Mixed Differential Privacy in Computer Vision

Aditya Golatkar[*,2] Alessandro Achille[1] Yu-Xiang Wang[1,3]
Aaron Roth[1,4] Michael Kearns[1,4] Stefano Soatto[1,2]
[1]AWS AI Labs [2]UCLA [3] UCSB [4] Penn
[*]adityagolatkar@ucla.edu aachille@amazon.com yuxiangw@cs.ucsb.edu
aaroth@cis.upenn.edu mkearns@cis.upenn.edu soattos@amazon.com

Abstract

We introduce AdaMix, an adaptive differentially private algorithm for training deep neural network classifiers using both private and public image data. While pre-training language models on large public datasets has enabled strong differential privacy (DP) guarantees with minor loss of accuracy, a similar practice yields punishing trade-offs in vision tasks. A few-shot or even zero-shot learning baseline that ignores private data can outperform fine-tuning on a large private dataset. AdaMix incorporates few-shot training, or cross-modal zero-shot learning, on public data prior to private fine-tuning, to improve the trade-off. AdaMix reduces the error increase from the non-private upper bound from the 167-311% of the baseline, on average across 6 datasets, to 68-92% depending on the desired privacy level selected by the user. AdaMix tackles the trade-off arising in visual classification, whereby the most privacy sensitive data, corresponding to isolated points in representation space, are also critical for high classification accuracy. In addition, AdaMix comes with strong theoretical privacy guarantees and convergence analysis.

1. Introduction

When training a deep neural network for visual classification, it is critical to protect privacy by limiting the amount of information that could be gleaned about any individual training sample. Differential Privacy (DP) [15] is a theoretical framework to provide strong guarantees on the maximum amount of information that any attacker can extract about an individual training sample. In particular, DP allows users to select the desired trade-off between privacy and accuracy, mediated by a *privacy parameter* ϵ, which typically depends on the application scenario.

Training large machine learning models while guaranteeing strong privacy for each sample is challenging. In practice, however, one often has a pool of data available for which there are no privacy concerns. This could be a synthetic datasets or datasets designed to be available for public use. Such *public* data are distinct from the *private* ones, for which we seek strong privacy guarantees.[1] In particular, using large amounts of generic public data for pre-training has recently enabled the creation of language models that achieve DP on the target task while remaining close to state-of-the-art performance [34, 57]. The same strategy in vision [49], however, still yields punishing error increases (Tab. 1) ranging from 311% to 167% on average across datasets, depending on the desired level of privacy.

Can we, then, make use of public data that is better suited for vision, so we can retain close-to-paragon performance while ensuring privacy?

One sure way of preserving privacy is not to use private data altogether, and recent literature has given us multiple ways to do so. For example, using zero-shot learning [19, 29, 30, 33, 46] one can leverage public data from a different modality (e.g., text) to train a visual model without ever seeing the private data. More generally, one can source or synthesize a few samples of labeled public data from the task distribution, and train using few-shot learning [54], still without using private data. Surprisingly, simple few-shot techniques outperform private training on much larger dataset (Fig. 2, left).

Of course, there may be subtle domain shifts between the public and private data, so ignoring the latter is not a desirable strategy to maintain privacy. The question, then, is how to use even small amounts of public data, in addition to the private data, to break the trade-off between privacy and accuracy. To do so, we change the setting from most work on DP to include *labeled* public data sourced with the *same* labels as the target task. We call this setting *mixed differential privacy*, or MixDP.

To address MixDP, we propose to use the public data not just for pre-training the backbone, but for few-shot or zero-

[*] work done during an Amazon internship

[1] Note that here public data is not the same as data from public sources, as the latter may still require privacy guarantees.

shot learning of a classifier on the target tasks, prior to private fine-tuning (Sec. 3.2). In Tab. 1, we show that indeed in the MixDP setting it is possible to achieve significant gains compared to training with only the private or only the public data, even using a small amount of the latter. To do so, we have to modify existing DP training algorithms to the mixed setting, which leads us to the development of AdaMix, a method for MixDP that uses public data to tune and adapt all major steps of private training, in particular model initialization, clipping of the gradients, and projection onto a lower-dimensional sub space. Some of these ideas to use auxiliary public data were applied in different contexts in isolation, but have suboptimal privacy/performance tradeoffs for visual tasks.

In visual classification tasks, the long-tails of the data play an important role in achieving high classification performance [17, 18]. However, DP is a worst-case framework which is highly-influenced by outliers or long-tails. MixDP eases the problem, since it allows to collect public data to ensure that each sub-population is sufficiently covered. This reduced the cost of privacy for the long tails, as we show using a per-instance DP (pDP) analysis of [55].

Unlike most related work, we use NoisyGD rather than DP-SGD as we find that it is faster and more private for our use case. We give a tight analysis of its (ϵ, δ)-DP properties using the Analytical Gaussian mechanism [5, 14] (Sec. 3.1). In addition, we present a convergence proof for our algorithm together with new stronger bound for the strongly convex case. This also allows us to describe the utility of the algorithm relative to its non-private counterpart.

To summarize, our contributions are: **(1)** we introduce AdaMix, a state-of-the-art adaptive method for mixed privacy learning; **(2)** we show significant improvement w.r.t. baselines on a novel benchmark for MixDP vision tasks; **(3)** we show that the zero-shot text information can improve the performance of private models (Sec. 4); **(4)** we analyze the goodness of our method both theoretically (new theorems) and empirically (pDP analysis).

2. Related Work

Most work on Differential Privacy [6,10,22,24,32,44,47, 53, 53, 57] uses public data either for: generic pre-training unrelated to the task [49], or to tune parameters [25, 58, 60], or as additional *unlabeled* data [37, 38]. Instead, we use a *small amount of labeled public data related to the task* to improve the accuracy of a private model under a given privacy parameter ϵ.

Unlabeled public data. PATE [37, 38, 50] trains teacher models on different partitions of the data to predict labels, and privately vote on how to label public (unlabelled) data. Then, a student model is trained in semi-supervised fashion on public data, now partially labelled using the private data. This strategy does not directly allow to train a model on a mix of labeled public and private data. [62] noted that the number of partitions required for meaningful privacy guarantees may be prohibitive for vision datasets. As an alternative, they suggest labeling with a differentially private KNN. In MixDP, where we assume that the public data is already labeled, these algorithms would perform at best similarly to our *Only-Public* baseline. Instead, what we are interested in effectively using labeled private data to improve performance relative to already labeled public data.

DP for Language models. [34, 57] show that a large language model pre-trained on generic public data can be fine-tuned on task-specific private data with only modest loss in accuracy. In contrast, our focus is on using small amounts of public data for fine-tuning. We note that in language models strong transfer of information from the pre-training is facilitated by the fact that the output space (a sequence of tokens) is the same for all tasks. Conversely, vision models need to relearn different output spaces based on the task (i.e., the last layer is always trained from scratch) reducing the transfer of information. We show that we can partly recover this advantage by either using an initialization based on public data *of the same task*, or a *multi-modal* vision-language model. We use CLIP [41] for zero-shot learning.

Adaptive clipping. Similarly to us, [52] studies DP training with access to a small amount of public data, and use adaptive gradient clipping [3, 4, 51, 59] based on the public gradients. Contrary to us, they first train a private model separately and then a mixed public-private model using the private one as a regularizer with compelling results on tabular data. We found this approach ineffective for computer vision, where usually models are pre-trained on public data. In Sec. 4 we show that AdaMix works better on vision tasks, and is further improved with multi-modal data.

Adaptive projection. [25, 58, 60] suggest improving DP-SGD by projecting the gradients into a lower-dimensional space estimated from public data. However, these methods do not leverage the performance improvements derived from training on public data, which are needed to make vision models viable. Using public data both to train and to estimate the subspace introduces more challenges, since the gradients of the public data and private data will then behave differently. Moreover, while they perform an SVD on the individual public sample gradients, we do the SVD of the total gradient matrix. Unlike the SVD of the matrix of individual sample gradients, this is cheaper to compute at each step.

Analysis. [2] studies the theoretical limits of mixed privacy learning, but does not provide practical algorithms. We provide both a method for MixDP and a proof of convergence. Our analysis leverages the results of [28] which shows that a particular non-uniform averaging scheme of SGD converges at an $O(1/t)$ rate, rather than the classical $O(\log t/t)$ rate. In the context of differentially private learning, [6] showed that NoisySGD achieves the optimal rates for (convex) empirical risk minimization (ERM) [11]. [48] provided the first analysis of NoisyGD in the general convex case. To the best of our knowledge, we are the first to formally establish that the use of an arbitrarily small public data gives provably smaller sample complexity in learning compared to the standard private learning setting.

3. Method

Notation. Let $z \in \mathcal{Z}$ be a datum, with $z = (x, y)$ being the feature/label pair; $\mathcal{Z}^* = \cup_{n=0}^{\infty} \mathcal{Z}^n$ is the universe of datasets with arbitrary size. For learning, we have access to a private dataset $D_{\text{pri}} := \{z_1, ..., z_{N_{\text{pri}}}\}$ and a public one $D_{\text{pub}} := \{\tilde{z}_1, ..., \tilde{z}_{N_{\text{pub}}}\}$. The loss function $\ell : \mathcal{Z} \times \mathcal{W} \to \mathbb{R}$ takes data $z \in \mathcal{Z}$ and "weights" $w \in \mathcal{W} \subset \mathbb{R}^d$ to yield the total loss on the private and public dataset $\mathcal{L}_{\text{pri}}(w) := \sum_{i=1}^{N_{\text{pri}}} \ell_i(w)$ and $\mathcal{L}_{\text{pub}}(w) := \sum_{j=1}^{N_{\text{pub}}} \tilde{\ell}_j(w)$, with ℓ_i and $\tilde{\ell}_j$ short-hands for $\ell(w, z_i)$ and $\ell(w, \tilde{z}_j)$ respectively. Notice that $\mathcal{L}_{\text{pri}}$ and $\mathcal{L}_{\text{pub}}$ are sums, not averages, of the loss functions. Their gradients are indicated by $g_i^{\text{pri}}(w) = \nabla_w \ell_i(w)$, $g_j^{\text{pub}}(w) = \nabla_w \tilde{\ell}_j(w)$, $G^{\text{pri}}(w) = \nabla_w \mathcal{L}_{\text{pri}}(w)$, $G^{\text{pub}}(w) = \nabla_w \mathcal{L}_{\text{pub}}(w)$. In addition, we say $N = N_{\text{pri}} + N_{\text{pub}}$ and $\mathcal{L}(w) = \mathcal{L}_{\text{pri}}(w) + \mathcal{L}_{\text{pub}}(w)$.

The average *empirical risk* is written as $\hat{\mathcal{R}}(w) := \mathcal{L}(w)/N$. When the data are drawn from some distribution $\mathcal{D}$, the risk, i.e., generalization error, $\mathcal{R}(w) := \mathbb{E}_{z\sim\mathcal{D}}[\ell(z, w)]$. Our goal is to design a differentially private algorithm that returns $\hat{w}$ which minimizes $\mathcal{L}(\hat{w})$. Its performance is measured by the *excess empirical risk*, i.e., $\hat{\mathcal{R}}(\hat{w}) - \min_{w\in\mathcal{W}} \hat{\mathcal{R}}(w)$; and by the *excess risk* $\mathcal{R}(\hat{w}) - \min_{w\in\mathcal{W}} \mathcal{R}(w)$. In the experiments, the latter is measured by the error on a test set.

3.1. Differential privacy

Two datasets D, $D' \in \mathcal{Z}^*$ are said to be *neighboring*, written as $D \simeq D'$, if we can obtain D' from D by *adding or removing* a single data point.

Definition 1 (Differential privacy [15, 16])**.** *For $\epsilon > 0$ and $\delta \geq 0$, a randomized algorithm $\mathcal{M}$ is (ϵ, δ)-differentially private if for any neighboring datasets $D \simeq D'$ and any measurable $S \subseteq \text{range}(\mathcal{M})$,*

$$\mathsf{Pr}[\mathcal{M}(D) \in S] \leq e^{\epsilon} \cdot \mathsf{Pr}[\mathcal{M}(D') \in S] + \delta.$$

In our context, $\mathcal{M}$ is the randomized learning algorithm, and $\mathcal{Y}$ is the space of model weights. This condition bounds the maximum amount of information that an attacker, having access to the weights of the model, can extract about one of the training samples. The parameter ϵ is referred to as the *privacy parameter*: lower values of ϵ carry better privacy guarantees, but also make it harder to train an accurate model, as it reduces the amount of information about the training samples that can be represented via the parameters of the model. The goal of a good DP training algorithm A is to train an accurate model while satisfying the (ϵ, δ)-privacy constraint selected by the user where, ideally, $\epsilon \approx 1$ and $\delta = o(1/n)$.

```
Algorithm 1: (Projected) Noisy Gradient Descent
 Data: Constraint set $\mathcal{W}$, initialization $w_1$, noise
       level $\sigma$, clipping threshold $\tau$, number of
       iteration $T$, weight decay parameters $\lambda$ and
       $w_{\text{ref}}$, learning rates $\eta_t$.
 Result: $w_1, ..., w_{T+1}$
 for $t = 1, \ldots, T$ do
     $G_t = \sum_{i=1}^{N_{\text{pri}}} \text{clip}_{\tau}(g_i(w_t))$;
     $n_t \sim N(0, \sigma^2\tau^2 I)$;
     $w_{t+1} \leftarrow w_t - \eta_t(G_t + n_t + \lambda(w_t - w_{\text{ref}}))$;
     $w_{t+1} \leftarrow \Pi_{\mathcal{W}}(w_{t+1})$;
 end
```

NoisyGD. NoisyGD is defined in Algorithm 1. We now show that NoisyGD is differentially private and that, in the convex case, it converges to the optimal solution.

Proposition 2 (NoisyGD is (ϵ, δ)-DP, informal)**.** *Algorithm 1 with parameter T, σ^2 such that $\rho := \frac{T^2}{2\sigma^2}$ satisfies the $(\epsilon, \delta(\epsilon))$-DP, where $\delta(\epsilon) := \Phi(\frac{\mu}{2} - \frac{\epsilon}{\mu}) - e^{\epsilon}\Phi(-\frac{\mu}{2} - \frac{\epsilon}{\mu})$, $\mu = \sqrt{2\rho}$ and Φ is the CDF of the normal distribution.*

For any prescribed privacy parameter (ϵ, δ), it suffices to choose this ρ parameter, thus from here onward we will use ρ as the privacy parameter of interest (note that, asymptotically, $\rho \asymp \min\{\epsilon, \epsilon^2/\log(1/\delta)\}$).

Mixed private learning. We define the problem of differentially private learning when we have access to an additional public dataset *mixed private learning*. We say the algorithm $\mathcal{M}$ (now taking two arguments D_{pri} and D_{pub}) satisfies (ϵ, δ)-DP if $\mathcal{M}(\cdot, D_{\text{pub}})$ satisfies (ϵ, δ)-DP (Def. 1) for every fixed D_{pub}. [2] shows that a small public dataset from the same distribution allows one to privately PAC-learn any classes with bounded VC-dimension, but with an inefficient algorithm. We focus on designing practical DP algorithm that can effectively use both public and private data under the setting of convex empirical risk minimization [11] and convex fine-tuning of deep models [1].

3.2. AdaMix

Model. Following [49], we use a linear logistic regression model trained on top of the last layer features of an ImageNet pre-trained network. This reduces the expressivity of the model compared to training a full network. However, the significant reduction in number of parameters makes it a better choice DP algorithms and recent deep networks allow for competitive performance on most tasks even when training only the last layer. We normalize the features before feeding them to the logistic model, which both improves the accuracy and bound the gradients at each step.

Initialization and regularization. The choice of initialization is normally of minor concern for logistic regression: since the problem is convex we will converge to a global optimum regardless of the initialization. L2 regularization (weight decay), when employed, is usually centering at 0 by default. However, centering the regularizer is critical in differential privacy, as we prove in the following result:

Theorem 3 (Convergence of NoisyGD on strongly convex problems)**.** *Assume $\mathcal{W}$ is a convex set satisfying $\sup_{w\in\mathcal{W}}\|w\| \leq B$. Let $w_1, ..., w_{T+1}$ be the parameter vectors returned by running Algorithm 1 on the following regularized loss $J(w) = \mathcal{L}(w) + \frac{\lambda}{2}\|w - w_{ref}\|^2$ with $w_1 = w_{ref} \in \mathcal{W}$, learning rate $\eta_t = \frac{2}{\lambda(t+1)}$ and $\lambda = \sqrt{\frac{\rho}{2\|w^*-w_{ref}\|dL^2}}$. Then, the ensemble solution:*

$$\bar{w} = \tfrac{2}{T(T+1)} \textstyle\sum_{t=1}^{T} t\, w_t$$

obeys the bound

$$\mathbb{E}[\hat{\mathcal{R}}(\bar{w})] - \hat{\mathcal{R}}(w^*) \leq \tfrac{4(N\tau+\lambda B)^2}{\lambda T N} + \tfrac{\sqrt{d}\tau\|w^*-w_{ref}\|}{\sqrt{2\rho}N}$$

for any $w^* \in \mathcal{W}$, where ρ is the privacy parameter of the algorithm.

Note that as $T \to +\infty$ the first term vanishes and the second term matches the information-theoretic limit for privately solving general convex ERM problems when $w_{\text{ref}} = 0$ [6]. However, this can improve if we choose w_{ref} according to the public data such that w_{ref} is closer to w^* than 0. This motivated us to propose choosing w_{ref} by training on the public data. Then, we proceed to fine-tune the model on the concatenation of private and public data, using NoisyGD while preserving (ϵ, δ)-DP.

Provable benefits of public data access. Next we provide a theoretical guarantee for the above approach under standard statistical learning assumptions.

Theorem 4 (Generalization with access to public data, informal)**.** *Assume that private 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_{ref} be obtained by training for one epoch on the public dataset and let $\bar{w}$ be as in Thm. 3, then at the limit when T is sufficiently large, the* excess risk *obeys*

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

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

To the best of our knowledge, Theorem 4 is the first demonstrated claim that access to even a very small public dataset can improve private learning in the convex ERM setting. We note that the generalization bound decomposes in a first term, which describes the generalization cost incurred by using DP, and a second part which is common to non-private learners. Interestingly, in the Appendix we show that in MixDP the cost of privacy asymptotically vanishes even if the percentage of public data becomes neglegible (i.e., when $N_{\text{pub}} \to \infty$, $N_{\text{pub}}/N \to 0$).

Besides providing a good initialization, access to a small public dataset also enables ways to further improve NoisyGD in practice, as we will now see.

Adaptive threshold. Since the norm of the gradients changes significantly during training, a fixed clipping threshold τ is suboptimal throughout training. [52] suggests to tune this threshold adaptively using public data. However, the norm of the average gradient of the public sample, may not yield good clipping thresholds for the *per-sample* gradients. Instead, we find the following adaptive threshold to be more robust:

$$\tau = \text{quantile}_{90}(\{\|g_i^{\text{pub}}(w)\|\}_{i=i}^{N_{\text{pub}}}),$$

that is, we take the 90-th quantile of the norm of the public gradients, where $g_i^{\text{pub}}(w) = \nabla_w \tilde{\ell}_i(w)$ is the gradient of the i-th public sample. Then the clipped gradient we use is

$$\tilde{g}_i^{\text{pri}}(w) = \frac{g_i^{\text{pri}}(w)}{\|g_i^{\text{pri}}(w)\|}\min(\|g_i^{\text{pri}}(w)\|, \tau).$$

This quantile rule is also used by [3], but we differ in the use of public data to determine τ, rather than allocating part of the privacy-budget for releasing the quantile. It was established in [12] that gradient clipping has the effect of Huberization of the loss function in the GLM cases. The convergence to the optimal solution of a modified objective function is guaranteed if we fix the clipping parameter. Our approach on the adaptive clipping has the effect of selecting the hyperparameter adaptively in a homotopy style algorithm, which shows further empirical benefits.

Adaptive subspace projection. From Thm. 4, the utility of a DP algorithm decreases with the number of parameters d

of the model. In view of this, we do not train the whole network but rather use it as a feature extractor and only train a linear model on top. However, for modern vision architectures, even the number of parameters of the final layer is in the order of tens of thousands. Therefore, we use an adaptive mechanism to project private gradients onto a low-dimensional subspace before adding noise. More precisely,

$$G^{\text{pub}}(w) = \nabla_w L_{D_{\text{pub}}}(w) = \sum_{i=1}^{N_{\text{pub}}} \nabla_w \ell(\tilde{z}_i, w)$$

be the total gradient of the public dataset D_{pub}. Since we are using a linear logistic model, $G^{\text{pub}}(w)$ is a $D \times C$ matrix where D is the number of features and C is the number of classes. Consider the SVD decomposition:

$$G^{\text{pub}}(w) = USV.$$

We now project each (clipped) private gradient matrix $\tilde{g}_i^{\text{pri}}$ on the top P components, and obtain a matrix $\hat{g}_i^{\text{pri}} = U^t \tilde{g}_i^{\text{pri}}$ of size $P \times C$. Other methods projecting the gradient using public data [25, 58, 60] do not compute the SVD of the total gradient matrix, as noted in Sec. 2. Rather, use the vectorized *per-sample* gradients of the public examples to compute the principal components to project. Since, in vision, the size of matrix of all per-sample gradients has a much larger size $N \times FC$ (rather than our $F \times C$), several approximations would be required to carry out this strategy.

Final algorithm. The final gradient update step at time t is:

$$w_{t+1} \leftarrow w_t - \eta_t(G^{\text{pub}} + U\hat{G}^{\text{pri}} + \lambda \cdot w_t)$$

where

$$\hat{G}^{\text{pri}} = \textstyle\sum_{i=1}^{N_{\text{pri}}} U^T \tilde{g}_i^{\text{pri}} + n_t.$$

and $n_t \sim \mathcal{N}(0, \sigma^2\tau^2 I_C)$. We then train for T steps. The pseudo-code of algorithm is given in Algorithm 2.

Algorithm 2: AdaMix training algorithm.

```
Data: Public dataset $D_{\text{pub}}$, private dataset $D_{\text{pri}}$,
      privacy parameter $(\epsilon, \delta)$, noise variance $\sigma$,
      learning rate $\eta$, $L_2$ reg. coefficient $\lambda$
Result: $w = A(S)$
$T_{\text{max}} \leftarrow \text{Calibrate}(\epsilon, \delta, \sigma)$;
$w \leftarrow \text{Pretrain}(S_u)$;
for $t = 1, \dots, T_{max}$ do
    // Adaptive clipping
    $\tau_t \leftarrow \text{quantile}_{90}\left(\{\|g_i^{\text{pub}}(w)\|\}_{i=1}^{N_{\text{pub}}}\right)$;
    // Adaptive projection
    $U, S, V \leftarrow \text{SVD}(G_{\text{public}}(w))$;
    // Clip, project and add noise
       to private gradients
    $n_t \sim N(0, \sigma^2\tau_t^2 I)$;
    $\tilde{g}_i^{\text{pri}}(w) \leftarrow \frac{g_i^{\text{pri}}(w)}{\|g_i^{\text{pri}}(w)\|} \min(\tau_t, \|g_i^{\text{pri}}(w)\|)$;
    $\tilde{G}^{\text{pri}}(w) = \sum_{i=1}^{N_{\text{pri}}} U^T \tilde{g}_i^{\text{pri}}(w) + n_t$;
    // Final weights update
    $w \leftarrow w - \eta(G^{\text{pub}}(w) + U\tilde{G}^{\text{pri}}(w) + \lambda \cdot w)$;
end
```

Proposition 5. *Algorithm 2 with parameter T, σ^2 such that $\rho := \frac{T^2}{2\sigma^2}$ satisfies the $(\epsilon, \delta(\epsilon))$-DP of a Gaussian mechanism in Theorem 8 [5] with parameter $\mu = \sqrt{2\rho}$.*

Note that the privacy guarantee is the same as long as the ratio $\sigma/\sqrt{T}$ remains constant, which gives a degree of freedom in choosing σ (hence T). Higher values of σ requires more training steps T, but, by Thm. 3, it also ensures better convergence (which we validate this empirically in Sec. 4). Hence, the user should generally pick the largest σ allowed by their computational budget.

3.3. Textual side information

CLIP [41] is an embedding trained so that the representation of an image is close to the representation of its textual description. This enables zero-shot learning, i.e., initializing an high-performance image classification model using only the name of the labels or a textual description of the classes, which is generally public. AdaMix can be easily adapted to work in this multi-modal MixDP setting, where the public data comes from a different modality. First, we initialize a linear logistic model using the names of the classes: Let c_i be the text associated to the i-th class, the weight matrix is [41]:

$$W = \beta[\text{enc}(c_1), \dots, \text{enc}(c_C)],$$

that is, each row is the normalized encoding of a label scaled by a parameter β (we use $\beta = 100$). Starting from this public initialization, we train with AdaMix on image data.

4. Results

Setting. Unless otherwise specified, we train a linear logistic model on top of the last layer features of a ResNet-50 [21] pre-trained on ImageNet [13]. For NoisyGD, we initalize the logistic weights to zero, fix $\sigma = 20$ and $\delta = 10^{-5}$. We train with $\lambda = 1e - 2$, weight decay centered at zero, learning rate $\eta \in \{0.0005, 0.001, 0.0025, 0.005\}$, and select the best learning rate for each value of ϵ. For AdaMix, we initialize the weights by training on the public data until convergence. We use the Gaussian mechanism implementation of `autodp`[2] to calibrate the parameters of the training algorithm, in particular $T_{\max}$, in order to achieve the desired (ϵ, δ)-DP requirement.

[2] https://github.com/yuxiangw/autodp

	Public Shots	Non-Private	Only-Public	Fully-Private	PPGD [52]	AdaMix	Fully-Private	PPGD [52]	AdaMix
		(paragon)	(baseline)		$\epsilon = 1$			$\epsilon = 3$	
Ox. Flowers [36]	2	12.34	40.16	95.12	40.99	**39.48**	81.42	39.22	**36.11**
CUB-200 [56]	2	31.97	64.01	96.79	64.20	**63.58**	81.61	61.59	**59.15**
Stanf. Dogs [26]	2	10.08	16.58	33.87	16.42	**15.38**	15.93	15.61	**12.72**
MIT-67 [40]	5	25.17	43.58	69.55	42.02	**40.46**	42.99	39.60	**33.03**
Oxford Pets [39]	5	7.17	12.83	26.02	12.85	**11.54**	12.37	11.15	**9.53**
Caltech-256 [20]	5	14.64	24.88	60.72	24.61	**23.74**	27.15	23.71	**20.85**

Table 1. **Mixed privacy, full privacy and AdaMix.** We report the result (test errors) of different methods on a diverse set of vision tasks (see text for description). Surprisingly **Only-Public**, which throws away private data and only trains a model on the small amount of public data outperforms **Fully-Private**, which trains on all data as if it was private. **AdaMix** is instead able to effectively use both the private and public data at the same time, and achieves a significant improvement even at relatively small value of ϵ (e.g., 10% improvement on MIT-67 w.r.t. Only-Public). Instead, the closest method to us, **PPGD** does not significantly improve over the Only-Public baseline.

Datasets. We test our algorithm on the following vision classification datasets commonly used in transfer learning (see Tab. 1). We split each training dataset into a "public" dataset consisting of N-samples per class (see Tab. 1), and consider the remaining training data as private. In all cases, the public data is less than 10% of the private data.

Baselines considered. We compare AdaMix with the following baselines: **(Fully-Private)** Ignore the MixDP setting and simply train on all data as if it was private, starting form a zero initialization and using NoisyGD; **(Only-Public)** Train a standard non-private model using only the few public samples and discarding the private ones; **(PPGD)** A version of PPSGD [52], but trained with NoisyGD rather than DP-SGD for easier comparison. Unlike AdaMix, PPGD has several hyper-parameters: we manually searched for the best combination to ensure the best comparison; **(NGD)** An ablated version of AdaMix which uses fixed hyper-parameters rather than the adaptive components.

4.1. Experiments

Mixed Privacy vs Full Privacy. In Table 1 we see that, perhaps surprisingly, Only-Public is always significantly better than Fully-Private even if it actually throws away most of the available data. This suggests that, in vision, collecting a small amount of public data may be preferred to collecting a much larger amount of private data (Fig. 2). However, we see that AdaMix is able to effectively use both the private and public data at the same time, and achieves a significant improvement even at relatively small value of ϵ (e.g., 10% improvement on MIT-67 w.r.t. Only-Public at $\epsilon = 3$). Conversely, the method closest to us, PPGD, does not significantly improve over the Only-Public baseline even after our changes to adapt it better to vision tasks.

Ablation study and finer comparison. In Fig. 1 (left and center) we plot the average accuracy obtained by various methods and ablations of AdaMix for different privacy parameters ϵ. Both AdaMix and the ablated NGD are able to improve over the Only-Public baseline. Interestingly, this holds true even when considering relatively low privacy parameters $\epsilon \approx 1$ which is usually considered too restrictive to train in a fully private setting (as seen in Tab. 1). On the other hand, thanks to the MixDP training we can obtain high-accuracy models even at restrictive privacy settings. We note that AdaMix performs uniformly better than other methods at all privacy regimes. From the ablation of AdaMix, we see that initializing the model with the public data has the largest impact, followed by adaptive clipping and adaptive projection.

Membership attacks. Differential privacy is a very strong requirement, and is it often assumed that using higher value of ϵ may still provide valuable defense to attacks [7–9, 31, 35, 42, 43, 45, 61] even if the theoretical guarantees are weak. Conversely, [23] suggests using attacks to give a lower-bound on the privacy parameter of the algorithm. In Fig. 1 (right) we plot the Area Under Curve of a membership attack executed against models for different values of ϵ. We use a thresholding based membership attack as proposed in [43]. The loss for samples in the private training set are labelled as 0 and test samples are labelled as 1. Then we simply compute the AUC for this. We find that the sorting of the algorithms does not change: algorithms with better theoretical bounds are also more robust to attacks.

Performance Boost and size of public data. We hypothesize that, when enough public data is present, the contribution of using private data may be less noticeable. To test this, we define the Performance Boost of a Mixed Privacy method as $\mathrm{PB} = (\mathrm{err}_{\mathrm{public}} - \mathrm{err}_{\mathrm{paragon}})/(\mathrm{err}_{\mathrm{DP}} - \mathrm{err}_{\mathrm{paragon}})$, where $\mathrm{err}_{\mathrm{paragon}}$ is the error obtained by training on non-private model on both public and private data, $\mathrm{err}_{\mathrm{public}}$ is the error obtained by training only on the public data, and $\mathrm{err}_{\mathrm{DP}}$ is the error obtained by using training a mixed privacy model. This measures how well we can learn from the private data in the MixDP setting compared to the non-private upper-bound. In Fig. 2 we plot the performance boost for different sizes of the public set. We observe that the performance boost derived of using mixed privacy is very high

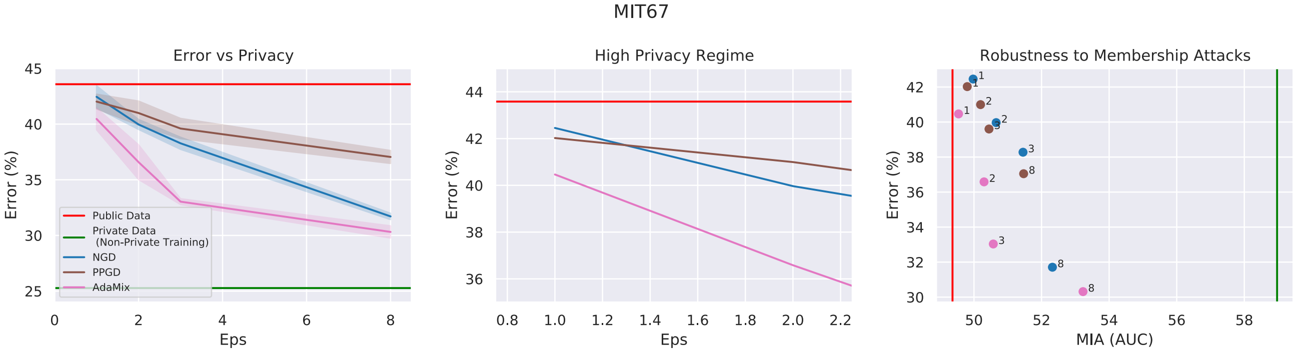

Figure 1. **Test error vs privacy for several methods. (Left)** On MIT-67, we show the test error obtained by different methods at different privacy parameters ϵ. The top horizontal red line shows the perfectly private Only-Public baseline (not using the private data at all). The bottom green line shows the non-private paragon (training on all data as if they where public). By changing the privacy parameter ϵ, we can interpolate between the two extrema. We see that AdaMix performs uniformly better than other methods, and can use private and public data together to obtain significant boost at all ϵ. **(Center)** Same as the before, but we zoom in to show the behavior of the different solutions in the low-ϵ regime. **(Right)** For the same algorithms, we plot the robustness to adversarial attacks (measured by AUC) vs test error obtained using different values of ϵ (annotated near each point). AdaMix obtains the best trade-off between accuracy and robustness.

when the public data is scarce, disappears when the amount of public data is comparable or larger than the private data.

Multi-modal initialization. In Fig. 4 we compare CLIP models initialized with (1) a random initialization, (2) a zero-shot initialization based on the public label names (Sec. 3.3), and (3) using few public image samples. While the latter performs the best at a given privacy level ϵ, the zero-shot initialization still improves significantly over the random initialization, thus showing that even just small amount of textual information (which is often available) can be used to significantly boost accuracy. Multi-modal models may also learn a better lower-dimensional representation of the data that is more amenable to DP. In Fig. 4 we compare AdaMix using an ImageNet pretrained ResNet-50 and using CLIP. This is not a direct comparison, but we see a significant performance boost by using CLIP at a given privacy level ϵ, suggesting that the move to multi-modal model may be a logical next step for DP.

Effect of differential privacy on individual samples. DP is a worst-case guarantee which may penalize the accuracy of a model on unusual data, or data on the long-tails of the distribution, which however may play an important role for generalization in vision [17,18]. MixDP may ease the problem, since it allows to collect public data to ensure that each sub-population is sufficiently covered, lessening the accuracy cost of making that data private. We show this effect using the per-instance DP (pDP) analysis of [55], which measures the privacy loss ϵ' incurred by each individual z during the private training. This allows us to provide a finer analysis of different methods by comparing the histogram of pDP loss ϵ for each individual in the dataset D on top of the worst case DP bounds, which we do in Fig. 3 (see Appendix for details and theoretical analysis). We observe that, when using AdaMix, the pDP losses of most samples are the same, and is close to the maximum privacy budget $\epsilon = 3$, indicating that we are using the prescribed privacy budget more effectively and more uniformly across samples, thereby improving utility.

Domain shift. Often, the private and public data may come from slightly different domains. To test the robustness of AdaMix to domain shifts, we use the DomainNet dataset, which aims to classify objects in different domains. In particular, we train a model to classify objects in painting, us-

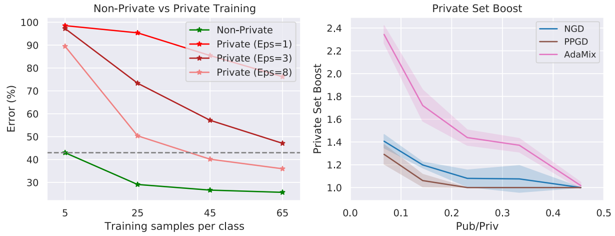

Figure 2. **(Left) Trade-off between public and private examples.** We show the accuracy reached on MIT-67 using N samples per class (x-axis) when they are public (green) or private (in red for different values of ϵ). Fully private training requires as much as 10 times more samples. This leaves users with a trade-off between collecting more private data or, if possible, a smaller amount of public data. **(Right) Performance boost using a private set for different methods.** We plot the Performance Boost (PB) for different DP methods as the ratio of public to private samples changes (see text for definition). We see that AdaMix achieves the best performance boost. As expected, when the public set is relatively large, the boost from adding private data decreases.

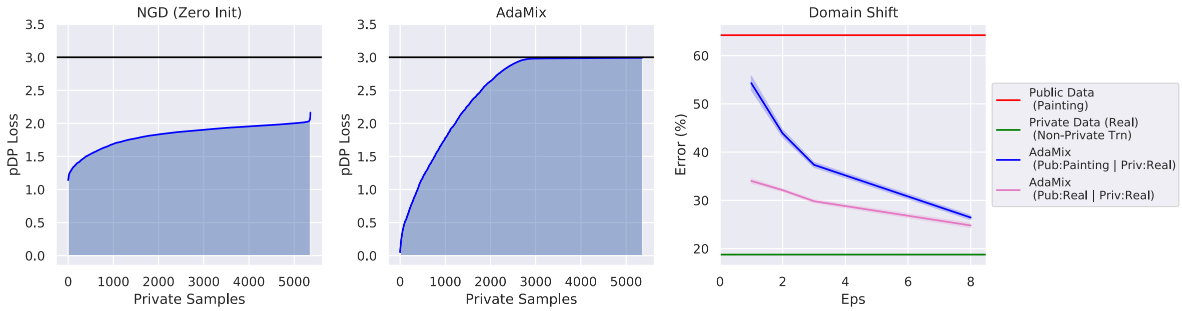

Figure 3. **(Left and center) Effect of public data on per-instance DP.** For $\epsilon = 3$, we plot the sorted pDP scores for each sample in the dataset with Fully-Private training **(left)** and with AdaMix **(center)**. AdaMix makes the the target ϵ (horizontal line) closer to pDP value used by most samples (blue curve). **(Right) MixDP and domain-shifts.** We plot the performance of AdaMix when (purple curve) the public data come from the same domain as the test (real images), and (blue curve) from different domains (public images are paintings). We see that AdaMix use the private data from the target domain (real images) to adapt the solution to the target task. This significantly improves over using only the public data from the wrong domain (red line) and for large ϵ approaches the non-private limit (green line).

ing however public data coming from real pictures (while the private data is in the painting domain). In Fig. 3 (right), we show that, even if the public comes from a different domain, AdaMix is still able to use the public data to significantly improve performance of private training.

Effect of σ. Outside of the learning rate η, the main other sensitive hyper-parameter of AdaMix is the noise variance σ. From Prop. 3 we expect that the best accuracy will be obtained using a large σ and training for a respectively larger amount of epochs T. In the Appendix, we plot the test error as a function of σ and we see that indeed, a larger σ give better results at the expense of longer training time.

5. Discussion

We study Mixed Differential Privacy learning in computer vision, and show that in this setting differential privacy can be used to provide high accuracy models while respecting a given privacy parameter. To do this, we introduce AdaMix, an adaptive differential privacy mechanism that uses the public data to initialize and compute, at each step, an optimal subspace and clipping threshold. We also show that multi-modal vision and text can significantly improve accuracy under a privacy constrain.

Limitations. In our analysis we train a linear model on pre-trained features, rather than fine-tuning the whole network. This is common in DP, since fine-tuning multiple layers may not significantly improve the accuracy while the additional parameters negatively affects the privacy bounds. This limitation is partially offset by new models that are trained to have generic and transferable last-layer features without the need of additional fine-tuning. We discuss several theoretical bounds that guide us in understanding the behavior of the algorithm. However, these bounds refer to a slighlty simplified version of AdaMix (see Appendix for details) and use hyper-parameters that are asymptotically optimal, but may not be practical.

Conclusions. Differentially Private models in computer vision often cannot reach the same level of performance as non-private model for realistic privacy parameters. Using AdaMix in the MixDP learning setting, we have shown that, assuming the presence of a small amount of public data, accurate networks can be produced that reach better accuracies than fully private training without compromising the privacy of the data. We hope that further research in this field may help the adoption of private models in more areas of computer vision.

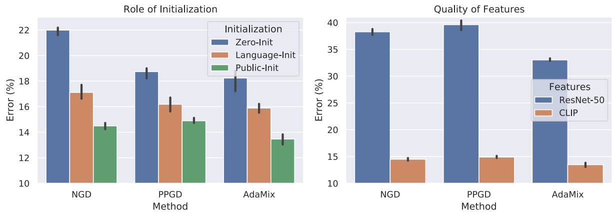

Figure 4. **(Left) Multi-modal initialization is better than zero initialization.** We compare the test accuracy on MIT-67 of a CLIP model trained with $\epsilon = 3$ using different initialization strategies. We observe that initializing the logistic weights to zero leads to the worst results under a given privacy parameter. Using the public label names to initialize the weights (language initialization) performs significantly better. However, using the few available public images to initialize the weights still outperforms the other choices. **(Right) Multi-modal models.** We compare the performance of ResNet-50 model, and a CLIP model. We see that CLIP performs significantly better under a given privacy parameter. While the ResNet-50 model and the CLIP model are not directly comparable due to different pre-training, this further reinforces that stronger pre-training can indeed benefit the privacy setting and that the feature space learned by multi-modal models may be better suited for privacy than standard vision networks.

References

[1] Alessandro Achille, Aditya Golatkar, Avinash Ravichandran, Marzia Polito, and Stefano Soatto. Lqf: Linear quadratic fine-tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15729–15739, 2021. 3

[2] Noga Alon, Raef Bassily, and Shay Moran. Limits of private learning with access to public data. In *Advances in Neural Information Processing Systems (NeurIPS'19)*, pages 10342–10352, 2019. 2, 3

[3] Galen Andrew, Om Thakkar, H Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. In *Advances in Neural Information Processing Systems (NeurIPS'21)*, 2021. 2, 4

[4] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32:15479–15488, 2019. 2

[5] Borja Balle and Yu-Xiang Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*, pages 394–403. PMLR, 2018. 2, 5, 15

[6] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014. 2, 3, 4

[7] Daniel Bernau, Philip-William Grassal, Jonas Robl, and Florian Kerschbaum. Assessing differentially private deep learning with membership inference. *arXiv preprint arXiv:1912.11328*, 2019. 6

[8] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, 2019. 6

[9] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2021. 6

[10] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In *NIPS*, volume 8, pages 289–296. Citeseer, 2008. 2

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

[12] Xiangyi Chen, Steven Z Wu, and Mingyi Hong. Understanding gradient clipping in private sgd: a geometric perspective. *Advances in Neural Information Processing Systems*, 33, 2020. 4

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 5

[14] Jinshuo Dong, Aaron Roth, and Weijie Su. Gaussian differential privacy. *Journal of the Royal Statistical Society*, 2021. 2, 15

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006. 1, 3

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

[17] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020. 2, 7

[18] Vitaly Feldman and Chiyuan Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2881–2891. Curran Associates, Inc., 2020. 2, 7

[19] Andrea Frome, Greg Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. 2013. 1

[20] G. Griffin, AD. Holub, and Pietro Perona. The caltech 256. 6

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[22] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019. 2

[23] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private sgd? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22205–22216. Curran Associates, Inc., 2020. 6

[24] Bargav Jayaraman and Lingxiao Wang. Distributed learning without distress: Privacy-preserving empirical risk minimization. *Advances in Neural Information Processing Systems*, 2018. 2

[25] Peter Kairouz, Mónica Ribero, Keith Rush, and Abhradeep Thakurta. Fast dimension independent private adagrad on publicly estimated subspaces. *arXiv preprint arXiv:2008.06570*, 2020. 2, 5

[26] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011. 6

[27] Tomer Koren and Kfir Y Levy. Fast rates for exp-concave empirical risk minimization. In *Neural Information Processing Systems*, pages 1477–1485, 2015. 19

[28] Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012. 2, 16, 17

[29] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009. 1

[30] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008. 1

[31] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 656–672. IEEE, 2019. 6

[32] Jaewoo Lee and Daniel Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1656–1665, 2018. 2

[33] Jimmy Lei Ba, Kevin Swersky, Sanja Fidler, et al. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4247–4255, 2015. 1

[34] Xuechen Li, Florian Tramèr, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021. 1, 2

[35] Yuzhe Ma, Xiaojin Zhu, and Justin Hsu. Data poisoning against differentially-private learners: Attacks and defenses. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4732–4738. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 6

[36] Maria-Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006. 6

[37] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *Proceedings of the International Conference on Learning Representations*, 2017. 2

[38] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Ulfar Erlingsson. Scalable private learning with PATE. In *International Conference on Learning Representations*, 2018. 2

[39] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6

[40] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420. IEEE, 2009. 6

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2, 5

[42] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. Membership inference attack against differentially private deep learning model. *Trans. Data Priv.*, 11(1):61–79, 2018. 6

[43] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019. 6

[44] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015. 2

[45] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017. 6

[46] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. 1

[47] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013. 2

[48] Shuang Song, Thomas Steinke, Om Thakkar, and Abhradeep Thakurta. Evading the curse of dimensionality in unconstrained private glms. In *International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2638–2646. PMLR, 13–15 Apr 2021. 3, 16

[49] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). *ICLR*, 2021. 1, 2, 3

[50] Archit Uniyal, Rakshit Naidu, Sasikanth Kotti, Sahib Singh, Patrik Joslin Kenfack, Fatemehsadat Mireshghallah, and Andrew Trask. Dp-sgd vs pate: Which has less disparate impact on model accuracy? *arXiv preprint arXiv:2106.12576*, 2021. 2

[51] Koen Lennart van der Veen, Ruben Seggers, Peter Bloem, and Giorgio Patrini. Three tools for practical differential privacy. *PPML18: Privacy Preserving Machine Learning, NIPS 2018 Workshop*, 2018. 2

[52] Jun Wang and Zhi-Hua Zhou. Differentially private learning with small public data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6219–6226, 2020. 2, 4, 6

[53] Lingxiao Wang and Quanquan Gu. Differentially private iterative gradient hard thresholding for sparse learning. In *28th International Joint Conference on Artificial Intelligence*, 2019. 2

[54] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020. 1

[55] Yu-Xiang Wang. Per-instance differential privacy. *Journal of Privacy and Confidentiality*, 9(1), 2019. 2, 7, 15

[56] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 6

[57] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021. 1, 2

[58] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations*, 2021. 2, 5

[59] Xinwei Zhang, Xiangyi Chen, Mingyi Hong, Zhiwei Steven Wu, and Jinfeng Yi. Understanding clipping for federated learning: Convergence and client-level differential privacy. *arXiv preprint arXiv:2106.13673*, 2021. 2

[60] Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private {sgd} with gradient subspace identification. In *International Conference on Learning Representations*, 2021. 2, 5

[61] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated learning*, pages 17–31. Springer, 2020. 6

[62] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020. 2