Rethinking Benchmarks for Differentially Private Image Classification*

Sabrina Mokhtari
University of Waterloo
s4mokhtari@uwaterloo.ca

Sara Kodeiri
University of Waterloo
skodeiri@uwaterloo.ca

Shubhankar Mohapatra
University of Waterloo
s3mohapa@uwaterloo.ca

Florian Tramèr
ETH Zürich
florian.tramer@inf.ethz.ch

Gautam Kamath
University of Waterloo
g@csail.mit.edu

Abstract

We revisit benchmarks for differentially private image classification. We suggest a comprehensive set of benchmarks, allowing researchers to evaluate techniques for differentially private machine learning in a variety of settings, including with and without additional data, in convex settings, and on a variety of qualitatively different datasets. We further test established techniques on these benchmarks in order to see which ideas remain effective in different settings. Finally, we create a publicly available leader board for the community to track progress in differentially private machine learning.

1 Introduction

Machine learning (ML) models have been repeatedly demonstrated to leak sensitive information pertaining to their training data. These issues manifest through a number of different types of attacks, including membership inference [31, 56], model inversion [26], and even training data extraction [13, 14, 57]. This can be problematic if the training data contains privacy-sensitive information belonging to people. To alleviate such concerns, a popular solution is differential privacy (DP) [23]. DP is a rigorous notion of individual data privacy, which can be used to mask the presence or absence of any single training data point when observing a trained model. In particular, training a model with DP provably prevents all the aforementioned attacks.

The past decade has seen significant effort and success in training ML models with DP, including image classifiers [1, 18, 48, 62], large language models [2, 44, 69], and other generative models [6, 8, 12, 20, 29, 67]. However, in a recent position paper, Tramèr, Kamath, and Carlini critique a number of trends in DP ML [63]. Most pertinent to our work, they question whether benchmarks used in DP ML are truly measuring progress in the field, specifically in the context of DP image classification, which will be our focus. The most common benchmark datasets used in DP image classification include MNIST [43],

*Authors SMok, SK, and SMoh have equal contributions and are listed alphabetically in order of *first* name. Authors FT and GK are listed in reverse alphabetical order.

CIFAR-10 [41], and ImageNet [19]. While significant progress has been made on each, TKC question whether this progress generalizes to privacy-sensitive settings where DP may be deployed. For example, CIFAR-10 and ImageNet are both composed primarily of natural images of everyday objects. While these datasets indeed have some privacy concerns [9], it is less clear whether they resemble domains where DP is of high practical concern, such as, e.g., medical images. Since, informally speaking, medical images appear to qualitatively differ from those in the aforementioned datasets, it is unclear whether techniques previously established to be effective remain so in these settings. This question is even more pronounced when models are pre-trained on public data (i.e., supplementary data which is not subject to any privacy constraints), a popular trend in private ML. In such settings, the chosen "public" datasets are often visually similar to the private ones – as a representative example, [18] treat ImageNet as public and privately fine-tune on CIFAR-10. On the other hand, for domains such as medical images, private images may be specialized and ill-represented in public pre-training datasets. Finally, further muddying the waters is the fact that results on these benchmark datasets are often reported for incomparable settings, in particular, with vastly differing public pre-training datasets. Overall, these issues make it difficult to isolate which ideas and techniques are truly effective in privacy-critical settings.

Our contributions are as follows:

- We propose standardized benchmark datasets and evaluation settings to measure progress in DP image classification, with a particular focus on privacy-sensitive domains;
- We release a public leaderboard for DP ML, for the community to track improvements on these benchmarks;
- We evaluate previously established techniques for DP image classification across a variety of settings to see which are and are not broadly effective.

2 Preliminaries

We recall the celebrated notion of differential privacy.

Definition 1 ([22, 23]) *An algorithm $M : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-differentially private if, for all neighboring datasets (i.e., datasets that differ in exactly one entry) X and X' and all events $S \subseteq \mathcal{Y}$, we have that* $\Pr[M(X) \in S] \leq e^{\varepsilon} \Pr[M(X') \in S] + \delta$.

DP is a quantitative definition of individual data privacy. The privacy cost is measured by the parameters (ϵ, δ), also called the privacy budget. Smaller values of ϵ correspond to stricter privacy guarantees, and it is standard in the literature to set $\delta \ll \frac{1}{n}$, where n is the size of the database. Complex DP algorithms can be built from the basic algorithms following two important properties of differential privacy: 1) Post-processing states that for any function g defined over the output of the mechanism $\mathcal{M}$, if $\mathcal{M}$ satisfies (ϵ, δ)-DP, so does $g(\mathcal{M})$; 2) Basic composition states that if for each $i \in [k]$, mechanism $\mathcal{M}_i$ satisfies (ϵ_i, δ_i)-DP, then a mechanism sequentially applying $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ satisfies $(\sum_{i=1}^k \epsilon_i, \sum_{i=1}^k \delta_i)$-DP.

Given a function $f : \mathcal{D} \to \mathbb{R}^d$, the *Gaussian mechanism* adds noise drawn from a normal distribution $\mathcal{N}(0, S_f^2 \sigma^2)$ to each dimension of the output, where S_f is the ℓ_2-sensitivity of f, defined as $S_f = \max_{D,D' \text{differ in a row}} \|f(D) - f(D')\|_2$. For $\epsilon \in (0, 1)$, if $\sigma \geq \sqrt{2 \ln(1.25/\delta)}/\epsilon$, then the Gaussian mechanism satisfies (ϵ, δ)-DP.

We focus on training ML models subject to DP, which (due to its post-processing property) allows the trained model to be publicly released without further privacy concerns. The most popular method for DP training of ML models is differentially private stochastic gradient descent (DPSGD) [1, 5, 58]. In contrast to non-private SGD where batches are sliced from the training dataset, DPSGD at each

iteration works by sampling "lots" from the training with probability L/n, where L is the (expected) lot size and n is the total data size. A set of queries are computed over those samples. These queries include gradient computation, updates to batch normalization, or accuracy metric calculations. As there is no a priori bound on these query outputs, the sensitivity S_f is set by clipping the maximum ℓ_2 norm of the gradient to a user-defined parameter C. The gradient of each point is then noised and published. All DP optimizers follow the same framework in which they take steps on the computed noisy gradient as in its non-private counterpart. The privacy cost of the whole training procedure is calculated using privacy accounting techniques. We discuss the specifics of DPSGD for our experiments in Section 3.

3 Benchmark design

In this section, we report our specific prescriptions for benchmarks, including datasets, parameters, and best practices, in a variety of settings, in order to standardize and (ideally) propel progress in DP image classification in privacy-critical settings. We note that we (intentionally) do not introduce any new datasets, and instead appeal to existing ones. This is because using established datasets allows for easier comparisons between the private and non-private setting, and introducing an entirely new dataset would serve no benefit for our setting.

Datasets We prescribe using the following two medical image datasets (which have been commonly used in other areas of machine learning) as benchmarks for DP ML: a) CheXpert [33], a chest X-ray dataset; and b) EyePACS [25], a diabetic retinopathy dataset. These datasets are primarily chosen due their privacy-critical domain. We hope that progress on these benchmarks would align with progress (i.e., increased utility) on truly private tasks in such settings. Secondarily, we choose these datasets due to diversity in their sizes, balance of classes, and in the case of CheXpert, for inclusion of a multilabel dataset. Further description of these datasets and justification of these choices appears in Section 4. In addition, we recommend continuing to use CIFAR-10 [41] and ImageNet [19] as benchmarks for training DP ML models *from scratch*, without any pretraining data. Indeed, keeping the caveats of [63] in mind, the popularity of these datasets still allows for direct comparison of accuracy on these tasks, and thus to track "how far behind" DP ML is behind the non-private setting.

Public datasets One of the most successful ways to improve the utility of DP ML has been pre-training the model on "public" data (i.e., data free of any privacy constraints). As discussed by [63], the size and nature of the pre-training data can dramatically affect the downstream utility of a privately fine-tuned model. Therefore, for fair comparison between different techniques, we prescribe tracking progress with the following datasets treated as public: a) no public data, for the "purest" measure of progress in DP ML; b) ImageNet-1K, perhaps the most commonly used large image classification dataset c) LAION-2B, due to it being the pre-training data for OpenCLIP's ViT-G/14 (representing the common use-case of privately fine-tuning a pre-trained CLIP model), and d) "anything goes." To elaborate on the last of these, we use "anything goes" to refer to the case when public pre-training data is unrestricted (barring data-leakage-like considerations where the private dataset contaminates the public one): it may include large-scale Internet datasets, additional domain-specific data, etc. As mentioned before, results in this category may not be directly comparable with each other. Nonetheless, they serve as a measure of absolute progress on a benchmark.

Privacy parameters It is not clear how to compare results on DP image classification at varying levels of the privacy parameters ε and δ. For example, is 90% accuracy at $\varepsilon = 1$ better or worse than 95% at $\varepsilon = 2$? We propose fixing the value of ε to be $1, 3, 5$ and 8 to facilitate direct comparisons between

results. This set of ε covers both high and low privacy regimes across the range usually considered in DP ML. We additionally propose fixing δ to to be the largest power of 10 that is at most the inverse of the training set size (consistent with previous parameter settings), though in many parameter regimes, δ can be dramatically increased or decreased with minor effect on the value of ε.

Privacy accounting. Every DP algorithm is associated with a proof of privacy, which provides an upper bound on the value of ε and δ. For DPSGD, this is generally automated using "privacy accountants," which take as input various hyperparameters and δ, and outputs the value of ε. Over time, improved accounting methods have given increasingly tight analyses, culminating in "exact" privacy accounting techniques [1, 28, 40, 45, 46]. However, as highlighted by some recent works [17, 42, 51], simply using a tighter accountant may give the illusion of an improved result, even if the training procedure is identical. Therefore, we recommend that the privacy accounting method (or, if not using DPSGD, the specific proof followed) is reported in order to keep track of such discrepancies (ideally, all future DPSGD works ought to use exact privacy accountants).

Applicable techniques. The most popular algorithm for DP ML is DPSGD [1, 5, 58], in part due to its flexibility: it can be used to privately train any differentiable model, even non-convex ones. Other methods, such as objective perturbation [15, 34, 38, 54], are usually applicable only to convex models. Consequently, in addition to several non-convex settings, we suggest some standardized convex settings so that a wider variety of methods may be compared and evaluated. We recommend linear probe (i.e., logistic regression) on a) Wide ResNet-28-10 pre-trained on ImageNet-1K;[1] and b) OpenCLIP's ViT-G/14 pre-trained on LAION-2B.

"Anything goes" zero-shot Parallel to the literature on DP ML, the general ML community has studied the challenging "zero-shot" setting, in which goal is to correctly classify a test image without seeing a single image in its training set. Naturally, this requires large-scale public pre-training to achieve acceptable results. In terms of DP, this corresponds to $\varepsilon = 0$ but with "anything goes" pre-training (described above). We suggest tracking the current SOTA for such settings, as a) they serve as an important measure of absolute progress on benchmarks; and b) it is otherwise easy to report a DP result with "anything goes" public data and $\varepsilon > 0$ as SOTA, despite being already dominated by existing zero-shot results.

Overall, we remind that our community's goal ought *not* be to get the highest numbers on these specific datasets, but instead to improve our techniques and understanding of DP image classification for settings that may generalize to those used in practice. We thus focus on a breadth of settings to hopefully cover a range of conditions in which DP classifiers may be deployed. Even if a model can achieve high utility on a benchmark in the "anything goes" zero-shot setting, this does not mean the problem is necessarily "solved." For instance, due to legal, ethical, computational, or safety reasons, depending on the specific setting, it may not be possible to use large, uncurated public datasets for pre-training in a real-world deployment. Therefore, we consider all settings outlined above to be of potential practical or technical interest, and do not identify any of them as "canonical" or more important than another.

3.1 Leaderboard

Tracking progress on benchmark datasets via leaderboards is an established practice in (non-private) ML.[2] This is not yet the case for DP ML: a broad and up-to-date knowledge of the literature is required

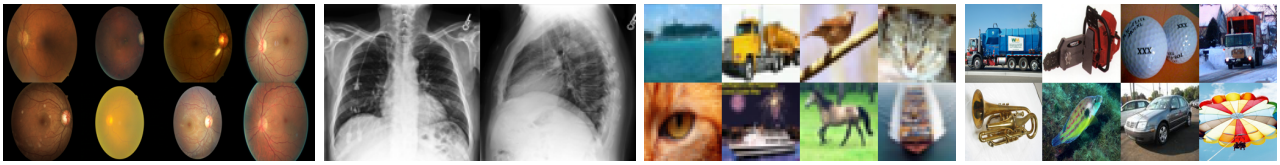
[1] Inspired by [18]. While they release their weights in JAX, we release comparable PyTorch weights with the code https://github.com/mshubhankar/DP-Benchmarks.

[2] See, e.g., `https://paperswithcode.com/sota`

to keep track of the latest results, making entering the field especially challenging and intimidating for newcomers. As one of our contributions, we alleviate this issue by creating and maintaining a leaderboard for DP ML.[3]

Due to the particulars of the DP setting, it is unnatural to simply incorporate results into an existing leaderboard for the non-private setting. Specifically, beyond just the specific dataset, a leaderboard for DP ML would need to track many of the considerations already discussed, including the privacy parameters (ε, δ), which privacy accountant was used, and which public datasets were used. Another difference from the non-private setting is the issue of *correctness*. For a proposed algorithm, the DP guarantees must be mathematically *proven*, and a claimed result could be false if there is a bug in the proof. This is in addition to existing concerns from the non-private setting on whether results are independently reproducible or not. However, since it is notoriously easy to have bugs in a proof of DP, we incorporate a *verification* system to our leaderboard. By default, all results are unverified when added. However, anyone is able to submit a pull request to our GitHub to verify that they reproduced the result, and believe correctness of the privacy proof (if applicable).

At present, our leaderboard focuses exclusively on DP image classification (as does this paper), though it may be extended to other problems (e.g., DP natural language understanding or generation).



(a) EyePACS (b) CheXpert (c) CIFAR-10 (d) ImageNet

Figure 1: EyePACS and CheXpert qualitatively look different than common benchmark datasets such as CIFAR-10 and ImageNet.

4 Datasets and Architectures

Here, we describe the relevant datasets and architectures, which are later explored in the experiment section.[4]

4.1 Overview of datasets

CheXpert The CheXpert dataset [33] has 224,316 chest X-ray images of size 390×320 from 64,540 patients. Images may have multiple labels, where the possible labels correspond to five pathology classes: 'Cardiomegaly', 'Edema', 'Consolidation', 'Atelectasis', and 'Pleural Effusion'. In our work, following prior state-of-the-art training, we re-scale all images to size 224×224 and augment the dataset using random affine transformations [70, 71].

EyePACS Kaggle EyePACS [25] contains retinal images of diverse populations with various degrees of diabetic retinopathy (DR). Each image is classified into one of five classes depending on the severity of the disease. The classification task is diagnostic of DR, as measured on a scale from 0 (no DR) to 4 (severe DR). The training set consists of 35,126 and the test set contains 53,576 color eye fundus images.

[3] Our leaderboard is available at https://private-machinelearning.github.io/

[4] Any omitted hyperparameter or architectural details appear in the code repository https://github.com/mshubhankar/DP-Benchmarks

To speak to these particular dataset selections: as mentioned before, we chose medical images to address a privacy-critical setting where DP may be deployed. Within this area, chest X-rays and fundus images are two of the most common domains, so we chose one of the most popular datasets from each of these domains. Additionally, we took guidance from [53], which also focuses on medical image classification, and studies CheXpert and a Google-proprietary DR dataset. While there are several public fundus photography datasets, most of them are very small (< 100 images) and thus not settings we would expect DP to function well: EyePACS is the most popular one of an acceptable size.

4.2 Overview of architectures and techniques

ScatterNets ScatterNets [47] (SN) are convolutional neural networks (CNNs) that utilize pre-defined wavelets for their architecture and filters. In other words, the features are "hand-crafted" rather than learned from data, and thus use neither public nor private data. Tramèr and Boneh [62] employ this architecture for DP image classification, using DPSGD to train either linear or convolutional layers acting on these features, and demonstrate compelling results on MNIST and CIFAR-10, particularly for small values of ε. We exclusively use ScatterNets without any public data.

Wide-ResNets The Wide-ResNet [72] (WRN) is a variant of the ResNet[30] that reduces issues of vanishing and exploding gradients by making the model wider instead of deeper. De et al. [18] use them to reach DP SOTA in multiple settings on CIFAR-10. They consider both DP training from scratch, and DP fine-tuning after being (publicly) pre-trained on ImageNet-1K (downsampled to 32×32, which we call IN-32 [16]).[5] To allow direct comparison, we emulate their setting as much as possible, e.g., using weight standardization [10], group normalization, and their choices of hyperparameters for pre-training. We use both without any public data, and pre-trained on ImageNet-1K.

Additionally, Tang et al. [60] utilize WRN-16-4 to achieve DP SOTA performance on CIFAR-10, when no extra public data is used for pretraining. They leverage image priors generated by random processes [3] instead of starting from random initialization, outperforming [62] and [18] when they only train from scratch. Moreover, they achieve SOTA performance using only a linear probe, making for a direct comparison to the linear ScatterNet method of [62]. We adopt the same architecture and replicate their settings to the greatest extent possible, incorporating techniques such as augmentation multiplicity and normalization. Tang et al. [60] build on the approach of De et al. [18] by using the third-to-last layer of the network, which has a dimension of 4096. We adopt a similar strategy but reduce the dimensionality to 2048. This adjustment is necessary due to the larger image sizes in our datasets (CheXpert and EyePACS with 224×224 images) compared to CIFAR-10 (32×32 images) and resource constraints.

CLIP-based models CLIP [52] is a popular contrastive learning pre-training technique, which allows one to jointly train a language and image encoder. CLIP has been observed to enable robust zero-shot image classification when pre-training on very large Internet datasets. We use two ViT [21] models pre-trained using CLIP: OpenAI's ViT-B/16 (pre-trained on the proprietary WebImageText (WIT) dataset) and OpenCLIP's ViT-G/14 model (pre-trained on LAION-2B [55]).[6] Besides pre-training data, these models differ in their size (12 and 48 layers, respectively) and patch size (16 and 14, respectively). For zero-shot experiments we use these models as-is, for DP fine-tuning experiments, we use only the image encoder as a feature extractor, and on top of that, apply either a linear layer (i.e., logistic

[5] They use WRN-16-4 and WRN-40-4 for from-scratch experiments and WRN-28-10 for fine-tuning experiments. For simplicity, we use WRN-28-10 in all our experiments.

[6] `https://github.com/mlfoundations/open_clip`

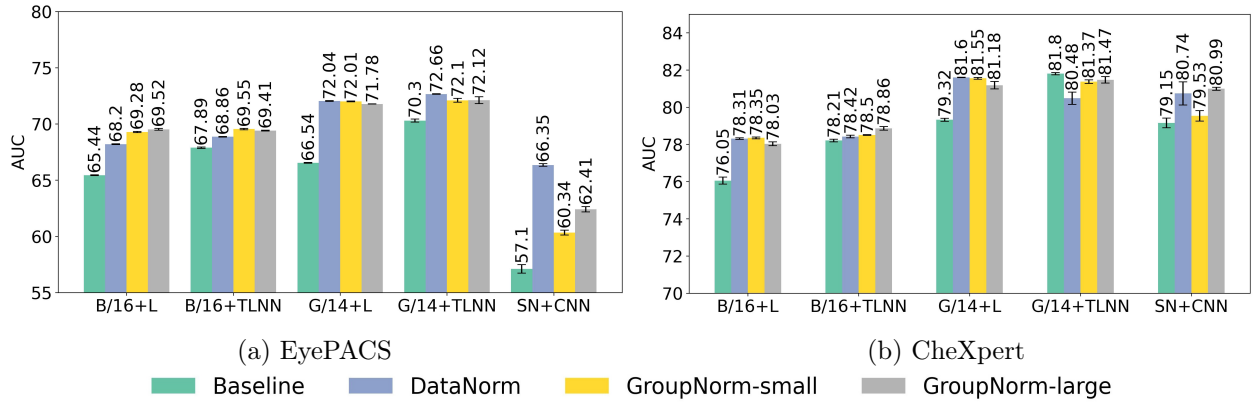


Figure 2: Normalization generally improves the final performance of all models. For CLIP-ViT models, GroupNorm small and large are groups of 8 and 16, respectively. For ScatterNets, GroupNorm small is 9 and large is 27. The choice of 27 over 81 is due to its superior performance. All experiments are done at $\varepsilon = 3$.

regression) or a two-layer neural network (TLNN, featuring tanh/tempered sigmoid activations [48]). We exclusively use CLIP-based models with their respective public pre-training datasets.

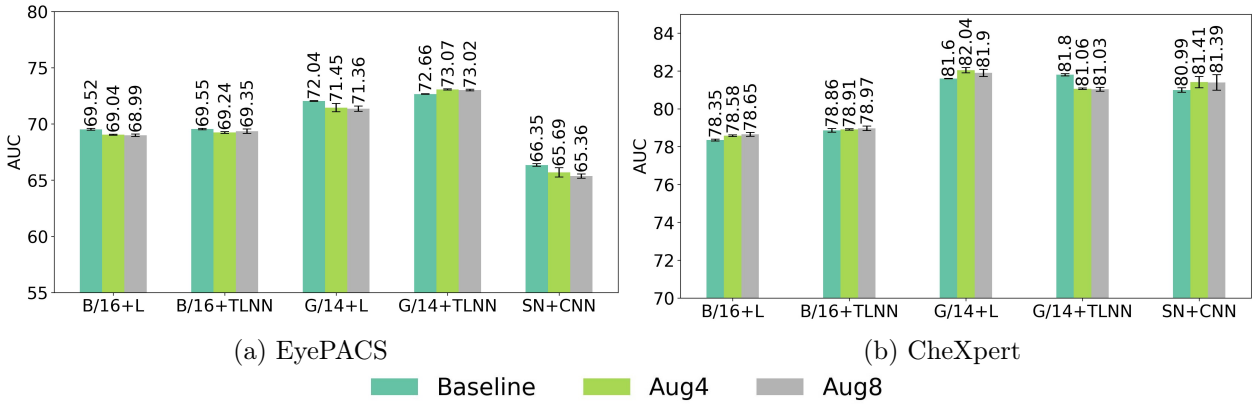
5 Experiments

Beyond proposing a variety of datasets and evaluation settings for benchmarking, we experimentally investigate techniques and the resulting utility obtained therein. Some of the key questions guiding our exporation: how many of the lessons learned in DP image classification on datasets like CIFAR-10 and ImageNet transfer to the privacy-critical setting of medical images? How much and when does public data help for such datasets, which may be ill-represented in public data? And, in absolute terms, how well can we do on these datasets with DP, in various evaluation settings?

After describing our experimental setup (Section 5.1), we revisit the efficacy of several ablations commonly employed in DP settings (Section 5.2). Finally, we make more broad conclusions about DP image classification based on our results (Section 5.3). Our code is included in the code repository.

5.1 Experimental Setup

We use PyTorch [49], and the Opacus library [68] for DP ML. We employed the Adam optimizer [39] across all experiments, both private and non-private, with a default learning rate of 0.001. We run our experiments at a variety of privacy levels ($\varepsilon \in [1, 3, 5, 8]$) with fixed delta values proportional to the inverse of the dataset size (10^{-6} for CheXpert and 10^{-5} for EyePACS), as we prescribed earlier. Batch size and total training epochs were fixed at 1024 and 20, respectively. A hyper-parameter search was performed to identify the optimal clipping norm within the range $[0.001, 0.01, 0.1, 1, 10]$. Following established metrics for all these datasets, we use AUC for CheXpert and EyePACS, and accuracy for CIFAR-10. We report mean and standard deviation over three independent runs. We used early stopping for non-private numbers due to overfitting, a phenomenon we did not observe for the DP setting due to its natural regularization properties [37].



(a) EyePACS (b) CheXpert

Figure 3: Augmentation multiplicity helps in general for CheXpert but not for EyePACS. We evaluate augmentation multiplicity by adding 4 and 8 augmentations of each image in the training data. All experiments are done at $\varepsilon = 3$.

5.2 Revisiting DP ablations

One of the most comprehensive ablation studies for DP image classification is by [18]. By using group normalization, large batches, weight standardization, augmentation multiplicity, and parameter averaging, they manage to raise CIFAR-10 accuracy on a validation set from 50.8% to an impressive 79.7%. We fix $\varepsilon = 3$ and, focusing on the CLIP ViTs and ScatterNets, run the exact same ordered sequence of ablations, without carrying forward the latest technique if it does not show improved utility. Broadly speaking, while [18]'s techniques proved highly effective for CIFAR-10, our results reveal mixed outcomes depending on various parameters.

Normalization Batch normalization [32] is not compatible with DPSGD because it combines information across a batch, making it impossible to bound the impact of a single image in the dataset. Instead, prior work has shown that variants including group normalization [66] and data normalization can be suitable replacements [7, 18, 24, 62].

Group normalization splits the channels of the hidden activations of an image into groups and normalizes the activations within each group. For the CLIP ViT models, with an input dimension of 512 and 1280 for the B/16 and G/14 respectively, we experiment with 8 and 16 groups. For Scatter features, with dimension (243, H/4, W/4) for RGB images and following [62], we use 9, 27, and 81 groups. Data normalization works on data channels by normalizing using the corresponding mean and variance across the training data. Normalizing in such a way, however, incurs a privacy cost as the per-channel means and variances must be privately estimated. We use Gaussian noise with ($\sigma = 8$) to estimate these means and variances for all runs, following [62].

In Figure 2 we show that normalization generally improves the final performance of all models, though the most effective normalization differs across architecture and dataset. Interestingly, the four experiments where data normalization was superior involved models with larger unclipped gradients. In these cases, the optimal clipping norm chosen during hyperparameter tuning was also the highest value (10). This suggests that data normalization can effectively manage large gradient magnitudes, especially when clipping underestimates the true gradient norms. Detailed results for our experiments are given in Table 1 and Table 2.

Table 1: Studying the impact of normalization for ScatterNet + CNN, normalization consistently improves performance. Data normalization tends to outperform group normalization for EyePACS and CIFAR-10, particularly due to the large gradients of their Scatter features.

Dataset	Model	Baseline	DataNorm	GroupNorm9	GroupNorm27	GroupNorm81
EyePACS (AUC)	SN + CNN	57.1 ± 0.38	$\mathbf{66.35} \pm 0.12$	60.34 ± 0.22	62.41 ± 0.24	63.68 ± 0.15
CheXpert (AUC)	SN + CNN	79.15 ± 0.26	80.74 ± 0.62	79.53 ± 0.28	$\mathbf{80.99} \pm 0.08$	80.72 ± 0.35
CIFAR-10 (Acc)	SN + CNN	55.18 ± 0.28	$\mathbf{68.29 \pm 0.17}$	65.97 ± 0.13	66.26 ± 0.11	66.45 ± 0.45

Table 2: Normalization impact for CLIP ViT models: Normalization generally improves performance, but it also depends on the architecture and dataset. Normalizations marked in red show a drop in performance compared to the baseline.

Dataset	Model	Baseline	DataNorm	GroupNorm8	GroupNorm16
EyePACS	B/16 + Linear	65.44 ± 0.04	68.2 ± 0.04	69.28 ± 0.06	$\mathbf{69.52 \pm 0.08}$
EyePACS	B/16 + TLNN	67.89 ± 0.06	68.86 ± 0.03	$\mathbf{69.55 \pm 0.06}$	69.41 ± 0.04
EyePACS	G/14 + Linear	66.54 ± 0.04	$\mathbf{72.04 \pm 0.03}$	72.01 ± 0.04	71.78 ± 0.01
EyePACS	G/14 + TLNN	70.30 ± 0.13	$\mathbf{72.66 \pm 0.03}$	72.1 ± 0.17	72.12 ± 0.31
EyePACS	G/14(CLIPA) + Linear	63.88 ± 0.08	$\mathbf{73.02 \pm 0.2}$	70.7 ± 0.06	70.62 ± 0.07
EyePACS	G/14(CLIPA) + TLNN	64.9 ± 0.2	$\mathbf{72.9 \pm 0.1}$	70.87 ± 0.25	70.8 ± 0.2
CheXpert	B/16 + Linear	76.05 ± 0.19	78.31 ± 0.04	$\mathbf{78.35 \pm 0.05}$	78.03 ± 0.1
CheXpert	B/16 + TLNN	78.21 ± 0.07	78.42 ± 0.07	78.5 ± 0.02	$\mathbf{78.86 \pm 0.1}$
CheXpert	G/14 + Linear	79.32 ± 0.08	$\mathbf{81.6 \pm 0.01}$	81.55 ± 0.05	81.18 ± 0.2
CheXpert	G/14 + TLNN	$\mathbf{81.80 \pm 0.06}$	80.48 ± 0.33	81.37 ± 0.1	81.47 ± 0.17
CheXpert	G/14(CLIPA) + Linear	72.39 ± 0.07	76.12 ± 0.4	$\mathbf{77.34 \pm 1.1}$	77.17 ± 0.5
CheXpert	G/14(CLIPA) + TLNN	77.65 ± 0.89	75.74 ± 1.1	77.38 ± 0.3	$\mathbf{77.43 \pm 0.7}$
CIFAR10	CLIP + Linear	99.64(93.91)	99.76(94.41)	99.75(94.43)	**99.75(94.57)**
CIFAR10	CLIP + TLNN	99.69(94.10)	99.74(94.15)	99.74(94.36)	**99.75(94.51)**

Larger Batch Size The impact of larger batch sizes in differentially private training has been observed both theoretically [4, 59] and empirically [2, 18]. In Table 3, scaling the batch size from 1024 to 4096 showed that CheXpert benefited in 80% of experiments, while EyePACS did not. This disparity is likely due to CheXpert having a training set six times larger than EyePACS, resulting in fewer model update steps for EyePACS and potential underfitting with a fixed number of epochs. We further observed that increasing the number of epochs showed a positive impact of larger batch sizes on EyePACS when using the ScatterNet model.

Weight Standardization We experiment with weight standardization (WS) on the Scatternet + CNN model as it applies to only convolution layers. From our results in Table 3, we observe that weight standardization does not help with EyePACS but helps with CheXpert and CIFAR-10. As alluded by prior work [10, 18], we also observe a positive correlation of group normalization with WS. However, due to a limited number of experiments, we do not have strong evidence either way.

Augmentation Multiplicity We apply a sequence of augmentations to our benchmark datasets: reflect padding, random cropping, and random horizontal flipping. While [18] recommend 16 augmentations per image, due to computational constraints with large datasets, we use 4 and 8 augmentations. As shown in Figure 3, contrary to [18]'s findings, augmentation multiplicity (augmult) does not consistently yield positive effects. Except for one experiment, (ViT-G/14+TLNN), augmentations generally benefit

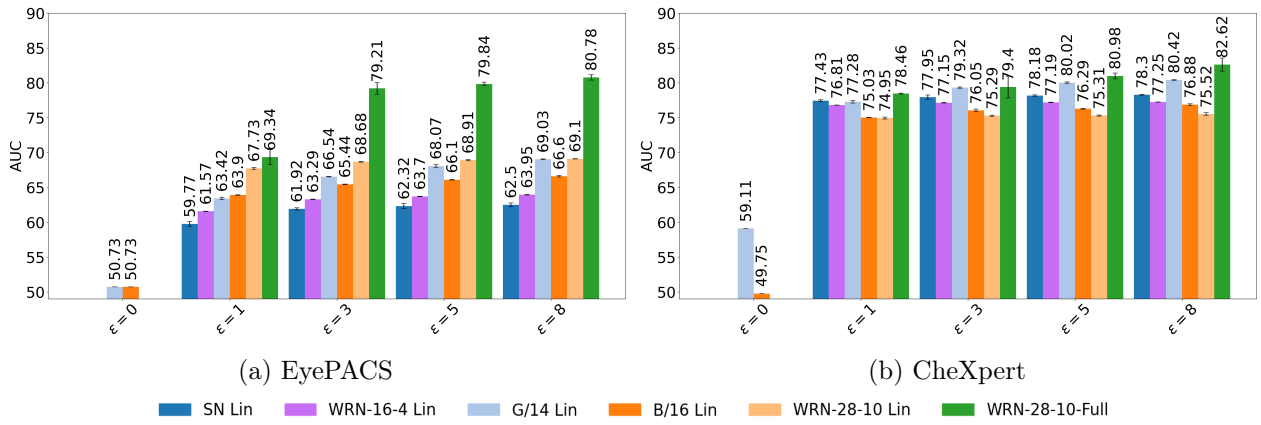


Figure 4: Pre-training datasets have different impacts: Wide-ResNet, pretrained on ImageNet, performs best on EyePACS, while ViT-G/14 with linear probe surpasses Wide-ResNet 28-10 linear probe across all ε values on CheXpert. Furthermore, ViT-G/14 achieves near-random performance on EyePACS in zero-shot settings but attains a non-trivial 59.11% AUC on CheXpert.

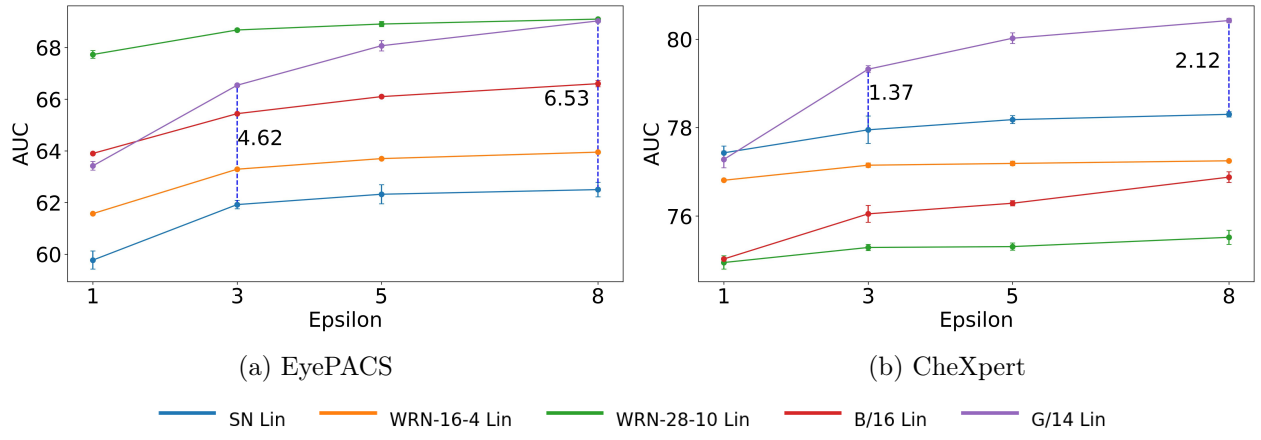


Figure 5: Pre-training public data is more beneficial with higher ε values. For CheXpert, ScatterNet performs better at smaller ε values, while pretrained models show marginal improvements at larger ε values. Similarly, for EyePACS, CLIP ViT-G/14 linear performs better as ε value increases.

Table 3: Studying all ablations together, we observe an almost consistent improvement in performance for CIFAR-10, whereas this pattern is not observed with the other datasets.

Dataset	Model	Best Normalization	+Larger Batch	+WS	+Best Augmult	+EMA
EyePACS	SN + CNN	66.35 ± 0.12	66.24 ± 0.23	65.81 ± 0.06	65.69 ± 0.42	$\mathbf{66.65 \pm 0.16}$
EyePACS	B/16 + L	$\mathbf{69.52 \pm 0.08}$	68.95 ± 0.24	-	69.04 ± 0.05	69.09 ± 0.12
EyePACS	B/16 + TLNN	$\mathbf{69.55 \pm 0.06}$	69.16 ± 0.28	-	69.24 ± 0.08	69.3 ± 0.21
EyePACS	G/14 + L	$\mathbf{72.04 \pm 0.03}$	71.59 ± 0.24	-	71.45 ± 0.37	71.29 ± 0.17
EyePACS	G/14 + TLNN	72.66 ± 0.03	$\mathbf{73.07 \pm 0.06}$	-	73.07 ± 0.05	73 ± 0.03
EyePACS	CLIPA + L	$\mathbf{73.02 \pm 0.2}$	68.5 ± 0.07	-	68.37 ± 0.05	70.94 ± 0
EyePACS	CLIPA + TLNN	72.09 ± 0.1	$\mathbf{72.17 \pm 0.1}$	-	72.04 ± 1.6	71.9 ± 0.07
CheXpert	SN + CNN	80.99 ± 0.08	81.54 ± 0.34	82.11 ± 0.29	81.41 ± 0.3	$\mathbf{82.24 \pm 0.22}$
CheXpert	B/16 + L	78.35 ± 0.05	78.42 ± 0.07	-	$\mathbf{78.65 \pm 0.1}$	$\mathbf{78.65 \pm 0.04}$
CheXpert	B/16 + TLNN	78.86 ± 0.1	78.86 ± 0.1	-	78.97 ± 0.12	$\mathbf{79.01 \pm 0.1}$
CheXpert	G/14 + L	81.6 ± 0.01	81.76 ± 0.05	-	$\mathbf{82.04 \pm 0.14}$	82.01 ± 0.05
CheXpert	G/14 + TLNN	81.8 ± 0.06	81.04 ± 0.16	-	81.06 ± 0.34	81.14 ± 0.26
CheXpert	CLIPA + L	77.34 ± 1.1	80.17 ± 0.08	-	$\mathbf{80.38 \pm 1.5}$	80.34 ± 0.15
CheXpert	CLIPA + TLNN	77.43 ± 0.7	80.4 ± 0.2	-	$\mathbf{80.75 \pm 0.18}$	80.52 ± 0.02
CIFAR10	B/16 + L	99.75(94.57)	99.74(94.49)	-	**99.77(94.76)**	99.76(94.67)
CIFAR10	B/16 + TLNN	99.75(94.51)	99.76(94.55)	-	**99.79(94.81)**	99.78(94.76)
CIFAR-10	SN + CNN	68.29 ± 0.17	66.47 ± 0.31	68.96 ± 0.26	$\mathbf{69.16 \pm 0.08}$	68.07 ± 0.24

CheXpert but not EyePACS. Future work may explore the effectiveness of dataset-specific augmentations, which could potentially yield more beneficial results. We show detailed experiment results in Table 4.

Parameter Averaging The final ablation that [18] suggests is the exponential moving average (EMA)[50] of all the parameters in the model. In Table 3, we notice that EMA occasionally improves performance, which contradicts the findings of [18] that it consistently enhances results across all experiments.

5.3 Experimental findings

We highlight some findings from our experimental results.

Different pre-training datasets offer varying degrees of improvement depending on the private data We compare representative models publicly pre-trained on a variety of datasets on both CheXpert and EyePACS. Results are displayed in Figure 4. For the case of no pre-training data, we choose ScatterNet+Linear, due to its consistently superior utility compared to Wide-ResNet trained from scratch, particularly for high privacy (i.e., low ε) settings.

On the other end of the spectrum, when we allow large-scale public pre-trarining, the CLIP ViT models provide a good indication of zero-shot performance (i.e., $\varepsilon = 0$).

When analyzing CheXpert, ViT-B/16 performs close to random in the zero-shot setting, whereas ViT-G/14 achieves an AUC of 59.11%, moderately better than random. Moving from ScatterNet+linear to Wide-ResNet+Linear, there is a noticeable decrease in AUC, yet ViT-G/14 consistently outperforms across various ε values, indicating that ViT-G/14 is a better fit for CheXpert. Notably, at $\varepsilon = 8$, Wide-ResNet with full fine-tuning exceeds the performance of ViT-G/14. However, considering that Wide-ResNet is fully fine-tuned while ViT-G/14 is not, this doesn't necessarily make Wide-ResNet better suited for CheXpert. Nevertheless, for smaller ε, it is clear that ViT-G/14 is the superior model with only linear fine-tuning.

Table 4: Studying the impact of augmentation multiplicity, we find that it consistently improves performance for CIFAR-10. However, looking at EyePACS and CheXpert, we observe inconsistent behavior, except that it generally seems to reduce performance with EyePACS. For the third column, we take the best result from Table 3 after best normalization, larger batch size, and weight standardization.

Dataset	Model	Norm + Larger BS + WS	+Augmult(4)	+Augmult(8)
EyePACS	SN + CNN	**66.35 ± 0.12**	65.69 ± 0.42	65.36 ± 0.19
EyePACS	B/16 + L	**69.52 ± 0.08**	69.04 ± 0.05	68.99 ± 0.11
EyePACS	B/16 + TLNN	**69.55 ± 0.06**	69.24 ± 0.08	69.35 ± 0.2
EyePACS	G/14 + L	72.04 ± 0.03	71.45 ± 0.37	71.36 ± 0.24
EyePACS	G/14 + TLNN	73.07 ± 0.05	**73.07 ± 0.05**	73.02 ± 0.07
EyePACS	G/14 (CLIPA)+ L	73.02 ± 0.2	68.37 ± 0.05	68.14 ± 1.2
EyePACS	G/14 (CLIPA)+ TLNN	72.9 ± 0.1	72.04 ± 1.6	71.98 ± 0.3
CheXpert	SN + CNN	82.11 ± 0.29	81.41 ± 0.3	81.39 ± 0.41
CheXpert	B/16 + L	78.42 ± 0.07	78.58 ± 0.04	**78.65 ± 0.1**
CheXpert	B/16 + TLNN	78.86 ± 0.1	78.91 ± 0.04	**78.97 ± 0.12**
CheXpert	G/14 + L	81.76 ± 0.05	**82.04 ± 0.14**	81.9 ± 0.19
CheXpert	G/14 + TLNN	81.8 ± 0.06	81.06 ± 0.04	81.03 ± 0.1
CheXpert	G/14 (CLIPA)+ L	77.34 ± 1.1	80.34 ± 0.2	**80.38 ± 1.5**
CheXpert	G/14 (CLIPA)+ TLNN	77.43 ± 0.7	80.5 ± 0.1	**80.75 ± 0.18**
CIFAR-10 (ACC)	SN + CNN	68.96 ± 0.26	69.07 ± 0.2	**69.16** ± 0.08
CIFAR-10 (ACC)	B/16 + L	99.75(94.57)	99.76(94.68)	**99.77(94.76)**
CIFAR-10 (ACC)	B/16 + TLNN	99.76(94.55)	99.78(94.75)	**99.79(94.81)**

Looking at Figure 4 for EyePACS, both CLIP ViT models show random performance in the zero-shot setting, indicating no improvement from pretraining. Conversely, Wide-ResNet linear exhibits a significant performance boost when transitioning from ScatterNet linear to Wide-ResNet linear, maintaining its superiority across all ε values. Although we notice that as we move toward less private regimes, the power of pre-trained ViT-G/14 becomes more evident, particularly from $\varepsilon = 1$ to $\varepsilon = 3$, approaching the performance of Wide-ResNet linear. However, there remains a substantial gap between fully fine-tuned Wide-ResNet and the other models, unlike CheXpert, suggesting that Wide-ResNet is better suited for EyePACS.

Public pre-training data helps more with higher ε values We compare feature generation methods in Figure 5 since, in all cases, there is a linear classifier on top of diverse feature extractors. On CheXpert, linear fine-tuning with ScatterNet shows the best performance at $\varepsilon = 1$. However, as ε increases, pretrained models, especially ViT-G/14, begin to outperform other methods significantly. While full fine-tuning of CLIP has not been explored, a direct comparison of features shows ViT-G/14's superiority when ε is sufficiently large. As ε value increases further, ViT-G/14's performance improves notably, highlighting its strong pretrained performance under less stringent privacy constraints.

When comparing the best performance on CheXpert across our proposed methods, ScatterNet achieves superior results compared to CLIP ViT models and Wide-ResNet on $\varepsilon = 3$, as shown in Figure 5. However, as ε values increase, pretrained models begin to perform better, and the performance gap between ScatterNet and the other models widens.

For EyePACS, we don't see the same pattern, likely because EyePACS is a much smaller dataset (about one-sixth the size) and Scatter features have high dimensionality, making it hard to balance this complexity with private training. We use ScatterNet linear as the baseline for the no pretraining regime and compare it to other architectures' linear fine-tuning for a fair comparison.

Table 5: Test AUC for EyePACS at different epsilons. Baselines include ScatterNet (SN), WideResNet (WRN) and CLIP models on datasets with different public data pre-training. The SOTA is due to [64].

Public data	Model	Test AUC (%)				
		$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 8$	$\varepsilon = \infty$
None	SN + L	59.77 ± 0.35	61.92 ± 0.16	62.32 ± 0.37	62.5 ± 0.28	69.70 ± 0.11
None	SN + CNN	63.73 ± 0.11	66.36 ± 0.17	66.59 ± 0.43	67.37 ± 0.27	69.28 ± 0.20
None	WRN-16-4+L	61.57 ± 0.02	63.29 ± 0.04	63.7 ± 0.04	63.95 ± 0.04	67.74 ± 0.01
None	WRN (Scratch)	55.45 ± 0.18	56.53 ± 0.08	57.14 ± 0.34	57.65 ± 0.22	61.97 ± 0.09
IN-32	WRN + Linear	67.73 ± 0.15	68.68 ± 0.05	68.91 ± 0.10	69.10 ± 0.03	73.21 ± 0.09
IN-32	WRN (Full)	69.34 ± 1.09	79.21 ± 0.83	79.84 ± 0.27	80.78 ± 0.38	83.61 ± 0.03
WIT	B/16 + Linear	63.9 ± 0.04	65.44 ± 0.04	66.1 ± 0.05	66.6 ± 0.12	69.93 ± 0.01
WIT	B/16 + TLNN	65.12 ± 0.04	67.89 ± 0.06	69.22 ± 0.1	69.84 ± 0.02	70.54 ± 0.01
LAION	G/14 + Linear	63.42 ± 0.17	66.54 ± 0.04	68.07 ± 0.2	69.03 ± 0.06	69.88 ± 0.2
LAION	G/14 + TLNN	65.47 ± 0.02	70.30 ± 0.13	71.74 ± 0.3	72.3 ± 0.19	73.36 ± 0.15
DataComp1B	G/14(CLIPA) + Linear	63.42 ± 0.06	63.88 ± 0.08	63.8 ± 0.12	64.33 ± 0.32	70.87 ± 0.01
DataComp1B	G/14(CLIPA) + TLNN	64.41 ± 1.00	64.9 ± 0.20	65.07 ± 0.26	65.67 ± 0.08	75.42 ± 0.08
IN-1K	SOTA	-	-	-	-	95.1

As illustrated in Figure 5, increasing the ε value amplifies ViT-G/14's performance advantage over the ScatterNet baseline, widening the gap. However, we do not observe any significant changes in ViT-B/16 and Wide-ResNet linear. ViT-B/16 appears to perform poorly regardless of privacy settings. On the other hand, Wide-ResNet linear consistently maintains a significant gap between its linear model and ScatterNet. This can be explained by the fact that Wide-ResNet linear can already achieve high AUC in the $\varepsilon = 1$ case, leaving little room for improvement.

The fact that Wide-ResNet maintains its advantage from the start is not surprising, given that as discussed earlier, the pre-trained model seems to help with EyePACS the most. However, ViT-G/14's performance improves more as the ε value increases. The detailed numbers for this experiment are provided in Table 5 and Table 6.

Progress on CIFAR10 does not translate to progress on benchmark datasets Looking at Figure 4, we notice that ViT-G/14 achieves an astonishing 99.75% zero-shot accuracy on CIFAR-10. In stark contrast, the same model's zero-shot performance on CheXpert and EyePACS is significantly lower, with AUC scores of 59.11% and 50.73%, respectively—the latter essentially equating to random guessing. Additionally, Wide-ResNet achieves 94.7% accuracy on CIFAR-10 at $\varepsilon = 1$, yet only 78.52% and 71.00% AUC on CheXpert and EyePACS, respectively.

Upon reviewing the ablation experiments in section 5.2, it becomes evident that the techniques beneficial for CIFAR-10 do not necessarily yield similar advantages for EyePACS and CheXpert datasets. The patterns observed in CIFAR-10 did not replicate in these medical image datasets, and notably, performance on CheXpert and EyePACS showed inconsistency.

Additionally, we observe that incorporating synthetic data as demonstrated by Tang et al. [60], leads to SOTA performance on CIFAR-10 without pretraining. However, in our experiments, ScatterNet outperforms [60]'s approach on CheXpert, whereas on EyePACS, Tang et al. achieve better results.

6 Related Work

Several works have evaluated the privacy-utility tradeoffs for DPML algorithms [35, 36, 73]. Jayaraman et al. [36] explored the impact of various variants of DP for ML algorithms. They explored the privacy

Table 6: Test AUC for CheXpert at different epsilons. Baselines include ScatterNet (SN), WideResNet (WRN) and CLIP models on datasets with different public data pre-training. The SOTA is from [7](Private) and [70](Non-Private).

Public data	Model	Test AUC (%)				
		$\varepsilon = 1$	$\varepsilon = 3$	$\varepsilon = 5$	$\varepsilon = 8$	$\varepsilon = \infty$
None	SN + CNN	78.16 ± 0.22	79.15 ± 0.26	79.16 ± 0.18	79.68 ± 0.04	80.65 ± 0.12
None	SN + Linear	77.43 ± 0.15	77.95 ± 0.31	78.18 ± 0.09	78.30 ± 0.06	78.94 ± 0.13
None	WRN-16-4+L	76.81 ± 0.02	77.15 ± 0.05	77.19 ± 0.05	77.25 ± 0.01	77.49 ± 0.03
None	WRN (Scratch)	76.9 ± 0.02	77.8 ± 0.04	77.89 ± 0.05	78.68 ± 0.10	87.31 ± 0.07
IN-32	WRN + Linear	74.95 ± 0.15	75.29 ± 0.07	75.31 ± 0.08	75.52 ± 0.16	75.91 ± 0.08
IN-32	WRN (Full)	78.46 ± 0.07	79.40 ± 1.57	80.98 ± 0.42	82.62 ± 0.94	87.62 ± 0.09
WIT	B/16 + Linear	75.03 ± 0.03	76.05 ± 0.19	76.29 ± 0.06	76.88 ± 0.12	76.89 ± 0.01
WIT	B/16 + TLNN	77.28 ± 0.19	78.21 ± 0.07	78.33 ± 0.04	78.54 ± 0.06	78.56 ± 0.01
LAION	G/14 + Linear	77.28 ± 0.19	79.32 ± 0.08	80.02 ± 0.12	80.42 ± 0.05	80.48 ± 0.02
LAION	G/14 + TLNN	80.63 ± 0.4	81.80 ± 0.06	82.25 ± 0.02	$82.27 \pm 0.0.4$	82.28 ± 0.01
DataComp1B	G/14(CLIPA) + Linear	71.45 ± 0.27	72.39 ± 0.07	72.98 ± 0.38	72.37 ± 0.41	78.35 ± 1.3
DataComp1B	G/14(CLIPA) + TLNN	77.3 ± 0.60	77.65 ± 0.89	77.67 ± 0.25	77.51 ± 0.85	80.62 ± 0.06
IN-21K	SOTA	86.3	-	-	89.2	-
IN-1K	SOTA	-	-	-	-	93 [7]

leakage concerning the privacy parameter ϵ for the same algorithm. The work of Zhao et al. [73] and Jarin et al. [35] similarly study the privacy-utility tradeoffs for different DP ML algorithms and evaluate them against membership inference attacks. There have also been some attempts at benchmarking DP algorithms [27, 61, 65]. Tao et al. [61] and Gong et al. [27] benchmark different synthetic data generation algorithms for tabular data and image data respectively. The work of Wei et al. [65] is closest to our work, where they benchmark different DPML algorithms on standard ML datasets such as MNIST/CIFAR-10 and comment on the effects of improvements made in DPML literature. In our work, we take a different stance than them and propose a new benchmark based on privacy-critical medical datasets. Compared to their work, we also experimented with more established architectures based on various techniques, such as Scatternets and CLIP-based models.

7 Future Work

While our work focused on image classification, future research should explore benchmarks in other areas such as Natural Language Understanding and Generation. In addition, to ensure fair comparisons, future work could investigate the use of more advanced model architectures. For instance, experiments using the NFNet-F7 [11] model pre-trained on ImageNet-1K could be compared with our Wide-ResNet experiments.

Future research should also extend to a wider range of datasets, both within and beyond the medical domain. This exploration will help in understanding the generalizability of DP ML techniques and identifying domain-specific challenges.

The continued maintenance and updating of the leaderboard we have established will be crucial for tracking long-term progress in the field and identifying emerging trends or breakthroughs. This ongoing effort will provide valuable insights into the evolution of DP ML techniques over time.

8 Conclusion

We suggest a number of standardized settings for benchmarking DP image classification, particularly with a focus on privacy-critical domains such as medical images. We also provide a leaderboard to help track progress on image classification benchmarks. In our experimental investigation, we find that several of the techniques which have enjoyed great success for DP ML are *not* universally effective across datasets and architectures, and furthermore that progress on standard benchmarks like CIFAR-10 do *not* transfer to medical images. Of course, it is hard and rare to design universally effective techniques. Indeed, our experiments are for a limited number of datasets and a limited number of architectures, so it is impossible to make a conclusion broad enough to encompass the entire field of DP image classification. However, it is clear that present work leaves the door open for new ideas and techniques that push the envelope on private image classification in these settings.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM Conference on Computer and Communications Security*, CCS '16, pages 308–318, New York, NY, USA, 2016. ACM.

[2] R. Anil, B. Ghazi, V. Gupta, R. Kumar, and P. Manurangsi. Large-scale differentially private BERT. *arXiv preprint arXiv:2108.01624*, 2021.

[3] M. Baradad Jurjo, J. Wulff, T. Wang, P. Isola, and A. Torralba. Learning to see by looking at noise. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, pages 2556–2569. Curran Associates, Inc., 2021.

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

[5] R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '14, pages 464–473, Washington, DC, USA, 2014. IEEE Computer Society.

[6] B. K. Beaulieu-Jones, Z. S. Wu, C. Williams, R. Lee, S. P. Bhavnani, J. B. Byrd, and C. S. Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7):e005122, 2019.

[7] L. Berrada, S. De, J. H. Shen, J. Hayes, R. Stanforth, D. Stutz, P. Kohli, S. L. Smith, and B. Balle. Unlocking accuracy and fairness in differentially private image classification. *arXiv preprint arXiv:2308.10888*, 2023.

[8] A. Bie, G. Kamath, and G. Zhang. Private GANs, revisited. *Transactions on Machine Learning Research*, 2023.

[9] A. Birhane and V. U. Prabhu. Large image datasets: A pyrrhic win for computer vision? In *2021 IEEE Winter Conference on Applications of Computer Vision*, WACV '21, pages 1536–1546. IEEE, 2021.

[10] A. Brock, S. De, and S. L. Smith. Characterizing signal propagation to close the performance gap in unnormalized resnets. *CoRR*, abs/2101.08692, 2021.

[11] A. Brock, S. De, S. L. Smith, and K. Simonyan. High-performance large-scale image recognition without normalization, 2021.

[12] T. Cao, A. Bie, A. Vahdat, S. Fidler, and K. Kreis. Don't generate me: Training differentially private generative models with sinkhorn divergence. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, pages 12480–12492. Curran Associates, Inc., 2021.

[13] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, B. Balle, D. Ippolito, and E. Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium*, USENIX Security '23, pages 5253–5270. USENIX Association, 2023.

[14] N. Carlini, F. Tramèr, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel. Extracting training data from large language models. In *30th USENIX Security Symposium*, USENIX Security '21, pages 2633–2650. USENIX Association, 2021.

[15] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(29):1069–1109, 2011.

[16] P. Chrabaszcz, I. Loshchilov, and F. Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. *CoRR*, abs/1707.08819, 2017.

[17] L. Chua, B. Ghazi, P. Kamath, R. Kumar, P. Manurangsi, A. Sinha, and C. Zhang. How private are DP-SGD implementations? In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24. JMLR, Inc., 2024.

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

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '09, pages 248–255, Washington, DC, USA, 2009. IEEE Computer Society.

[20] T. Dockhorn, T. Cao, A. Vahdat, and K. Kreis. Differentially private diffusion models. *Transactions on Machine Learning Research*, 2023.

[21] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

[22] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proceedings of the 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, EUROCRYPT '06, pages 486–503, Berlin, Heidelberg, 2006. Springer.

[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, Berlin, Heidelberg, 2006. Springer.

[24] F. Dörmann, O. Frisk, L. N. Andersen, and C. F. Pedersen. Not all noise is accounted equally: How differentially private learning benefits from large sampling rates, 2021.

[25] Eyepacs. Diabetic retinopathy detection, 2015. data retrieved from Kaggle, https://www.kaggle.com/c/diabetic-retinopathy-detection.

[26] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 2015 ACM Conference on Computer and Communications Security*, CCS '15, pages 1322–1333. ACM, 2015.

[27] C. Gong, K. Li, Z. Lin, and T. Wang. Dpimagebench: A unified benchmark for differentially private image synthesis, 2025.

[28] S. Gopi, Y. T. Lee, and L. Wutschitz. Numerical composition of differential privacy. In *Advances in Neural Information Processing Systems 34*, NeurIPS '21, pages 11631–11642. Curran Associates, Inc., 2021.

[29] F. Harder, M. Jalali, D. J. Sutherland, and M. Park. Pre-trained perceptual features improve differentially private image generation. *Transactions on Machine Learning Research*, 2023.

[30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[31] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4(8):1–9, 2008.

[32] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, ICML '15, pages 448–456. JMLR, Inc., 2015.

[33] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI '19, pages 590–597, 2019.

[34] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang. Towards practical differentially private convex optimization. In *Proceedings of the 40th IEEE Symposium on Security and Privacy*, SP '19, pages 299–316, Washington, DC, USA, 2019. IEEE Computer Society.

[35] I. Jarin and B. Eshete. Dp-util: comprehensive utility analysis of differential privacy in machine learning. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy*, pages 41–52, 2022.

[36] B. Jayaraman and D. Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.

[37] C. Jung, K. Ligett, S. Neel, A. Roth, S. Sharifi-Malvajerdi, and M. Shenfeld. A new analysis of differential privacy's generalization guarantees. In *Proceedings of the 11th Conference on Innovations in Theoretical Computer Science*, ITCS '20, pages 31:1–31:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[38] D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Proceedings of the 25th Annual Conference on Learning Theory*, COLT '12, pages 25.1–25.40, 2012.

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR '15, 2015.

[40] A. Koskela, J. Jälkö, and A. Honkela. Computing tight differential privacy guarantees using FFT. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, AISTATS '20, pages 2560–2569. JMLR, Inc., 2020.

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

[42] C. J. Lebeda, M. Regehr, G. Kamath, and T. Steinke. Avoiding pitfalls for privacy accounting of subsampled mechanisms under composition. *arXiv preprint arXiv:2405.20769*, 2024.

[43] Y. LeCun, C. Cortes, and C. Burges. MNIST handwritten digit database, 2010.

[44] X. Li, F. Tramèr, P. Liang, and T. Hashimoto. Large language models can be strong differentially private learners. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

[45] I. Mironov. Rényi differential privacy. In *Proceedings of the 30th IEEE Computer Security Foundations Symposium*, CSF '17, pages 263–275, Washington, DC, USA, 2017. IEEE Computer Society.

[46] I. Mironov, K. Talwar, and L. Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[47] E. Oyallon and S. Mallat. Deep roto-translation scattering for object classification. *CoRR*, abs/1412.8659, 2014.

[48] N. Papernot, A. Thakurta, S. Song, S. Chien, and Ú. Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, AAAI '21, pages 9312–9321, 2021.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, NeurIPS '19, pages 8026–8037. Curran Associates, Inc., 2019.

[50] B. Polyak and A. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30:838–855, 07 1992.

[51] N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. G. Thakurta. How to DP-fy ML: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

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

[53] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio. Transfusion: Understanding transfer learning for medical imaging, 2019.

[54] R. Redberg, A. Koskela, and Y.-X. Wang. Improving the privacy and practicality of objective perturbation for differentially private linear learners. In *Advances in Neural Information Processing Systems 36*, NeurIPS '23, pages 13819–13853. Curran Associates, Inc., 2023.

[55] C. Schuhmann, R. Beaumont, R. Vencu, C. W. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. R. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[56] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*, SP '17, pages 3–18, Washington, DC, USA, 2017. IEEE Computer Society.

[57] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the 2023 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '23, pages 6048–6058. IEEE Computer Society, 2023.

[58] S. Song, K. Chaudhuri, and A. D. Sarwate. Stochastic gradient descent with differentially private updates. In *Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing*, GlobalSIP '13, pages 245–248, Washington, DC, USA, 2013. IEEE Computer Society.

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

[60] X. Tang, A. Panda, V. Sehwag, and P. Mittal. Differentially private image classification by learning priors from random processes, 2023.

[61] Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, and G. Miklau. Benchmarking differentially private synthetic data generation algorithms, 2022.

[62] F. Tramèr and D. Boneh. Differentially private learning needs better features (or much more data). In *Proceedings of the 9th International Conference on Learning Representations*, ICLR '21, 2021.

[63] F. Tramèr, G. Kamath, and N. Carlini. Position: Considerations for differentially private learning with large-scale public pretraining. In *Proceedings of the 41st International Conference on Machine Learning*, ICML '24. JMLR, Inc., 2024.

[64] M. Voets, K. Møllersen, and L. A. Bongo. Reproduction study using public data of: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *PLOS ONE*, 14(6):e0217541, June 2019.

[65] C. Wei, M. Zhao, Z. Zhang, M. Chen, W. Meng, B. Liu, Y. Fan, and W. Chen. Dpmlbench: Holistic evaluation of differentially private machine learning. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2621–2635, 2023.

[66] Y. Wu and K. He. Group normalization. *CoRR*, abs/1803.08494, 2018.

[67] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

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

[69] D. Yu, S. Naik, A. Backurs, S. Gopi, H. A. Inan, G. Kamath, J. Kulkarni, Y. T. Lee, A. Manoel, L. Wutschitz, S. Yekhanin, and H. Zhang. Differentially private fine-tuning of language models. In *Proceedings of the 10th International Conference on Learning Representations*, ICLR '22, 2022.

[70] Z. Yuan, Y. Yan, M. Sonka, and T. Yang. Large-scale robust deep auc maximization: A new surrogate loss and empirical studies on medical image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3040–3049, 2021.

[71] Z. Yuan, D. Zhu, Z.-H. Qiu, G. Li, X. Wang, and T. Yang. Libauc: A deep learning library for x-risk optimization. In *29th SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023.

[72] S. Zagoruyko and N. Komodakis. Wide residual networks. *CoRR*, abs/1605.07146, 2016.

[73] B. Z. H. Zhao, M. A. Kaafar, and N. Kourtellis. Not one but many tradeoffs: Privacy vs. utility in differentially private machine learning. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 15–26, 2020.