Differentially Private Prototypes for Imbalanced Transfer Learning

Dariush Wahdany[1,3*], **Matthew Jagielski**[2], **Adam Dziedzic**[3], **Franziska Boenisch**[3]

[1]Fraunhofer AISEC,
[2]Google DeepMind,
[3]CISPA Helmholtz Center for Information Security
dariush.wahdany@aisec.fraunhofer.de, jagielski@google.com, adam.dziedzic@cispa.de, boenisch@cispa.de

Abstract

Machine learning (ML) models have been shown to leak private information from their training datasets. Differential Privacy (DP), typically implemented through the differential private stochastic gradient descent algorithm (DP-SGD), has become the standard solution to bound leakage from the models. Despite recent improvements, DP-SGD-based approaches for private learning still usually struggle in the high privacy ($\varepsilon \leq 1$) and low data regimes, and when the private training datasets are imbalanced. To overcome these limitations, we propose Differentially Private Prototype Learning (DPPL) as a new paradigm for private transfer learning. DPPL leverages publicly pre-trained encoders to extract features from private data and generates DP prototypes that represent each private class in the embedding space and can be publicly released for inference. Since our DP prototypes can be obtained from only a few private training data points and without iterative noise addition, they offer high-utility predictions and strong privacy guarantees even under the notion of *pure DP*. We additionally show that privacy-utility trade-offs can be further improved when leveraging the public data beyond pre-training of the encoder: in particular, we can privately sample our DP prototypes from the publicly available data points used to train the encoder. Our experimental evaluation with four state-of-the-art encoders, four vision datasets, and under different data and imbalancedness regimes demonstrate DPPL's high performance under strong privacy guarantees in challenging private learning setups.

1 Introduction

Machine learning (ML) models are known to leak private information about their training datasets (Carlini et al. 2022; Fredrikson, Jha, and Ristenpart 2015; Shokri et al. 2017). As a solution to provably upper-bound privacy leakage, differential privacy (DP) (Dwork et al. 2006) has emerged as the de-facto standard for private training. It is usually implemented in ML through the differential private stochastic gradient descent (DP-SGD) algorithm which bounds the contribution of each data point during training and iteratively injects controlled amounts of noise (Abadi et al. 2016). Thereby, DP-SGD has been shown to increase training time and decrease the final model's utility.

While, over the last years, there has been significant progress in improving both computational efficiency (Bu et al. 2021; He et al. 2022a; Li et al. 2021; Lee and Kifer 2021; Subramani, Vadivelu, and Kamath 2021) and privacy-utility trade-offs (Bu, Mao, and Xu 2022; De et al. 2022), there are a few relevant setups where DP training still yields unfavorable results. These include the *high privacy regime* (expressed in DP with small values of the privacy parameter ε, such as $\varepsilon \leq 1$), the low data regime, *i.e.,* when only a *few private data points* are available for training, and when the training dataset is *imbalanced*, *i.e.,* when some classes have significantly more data points than others (Buda, Maki, and Mazurowski 2018; Liu et al. 2019; Reed 2001).

There are various reasons why DP training is challenging in these setups (Feldman 2020; Esipova et al. 2023). One of these is that DP protects small sets of examples due to its "group privacy" property, providing a provable bound on how much a DP algorithm can learn from small data (Feldman 2020). Beyond this concern, the iterative noise addition weakens the signal from the training data, especially when only a few training data points are available. Moreover, standard approaches for learning in imbalanced setups, such as changing the sampling (Domingos 1999; Kubat, Matwin et al. 1997; Japkowicz 2000; Lewis and Catlett 1994; Ling and Li 1998; Zada, Benou, and Irani 2022), generating synthetic data for the minority classes (Chawla et al. 2002), or weighing the training loss (Cao et al. 2019) are not directly compatible with DP or incur additional privacy costs. In a similar vein, each training iteration with DP training incurs additional privacy costs (Abadi et al. 2016), making it hard to keep ε low, *i.e.,* to stay in the high privacy regime.

To address all of these challenges, we propose *Differential Private Prototype Learning* (`DPPL`), a novel approach for private learning that combines prototypical networks (Snell, Swersky, and Zemel 2017), a standard algorithm for non-private few shot learning, with recent advances in training high-performance private models with DP that leverage powerful encoder models pre-trained on public data (Caron et al. 2021; He et al. 2022b; Radford et al. 2021) combined with private transfer learning (Li et al. 2021; Yu et al. 2021; Gu, Kamath, and Wu 2022; Tramèr, Kamath, and Carlini 2022; Ganesh et al. 2023; Hu et al. 2021; Houlsby et al. 2019; Li et al. 2023; Mehta et al. 2023). The main idea of our `DPPL` is to use the encoder as a feature extractor for the

[*]Part of the work was conducted as visiting researcher at CISPA

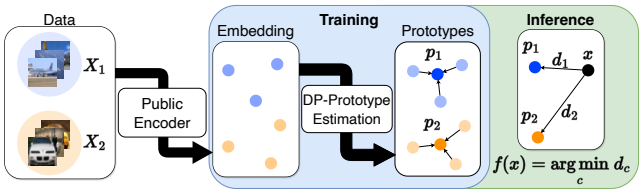


Figure 1: **Overview of DPPL.** We split the private data $\mathbf{X}$ per class c into $\mathbf{X}_c$'s, infer them through a publicly pre-trained encoder, and estimate per-class prototypes $\mathbf{p}_c$ in the embedding space with DP. Classification of samples is performed by returning the label of the closest prototype $\mathbf{p}_c$ in the embedding space according to some distance function d.

private data and to generate DP prototypes in the embedding space for each private class. To classify new data points, we then simply have to infer these points through the encoder and to return the label of the closest prototype.

Relying on DP prototypes for private learning offers significant advantages over iterative private training or fine-tuning. First, our prototypes do not require iterative noise addition. This enables to obtain less noisy predictions at lower privacy costs and improves privacy-utility trade-offs in the high privacy regime. Second, the prototypes are inherently balanced, *i.e.,* it is possible to obtain good prototypes also at the low data regime or for underrepresented classes from imbalanced private training datasets. Third, DP prototypes are fast to obtain, enable fast inference, and, due to the DP post-processing guarantees—which express that no query to them will incur additional privacy costs—can be publicly released for performing predictions.

We propose multiple algorithms for obtaining DP prototypes, and find that prior approaches for training models with DP do not yet leverage the full capacity of the public data (Li et al. 2021; Yu et al. 2021; Mehta et al. 2023): these prior approaches use the public data only for pre-training the encoder. Yet, we make the observation that we can leverage the public data additionally during the transfer learning step. By privately selecting per-class private prototypes from the public data, we can significantly decrease the privacy costs of our prototypes (even under the strong notion of *pure DP*, *i.e.,* ε-DP) and further improve privacy-utility trade-offs.

By performing thorough experimentation with four state-of-the-art encoders and four standard vision datasets, we show that DPPL provides strong utility in the high privacy regimes. Additionally, we highlight that DPPL is able to provide good privacy-utility trade-offs when only a few private training data points are available and that it yields state-of-the-art performance on imbalanced classification tasks. Thereby, DPPL represents a new powerful learning paradigm for private training with DP.

In summary, we make the following contributions:

- We propose DPPL, a novel alternative to private fine-tuning that combines recent advances in DP transfer learning with private few shot learning and can even yield pure DP guarantees.
- We perform extensive empirical evaluation which highlights that DPPL yields strong privacy-utility trade-offs, in particular in the high privacy regime and for imbalanced data.
- To further improve DP transfer learning, we show that we can leverage the public data beyond the pre-training step of the feature encoder by privately selecting public prototypes from it.

2 Background

Transfer Learning. We consider transfer learning where a publicly available pre-trained encoder $\hat{E}$ is used to extract features $\hat{\mathbf{X}}$ from a (private) dataset $D = (\mathbf{X}, \mathbf{y})$. Those features are then used to perform downstream classification by learning a function f maximizing $\Pr_{\mathbf{x},y\in D}[f(\hat{E}(\mathbf{x})) = y]$.

Differential Privacy. Differential privacy (DP) (Dwork et al. 2006) is a mathematical framework that provides privacy guarantees in ML by formalizing the intuition that a learning algorithm $\mathcal{A} : I \to S$, executed on two neighboring datasets D, D' that differ in only one data point,*i.e.,*, $D = D' \cup \{x\}$ (*add/remove DP*), will yield roughly the same output, *i.e.,* $\Pr[\mathcal{A}(D) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(D') \in S] + \delta$ (*approximate DP*). In this inequality, ε is the privacy budget that specifies by how much the output is allowed to differ and δ is the probability of differing more. If $\delta = 0$, we refer to it as *pure DP*, a strictly stronger notion of privacy. We will also refer to zero-concentrated DP (zCDP) (Bun and Steinke 2016), which requires that $D_\alpha(\mathcal{A}(D)||\mathcal{A}(D')) \leq \xi + \rho\alpha \ \forall_{\alpha\in(1,\infty)}$, where D_α is the Rényi divergence of order α. zCDP is a relaxation of pure DP, but stricter than approximate DP. $(0, \rho)$-zCDP can also be expressed simply as ρ-zCDP. We provide more details on DP in Appendix A.1. The standard approach for learning ML models with DP guarantees is differentially private stochastic gradient descent (DPSGD) (Song, Chaudhuri, and Sarwate 2013; Abadi et al. 2016). DPSGD clips model gradients to a given norm to limit the impact of individual data points on the model updates and adds a controlled amount of Gaussian noise to implement formal privacy guarantees during training.

Exponential Mechanism. The exponential mechanism (McSherry and Talwar 2007) offers a way to implement pure DP guarantees. Given a set of possible outputs $\mathbf{X}'$, it samples an output $\mathbf{x}'$ according to some utility function u with probability $\Pr[\text{EM}_u(\mathbf{X}) = \hat{\mathbf{x}}] \propto \exp\left(\frac{\epsilon}{\Delta_u} u(\mathbf{X}, \hat{\mathbf{x}})\right)$. This algorithm satisfies 2ϵ-DP. Appendix A.3 shows more details on the exponential mechanism and utility function.

Prototypical Networks. Prototypical networks (Snell, Swersky, and Zemel 2017) are used for few-shot classification, *i.e.,* they provide a way on adapting a classifier to new unseen classes with access only to a small number of data points from each new class. Their main components are a set of prototypes $\mathbf{p}_c \in \mathbb{R}^M$ and an embedding function $f_\phi : \mathbb{R}^D \to \mathcal{R}^M$. Each prototype for a class c is the mean of the embedded points belonging to that class, *i.e.,* $\mathbf{p}_c = \frac{1}{|\mathbf{X}_c|} \sum_{\mathbf{x}\in\mathbf{X}_c} f_\phi(\mathbf{x})$. Given a distance function $d : \mathbb{R}^M \times \mathbb{R}^M \to [0, \infty)$, the model classifies a point $\mathbf{x}$ based on its nearest prototype in the embedded space as $\hat{y}(\mathbf{x}) = \arg\min_c d(f_\phi(\mathbf{x}), \mathbf{p}_c)$.

DP Mean Estimation. Obtaining differentially private means $\mu = 1/n \sum_n \mathbf{x}$ for $\mathbf{x} \in \mathbb{R}^d$ is challenging in high dimensions. A straightforward approach (Kamath and Ullman 2020) consists of clipping all samples to some ℓ_2 norm, adding noise scaled according to the clip norm and then reporting the noisy mean of the clipped samples. *FriendlyCore* (Tsfadia et al. 2022) is a framework for pre-processing the input data of private algorithms, such that the algorithms being executed on this pre-processed data need to be private only for relaxed conditions. It improves especially for the cases where the samples have a high ℓ_2 norm and high dimensionality d. The *CoinPress* algorithm (Biswas et al. 2020) estimates the mean iteratively, clipping the samples not w.r.t. the origin but to the estimated mean of the previous step. This approach is especially useful when the mean is far away from the origin and generally considered state-of-the-art for dimensionalities in the low thousands. Figure 11 shows that the straightforward approach outperforms all other methods given strong priors on the ℓ_2 norms of the samples. We provide more details in Appendix A.4.

3 Related Work

Private Transfer Learning. Standard approaches for DP transfer learning rely on the DPSGD algorithm to train a classifier on top of the representations output by a pre-trained encoder, and potentially also to privately update existing or added model parameters on the sensitive data (Yu et al. 2021; De et al. 2022; Li et al. 2021; Mehta et al. 2022). Notably, there also exists an approach for transfer learning from few samples, DP-FiLM introduced by Tobaben et al. (2023). Such approaches have been shown effective, for loose DP guarantees (*i.e.,* large ε), yet suffer from severe utility drops in strong privacy regimes (*i.e.,* with small ε). This is because of the iterative nature of the DPSGD algorithm with multiple rounds of noise addition that negatively impact performance. To overcome these limitations, Mehta et al. (2023) proposed Differentially Private Least Squares (DP-LS), DP-Newton and DPSGD with Feature Covariance (DP-FC). DP-LS takes advantage of the closed form solution for least squares to avoid running many iterations of gradient descent. DP-Newton employs a second-order optimization to solve the smaller problem of transfer learning more efficiently. DP-FC integrates second order information by utilizing the covariance of the features without paying the composition cost of DP-Newton. All methods have three hyperparameters. In contrast to theirs, our method only has a single optional hyperparameter, does not rely on higher order optimization and utilizes parallel composition to solve each class independently in a single iteration, resulting in lower privacy costs especially for imbalanced datasets.

Leveraging Public Data for Private Training. Public data has, so far, been leveraged for privacy-preserving knowledge transfer to protect sensitive data (Papernot et al. 2017, 2018), to reduce the sample complexity within DP distribution learning (Bie, Kamath, and Singhal 2022; Ben-David et al. 2024), and for pre-training public encoders to then perform private transfer learning (Li et al. 2021; Yu et al. 2021; Gu, Kamath, and Wu 2022; Tramèr, Kamath, and Carlini 2022; Ganesh et al. 2023; Hu et al. 2021; Houlsby et al. 2019; Li et al. 2023; Mehta et al. 2023). In a similar vein as previous work that determines the importance of public samples to private data (Ji and Elkan 2013), our approach goes beyond the latter and additionally leverages the public pre-training data of the encoder during the private transfer learning step by selecting public prototypes to represent our private classes.

Private Training on Unbalanced Datasets. DP has been shown to disproportionately harm utility for underrepresented sub-groups, *i.e.,* groups with fewer data points (Bagdasaryan, Poursaeed, and Shmatikov 2019; Suriyakumar et al. 2021). This is because the weak signal from these groups is more affected by the added noise. Additionally, the clipping operation in DPSGD changes the direction of the overall gradient, which adds a compounding bias over the runtime of the training, that disproportionally affects minority classes (Esipova et al. 2023). To mitigate this issue, Esipova et al. (2023) propose DPSGD-Global-Adapt, which clips only some gradients and instead scales most gradients, thus preserving the overall direction. The algorithm adaptively learns the clipping threshold, keeping the amount of clipped gradients low. Another approach is to add fairness through in- or post-processing (Jagielski et al. 2019), which trades off accuracy against fairness and requires additional privacy budget. In a non-private setting, solutions for improving utility of small subgroups include changing the sampling (Domingos 1999; Kubat, Matwin et al. 1997; Japkowicz 2000; Lewis and Catlett 1994; Ling and Li 1998; Zada, Benou, and Irani 2022), generating synthetic data for the minority classes (Chawla et al. 2002; Wang et al. 2018), or weighing the training loss (Cao et al. 2019). However, these approaches are not directly compatible with DP or incur additional privacy costs.

4 Differentially Private Prototyping

Setup and Assumptions. We aim at learning a private classifier based on a sensitive labeled dataset $D = (\mathbf{X}, \mathbf{y})$ with C different classes. We assume the availability of a standard public pre-trained vision encoder $\hat{M}$, such as DINO[1] or MAE[2] encoders, that return high-dimensional feature vectors for their input data points. Additionally, we assume the availability of a general purpose public dataset $\hat{D} = (\hat{\mathbf{X}}, \dots)$, such as ImageNet (Deng et al. 2009). Note that $\hat{D}$ can also be from a different distribution than D and $\hat{M}$'s pre-training data, as we show experimentally in Figure 8a, and does not require labels. In case of available labels for $\hat{D}$, we just discard them.

Overview. Our goal is to obtain private prototypes $\mathbf{p}_1, \dots \mathbf{p}_C$ that represent every class C from the private dataset D in the embedding space. To classify a new unseen data point $\mathbf{x}'$, we simply have to retrieve the most representative prototype and return its label. Concretely, we have to infer $\mathbf{x}'$ through the encoder $\hat{M}$, retrieve the prototype with the minimum distance in embedding space to $\mathbf{x}'$ and return

[1] https://github.com/facebookresearch/dinov2

[2] https://github.com/facebookresearch/mae

its label as the prediction $y' = \min_{c \in C} d(\hat{M}(\mathbf{x}'), \mathbf{p}_c)$. We detail the general approach in Figure 1.

Note that if the private prototypes are obtained with DP guarantees, using them for predictions will not incur additional privacy costs due to the DP post-processing guarantees. Hence, our DP prototypes can be publicly released, similar to privately trained ML models. We experimented with multiple ways for implementing DP prototypes and identified the two most promising approaches: `DPPL-Mean` generates a private prototype by calculating a DP mean on all data points of a given class in the embedding space. Our `DPPL-Public` takes advantage of the public dataset $\hat{D}$ and privately selects a data point from $\hat{D}$ to act as a prototype for each private class.

4.1 DPPL-Mean: Private Means

Intuition. Non-private prototypical networks (Snell, Swersky, and Zemel 2017) consist of two steps, namely the training of a projection layer at the output of the encoder and the estimation of the class prototypes. In the private setup, both these steps would depend on the private data and therefore each incur additional privacy costs. To keep privacy cost low, we forgo projection layer training, as we find it is unnecessary when given a strong pre-trained encoder (see Appendix C.6). Hence, for our private `DPPL-Mean`, we only implement the estimation of the prototypes without projection.

Non-Private Means. Given a training class c and corresponding samples $\mathbf{X}_c \in \mathbb{R}^{n_c \times d}$, the non-private prototype of each class is the mean of the embeddings $\frac{1}{n_c} \sum_{i=0}^{n_c} \hat{M}(\mathbf{x}_i)$

Our DPPL-Mean: Private Means. To privately estimate the means, we rely on the Gaussian Mechanism. We first clip each $\mathbf{x}_i \in \mathbb{R}^d$ to a ℓ_2 norm r. The estimate is then

$$\mathbf{p}_c = \mathcal{N}\left(\mathbf{0}, \frac{2r^2}{n_c^2 \rho}\right) + \frac{1}{n_c} \sum_{i=0}^{n_c} \hat{M}(clip_{\ell_2}(\mathbf{x}_i), r) \quad (1)$$

where ρ is the zCDP privacy budget. To improve the utility at strict privacy budgets, we include a single optional hyperparameter $k_{\text{pool}} \geq 1$, describing the kernel size of an average pooling layer before the mean estimation to reduce dimensionality, reducing the dimension from d to d/k_{pool}.

Privacy Analysis. The privacy analysis of `DPPL-Mean` follows the analysis of the Gaussian Mechanism. By clipping each sample to ℓ_2 norm of r we obtain $\Delta \mathbf{p}_c = 2r/n$, since the ℓ_2 distance between the previous mean and any new sample can be $2r$ at maximum and its influence diminishes with the number of samples n. We use parallel composition: each disjoint class computes a ρ-zCDP mean prototype, making the privacy cost ρ for the entire private dataset.

4.2 DPPL-Public: Privately Selecting Public Prototypes

Intuition. Our main idea for `DPPL-Public` is to leverage public data beyond the pre-training stage for learning a private classifier based on the sensitive data. Therefore, we privately select public prototypes for each training class, *i.e.,* a data point from the public dataset that represent the given class well.

Non-Private Selection. A good public prototype $\hat{\mathbf{x}}_c$ for a given training class c represents that class well in the embedding space of encoder $\hat{M}$. To select such a good prototype per class, we first calculate the embeddings $\mathbf{E} = \hat{M}(\mathbf{X})$ and $\hat{\mathbf{E}} = \hat{M}(\hat{\mathbf{X}})$ for the private and public data points, respectively. Then, based on the private labels $\mathbf{y}$, we split the embeddings of $\mathbf{X}$ in C subsets $\mathbf{E}_1, \ldots, \mathbf{E}_C$. Without any privacy considerations, a public prototype $\hat{\mathbf{x}}_c$ for class c could then be chosen as the data point that minimizes the average distance according to metric d, to all training data points $\mathbf{x}_i$ in class c as

$$\hat{\mathbf{x}}_c = \min_{\hat{\mathbf{x}} \in \hat{\mathbf{X}}} \frac{\sum_{i=0}^{|\mathbf{X}_c|} d(\hat{M}(\mathbf{x}_i), \hat{M}(\hat{\mathbf{x}}))}{|\mathbf{X}_c|}. \quad (2)$$

Our DPPL-Public: Private Selection. The previously described approach, however, does not take any privacy of the training data D into account. To perform public prototype selection with ε-DP guarantees, we rely on the exponential mechanism (McSherry and Talwar 2007). We use the cosine similarity and add $+1$ as our distance metric d, therefore obtaining a bounded and non-negative metric in $[0, 2]$. The utility function that indicates the goodness of each each public sample for a given class c is

$$u(\hat{\mathbf{x}}, c) = \sum_{i=0}^{|\mathbf{X}_c|} 1 + \frac{\hat{M}(\mathbf{x}_i) \cdot \hat{M}(\hat{\mathbf{x}})}{\|\hat{M}(\mathbf{x}_i)\| \|\hat{M}(\hat{\mathbf{x}})\|}. \quad (3)$$

To improve utility at strict privacy budgets, we include two optional hyperparameters $d_{\text{max}} \in (0, 2]$ and $d_{\text{min}} \in [0, d_{\text{max}})$, which clips the distances to $[d_{\text{min}}, d_{\text{max}}]$, reducing the sensitivity to $\Delta u = d_{\text{max}} - d_{\text{min}}$.

We detail the full algorithm for privately selecting public prototypes in Algorithm 1.

Algorithm 1: Privately Select Public Prototypes

Input: Private dataset $D = (\mathbf{X}, \mathbf{y})$ with C classes, privacy budget ϵ, public pre-trained encoder $\hat{M}$, public dataset $\hat{D} = (\hat{\mathbf{X}}, \ldots)$, hyperparameters d_{max},d_{min}

Output: Prototypes $\mathbf{P} = \{\mathbf{p}_c \in \hat{D} | c \in \mathbf{y}\}$

Function `SelectPublicPrototypes()`**:**

$\mathbf{E} \leftarrow M(\mathbf{X})$ $\hat{\mathbf{E}} \leftarrow M(\hat{\mathbf{X}})$ **foreach** *class* $c \in C$ **do**

$\mathbf{E}_c \leftarrow \{\mathbf{e}_i \in \mathbf{E} | y_i = c\}$

$u_c(\hat{\mathbf{x}}_i) = \sum_{\mathbf{e} \in \mathbf{E}_c} \text{clip}(1 + \frac{\mathbf{e} \cdot \hat{\mathbf{e}}_i}{\mathbf{e}\hat{\mathbf{e}}_i}, d_{\text{max}}, d_{\text{min}})$;

$\mathbf{p}_c \propto \exp\left(\frac{\epsilon u_c}{d_{\text{max}} - d_{\text{min}}}\right)$

return $\{\mathbf{p}_c | c \in \mathbf{y}\}$;

Privacy Analysis. Our proposed `DPPL-Public` fulfills ε-DP. We provide a sketch of the full proof from Appendix D.1 here. We first note that $\Delta u = d_{\text{max}} - d_{\text{min}}$. As mentioned above, we add +1 to each cosine similarity, which is therefore non-negative. Therefore, u is positively monotonic w.r.t. $\mathbf{X}$. The exponential mechanism with

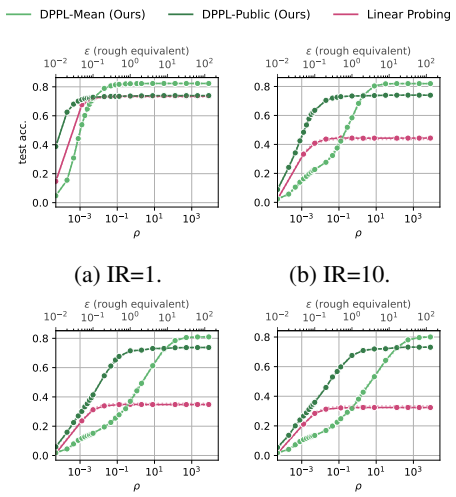
$\Pr[\text{EM}(\mathbf{X}) = \hat{\mathbf{x}}] \propto \exp \frac{\varepsilon u(\mathbf{X}, \hat{\mathbf{x}})}{\Delta u}$ is ε-DP if u is monotonic w.r.t. the private data $\mathbf{X}$ (McSherry and Talwar 2007). Therefore, executing `DPPL-Public` on a single class yields ε-DP. Additionally, since we calculate prototypes per-class and the classes are non-overlapping, parallel composition applies, *i.e.,*, the total privacy costs are also ε-DP.

5 Empirical Evaluation

Experiment Setup. We experiment with CIFAR10 (Krizhevsky 2009), CIFAR100 (Krizhevsky 2009), STL10 (Coates, Ng, and Lee 2011) and FOOD101 (Bossard, Guillaumin, and Van Gool 2014) as private datasets. From these datasets, we construct exponentially long-tailed imbalanced datasets with various imbalance ratios (IRs), the ratio between the number of samples in the largest and smallest class, following Cui et al. (2019) and Cao et al. (2019). Concretely, the number of samples in each class decreases exponentially with a factor of $n(c) = \exp(-c\lambda)$, where $\lambda = \log(\text{IR})/C$. For the balanced case(IR $= 1$), $\lambda = \log(1)/C = 0$ and therefore $n(c) = 1 \, \forall_c$. We detail the imbalancing process and depict the effect on the resulting absolute class sizes per dataset further in Appendix B.2. We compare our DP prototypes on the features obtained from three vision transformers based on the original architecture from Dosovitskiy et al. (2020) Vit-B-16 (Singh et al. 2022), namely ViT-L-16 (Oquab et al. 2023), ViT-H-14 (Singh et al. 2022) and a ResNet-50 (Caron et al. 2021; He et al. 2016). All models, except for ViT-L-16, which is trained on LVD-142M introduced by Oquab et al. (2023), are trained on ImageNet-1K (Deng et al. 2009). For `DPPL-Public`, we use the 64×64 downscaled version of ImageNet-1K upscaled to between 128×128 and 512×512 depending on the encoder. Notably, we evaluate our methods on the standard *balanced* test set. This corresponds to reporting a *balanced accuracy* for the imbalanced setups. A full description of our experimental setup is provided in Appendix B.

Baselines. For the baseline comparisons we compare to standard linear probing with DPSGD, as it's a common way of DP transfer-learning. Furthermore, we compare to DP-LS from Mehta et al. (2023) which is the current state-of-the-art for DP transfer learning across all privacy regimes and to DPSGD-Global-Adapt from Esipova et al. (2023) as it is specifically designed for for training on imbalanced datasets. We outline in Appendix E.4 the experimentally-supported reasons against including DP-FC and DP-FiLM.

Comparing Results over Different Notions of DP. Since we are comparing our new proposed methods that implement *pure DP* or *pure* ρ-zCDP guarantees against other baselines that also implement zCDP, we convert all privacy guarantees to ρ-zCDP. We also present a pure-DP ε equivalent by inverting the $\rho = \epsilon^2/2$ conversion from pure DP to zCDP. However, this does not imply that these algorithms fulfil pure DP. We detail all comparison implementations and conversion theorems used in Appendix D.2.



(a) IR=1. (b) IR=10.

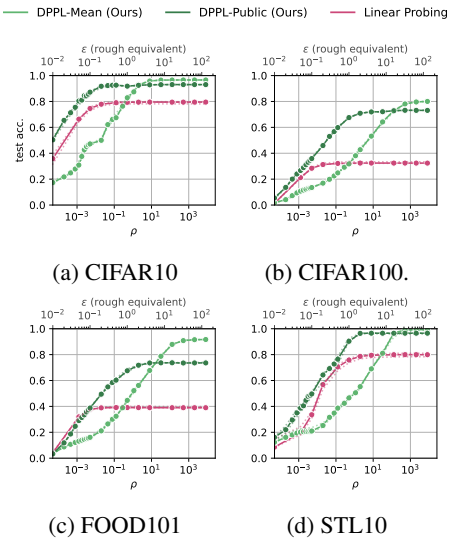
(c) IR=50. (d) IR=100.

Figure 2: **DP Prototypes on CIFAR100.** We present the balanced test accuracy of our methods vs. standard linear probing with DP-SGD on CIFAR100 and ViT-H-14 at different levels of imbalance rations (IR), using ImageNet as public data for `DPPL-Public`. We plot the mean test accuracy over multiple runs and represent the upper and lower quantiles for all methods by the dotted lines.

5.1 DP Prototypes: High Utility in High Privacy and Extreme Imbalance

We evaluate our DP prototypes at different privacy regimes in the range corresponding to standard approximate DP (Dwork et al. 2006) of $0.01 < \varepsilon < 100$ and under different IRs. In Figure 2, we benchmark our methods vs. standard DP linear probing on CIFAR100 with ViT-H-14. Our results highlight that over all levels of IRs larger than 1, our `DPPL-Public` significantly outperforms linear probing in all privacy regimes. Additionally `DPPL-Public` yields strong performance already at very low ε, such as $\varepsilon = 0.1$ for IR=1. As data becomes more imbalanced, all methods require larger privacy budgets to yield similarly high performance. Further, our results indicate that our `DPPL-Mean` method underperforms `DPPL-Public` and DP linear probing for low ε. We find that this results from the noise added during the mean calculation leading diverging estimations. We provide further detail on the cause in Appendix C.2. Yet, we observe that at higher epsilon, `DPPL-Mean` outperforms `DPPL-Public`. This suggests that the most beneficial way for leveraging DP prototypes might be an adaptive method where `DPPL-Public` is chosen for high performance in the high privacy regimes and `DPPL-Mean` can further boost performance for larger ε.

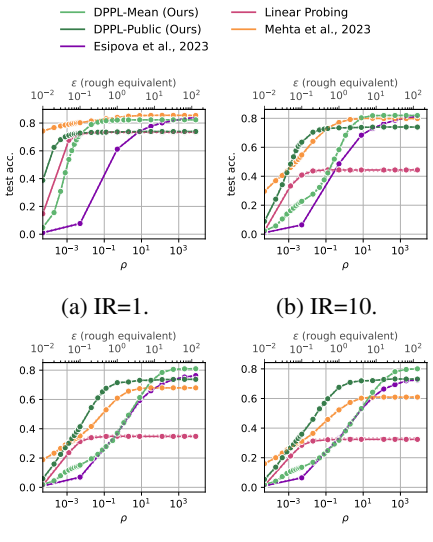
We further assess whether the observed trend holds over different datasets. Therefore, we depict the results of our methods vs. standard DP linear probing for different datasets and the ViT-H-14 encoder under the most challenging setup with IR$= 100$ in Figure 3. The observed trends are indeed consistent between all datasets. We provide full results over



(a) CIFAR10 (b) CIFAR100.

(c) FOOD101 (d) STL10

Figure 3: **DP Prototypes on various imbalanced datasets.** We present the balanced test accuracy for CIFAR10, CIFAR100, FOOD101 and STL10 at an imbalance ratio of 100 on ViT-H-14, using ImageNet as public data for `DPPL-Public`. We compare to standard Linear Probing with DP-SGD. We plot the mean test accuracy over multiple runs and represent the upper/lower quantiles by the dotted lines. Appendix E.1 shows more results.



(a) IR=1. (b) IR=10.

(c) IR=50. (d) IR=100.

Figure 4: **Comparing against baselines on CIFAR100.** We present the results of our methods vs. state-of-the-art methods (DP-LS and DPSGD-Global-Adapt) on the CIFAR100 dataset using ViT-H-14 under different IRs. `DPPL-Public` uses ImageNet as public data. Dotted lines represent the upper/lower quantiles. Similar results for CIFAR10, Food101, and STL10 are presented in Appendix E.1.

all datasets and IRs in Appendix E.1.

5.2 DP Prototypes Improve over State-of-the-Art Baselines in Imbalanced Setups

We further benchmark our methods against DP-LS, the current state-of-the-art method for private transfer learning by Mehta et al. (2023), and DPSGD-Global-Adapt by Esipova et al. (2023), a DP method deliberately designed to achieve high utility under imbalancedness of the private data. Our results in Figure 4 highlight that while in the balanced setup, Mehta et al. (2023) outperforms the other methods, our DP prototypes outperform all other methods the more imbalanced the setup becomes. In particular `DPPL-Public` outperforms in high privacy regimes (*i.e.,* for small ε), while `DPPL-Mean` is better in lower privacy regimes, mostly outperforming even DPSGD-Global-Adapt.

The advantage of our methods against the baselines become even more obvious as we do not consider the accuracy over the entire balanced test set (equivalent to balanced accuracy), but look specifically at accuracy on minority classes, see Figure 5. Therefore, we take the smallest 25% of training classes in terms of number of their training data points and measure their accuracy on a balanced test set consisting of only those classes. Our results for CIFAR100 on ViT-H-14 under IR$=50$ in Figure 5 highlight that our DP prototypes significantly outperform all baselines. Full results for the minority classes are depicted in Appendix E.2.

5.3 Understanding the Success of DP Prototypes

To better understand the success of our DP prototypes, we perform various ablations. The full set of ablations and their

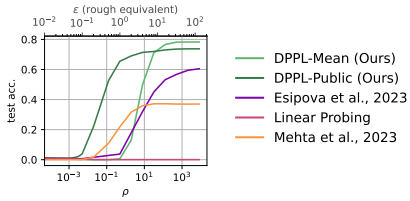
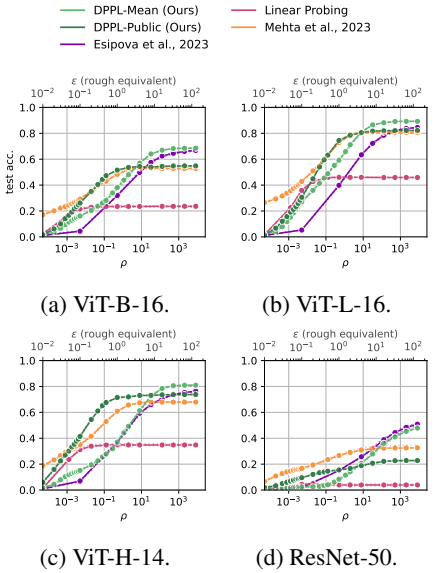


Figure 5: **Accuracies of the minority classes.** We depict the test accuracy on CIFAR100 with ViT-H-14 embeddings for the minority classes (smallest 25% of classes) at IR $= 50$.

results is presented in Appendix C.

Effect of the Publicly Pre-trained Encoder. We first assess the impact of the encoder used as a feature extractor. Therefore, we apply our method and the baselines with different encoder architectures. Our results in Figure 6 highlight that the encoder performance impacts all methods alike. In particular, we observe that all methods obtain better results with stronger encoders. For example, the ViT-H-14 yields to significantly higher private prediction accuracy that the much smaller ViT-B-16. Additionally, none of the methods yields satisfactory results using the ResNet50.

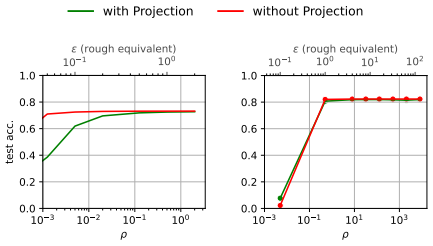
Impact of the Projection Layer. We evaluate whether a projection layer, usually part of a prototypical network, can increase the utility. We present our results in Figure 7 for CIFAR100 on ViT-H-14, using ImageNet as public for `DPPL-Public`. They highlight that `DPPL-Public` does



(a) ViT-B-16. (b) ViT-L-16.

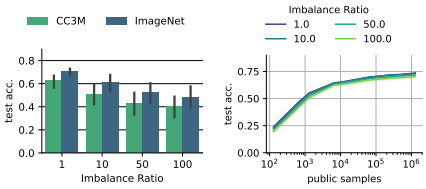
(c) ViT-H-14. (d) ResNet-50.

Figure 6: **Choice of Encoder.** We report the test accuracy for our methods and the baselines for CIFAR100, using ImageNet as public data for `DPPL-Public`. We observe that the success of all methods depends on the quality of the underlying encoder.



(a) `DPPL-Public`. (b) `DPPL-Mean`.

Figure 7: **Impact of the Projection Layer.**



(a) Public Distribution. (b) Public Data Size.

Figure 8: **Impact of the Public Data.**

not benefit from the projection. Even non-privately ($\epsilon = \infty$), the accuracy of `DPPL-Public` with projection is 72.6% and without projection is 74.0%, showing that it is not just the decreased privacy budget for the sampling that reduces the utility, but a fundamental misalignment between how the projection is trained and the public prototyping. We observe the same effect for `DPPL-Mean` and conclude that, although this limits adaptability (see Appendix F.2), with a strong enough pre-trained encoder, it is sufficient for `DPPL` to estimate prototypes without projection.

Improving through Multiple Per-Class Prototypes. We experiment with extending our `DPPL-Public` beyond a single per-class prototype—the common standard for prototypical networks. Therefore, we introduce the variation `DPPL-PublicK` which selects the top-K public prototypes per class. We extend the algorithm from Gillenwater et al. (2022) to sample multiple prototypes jointly using the exponential mechanism. Then, we classify based on the mean distance to each class's prototype. Our results in Figure 14 show that `DPPL-PublicK`'s privacy-utility trade-offs are between `DPPL-Mean` and `DPPL-Public`, indicating that the private means can be —to a certain degree— approximated by multiple public prototypes. `DPPL-PublicK` could therefore replace `DPPL-Mean` in cases of high dimensionality, where a mean estimation is not feasible. We provide more details, privacy proof, and a full set of results in Appendix C.3.

Impact of the Public Data for Prototype Selection. To assess whether the public data for prototype estimation needs to be from the same distribution as the one used to pre-train the encoder, we conduct experiments with a different public dataset. We evaluate `DPPL-Public` using 2,298,112 samples from CC3M introduced by Sharma et al. (2018) as public data instead of ImageNet which is used to pre-train the encoder. We show the relation between the accuracy using ImageNet as public data and CC3M in Figure 8a for CIFAR100 on ViT-H-14. We find that `DPPL-Public` works well with both public datasets, highlighting the flexibility of our approach. Still, we observe that ImageNet yields better results which suggests that it is particularly beneficial to leverage the public data already available for pre-training also in the private transfer learning step.

Size of the Public Dataset for DPPL-Public. We also evaluate the success of `DPPL-Public` for different sizes of the public dataset that the public prototypes can be chosen from. Therefore, we randomly draw subsets of different sizes from ImageNet and apply `DPPL-Public`. Our results for CIFAR100 (100 classes) and ViT-H-14 in Figure 8b indicate that with growing public dataset size, our method's success increases. Figure 10 shows that tasks less similar to the pretraining data are more sensitive to dataset size. Note that even for public dataset of more than one million images, the selection of our public prototypes for 100 classes (*i.e.,* the "training") takes 34.3 seconds on a single GPU as we depict in Appendix E.3. For 10 classes (*e.g.,* CIFAR10), it takes 5 seconds. Hence, choosing a larger public dataset does not represent a practical limitation.

6 Conclusions and Future Work

We propose `DPPL` as a novel alternative to private fine-tuning with DP. `DPPL` builds DP prototypes on top of features extracted by a publicly pre-trained encoder, that can be later used as a classifier. The prototypes can be obtained without iterative noise addition and yield high utility even in

high-privacy regimes, with few private training data points, and in unbalanced training setups. We show that we can further boost performance of our DP prototypes by leveraging the public data beyond training of the encoder and using them to draw the public prototypes from (DPPL-Public). Future work at improving utility of high-dimensional DP mean estimation will benefit our DPPL-Mean, which can, in the future, serve as an additional benchmark for private mean estimation algorithms.

Acknowledgements

This research project was supported by the Google Safety Engineering Center. The project on which this paper is based was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project number 550224287. We thank the Helmholtz Information and Data Science Academy (HIDA) for supporting this research through the Helmholtz Visiting Researcher Grant.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 308–318. Vienna Austria: ACM. ISBN 978-1-4503-4139-4.

Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; and Koyama, M. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, 2623–2631. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-4503-6201-6.

Bagdasaryan, E.; Poursaeed, O.; and Shmatikov, V. 2019. Differential Privacy Has Disparate Impact on Model Accuracy. *Advances in Neural Information Processing Systems*, 32.

Ben-David, S.; Bie, A.; Canonne, C. L.; Kamath, G.; and Singhal, V. 2024. Private distribution learning with public data: The view from sample compression. *Advances in Neural Information Processing Systems*, 36.

Bergstra, J.; Bardenet, R.; Bengio, Y.; and Kégl, B. 2011. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Bie, A.; Kamath, G.; and Singhal, V. 2022. Private estimation with public data. *Advances in neural information processing systems*, 35: 18653–18666.

Biswas, S.; Dong, Y.; Kamath, G.; and Ullman, J. 2020. CoinPress: Practical Private Mean and Covariance Estimation. In *Advances in Neural Information Processing Systems*, volume 33, 14475–14485. Curran Associates, Inc.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101 – Mining Discriminative Components with Random Forests. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, 446–461. Cham: Springer International Publishing. ISBN 978-3-319-10599-4.

Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M. J.; Leary, C.; Maclaurin, D.; Necula, G.; Paszke, A.; VanderPlas, J.; Wanderman-Milne, S.; and Zhang, Q. 2018. JAX: composable transformations of Python+NumPy programs.

Bu, Z.; Gopi, S.; Kulkarni, J.; Lee, Y. T.; Shen, H.; and Tantipongpipat, U. 2021. Fast and memory efficient differentially private-sgd via jl projections. *Advances in Neural Information Processing Systems*, 34: 19680–19691.

Bu, Z.; Mao, J.; and Xu, S. 2022. Scalable and efficient training of large convolutional neural networks with differential privacy. *Advances in Neural Information Processing Systems*, 35: 38305–38318.

Buda, M.; Maki, A.; and Mazurowski, M. A. 2018. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106: 249–259.

Bun, M.; and Steinke, T. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In Hirt, M.; and Smith, A., eds., *Theory of Cryptography*, Lecture Notes in Computer Science, 635–658. Berlin, Heidelberg: Springer. ISBN 978-3-662-53641-4.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; and Tramer, F. 2022. Membership Inference Attacks from First Principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, 1519–1519. IEEE Computer Society.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Cesar, M.; and Rogers, R. 2021. Bounding, Concentrating, and Truncating: Unifying Privacy Loss Composition for Data Analytics. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, 421–457. PMLR. ArXiv:2004.07223 [cs].

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16: 321–357.

Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 215–223. JMLR Workshop and Conference Proceedings. ISSN: 1938-7228.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-Balanced Loss Based on Effective Number of Samples. 9268–9277.

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

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. ISSN: 1063-6919.

Domingos, P. 1999. MetaCost: a general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 155–164. San Diego California USA: ACM. ISBN 978-1-58113-143-7.

Dong, J.; Durfee, D.; and Rogers, R. 2020. Optimal Differential Privacy Composition for Exponential Mechanisms. In *Proceedings of the 37th International Conference on Machine Learning*, 2597–2606. PMLR. ISSN: 2640-3498.

Dong, J.; Roth, A.; and Su, W. J. 2019. Gaussian Differential Privacy. ArXiv:1905.02383 [cs, stat].

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Durfee, D.; and Rogers, R. M. 2019. Practical Differentially Private Top-k Selection with Pay-what-you-get Composition. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In Halevi, S.; and Rabin, T., eds., *Theory of Cryptography*, Lecture Notes in Computer Science, 265–284. Berlin, Heidelberg: Springer. ISBN 978-3-540-32732-5.

Esipova, M. S.; Ghomi, A. A.; Luo, Y.; and Cresswell, J. C. 2023. Disparate impact in differential privacy from gradient misalignment. In *The eleventh international conference on learning representations*.

Feldman, V. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, 954–959. Chicago IL USA: ACM. ISBN 978-1-4503-6979-4.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1322–1333.

Ganesh, A.; Haghifam, M.; Nasr, M.; Oh, S.; Steinke, T.; Thakkar, O.; Thakurta, A. G.; and Wang, L. 2023. Why is public pretraining necessary for private model training? In *International Conference on Machine Learning*, 10611–10627. PMLR.

Gillenwater, J.; Joseph, M.; Munoz, A.; and Diaz, M. R. 2022. A Joint Exponential Mechanism For Differentially Private Top-k. In *Proceedings of the 39th International Conference on Machine Learning*, 7570–7582. PMLR. ISSN: 2640-3498.

Gu, X.; Kamath, G.; and Wu, S. 2022. Choosing Public Datasets for Private Machine Learning via Gradient Subspace Distance. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*.

He, J.; Li, X.; Yu, D.; Zhang, H.; Kulkarni, J.; Lee, Y. T.; Backurs, A.; Yu, N.; and Bian, J. 2022a. Exploring the Limits of Differentially Private Deep Learning with Group-wise Clipping. In *The Eleventh International Conference on Learning Representations*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022b. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16000–16009.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. 770–778.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Jagielski, M.; Kearns, M.; Mao, J.; Oprea, A.; Roth, A.; Sharifi Malvajerdi, S.; and Ullman, J. 2019. Differentially Private Fair Learning. In *Proceedings of the 36th International Conference on Machine Learning*, 3000–3008. PMLR. ISSN: 2640-3498.

Japkowicz, N. 2000. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on artificial intelligence*, volume 56, 111–117.

Ji, Z.; and Elkan, C. 2013. Differential privacy based on importance weighting. *Machine Learning*, 93(1): 163–183.

Kamath, G.; and Ullman, J. 2020. A Primer on Private Statistics. ArXiv:2005.00010 [cs, math, stat].

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Kubat, M.; Matwin, S.; et al. 1997. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, 179. Citeseer.

Lee, J.; and Kifer, D. 2021. Scaling up differentially private deep learning with fast per-example gradient clipping. *Proceedings on Privacy Enhancing Technologies*.

Lewis, D. D.; and Catlett, J. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning. In Cohen, W. W.; and Hirsh, H., eds., *Machine Learning Proceedings 1994*, 148–156. San Francisco (CA): Morgan Kaufmann. ISBN 978-1-55860-335-6.

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

Li, Y.; Tsai, Y.-L.; Yu, C.-M.; Chen, P.-Y.; and Ren, X. 2023. Exploring the benefits of visual prompting in differential privacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5158–5167.

Ling, C. X.; and Li, C. 1998. Data mining for direct marketing: Problems and solutions. In *Kdd*, volume 98, 73–79.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2537–2546.

McSherry, F.; and Talwar, K. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, 94–103. IEEE.

McSherry, F. D. 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, 19–30. New York, NY, USA: Association for Computing Machinery. ISBN 978-1-60558-551-2.

Medina, A. M.; and Gillenwater, J. 2021. Duff: A Dataset-Distance-Based Utility Function Family for the Exponential Mechanism. ArXiv:2010.04235 [cs].

Mehta, H.; Krichene, W.; Thakurta, A. G.; Kurakin, A.; and Cutkosky, A. 2023. Differentially private image classification from features. *Transactions on Machine Learning Research*. ArXiv:2211.13403 [cs].

Mehta, H.; Thakurta, A.; Kurakin, A.; and Cutkosky, A. 2022. Large Scale Transfer Learning for Differentially Private Image Classification. ArXiv:2205.02973 [cs].

Mironov, I. 2017. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 263–275. Santa Barbara, CA: IEEE. ISBN 978-1-5386-3217-8.

Moritz, P.; Nishihara, R.; Wang, S.; Tumanov, A.; Liaw, R.; Liang, E.; Elibol, M.; Yang, Z.; Paul, W.; Jordan, M. I.; and Stoica, I. 2018. Ray: A Distributed Framework for Emerging {AI} Applications. 561–577. ISBN 978-1-939133-08-3.

Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Assran, M.; Ballas, N.; Galuba, W.; Howes, R.; Huang, P.-Y.; Li,

S.-W.; Misra, I.; Rabbat, M.; Sharma, V.; Synnaeve, G.; Xu, H.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. 2023. DINOv2: Learning Robust Visual Features without Supervision. ArXiv:2304.07193 [cs].

Papernot, N.; Abadi, M.; Úlfar Erlingsson; Goodfellow, I.; and Talwar, K. 2017. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. In *Proceedings of the International Conference on Learning Representations*.

Papernot, N.; Song, S.; Mironov, I.; Raghunathan, A.; Talwar, K.; and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *International Conference on Learning Representations (ICLR)*.

Papernot, N.; and Steinke, T. 2021. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Reed, W. J. 2001. The Pareto, Zipf and other power laws. *Economics letters*, 74(1): 15–19.

Rocklin, M. 2015. Dask: Parallel Computation with Blocked algorithms and Task Scheduling. In *SciPy*, 126–132. Austin, Texas.

Rogers, R.; and Steinke, T. 2021. A Better Privacy Analysis of the Exponential Mechanism.

Sharma, P.; Ding, N.; Goodman, S.; and Soricut, R. 2018. Conceptual Captions: A Cleaned, Hypernymed, Image Alt-text Dataset For Automatic Image Captioning. In Gurevych, I.; and Miyao, Y., eds., *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2556–2565. Melbourne, Australia: Association for Computational Linguistics.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Singh, M.; Gustafson, L.; Adcock, A.; de Freitas Reis, V.; Gedik, B.; Kosaraju, R. P.; Mahajan, D.; Girshick, R.; Dollár, P.; and van der Maaten, L. 2022. Revisiting Weakly Supervised Pre-Training of Visual Perception Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 804–814.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Song, Q.; Peng, Z.; Ji, L.; Yang, X.; and Li, X. 2022. Dual Prototypical Network for Robust Few-shot Image Classification. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 533–537. Chiang Mai, Thailand: IEEE. ISBN 978-616-590-477-3.

Song, S.; Chaudhuri, K.; and Sarwate, A. D. 2013. Stochastic gradient descent with differentially private updates. In *2013 IEEE global conference on signal and information processing*, 245–248. IEEE.

Subramani, P.; Vadivelu, N.; and Kamath, G. 2021. Enabling fast differentially private sgd via just-in-time compilation and vectorization. *Advances in Neural Information Processing Systems*, 34: 26409–26421.

Suriyakumar, V. M.; Papernot, N.; Goldenberg, A.; and Ghassemi, M. 2021. Chasing your long tails: Differentially private prediction in health care settings. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 723–734.

Tobaben, M.; Shysheya, A.; Bronskill, J. F.; Paverd, A.; Tople, S.; Zanella-Beguelin, S.; Turner, R. E.; and Honkela, A. 2023. On the Efficacy of Differentially Private Few-shot Image Classification. *Transactions on Machine Learning Research*.

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

Tsfadia, E.; Cohen, E.; Kaplan, H.; Mansour, Y.; and Stemmer, U. 2022. FriendlyCore: Practical Differentially Private Aggregation. In *Proceedings of the 39th International Conference on Machine Learning*, 21828–21863. PMLR. ISSN: 2640-3498.

Wang, Y.-X.; Balle, B.; and Kasiviswanathan, S. P. 2019. Subsampled Renyi Differential Privacy and Analytical Moments Accountant. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, 1226–1235. PMLR. ISSN: 2640-3498.

Wang, Y.-X.; Girshick, R.; Hebert, M.; and Hariharan, B. 2018. Low-Shot Learning from Imaginary Data. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7278–7286. ISSN: 2575-7075.

Yadan, O. 2019. Hydra - A framework for elegantly configuring complex applications. Github.

Yousefpour, A.; Shilov, I.; Sablayrolles, A.; Testuggine, D.; Prasad, K.; Malek, M.; Nguyen, J.; Ghosh, S.; Bharadwaj, A.; Zhao, J.; Cormode, G.; and Mironov, I. 2021. Opacus: User-Friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298*.

Yu, D.; Naik, S.; Backurs, A.; Gopi, S.; Inan, H. A.; Kamath, G.; Kulkarni, J.; Lee, Y. T.; Manoel, A.; Wutschitz, L.; et al. 2021. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*.

Zada, S.; Benou, I.; and Irani, M. 2022. Pure Noise to the Rescue of Insufficient Data: Improving Imbalanced Classification by Training on Random Noise Images. In *Proceedings of the 39th International Conference on Machine Learning*, 25817–25833. PMLR. ISSN: 2640-3498.

Zhu, Y.; and Wang, Y.-X. 2019. Poission Subsampled Rényi Differential Privacy. In *Proceedings of the 36th International Conference on Machine Learning*, 7634–7642. PMLR. ISSN: 2640-3498.

A Extended Background

A.1 Differential Privacy

Definition 1 ((ξ, ρ)-zCDP from Bun and Steinke (2016)) *A randomised mechanism $M : \mathcal{X}^n \to \mathcal{Y}$ is (ξ, ρ)-zero-concentrated differentially private (henceforth (ξ, ρ)-zCDP) if, for all $x, x' \in \mathcal{X}^n$ differing on a single entry and all $\alpha \in (1, \infty)$,*

$$\mathrm{D}_\alpha \left(M(x) \| M\left(x'\right)\right) \leq \xi + \rho\alpha,$$

where $\mathrm{D}_\alpha \left(M(x) \| M\left(x'\right)\right)$ is the α-Rényi divergence between the distribution of $M(x)$ and the distribution of $M\left(x'\right)$.

$(0, \rho)$-zCDP can also be expressed simply as ρ-zCDP.

Definition 2 ((α, ϵ)-RDP from Mironov (2017)) *A randomized mechanism $f : \mathcal{D} \mapsto \mathcal{R}$ is said to have ϵ-Rényi differential privacy of order α, or (α, ϵ)-RDP for short, if for any adjacent $D, D' \in \mathcal{D}$ it holds that*

$$D_\alpha \left(f(D) \| f\left(D'\right)\right) \leq \epsilon.$$

Definition 3 (μ-GDP from Dong, Roth, and Su (2019)) *A randomized mechanism $M : \mathcal{D} \mapsto \mathcal{R}$ is said to have μ-Gaussian differential privacy, μ-GDP for short, if it operates on a statistic Θ as*

$$M(D) = \Theta(D) + \xi$$

where $\xi \sim \mathcal{N}(0, sens(\theta)^2/\mu^2)$

Theorem 1 (Parallel composition from McSherry (2009)) *Let M_i each provide ϵ-differential privacy. Let D_i be arbitrary disjoint subsets of the input domain D. The sequence of $M_i \left(X \cap D_i\right)$ provides ϵ-differential privacy.*

A.2 The Gaussian Mechanism

Proposition 2 (Cesar and Rogers (2021)) *Let $q : \mathcal{X}^n \to \mathbb{R}$ be a sensitivity-Δ query. Consider the mechanism $M : \mathcal{X}^n \to \mathbb{R}$ that on input $\mathbf{x}$, releases a sample from $\mathcal{N}\left(q(\mathbf{x}), \sigma^2\right)$. Then M satisfies $\left(\Delta^2/2\sigma^2\right)$-zCDP.*

A.3 The Exponential Mechanism

The exponential mechanism (McSherry and Talwar 2007) aims to give the best output $\hat{\mathbf{x}} \in \hat{\mathbf{X}}$ w.r.t. a utility function $u(\mathbf{X}, \hat{\mathbf{x}}) : \mathbf{X} \times \hat{\mathbf{X}} \to \mathbb{R}$ where $\mathbf{X}$ is a private dataset and $\hat{\mathbf{X}}$ a public dataset. It can be described as a randomized mapping $\mathrm{EM}_u : \mathbf{X} \to \hat{\mathbf{X}}$ where

$$P[\mathrm{EM}_u(\mathbf{X}) = \hat{\mathbf{x}}] \propto \exp\left(\frac{\epsilon}{\Delta u} u(\mathbf{X}, \hat{\mathbf{x}})\right) \tag{4}$$

Lemma 1 (McSherry and Talwar (2007)) *The exponential mechanism is 2ϵ-DP.*

Definition 4 *A utility function $U(D, \mathbf{x})$ is positively (negatively) monotonic if for any point $\mathbf{x}$ and any datasets D and D', $U(D, \mathbf{x}) \leq U(D \cup D', \mathbf{x})$ $(U(D, \mathbf{x}) \geq U(D \cup D', \mathbf{x}))$.*

Lemma 2 (McSherry and Talwar (2007)) *Given a monotonic utility function, the exponential mechanism is ϵ-DP.*

The exponential mechanism fulfils not only differential privacy, but also the stricter bounded range (Durfee and Rogers 2019). Using a monotonic utility function leads to an improved privacy bound because the sensitivity and range of monotonic functions are equal (Dong, Durfee, and Rogers 2020).

Lemma 3 (Cesar and Rogers (2021)) *The ϵ-DP exponential mechanism is $\epsilon^2/8$-zCDP.*

A.4 Private Mean Estimation

For the mean estimation, we investigated both *CoinPress* and a naïve estimator based on the Gaussian Mechanism. While both provide zCDP, *CoinPress* provides guarantees for a substitute neighborhood and the Gaussian Mechanism for a add/remove neighborhood, meaning they are not compared under the same guarantees.

CoinPress The *CoinPress* algorithm (Dong, Durfee, and Rogers 2020) aims to privately estimate the mean $\mu = 1/n \sum_n \mathbf{x}$ for some private $\mathbf{x} \in \mathbb{R}^d$. Each step is initiated with a center $\mathbf{c}_i$ and radius r_i with $||\mu - c_i||_2 \leq r_i$. Commonly used for (r_0, c_0) are $(\sqrt{d}, \mathbf{0})$. All points further away from $\mathbf{c}_i$ than $r_i + \gamma$, where γ is chosen s.t. $\Pr[||\mathcal{N}(\mathbf{0}, d)||_2 < \gamma] \geq 0.99$ are ℓ_2-clipped to $r_i + \gamma$. Finally, Gaussian noise is added to all points. Then $\mathbf{c}_{i+1}$ is the mean of the noised and clipped points and r_{i+1} defined through the new Gaussian tailbounds of the points.

Naïve We formulate the mean estimation problem as a private query $q(\mathbf{X}) : \mathbb{R}^{n \times d} \to \mathbb{R}^d$ that we want to release from the private database $\mathbf{X}$ where

$$q(\mathbf{X}) = \frac{1}{\|\mathbf{X}\|} \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{x} \tag{5}$$

Without further bounds, this query cannot satisfy Differential Privacy. We therefore clip all $\mathbf{x} \in \mathbf{X}$ to some ℓ_2 norm r to obtain $\overline{\mathbf{X}} = \{clip_{\ell_2}(\mathbf{x}, r) | \mathbf{x} \in \mathbf{X}\}$. From this we obtain

$$\Delta q(\overline{\mathbf{X}}) = \frac{2r}{n}. \tag{6}$$

and using Theorem 2 a ρ-zCDP mean estimation query q_ρ as

$$q_\rho(\mathbf{X}) = q(\overline{\mathbf{X}}) + \mathcal{N}(\mathbf{0}, 2r^2/(n^2\rho)). \tag{7}$$

To finally obtain a mean $\mathbf{p}_c$ for each class $c \in C$ while fulfilling ρ-zCDP with respect to the entirety of $\mathbf{X}$, we utilize parallel composition (Theorem 1) over the disjoint class subsets $\mathbf{X}_c$.

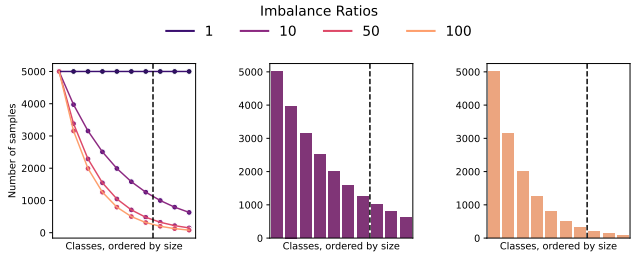
B Extended Experimental Setup

B.1 Computational Resources and Libraries

Our implementation is in Jax (Bradbury et al. 2018) v.0.4.31 with CUDA12. Private linear probing was conducted with PyTorch (Paszke et al. 2019) v.2.3.1 and made private using the Opacus (Yousefpour et al. 2021) v.1.5.2 privacy engine. We relied on Optuna (Akiba et al. 2019) v.3.6.1 using the Tree-structured Parzen Estimator (Bergstra et al. 2011) for all algorithmic hyperparameter optimizations. Converting and visualizing privacy guarantees was done in part with

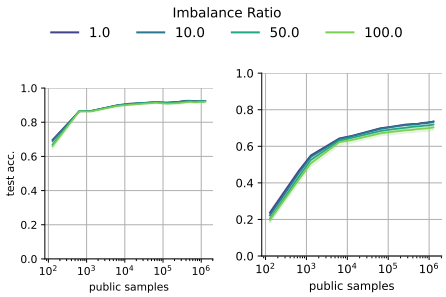
Dataset	Imbalance Ratio 1		10		50		100	
	Median	Min	Median	Min	Median	Min	Median	Min
CIFAR10	5000		1594	500	724	100	517	50
CIFAR100	500		158	50	71	10	50	5
FOOD101	750		237	75	106	15	75	8
STL10	500		160	50	73	10	52	5
FLOWERS101	1		3	1	7	1	10	1

Table 1: **Number of data points over different rations of imbalance.** Median and average number of samples per class for each dataset and imbalance ratio. We construct the imbalanced datasets as described in Cui et al. (2019) and subsequently Cao et al. (2019) (Exponential Long-Tailed).



(a) All IRs. (b) IR=10. (c) IR=100.

Figure 9: **Visualizing the effect of the imbalance on CIFAR10.** We order the classes by the number of corresponding samples and plot them. The classes right of the dotted line are considered to be the minority classes.



(a) CIFAR10. (b) CIFAR100.

(c) STL10. (d) FOOD101.

Figure 10: **Varying public dataset sizes for DPPL-Public**. We randomly subsample the public dataset, limiting `DPPL-Public`'s prototype selection to fewer samples, and evaluate the resulting changes in accuracy. Tasks with less similarity to the pretraining dataset, *e.g.,* FOOD101, are more sensitive to dataset size.

AutoDP (Wang, Balle, and Kasiviswanathan 2019; Zhu and Wang 2019) v.0.2.3.1. Scaling the experiments has been aided by Ray (Moritz et al. 2018) v.2.23.0 and Dask (Rocklin 2015) v.2024.8.0. Configurations were handled by Hydra (Yadan 2019) v.1.3.2. The experiments were conducted using NVIDIA A100 GPUs and an AMD EPYC 7742 64-Core Processor with 1TB of RAM on Ubuntu 22.04 in Python 3.11. In total, obtaining all results required approximately 300 GPU hours, resulting in roughly 105 kWh of electric energy usage.

B.2 Imbalanced Datasets

We present the distribution of the number of data points per class in Appendix B.2 and Figure 9

C Ablations

C.1 Public Dataset Size

The public data used for `DPPL-Public`, ImageNet, consists of 1,281,167 samples. We evalute the impact of using a smaller public dataset by randomly subsampling. Figure 10 shows the resulting accuracy. The accuracy and amount of public samples are positively correlated in all cases, but for some datasets, *i.e.,* CIFAR10, STL10, the accuracy seemingly asymptotically approaches a maximum accuracy. For FOOD101 it seems ImageNet is not large enough. We note that the imbalance ratio doesn't change the amount of public data required.

C.2 Comparing Different Mean Estimations

We find that the naïve estimator outperforms *CoinPress* given the strong priors on the ℓ_2 norms of the embeddings and the fact that they are generally close to the origin. Figure 11 compares the accuracy from both mean estimation methods.

Figure 12 shows how the *CoinPress* private mean estimation behaves for reasonable and too low privacy levels. For too low privacy values, the mean estimation breaks down. We identify the underlying cause as the divergence of the bounding radius and visualize it in Figure 13. Each successive radius is supposed to be decreasing in size, successively bounding the estimated mean to a smaller space. This is achieved by taking the mean of clipped and noised samples. The clipping decreases the average norm and thus reduces the radius. For very low privacy budgets, the noising of the samples outweighs this effect, and the norms instead grow with each step, leading to a diverging radius. As we take the mean of increasingly diverging samples, the estimates of the mean diverge.

While the naïve estimator also diverges at low privacy budgets, we find the minimum privacy budget required to be lower compared to *CoinPress*.

C.3 Top-K Public Prototyping

Prototypical Networks have been extended to two prototypes per class, leading to increased generalization and robustness (Song et al. 2022). We generalize this concept to K proto-

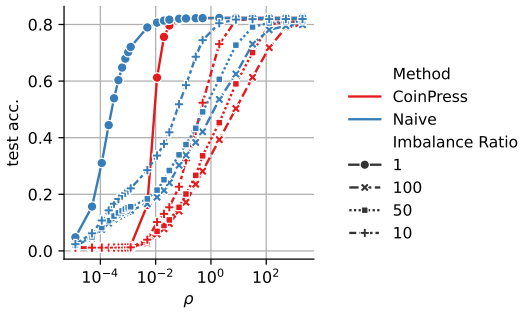


Figure 11: **Comparing *CoinPress* and naïve mean estimation results** for CIFAR100 and ViT-H-14 embeddings.

types per class. We propose our Differentially Private Unordered Top-K Selection as an adaption of the algorithm from Gillenwater et al. (2022) to sample these multiple prototypes jointly using the exponential mechanism.

Differentially Private Unordered Top-K Selection Let $(u_1, \ldots, u_n)$ be the utilities of the public samples in decreasing order and K the number of prototypes to select. Since the order of the prototypes is not important in this context, we define the utility $U(\mathbf{X}, S)$ of a set of prototypes $S = \{u_{S_1}, \cdots, u_{S_K}\}$ w.r.t. the private datasets $\mathbf{X}$ as

$$U(\mathbf{X}, S) = \begin{cases} -u_K + \min_{k \in [K]} u_{s_k} & \text{if } s_1, \ldots, s_K \\ & \text{are distinct.} \\ -\infty & \text{otherwise} \end{cases} \tag{8}$$

Lemma 4 $\Delta U = \Delta u$.

Proof. The choice of $-\infty$ for repeating sequences does not depend on the private data and therefore doesn't affect the sensitivity. Furthermore, the utility of a set is only dependent on the lowest utility in the set $u_{\min}$ and the Kth true best utility u_K. The utility of a set can thus be formulated as $U(\mathbf{X}, S) = u_K - u_{\min}$. u is monotonic and has sensitivity Δu, in other words, insertion of a private sample can only increase each utility u by a maximum of Δu. It follows that insertion or removal of a private sample can only change U by $\pm\Delta u$, *i.e.*, $\Delta U = \Delta u$

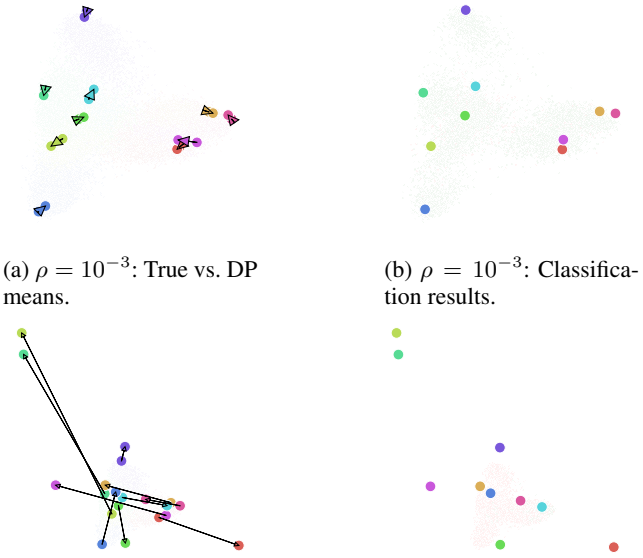
Therefore, each set has the utility of its worst entry, unless two entries repeat, in which case the utility is $-\infty$ and such set therefore never selected. If we select the true K-best prototypes, the utility is 0 and otherwise it's negative. Each utility is not unique. Instead, a utility can occur as many times as the number of possible combinations of samples with a higher utility. Given a utility u_y, we can therefore obtain the number of possible sets with that utility

$$m(u_y) = \binom{y}{K} \tag{9}$$

The entire algorithm then consists of

1. privately sampling a utility u_y with $P[EM(x) = y] \propto \binom{y}{K} \exp \frac{\epsilon u_y}{2\Delta U}$,
2. fixing the corresponding $\hat{\mathbf{x}}_y$ as part of the set, and
3. uniformly sampling the remaining $K-1$ prototypes without replacement, s.t. $\{\hat{\mathbf{x}}_i | u_i \geq u_y\}$.

Note that while Lemma 4 implies the sensitivity of u and U are the same, our effective privacy costs still double, since U is no longer monotonic. We perform the sampling using Proposition 5 from Medina and Gillenwater (2021).



(a) $\rho = 10^{-3}$: True vs. DP means.

(b) $\rho = 10^{-3}$: Classification results.

(c) $\rho = 10^{-4}$: True vs. DP means.

(d) $\rho = 10^{-4}$: Classification results.

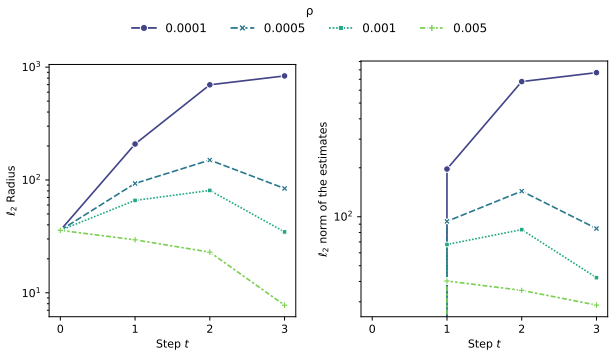
Figure 12: **Visualizing the mean estimation on CIFAR10.** We estimate the means using *CoinPress* (Biswas et al. 2020) at $\rho \in \{10^{-3}, 10^{-4}\}$. On the left, we show the non-private means and connect them with arrows to the privately estimated ones on top of the train set. Colors indicate the classes. On the right, we show the privately estimated means and the test set. Green points represent correctly classified samples, red points misclassified ones.

C.4 Classification with Multiple Prototypes

Given K multiple prototypes $\mathbf{P}_c \in \mathcal{R}^{K \times d}$ for each class $c \in [C]$, we classify $\mathbf{x}$ based on the minimum average distance

$$f(\mathbf{x}) = \arg\min_c \frac{1}{K} \sum_{i=1}^{K} d(\mathbf{x}, \mathbf{p}_{c,k}). \tag{10}$$

Results We sweep over $K \in [1, 2, 3, 5, 10, 20]$ and sort by balanced train accuracy to find the optimal K per privacy value, which Figure 19 shows. As we increase the privacy budget, the optimal accuracy is achieved at increasing K's. For $\rho > 100$ all K_{optimal} converge to $K = 10$. In this privacy regime, the Top-K selection behaves essentially as it would non-privately. In this case, it seems to be detrimental to pick too many prototypes, although we note the accuracy of $K \in \{5, 10, 20\}$ is almost on par, being 76.7%, 77.1% and 77.2% respectively for $\rho = 128$. Figure 14 shows the method requires a privacy budget somewhere between `DPPL-Mean` and `DPPL-Public`, while



(a) Radius. (b) Norm of the estimation.

Figure 13: **Analyzing the steps of *CoinPress*.** We estimate the means using *CoinPress* (Biswas et al. 2020) for different ρ. We see that for low values of ρ, the radius and the estimates diverge.

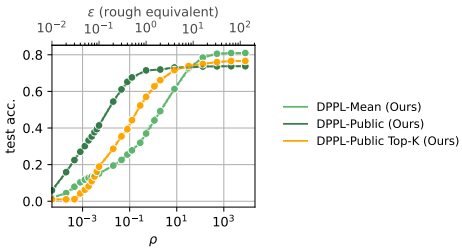


Figure 14: **Top-K**.

having a maximum accuracy also in between the maximum accuracy of those methods, closer to the higher accuracy of DPPL-Mean. We show the full set of results in Figures 15 to 18.

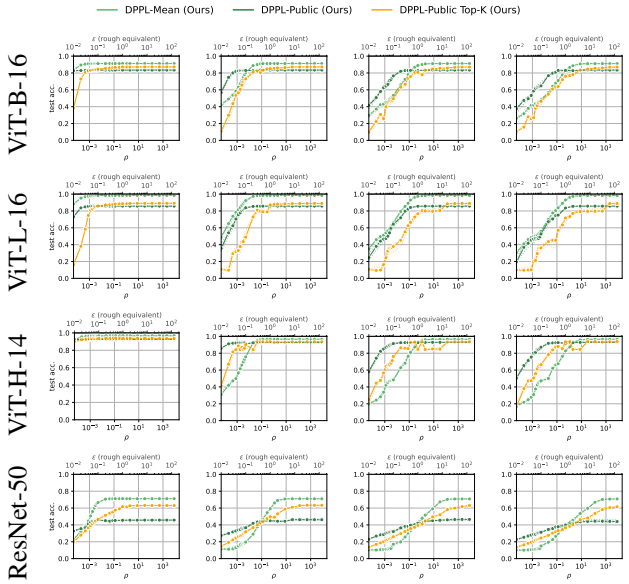
C.5 Sampling Mechanisms

We compare two different sampling mechanisms for DPPL-Public and find that both commonly utilized mechanisms, the Laplace and Exponential mechanism, produce very similar results. We show accuracies for various datasets and privacy budgets in Appendix C.5. In fact, if the Laplace mechanism is used with a Gumbel noise distribution, the output distribution of the Laplace mechanism and Exponential mechanism are identical. This is commonly referred to as the *Gumbel max trick* (Rogers and Steinke 2021).

We utilize the Exponential mechanism for its additional flexibility, which allows us to build DPPL-PublicK as described in Appendix C.3.

C.6 Projection

Setup The projection consists of a single layer linear network $f : \mathbb{R}^{d_{\text{avg}}} \to \mathbb{R}^{d_p}$ with no activation function and an average pooling layer with kernel size $p_{\text{before}} \in [1, 64]$ before the linear layer. As we leave the total privacy budget ρ unchanged, we introduce a hyperparameter $s \in [0.1, 0.9]$



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 15: **DPPL-Public with Top-K.** We present the results including DPPL-Public Top-K of CIFAR10 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for DPPL-Public, at different levels of imbalance rations (IR).

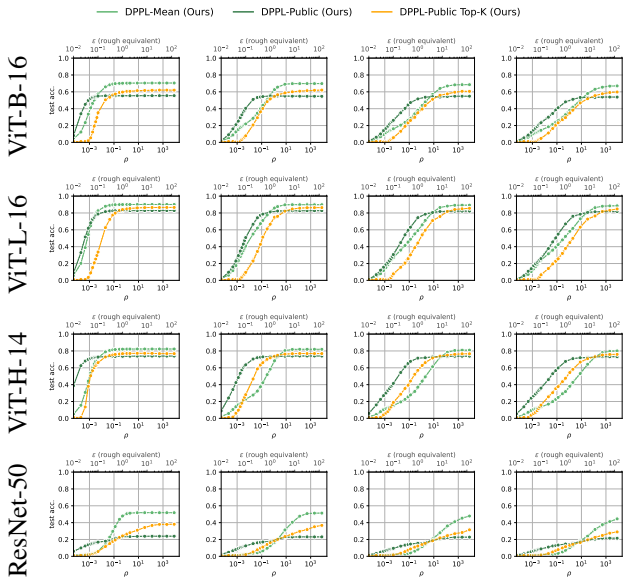
	Method $\backslash \rho$	0.005	0.02	0.125	0.5
CIFAR10	Exponential	91.45	91.33	91.27	91.25
	Laplace	91.45	91.33	91.28	91.26
CIFAR100	Exponential	36.36	70.47	72.22	72.75
	Laplace	35.89	70.48	72.26	72.72
Food101	Exponential	19.09	53.71	59.19	66.18
	Laplace	18.82	53.77	59.21	66.20

Table 2: **Comparison between Noisy Max (Laplace Mechanism) and Exponential Mechanism for sampling prototypes.**

which defines the privacy budget of the projection layer $\rho_l = s * \rho$ and of the prototype estimation $\rho_p = (1-s) * \rho$. In total, the hyperparameters are $s \in [0.1, 0.9]$, $p_{\text{before}} \in [1, 64]$, output dimension d_p, the number of augments per step n, batch-size, learning-rate, gradient clipping norm and number of training steps.

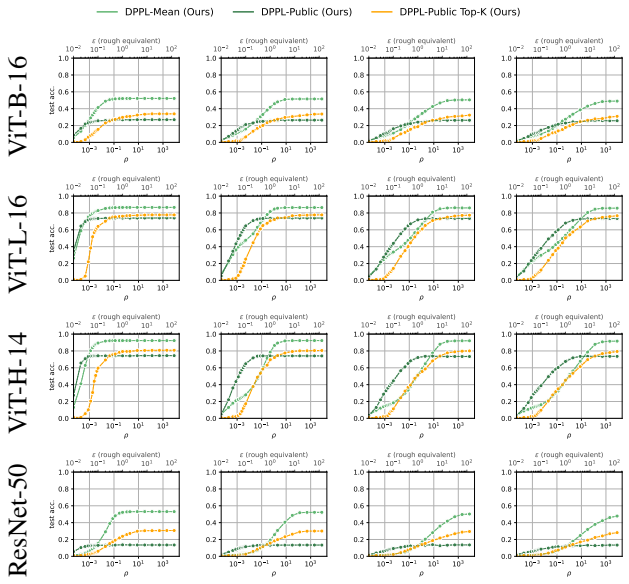
We train the projection with the original training rule from Snell, Swersky, and Zemel (2017), with some adaptions for privacy. We perform Bernoulli sampling (often more generally referred to as Poisson sampling) to receive a batch $B = (\mathbf{X}, \mathbf{y})$. We split $(\mathbf{X}, \mathbf{y})$ evenly into support and query $(\mathbf{X}_S, \mathbf{y}_S), (\mathbf{X}_Q, \mathbf{y}_Q)$, s.t. each part has the same number of samples per class. If a class has only one corresponding sample in $\mathbf{X}$, we drop it. The prototypes for each class $\mathbf{p}_c$ are estimated as the mean of samples in the support set $\mathbf{X}_S$ that have the corresponding class label in $\mathbf{y}_S$.

$$\mathbf{X}_{S,c} = \{\mathbf{x}_i \in \mathbf{X}_S | y_i = c\} \tag{11}$$



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 16: **DPPL-Public with Top-K.** We present the results including `DPPL-Public` Top-K of CIFAR100 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR).



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 17: **DPPL-Public with Top-K.** We present the results including `DPPL-Public` Top-K of FOOD101 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR).

$$\mathbf{p}_c = \frac{1}{|\mathbf{X}_{S,c}|} \sum_{\mathbf{x} \in \mathbf{X}_{S,c}} \mathbf{x} \tag{12}$$

Then, prototypes and query samples are projected with the linear layer.

$$\mathbf{X}'_Q = \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}_Q\} \tag{13}$$

$$\mathbf{p}'_c = f(\mathbf{p}_c) \tag{14}$$

Finally, the model aims to classify each sample in $\mathbf{X}'_Q$ by assigning it the label of the closest prototype

$$\hat{\mathbf{y}} = \{\arg\min_c d(\mathbf{p}'_c, \mathbf{x}') | \mathbf{x}' \in \mathbf{X}'_Q\} \tag{15}$$

We implement the classification training using a log-softmax over the distances to the prototypes and the negative log likelihood loss. This entire process, beginning with the split of B, is repeated n times, before aggregating the loss and conducting the private gradient descent on the projection layer weights.

DPPL-Mean For `DPPL-Mean` we expected the reduction in dimensionality to potentially improve to utility-privacy-tradeoff, but instead higher dimensions were strictly better, leading to the removal of p_{before} and d_p as hyperparameters. We show in Figure 20 the distribution of accuracies during the hyperparameter optimization and that no configuration reached the performance without projection.

DPPL-Public For `DPPL-Public`, we additionally need to project the public data embeddings, to find the prototypes in the projected latent space. Furthermore, we found that the utility is strictly lower than without projection. It seems that what the projection layers learns is fundamentally misaligned with the actual task. An obvious mismatch between training and application of the model is that we take the means as prototypes $\mathbf{p}_c$ during training, but we later pick these prototypes from public data. Even after accounting for this, and using the actual prototypes during the projection training, utility didn't improve.

For both methods, the utility with projection is always worse Figure 7 shows. The necessary accounting for the privacy costs of optimizing the additional hyperparameters (Papernot and Steinke 2021) would further reduce the utility.

D Privacy Proofs and Privacy Conversion

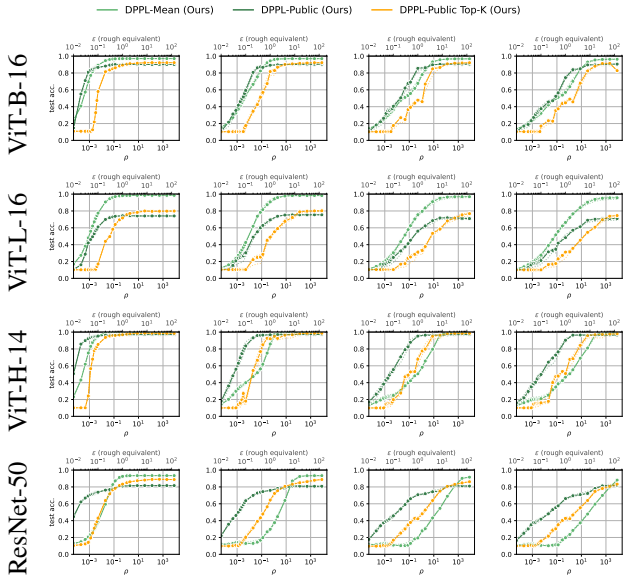
D.1 Full Proof for Privacy Guarantees of DPPL-Public

We recall that our utility function is

$$u(\hat{\mathbf{x}}, c) = \sum_{i=0}^{|\mathbf{X}_c|} 1 + \frac{\hat{M}(\mathbf{x}_i) \cdot \hat{M}(\hat{\mathbf{x}})}{\|\hat{M}(\mathbf{x}_i)\| \|\hat{M}(\hat{\mathbf{x}})\|} \tag{16}$$

where $\mathbf{X}_c \in \mathbf{X}$ are disjoint subsets of the private data $\mathbf{X}$ we want to keep private.

We choose u to be the sum and not, for example, the mean of the cosine similarity, to make u monotonic w.r.t. $\mathbf{X}$. It can be easily verified that the two different utility functions (mean and sum) lead to an identical mechanism, since the changes in Δu and u cancel each other out. As we exhaust the full range $[0, 2]$ of the cosine similarities, we clip each



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 18: **DPPL-Public with Top-K.** We present the results including `DPPL-Public` Top-K of STL10 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR).

similarity to $[d_{\min}, d_{\max}]$ and then substract $d_{\min}$. This gives us the adapted utility function

$$u(\hat{\mathbf{x}}, c) = \sum_{i=0}^{|\mathbf{X}_c|} \text{clip}\left(1 + \frac{\hat{M}(\mathbf{x}_i) \cdot \hat{M}(\hat{\mathbf{x}})}{\|\hat{M}(\mathbf{x}_i)\|\|\hat{M}(\hat{\mathbf{x}})\|}, d_{\min}, d_{\max}\right) - d_{\min} \tag{17}$$

Lemma 5 *$\Delta u = d_{max} - d_{min}$ and u is positively monotonic w.r.t. to $\mathbf{X}$.*

Proof. Since the cosine similarity's range is bound to $[0, d_{\max} - d_{\min}]$, each private sample contributes one non-negative summand in $[0, d_{\max} - d_{\min}]$. It immediately follows that $\Delta u = d_{\max} - d_{\min}$ and u is positively monotonic w.r.t. $\mathbf{X}$.

Theorem 3 *`DPPL-Public` is ε-DP.*

Proof. We sample the public prototypes independently for each class, using a utility function on disjoint sets $\mathbf{X}_c \in \mathbf{X}$ (each training data point only has one label), s.t. parallel composition applies. Each class prototype is sampled with the exponential mechanism, with probability $\Pr[\hat{\mathbf{x}}] \propto \exp\left(\epsilon u_{(\hat{\mathbf{x}},c)}/\Delta u\right)$ for outputting $\hat{\mathbf{x}}$ as the class prototype, with Δu denoting the sensitivity of $u_{(\hat{\mathbf{x}},c)}$. Our utility function is monotonic (Lemma 5) and the described exponential mechanism is ε-DP for monotonic utility functions (Lemma 2). Since parallel composition applies and each parallel algorithm is ε-DP, the overall algorithm is ε-DP.

D.2 Comparing Between Different Notions of DP

Note that to fairly compare our developed method that yields *pure DP* guarantees against related work that yield zCDP,

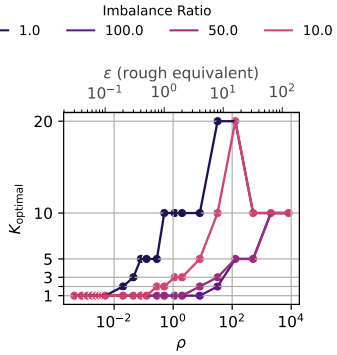


Figure 19: **Optimal K for Top-K**

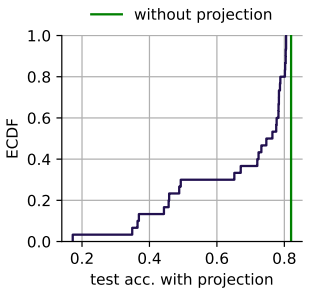


Figure 20: Results of the hyperparameter sweep of `DPPL-Mean` with projection.

we convert our method using Lemma 3. To obtain a ρ-zCDP guarantee for the Linear Probing and DPSGD-Global-Adapt baselines, we perform full batch training and obtain the guarantee from Theorem 2.

E Additional Experimental Results

E.1 Imbalanced Experiments

We compare the accuracies for all methods, all encoders and imbalance ratios in $[1, 10, 50, 100]$ in Figures 21 to 24.

E.2 Minority Class Accuracies

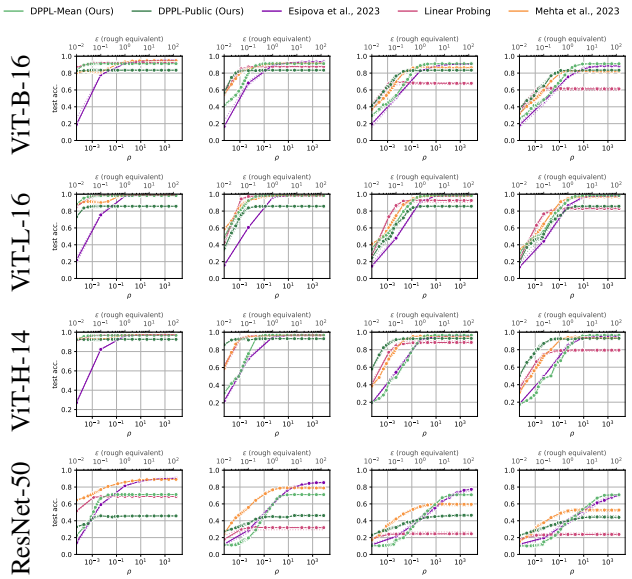
We compare the accuracies of the minority classes for all methods, all encoders and imbalance ratios in $[1, 10, 50, 100]$ in Figures 25 to 28.

E.3 Computational Runtimes

We compare the runtime for a single training in Appendix E.3. We chose to compare CIFAR10 and CIFAR100 because the runtime of all methods scale with the number of classes on otherwise equally large training datasets.

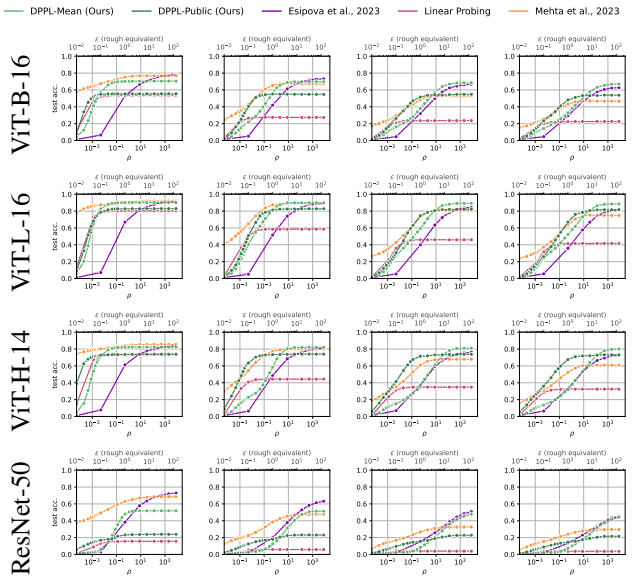
E.4 Potential Baselines

We further considered DP-FC introduced by Mehta et al. (2023) and DP-FiLM introduced by Tobaben et al. (2023) as baseline methods.



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 21: **DP Prototypes on CIFAR10.** We present the results for CIFAR10 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023). Plotted is the median over multiple runs and dotted lines represent the upper and lower quantiles for all methods.



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 22: **DP Prototypes on CIFAR100.** We present the results for CIFAR100 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023). Plotted is the median over multiple runs and dotted lines represent the upper and lower quantiles for all methods.

DP-FiLM While DP-FiLM exhibits strong learning potential from few samples, it is an iterative algorithm and comes with the same drawbacks for unbalanced tasks as DP-SGD. We conducted initial experiments for which we show the results in Table 4. We compare DP-FiLM on ViT-H-14 on CIFAR10 and CIFAR100. Both methods were trained at the same value of ϵ. We set δ for DP-FiLM to $1/2n$ where n is the number of training samples. Our method provides $\delta = 0$ pure DP. We note the significantly lower utility of DP-FiLM in imbalanced cases and considering the high computational costs — other methods require training in the range 10^{-1} to 10^{2} GPU-seconds, whereas DP-FiLM requires 10^{5} to 10^{6} GPU-seconds— decided against a comprehensive comparison.

DP-FC DP-FC is an iterative optimization algorithm that integrates second order information by utilizing the covariance of the features. Figure 29 shows results on imbalanced datasets. While DP-FC slightly outperforms DP-LS for very strict privacy budgets in balanced cases, it exhibits the same disparate effects on minority classes as DP-SGD, resulting in reduced utility for imbalanced cases. Given that it has only a small advantage in balanced cases and otherwise large disadvantage in unbalanced cases, we focused on DP-LS as a non-iterative baseline instead.

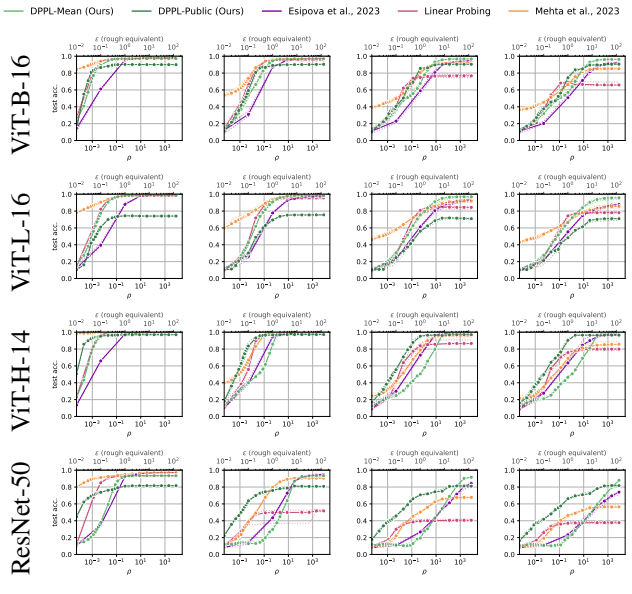
F Discussion

F.1 Broader Impacts

We expect the prevalence of machine learning and it's impact on society to ever increase. Our methods are especially useful at preserving the privacy of the training data, data that often consists of sensitive data from users. We consider contributing to an increase in the privacy of the training data and therefore protecting the users that contribute data to machine learning models to be a positive societal impact. Furthermore, our methods especially address the use case of imbalanced datasets. Real-world data is often long-tailed and models trained on unbalanced data can lead to unfair decisions w.r.t. to gender, ethnicity, disabilities, religion or social status, especially for minorities. We consider contributing to an improvement of the utility for minority classes as outlined in Figure 5 and Appendix E.2 to be a positive societal impact.

F.2 Limitations

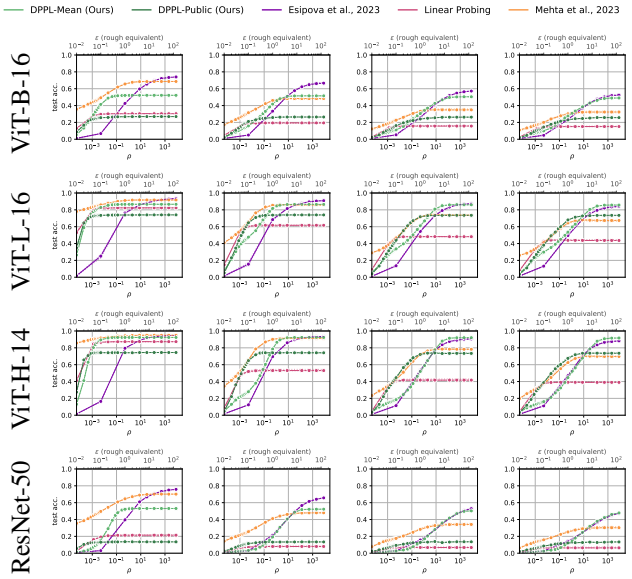
We introduce `DPPL` as a novel approach to private transfer learning. Like all transfer learning methods, our method depends on a suitable base model. We've seen that especially ResNet-50 poses significant challenges, while the vision transformers worked well. The largest vision transformer ViT-H-14 yielded the best results of the compared models. We note that the combination of less suitable base models



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 23: **DP Prototypes on STL10.** We present the results for STL10 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023). Plotted is the median over multiple runs and dotted lines represent the upper and lower quantiles for all methods.

in addition to further out-of-distribution tasks, relative to the pre-training data, has a larger negative effect on the performance of our method compared to other methods. When evaluating the most out-of-distribution dataset, FOOD101, in combination with using embeddings from ResNet-50 or ViT-B-16, our methods are outperformed (see Figure 24. We can still claim the highest accuracy for minority classes in that case (see Figure 28), although the significance of that given the low utility is questionable. As we didn't include the projection layer for our methods, the ability to adapt to these distribution shifts is limited and possibilities to include it need to be investigated further.



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

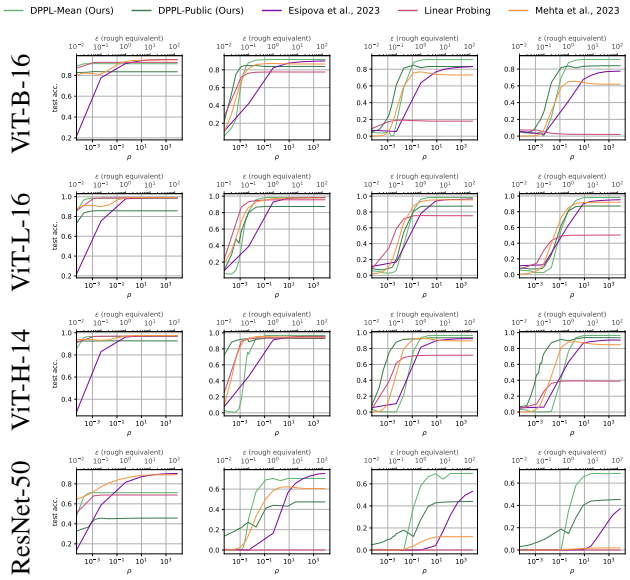
Figure 24: **DP Prototypes on FOOD101.** We present the results for FOOD101 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023). Plotted is the median over multiple runs and dotted lines represent the upper and lower quantiles for all methods.

	Step	Runtime [s] CIFAR10	CIFAR100
`DPPL-Mean` (Ours)	Mean Estimation	0.079	0.168
`DPPL-Public` (Ours)	Utility Calculation	5.0	34.3
	Private Sampling	0.0003	0.14
Linear Probing (DPSGD)	Iterative Training	5.49	6.3
Esipova et al., 2023 (DPSGD-Global-Adapt)	Iterative Training	174	242
Mehta et al., 2023 (DP-LS)	Setup	0.49	2.2
	Solving	0.28	2.5

Table 3: **Computational wall-time measurements** of a single training on a single machine, limiting each method to a single GPU. Where applicable, iterative training was limited to 15 epochs. `DPPL-Public`'s score computation was conducted for 1,281,167 public samples.

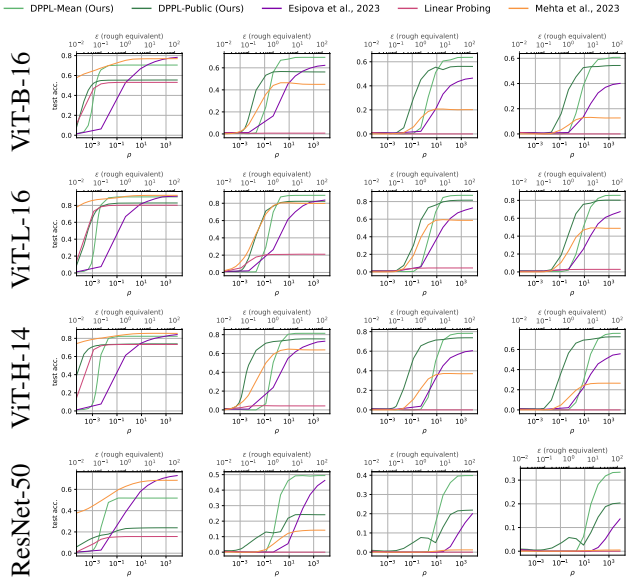
Dataset	Method ϵ	0.1	0.5	1.0	2.0
CIFAR10	`DPPL-Public`	66.1	92.9	92.5	92.9
	DP-FiLM	34.1	63.3	67.7	84.4
CIFAR100	`DPPL-Public`	27.4	51.2	59.9	70.0
	DP-FiLM	4.8	20.1	34.1	45.2

Table 4: **DP-FiLM vs. DPPL-Public** using ImageNet-1K as public data on CIFAR10 and CIFAR100 with imbalance ratio 100. We compare at the same ϵ value, although DP-FiLM provides approximate DP and our method provides more strict pure DP.



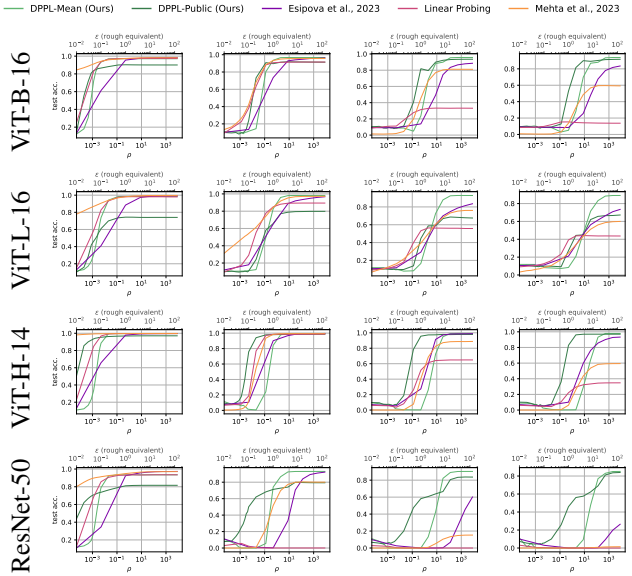
(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 25: **Minority class accuracies on CIFAR10.** We present the results for the minority classes (lower 25% quantile) of CIFAR10 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023).



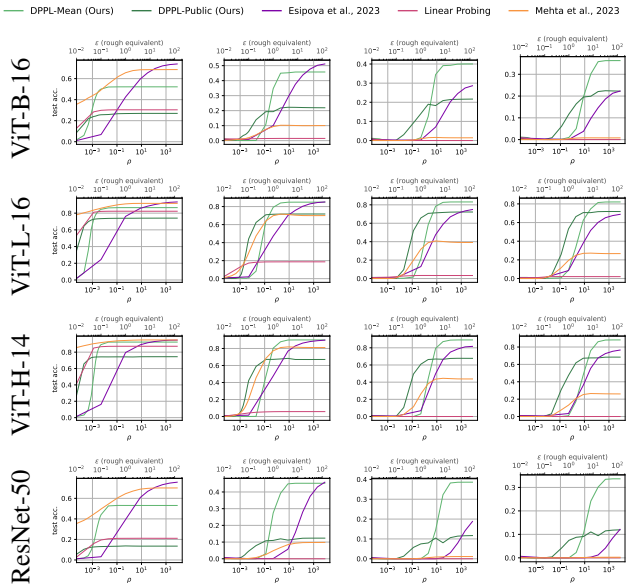
(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 26: **Minority class accuracies on CIFAR100.** We present the results for the minority classes (lower 25% quantile) of CIFAR100 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023).



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 27: **Minority class accuracies on STL10.** We present the results for the minority classes (lower 25% quantile) of STL10 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023).



(a) IR=1. (b) IR=10. (c) IR=50. (d) IR=100.

Figure 28: **Minority class accuracies on FOOD101.** We present the results for the minority classes (lower 25% quantile) of FOOD101 on ViT-B-16, ViT-H-14, ViT-L-16 and ResNet-50, using ImageNet as public data for `DPPL-Public`, at different levels of imbalance rations (IR). We compare to DP-LS by Mehta et al. (2023) and DPSGD-Global-Adapt by Esipova et al. (2023).

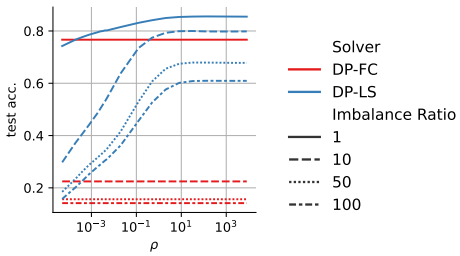


Figure 29: **DP-LS vs. DP-FC**.