lambda}{2}\|w\|^2$ with individual loss functions ℓ being L-Lipschitz on $\mathcal{W}$. Assume $\sup_{w \in \mathcal{W}} \|w\| \leq B$. Let the learning rate be $\eta_t = \frac{2}{\lambda(t+1)}$, then*

$$\mathbb{E}\left[J\left(\frac{2}{T(T+1)}\sum_{t=1}^{T} t w_t\right)\right] - J^* \leq \frac{2(NL + \lambda B)^2}{\lambda T} + \frac{2d\sigma^2}{\lambda T} \tag{3}$$

$$= \frac{2(NL + \lambda B)^2}{\lambda T} + \frac{dL^2}{\lambda \rho}, \tag{4}$$

where $\rho := \frac{TL^2}{2\sigma^2}$ is the privacy parameter of the algorithm ($\sqrt{2\rho}$-GDP).

Proof. First check that $NL + \lambda B$ upper bounds the Lipschitz constant of J on because the Lipschitz constant of $\frac{\lambda}{2}\|w\|^2$ is smaller than λB due to the bounded domain. Second, check that the noisy gradient oracle satisfies that it is unbiased, and the added noise has a variance of σ^2 per coordinate for all d coordinates. Thus

$$\mathbb{E}[\|g_t\|^2|w_t] = \mathbb{E}[\|\nabla J(w_t)\|^2|w_t] + \mathbb{E}[\|n_t\|^2|w_t] \leq (NL + \lambda B)^2 + d\sigma^2.$$

Thus by taking expectation on both sides we verify that we can take $G^2 = (NL + \lambda B)^2 + d\sigma^2$.

It remains to substitute these quantities and apply the first statement of Theorem 15. □

Corollary 17 (One-Pass SGD on public data). *Assume the public data with N samples are drawn from the same distribution of the private data. Assume that the (population risk)*

$$R(\theta) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(\theta^*, (x, y))]$$

is c-strongly convex at θ^* for some constant c. Then the one-pass SGD algorithm below

$$w_{t+1} = w_t - \frac{2}{c(t+1)} \nabla \ell(w_t, (x_t, y_t))$$

for $t = 1, ..., N_{pub}$ obeys that

$$\mathbb{E}[\|w_{N_{pub}+1} - w^*\|^2] \leq \frac{4L^2}{c^2 N_{pub}}$$

where $w^* = \text{argmin}_{w \in \mathcal{W}} \mathcal{R}(w)$.

Proof. First note that since the data is drawn iid, running one-pass SGD by going through the data points in a random order uses a *fresh sample* to update the parameters. This is is equivalent to optimizing the population risk directly. Check that for any fixed w and all $t = 1, ..., N_{\text{pub}}$ $\mathbb{E}_{(x_t,y_t)\sim\mathcal{D}}[\nabla_w \ell(w, (x_t, y_t))] = \nabla \mathcal{R}(w)$. Moreover, we need this stochastic gradient oracle to satisfy $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\|\nabla \ell(\theta, (x, y))\|^2] \leq G^2$. Notice that by our assumption $\|\nabla \ell(\theta, (x, y))\|^2 \leq L^2$, thus we can take $G = L$. By invoking the second statement of Theorem 15 the result follows. □

With these two corollaries stated, we are now ready to prove Theorem 3 and Theorem 18.

Proof of Theorem 3. The proof relies on Corollary 16, and the following argument. When additional regularization with parameter λ, the utility we consider should still be considered in terms of $\mathcal{L}(\hat{w}) - \mathcal{L}(w^*)$. Let w^* be any comparator satisfying $B > \|w^*\|$

$$\begin{aligned}
&\mathcal{L}(\bar{w}) - \mathcal{L}(w^*) = J(\bar{w}) - J_\lambda(w_\lambda^*) \\
&+ J_\lambda(w_\lambda^*) - J(w^*) + \frac{\lambda}{2}\|w^* - w_{\text{ref}}\|^2 - \frac{\lambda}{2}\|\hat{w} - w_{\text{ref}}\|^2 \\
\leq & J(\hat{w}) - J_\lambda(w_\lambda^*) + \frac{\lambda}{2}\|w^* - w_{\text{ref}}\|^2.
\end{aligned}$$

Take expectation on both sides and apply Corollary 16

$$\mathbb{E}[\mathcal{L}(\bar{w})] - \mathcal{L}(w^*) \leq \frac{2(NL + \lambda B)^2}{\lambda T} + \frac{dL^2}{\lambda \rho} + \frac{\lambda}{2}\|w^* - w_{\text{ref}}\|^2.$$

Finally, choosing $\lambda = \sqrt{\frac{\rho}{2\|w^* - w_{\text{ref}}\| dL^2}}$ yields

$$\mathbb{E}[\mathcal{L}(\hat{w})] - \mathcal{L}(w^*) \leq \frac{4(nL + \lambda B)^2}{\lambda T} + \frac{\sqrt{d}L\|w^* - w_{\text{ref}}\|}{\sqrt{2\rho}}$$

as claimed (dividing N on both sides to get $\hat{R}$). □

Theorem 18. *Assume the private data and public data are drawn i.i.d. from the same distribution $\mathcal{D}$ and that $R(w) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(w,(x,y))]$ is c-strongly convex in $\mathcal{W}$. Let $w_{\text{ref}} = w_{N_{pub}+1}$ — the last iterate of a single pass stochastic gradient descent on $\hat{\mathcal{R}}_{pub}(w)$ (initializing from $w_0 = 0$) that goes over the public dataset exactly once one data-point at a time with learning rate $\eta_t = \frac{2}{c(t+1)}$. Let w_{ref} be passed into Theorem 3's instantiation of NoisyGD, which returns $\bar{w}$ (The pseudo-code of this algorithm is summarized in Algorithm 3 in the appendix), then at the limit when T is sufficiently large, the* excess risk *obeys that*

$$\begin{aligned}
&\mathbb{E}[\mathcal{R}(\bar{w})] - \mathcal{R}(w^*) \\
\leq & \frac{4\sqrt{d}L^2}{c\sqrt{N_{pub}}N\sqrt{2\rho}} + \mathit{Gen}(\bar{w}, N) + \mathit{Gen}(w^*, N),
\end{aligned}$$

where $w^* = \operatorname{argmin}_{w\in\mathcal{W}} \mathcal{R}(w)$ and $\mathit{Gen}(w, N) := \left|\mathbb{E}[\mathcal{R}(w) - \hat{\mathcal{R}}(w)]\right|$ is the expected generalization gap of (a potentially data-dependent) w.

Proof of Theorem 18. Let $w^* = \arg\min_{w\in\mathcal{W}} \mathcal{R}(w)$. By Corollary 17, we have

$$\mathbb{E}[\|w_{\text{ref}} - w^*\|^2] \leq \frac{4L^2}{c^2 N_{\text{pub}}},$$

which implies (by Jensen's inequality) that $\mathbb{E}[\|w_{\text{ref}} - w^*\|] \leq \sqrt{\frac{4L^2}{c^2 N_{\text{pub}}}}$.

Now by plugging in the w^* in theorem, take expectation over the public dataset, and substitute the above bound, we get

$$\mathbb{E}[\mathbb{E}[\hat{\mathcal{R}}(\bar{w})] - \hat{\mathcal{R}}(w^*)] \leq \frac{4(NL + \lambda B)^2}{\lambda T N} + \frac{2\sqrt{d}L^2}{c\sqrt{N_{\text{pub}}}N\sqrt{2\rho}}.$$

Take T to be sufficiently large so that the second term dominates, we obtain the stated bound.

Finally, to convert the above bound into that of the excess risk:

$$\begin{aligned}
&\mathbb{E}[\mathbb{E}[\hat{\mathcal{R}}(\bar{w})] - \hat{\mathcal{R}}(w^*)] - (\mathbb{E}[\mathcal{R}(\bar{w})] - \mathcal{R}(w^*)) \\
\leq & |\mathbb{E}[\mathbb{E}[\hat{\mathcal{R}}(\bar{w})] - \mathcal{R}(\bar{w})]| + |\mathbb{E}[\hat{\mathcal{R}}(w^*)] - \mathcal{R}(w^*)| \\
:= & \text{Gen}(\bar{w}, N) + \text{Gen}(w^*, N),
\end{aligned}$$

which completes the proof. □

We make two additional remarks. First, we do not require the *empirical* objective $\mathcal{L}_{\text{pub}}$ to be strongly convex. In practice, we do not have strong convexity when $N_{\text{pub}} < d$. The assumption of c-strong convexity is on the *population-level* risk function $\mathcal{R}$. Second, our bound decomposes the excess (population) risk of the private learner into a (local) uniform-convergence bound (which is required by a non-private learner too) and an additional cost due to privacy. Note that $\text{Gen}(N)$ is usually $O(1/\sqrt{N})$ but could be $O(1/N)$ when certain data-dependent "fast rate" conditions are met, e.g., realizability, low-noise, or curvature (see, e.g., [27]). Our results suggest that the cost of privacy asymptotically vanishes (fix ρ, $N_{\text{pub}} \to \infty$, and $N_{\text{pub}}/N \to 0$) even under these fast rate conditions relative to the non-private rate.