

DPMLBench: Holistic Evaluation of Differentially Private Machine Learning

Chengkun Wei^{¶*} Minghu Zhao^{¶*} Zhikun Zhang^{‡§¶†} Min Chen[‡]

Wenlong Meng[¶] Bo Liu^{||} Yuan Fan[¶] Wenzhi Chen^{¶†}

[¶]Zhejiang University [‡]CISPA Helmholtz Center for Information Security

[§]Stanford University ^{||}DBAPPSecurity

Abstract

Differential privacy (DP), as a rigorous mathematical definition quantifying privacy leakage, has become a well-accepted standard for privacy protection. Combined with powerful machine learning techniques, differentially private machine learning (DPML) is increasingly important. As the most classic DPML algorithm, DP-SGD incurs a significant loss of utility, which hinders DPML's deployment in practice. Many studies have recently proposed improved algorithms based on DP-SGD to mitigate utility loss. However, these studies are isolated and cannot comprehensively measure the performance of improvements proposed in algorithms. More importantly, there is a lack of comprehensive research to compare improvements in these DPML algorithms across utility, defensive capabilities, and generalizability.

We fill this gap by performing a holistic measurement of improved DPML algorithms on utility and defense capability against membership inference attacks (MIAs) on image classification tasks. We first present a taxonomy of where improvements are located in the machine learning life cycle. Based on our taxonomy, we jointly perform an extensive measurement study of the improved DPML algorithms, over twelve algorithms, four model architectures, four datasets, two attacks, and various privacy budget configurations. We also cover state-of-the-art label differential privacy (Label DP) algorithms in the evaluation. According to our empirical results, DP can effectively defend against MIAs, and sensitivity-bounding techniques such as per-sample gradient clipping play an important role in defense. We also explore some improvements that can maintain model utility and defend against MIAs more effectively. Experiments show that Label DP algorithms achieve less utility loss but are fragile to MIAs. Machine learning practitioners may benefit from these evaluations to select appropriate algorithms. To support our evaluation, we implement a modular re-usable software, DPMLBench,¹ which enables sensitive data owners to deploy DPML algorithms and serves as a benchmark tool for researchers and practitioners.

*Chengkun and Minghu contributed equally to this work.

[†]Corresponding authors.

¹The implementation can be found at <https://github.com/DmsKinson/DPMLBench>

1 Introduction

As *machine learning* (ML) continues to evolve, numerous fields are leveraging its power to advance their development [1, 2]; however, this often involves the use of private data, such as medical records. Previous studies have revealed that the models trained on private data can leak information through a bunch of attacks, such as membership inference [3], model inversion [4], and attribute inference [5], which raises critical privacy and security concerns.

Differential privacy (DP) is a widely used notion to rigorously formalize and measure the privacy guarantee based on a parameter called *privacy budget*. Abadi *et al.* [6] proposed a general DPML algorithm called *differentially private stochastic gradient descent* (DP-SGD) by integrating per-sample clipping and noise perturbation to the aggregated gradient in the training process. However, models trained by DP-SGD normally perform badly with respect to model utility. Recently, researchers proposed many *improved algorithms* with better privacy-utility trade-off [7, 8, 9, 10, 11, 12, 13, 14, 15]. In the rest of this paper, we refer to DP-SGD as *vanilla DP-SGD* to distinguish between DP-SGD and the improved algorithms.

The improved algorithms modify the vanilla DP-SGD from different aspects but are evaluated in isolation with various settings, which cannot reveal the differences between each other. Furthermore, existing studies [16, 17, 18, 19] fail to report a complete and practical analysis of general DPML algorithms in practical scenarios. This motivates us to perform a holistic evaluation and analysis of these improved DPML algorithms.

1.1 Our Contributions

Algorithm Taxonomy. We first propose a new taxonomy for the state-of-the-art DPML algorithms based on their improved component in the ML pipeline. Concretely, we divide the ML pipeline into four phases: Data preparation, model design, model training, and model ensemble (see Section 2.1 for details), and categorize the DPML algorithms into each phase. We then perform a theoretical and empirical analysis to obtain an extensive view of the impact of differential privacy on machine learning.

Experimental Evaluation. In this paper, we concen-

trate on twelve state-of-the-art DPML algorithms for image classification tasks. We then conduct comprehensive experiments for these algorithms on four model architectures (ResNet20 [20], VGG16 [21], InceptionNet [22], and SimpleCNN) and four benchmark image datasets (MNIST [23], FashionMNIST [24], CIFAR-10 [25], and SVHN [26]) to jointly evaluate the tradeoff between privacy protection, model utility, and defense effectiveness. Furthermore, we evaluate the defensive capabilities of the DPML algorithms on both white-box and black-box *membership inference attacks* (MIAs). Our measurement aims to answer the following three research questions:

- RQ1.** What improvements in DPML algorithms are most effective in maintaining model utility?
- RQ2.** What improvements in DPML algorithms are most robust in defending membership inference attacks?
- RQ3.** What is the impact of dataset and model architecture on algorithms focusing on different stages?

In addition, our measurement covers two state-of-the-art label differential privacy (Label DP) algorithms, which is a variant DPML notion by relaxing the protection of the whole data sample to only protect the label. To the best of our knowledge, we are the first to analyze the Label DP algorithms on utility and defense empirically.

DPMLBench. We implement a toolkit called DPMLBench to support the comprehensive evaluation of DPML algorithms with respect to model utility and MIA defense. With a modular design, DPMLBench can easily integrate additional DPML algorithms, attacks, datasets, and model architectures by implementing new functional codes to the relevant modules. Our code will be publicly available, facilitating researchers to leverage existing DPML algorithms to provide DP guarantee or benchmark new algorithms.

Main Findings. Our work reveals several interesting findings:

- Different improvement techniques can affect the privacy-utility trade-offs of the algorithm from different perspectives. For example, we find that reducing the dimension of the parameter improves the performance of DPML on large models but may impair utility when the privacy budget is large. In addition, DP synthetic algorithms and algorithms in the model ensemble category are the most robust in defending against MIAs.
- DP can effectively defend against MIAs. Also, sensitivity-bounding techniques such as per-sample gradient clipping play an important role in defense.
- Some model architecture design choices for non-private ML models are ineffective for private ML models. For instance, using Tanh as the activation function and Group-Norm can reduce the utility loss on vanilla DP-SGD. However, we also find that using Tanh and GroupNorm together would have a negative effect.
- Compared to standard DP, Label DP has less utility loss but is more fragile to MIAs.

2 Preliminaries

2.1 Machine Learning Pipeline

Figure 1 illustrates a typical machine learning pipeline, which consists of four phases: Data preparation, model design, model training, and model ensemble.

The *data preparation* phase aims to explore the underlying distribution of data for learning algorithms. Commonly used techniques in this phase include data cleaning [27], data labeling [28], and feature extraction [29]. Feature extraction transforms the input data into a low-dimensional subspace that reveals the most relevant information [30]. Low dimensional information can downgrade the difficulty of the following training procedures [31, 7, 32].

In the *model design* phase, we aim to select components such as the model architectures, loss functions, and optimization algorithms that are appropriate for the task. There are plenty of studies on this topic [20, 21].

The *model training* phase is the process of computing the following optimization objective:

$$\arg \min_{\theta} \frac{1}{|D_{train}|} \sum_{(x,y) \in D_{train}} L(y, M(x; \theta)),$$

where (x, y) is the data sample in the training dataset D_{train} ; L and M represent loss function and model architecture, respectively. The parameters θ in model M are optimized to minimize the objective function L on training data during the model training phase.

The *model ensemble* phase combines multiple models while deploying the model. Previous studies show that aggregating multiple models' predictions can obtain better generalization performance than a single model [33].

2.2 Differential Privacy

Differential privacy (DP) [34] is a rigorous mathematical definition quantifying how much privacy preservation a mechanism can provide. DP provides a privacy guarantee by bounding the impact of a single input on the mechanism's output.

Definition 2.1 ((ϵ, δ) -Differential Privacy). *Given two neighboring datasets D and D' differing by one record, a mechanism M satisfies (ϵ, δ) -differential privacy if*

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

where ϵ is the privacy budget, and δ is the failure probability.

The privacy budget quantifies the maximum information a mechanism M can expose. A smaller privacy budget indicates better privacy preservation. δ indicates the probability that M fails to satisfy ϵ -DP. When $\delta = 0$, we achieve pure ϵ -DP, a stronger notion, and a more rigorous privacy guarantee.

Bounded DP and Unbounded DP. How to interpret neighboring datasets distinguishes between *bounded DP* and *unbounded DP* [35]. In *unbounded DP*, D and D' are neighbors if D can be obtained from D' by adding or removing

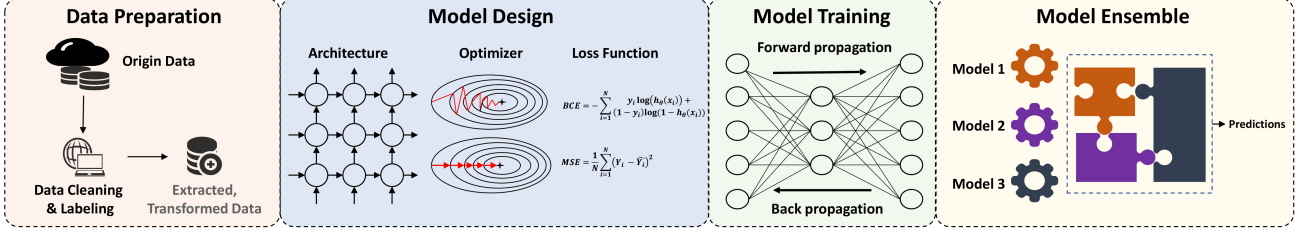


Figure 1: Machine learning pipeline.

one element. In *bounded DP*, D and D' are neighbors if D can be obtained from D' by replacing one element. When using *bounded DP*, two datasets should have the same number of elements. Furthermore, any algorithms that satisfy ϵ -*unbounded DP* also satisfy 2ϵ -*bounded DP* because replacing one element can be achieved by removing then adding one element. All algorithms in Table 1 satisfy the *unbounded DP*.

Gaussian Mechanism. Adding noise sampled from Gaussian distribution is a commonly used approach to achieve (ϵ, δ) -DP, known as Gaussian mechanism [36]. Formally, applying the Gaussian mechanism to a function f can be defined as:

$$M(d) = f(d) + \mathbf{N}(0, S_f^2 \cdot \sigma^2),$$

where $\mathbf{N}(0, S_f^2 \cdot \sigma^2)$ is the Gaussian distribution with mean 0 and standard deviation $S_f^2 \cdot \sigma^2$, where σ is called noise multiplier and S_f is the sensitivity of function f .

Definition 2.2. (Sensitivity). Given two neighboring datasets D and D' , the global sensitivity of a mechanism M , denoted by S_M , is given below

$$S_M = \max_{D, D'} |M(D) - M(D')|.$$

Composition. The composition theorems calculate the total privacy budget when we apply DP on the private dataset multiple times. The most straightforward composition strategy is summing up the privacy budget of each individual DP algorithm. Formally, for k DP algorithms with privacy budget $\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_k$, the total privacy budget is $\epsilon = \epsilon_1 + \epsilon_2 + \epsilon_3 + \dots + \epsilon_k$. Mironov [37] et al. propose Rényi differential privacy to achieve a tighter analysis of cumulative privacy budgets.

Definition 2.3 ((α, δ) -Rényi Differential Privacy (RDP) [37]). A randomized mechanism M is said to satisfy ϵ -Rényi differential privacy of order α (which can be abbreviated as (α, δ) -RDP), if for any adjacent datasets D, D' , it holds that

$$D_\alpha(M(D) || M(D')) \leq \epsilon,$$

where $D_\alpha(M(D) || M(D'))$ is the α -Rényi divergence between the distribution of $M(D)$ and the distribution of $M(D')$. Parameter α controls the momentum of the privacy loss random variable.

Note that larger α leads to more weight being assigned to worst-case events, e.g., (∞, ϵ) -RDP is equivalent to ϵ -DP.

If M satisfies (α, ϵ) -RDP, it also satisfies $(\epsilon + \frac{\log \frac{1}{\delta}}{\alpha - 1}, \delta)$ -DP. Applying k algorithms with (α, ϵ_1) -RDP, (α, ϵ_2) -RDP, \dots , (α, ϵ_k) -RDP on same dataset sequentially leads to an algorithm with $(\alpha, \epsilon_1 + \epsilon_2 + \dots + \epsilon_k)$ -RDP. By selecting α delicately, accumulating privacy loss in RDP and then converting to DP can derive a tighter upper bound than composite (ϵ, δ) -DP directly.

Post-processing. The post-processing property guarantees that no matter what additional processing one performs on the output of an algorithm that satisfies (ϵ, δ) -DP, the composition of the algorithm and the post-processing operations still satisfy (ϵ, δ) -DP.

2.3 Differentially Private Machine Learning

Abadi *et al.* [6] integrated differential privacy with stochastic gradient descent (SGD) and proposed a general learning algorithm named differential privacy stochastic gradient descent (DP-SGD). Compared to SGD, DP-SGD introduced a few modifications to make the algorithm satisfy differential privacy. Firstly, the sensitivity of each gradient is bounded by clipping each gradient in the l_2 norm.

$$\text{clip}(\mathbf{g}, C) = \mathbf{g} / \max(1, \frac{\|\mathbf{g}\|_2}{C}). \quad (1)$$

Per-sample clipping bounds the contribution of each sample to model parameters to C . Moreover, DP-SGD applies a Gaussian mechanism to the aggregated clipped gradient. Formally,

$$\tilde{\mathbf{g}} = \mathbf{g} + \mathbf{N}(0, C^2 \sigma^2), \quad (2)$$

where $\tilde{\mathbf{g}}$ is the noisy gradient used to update parameters and σ controls privacy level. After the above two steps, the gradients used to update the parameters satisfy DP.

Nevertheless, gradient clipping and noise perturbation introduce deviation in the training process, which impairs the model's utility. Recently, researchers proposed a number of improved DPML algorithms to reduce the utility loss incurred by vanilla DP-SGD [38, 11, 31, 12]. However, these improved DPML algorithms were evaluated on different models and datasets with different assumptions. Therefore, it is a pressing need to design a holistic benchmark to comprehensively evaluate these DPML algorithms to gain a deeper insight.

Table 1: Overview and comparison of DPML algorithms. *: Evaluation is based on subsequent private model training on generated data. ○: same as non-private training. ①: with modification but no noise adding. ●: with modification and noise adding.

| | Algorithms | Auxiliary Data | Private Data | Model Architecture | Gradient | Loss Function | Perturbation |
|------------------|----------------|----------------|--------------|--------------------|----------|---------------|--------------|
| Data Preparation | Hand-DP | ○ | ① | ○ | ● | ○ | Gradient |
| | PrivSet * | ○ | ● | ○ | ○ | ○ | Input |
| | DPGEN * | ○ | ● | ○ | ○ | ○ | Input |
| Model Design | TanhAct | ○ | ○ | ① | ● | ○ | Gradient |
| | FocalLoss | ○ | ○ | ○ | ● | ① | Gradient |
| Model Training | Vanilla DP-SGD | ○ | ○ | ○ | ● | ○ | Gradient |
| | RGP | ○ | ○ | ① | ● | ○ | Gradient |
| | GEP | ① | ○ | ○ | ● | ○ | Gradient |
| | AdpAlloc | ○ | ○ | ○ | ● | ○ | Gradient |
| | AdpClip | ○ | ○ | ○ | ● | ○ | Gradient |
| Model Ensemble | PATE | ● | ○ | ○ | ○ | ○ | Input |
| | Priv-kNN | ● | ○ | ○ | ○ | ○ | Input |

2.4 Membership Inference in Machine Learning Models

The MIAs have become one of the most widely studied [39, 40] attacks against ML models after Shokri *et al.* proposed in [3]. The MIA aims to infer whether a data sample is used to train the target ML model. Formally, MIA A can be defined as:

$$A : I, M, \mathbf{x} \rightarrow \{0, 1\},$$

where I is the auxiliary knowledge of adversary, M is the model to be attacked, and \mathbf{x} is a data sample. A can be seen as a binary classifier, where 1 means the data sample \mathbf{x} is used for training model M , namely a member, and 0 otherwise. It is natural to use MIAs to evaluate the defensive capabilities of DPML algorithms, as in many previous studies [16, 19, 41].

Based on the information an attacker can obtain, MIAs can be classified into two categories: White-box and black-box. The white-box attacks have full access to the target model, while black-box attacks only have query access to the target model and obtain the prediction confidence vector. We adopt both types of MIAs to comprehensively evaluate the defensive capabilities of the DPML algorithms (in Section 5.3).

3 Taxonomy

In this section, we provide an overview of our taxonomy and give survey-style descriptions of the DPML algorithms.

3.1 Overview

We first propose a new taxonomy for the DPML algorithms based on the component they improve in the ML pipeline discussed in Section 2.1. We introduce this taxonomy due to the following reasons: (1) The training phases of ML are independent, meaning the improvements in different phases might be combined to achieve better model utility. (2) It provides future researchers with a clear roadmap to improve the DPML algorithms, which we hope can benefit the community. (3) It is domain-agnostic and can be easily extended

to evaluate the DPML algorithms in other domains, such as graph and NLP data.

Table 1 summarizes all the improved DPML algorithms and their corresponding categories. We also discuss the properties of all the DPML algorithms. For instance, vanilla DP-SGD falls in the model training category and modifies the gradient to provide the DP guarantee, whereas PATE belongs to the model ensemble category and leverages auxiliary data to provide a DP guarantee. Auxiliary data generally refers to data with the same distribution as sensitive data but is publicly available, which is a common assumption in DPML [31, 9, 32].

Data Preparation. The algorithms in this category preprocess the original training data. Feature extraction and DP synthetic data are two typical approaches in this category. Feature extraction aims to reduce the difficulty of private training. Using a pre-trained network before classifier [6, 13, 42] can be seen as a variant of feature extraction. DP synthetic data aims to provide a DP guarantee for training data. Applying DP mechanisms to data directly, such as the Gaussian mechanism, downgrades the utility of data, especially when data is in high dimension (*e.g.*, image). DP synthetic data is an alternative that aims to generate data in a DP manner with a similar distribution as sensitive data. Training models on synthetic data with traditional machine learning algorithms can derive a model with DP guarantee according to post-processing property [43, 15, 44]. In this category, we pick three algorithms, of which Hand-DP [12] leverages a feature extractor, and the other two (PrivSet [45] and DPGEN [15]) belong to DP synthetic data algorithms.

Model Design. Algorithms in this category focus on designing more adapted model designs to DPML. Deep learning in non-private settings has been widely studied, and many rules have been summarized to train a standard neural network. However, these design guidelines do not perform well in vanilla DP-SGD [42] due to gradient clipping and noise perturbation. For instance, larger models often mean better performance in non-private settings. However, smaller models tend to get better performance on vanilla DP-SGD. Some existing studies focus on exploring more adapted model design rules to DPML [11, 46]. We select two algorithms in this

category, and they propose improvements from the activation function (TanhAct [11]) and loss function (FocalLoss [8]), respectively.

Model Training. Algorithms in this category explore DP mechanisms with less impact on model utility in the DP-SGD training phase. The vanilla DP-SGD [6] bounds the l_2 -norm of gradient g by clipping the gradient to the threshold C ; thus, a straightforward improvement strategy is to find an optimal clipping strategy [47, 48]. On the other hand, the noise perturbation leads to bias during model updating, which impairs the model’s utility. Therefore, designing a better noise perturbation mechanism to alleviate the noise effect is another optimization option [31, 32, 13]. In this category, we select four algorithms, excluding vanilla DP-SGD. AdpClip [48] proposes an improved clipping strategy, and the rest of them (RGP [7], GEP [31], and AdpAlloc [13]) explore better noise perturbation mechanisms.

Model Ensemble. This category contains algorithms providing DP guarantee through the model ensemble. The vanilla DP-SGD has poor scalability because it requires modifications to the training process. Papernot *et al.* [9] propose Private Aggregation of Teacher Ensemble (PATE) by leveraging model ensemble. PATE treats the training phase of the model as a black box so that it has better scalability than vanilla DP-SGD for less modification to the training process. DP mechanism is applied while aggregating the prediction of multiple models. Since then, many DPML algorithms based on the model ensemble have emerged [38, 10]. We select PATE and Priv-kNN in our measurement.

3.2 Data Preparation

Hand-DP [12]. Tramer *et al.* leverage Scattering Network (ScatterNet) [49], a feature extractor that encodes images using a cascade of wavelet transforms to extract the features. To achieve the DP guarantee, they fine-tuned a model on top of extracted features through DP-SGD.

DPGEN [15]. It is an instantiation of the DP variant of the Energy-based Model (EBM) [50, 51], which aims to privatize Langevin Markov Chain Monte Carlo (MCMC) sampling method [52] to synthesize images, of which an energy-based network guides the movement directions. DPGEN achieves DP by using Randomized Response (RR) in movement direction selection. Compared to other DP-SGD based synthesis methods [53, 44], DPGEN can generate higher-resolution images.

PrivSet [45]. It leverages dataset condensation to generate data in a differentially private manner. It directly optimizes for a small set of samples promising to derive approximate results under downstream tasks instead of imitating the complete data distribution. Specifically, they use DP-SGD to optimize a gradient-matching objective for the downstream task that minimizes the difference between the gradient on the real data and the generated data.

3.3 Model Design

TanhAct [11]. Considering the need for DP to bound sensitivity, Papernot *et al.* [11] replace ReLU with tempered sigmoid as the activation function. The authors found that the bounded property of tempered sigmoid functions, especially Tanh, can effectively limit the l_2 -norm of the gradient while training models with DP-SGD. Thus, less information can be lost in gradient clipping.

FocalLoss [8]. It introduces a loss function adapted to vanilla DP-SGD, which combines three terms: The summed squared error L_{Focal} , the focal loss L_{SSE} [54], and a regularization penalty on the intermediate pre-activations L_{Reg} . Finally, they proposed loss function L :

$$L = \alpha L_{\text{Focal}} + (1 - \alpha) L_{\text{SSE}} + \frac{(1 - \alpha)}{\beta} L_{\text{Reg}}, \quad (3)$$

where $\alpha = \text{Sigmoid}(e_c - e_t)$ (current epoch e_c , and threshold epoch e_t), β is the hyperparameter controlling the strength of the regularization. These terms consider convergence speed, emphasis on complex samples, and sensitivity during training. The new loss function can better control the gradient sensitivity in the training procedure.

3.4 Model Training

RGP [7]. It adopts a reparametrization scheme to replace the model weight in each layer with two low-dimensional weight matrices and a residual weight matrix:

$$\mathbf{W} \rightarrow \mathbf{L}\mathbf{R} + \tilde{\mathbf{W}}.\text{stop_gradient}(). \quad (4)$$

By making the gradient carriers $\{\mathbf{L}, \mathbf{R}\}$ consist of orthonormal vectors, a projection of the gradient of \mathbf{W} can be constructed from the noisy gradients of $\tilde{\mathbf{L}}$ and $\tilde{\mathbf{R}}$. $\{\mathbf{L}, \mathbf{R}\}$ are trained by DP-SGD separately to achieve the DP guarantee and finally combined to obtain the gradient for updating the model. Note that the dimensionality of \mathbf{L} and \mathbf{R} is much smaller than that of \mathbf{W} . Thus RGP can reduce the storage consumption and the noise added to the model.

GEP [31]. Yu *et al.* observe that the number of noise increases with the growth of model size in vanilla DP-SGD and figure out a solution, GEP [31], to reduce the dimension of the gradient before adding noise. GEP first computes an anchor subspace that contains some gradients of public data via the power method. Then, it projects the gradient of private data into the anchor subspace to produce a low-dimensional gradient embedding and a small-norm residual gradient. The two parts are applied with the DP mechanism separately and combined to update the original weight. Compared to RGP, GEP leverages public data to decompose the original model parameters for dimensionality reduction.

AdpAlloc [13]. It proposes a dynamic noise-adding mechanism instead of keeping noise multiplier σ constant every training epoch in vanilla DP-SGD. It replaces the variance in the Gaussian mechanism with a function of the epoch:

$$M(d) = f(d) + \mathbf{N}(0, S_f^2 \cdot \sigma_t^2), \quad (5)$$

the value of σ_t depends on the final privacy budget, epoch, and schedule function. The schedule function defines how the noise scale is adjusted during training. Yu *et al.* proposed several pre-defined schedules. We select *Exponential Decay* in our evaluation, which has the best average performance in [13]. The mathematical form of *Exponential Decay* is $\sigma_t = \sigma_0 e^{-kt}$, where $k(k > 0)$ is decay rate and σ_0 is the initial noise scale.

AdpClip [48]. It uses an adaptive clipping threshold mechanism, which sets the clip threshold to a specified quantile of the update norm distribution every epoch. Formally, clipping threshold C_t in epoch t can be computed as $C_t = C_{t-1} \cdot \exp(-\eta_C(\bar{b} - \gamma))$, where $\gamma \in [0, 1]$ is a quantile to be matched, $\bar{b} \triangleq \frac{1}{m} \sum_{i \in [m]} \mathbb{I}_{x_i \leq C}$ is the empirical fraction of samples with value at most C , and η_C is the learning rate with default value of 0.2 in [48]. To address the issue that \bar{b} reveals private information, Gaussian mechanism is applied to \bar{b} : $\bar{b}' = \frac{1}{m} (\sum_{i \in Q'} b'_i + N(O, \sigma_b^2))$. The method consumes a negligible privacy budget to track the quantile closely. AdpClip was originally designed for federated learning (FL) but can be extended to traditional centralized learning scenarios.

3.5 Model Ensemble

PATE [9]. It first trains multiple teacher models with disjoint private data. The teacher ensemble is later used to label the public data, and the noise perturbation is applied to the voting aggregation before generating a prediction. The student model, which gives the final output, is trained from labeled public data and cannot directly access private data. The privacy budget is determined by the noise added to the votes and the number of queries to the teacher ensemble. Additionally, PATE leverages a semi-supervised learning method to reduce the queries to the teacher ensemble.

Priv-kNN [10]. In PATE, a larger number of teacher models lead to a larger absolute lead gap while aggregating votes, potentially allowing for a larger noise level. At the same time, splitting data makes each teacher model hold only partial original training data, which causes a model utility drop. Thus, Zhu *et al.* [10] propose a data-efficient scheme based on the private release of k-nearest neighbor (kNN) queries to replace teacher ensemble, which avoids splitting the training dataset. For every given data sample from the public domain, Priv-kNN subsamples a random subset from the entire private dataset. Then it picks the k nearest neighbors from the subset in feature space, equivalent to k teachers' prediction in vanilla PATE.

4 DPMLBench

This section introduces DPMLBench, a modular toolkit designed to evaluate DPML algorithms' performance on utility and privacy. Figure 2 illustrates the four modules of DPMLBench.

1. **Input.** This module prepares the dataset and model for the following modules. For dataset, it involves dataset partition and preprocessing *e.g.*, normalization. For the

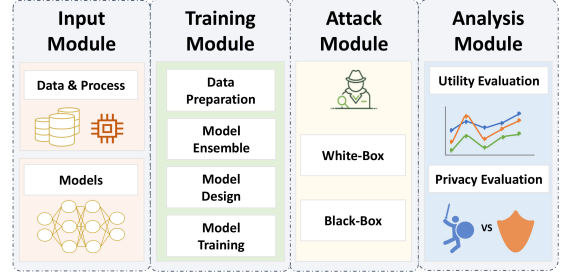


Figure 2: Overview of DPMLBench.

model, it constructs model architectures and does necessary modifications for private training (see Section 5.1).

2. **Training.** This module performs the DPML algorithms to train DPML models. It currently supports twelve different DPML algorithms into four categories (see Section 3).
3. **Attack.** This module performs two MIAs on models trained from the training module.
4. **Analysis.** This module evaluates the performance of DPML algorithms on utility and privacy.

DPMLBench follows a modular design that makes it flexible to integrate new algorithms, attacks, datasets, and models. We envisage that DPMLBench can be used for the following purposes:

- As we have implemented twelve representative DPML algorithms, DPMLBench enables data owners to train their privacy-preserving models with these DPML algorithms efficiently.
- DPMLBench comprehensively assesses different DPML algorithms in utility and privacy. Researchers can re-use DPMLBench as a benchmark tool to evaluate other DPML algorithms and attacks in the future.
- Since DPMLBench follows a modular design, modules are connected through abstract interfaces. To integrate a new DPML algorithm and attack or to extend DPMLBench into different domains, users can re-implement processing functions in the corresponding modules and reuse other modules directly.

5 Experiments

Based on the proposed taxonomy, we present a series of comprehensive experiments to answer the following questions:

- RQ1.** What improvements in DPML algorithms are most effective in maintaining model utility?
- RQ2.** What improvements in DPML algorithms are most robust in defending MIAs?
- RQ3.** What is the impact of dataset and model architecture on algorithms focusing on different stages?

Table 2: The testing accuracy, tailored AUC of MIAs in black-box/white-box of baseline models. The number of parameters follows each model name. (Accuracy (%)/black-box/white-box)

| Target Model | MNIST | FMNIST | SVHN | CIFAR-10 |
|----------------------|-----------------|-----------------|-----------------|-----------------|
| SimpleNet (0.17M) | 98.42/0.50/0.50 | 88.04/0.54/0.54 | 87.69/0.64/0.53 | 69.50/0.78/0.72 |
| ResNet (0.26M) | 99.12/0.50/0.50 | 89.16/0.52/0.54 | 92.88/0.57/0.59 | 66.56/0.77/0.63 |
| InceptionNet (1.97M) | 99.18/0.51/0.50 | 90.92/0.56/0.53 | 95.08/0.55/0.57 | 83.52/0.71/0.68 |
| VGG (128.8M) | 98.70/0.50/0.52 | 90.74/0.59/0.56 | 91.91/0.62/0.56 | 72.96/0.78/0.73 |

5.1 Experimental Setup

DPML Algorithms. We implement twelve DPML algorithms; their details can be found in Section 3. For GEP, RGP, Priv-kNN, DPGEN, and PrivSet, we use implementations of authors and modify codes to adapt for our evaluation. The rest of the algorithms are implemented by PyTorch [55] and Opacus [56].

Datasets. We conduct experiments on four datasets: MNIST [23], FashionMNIST [24], CIFAR-10 [25], and SVHN [26], which are widely used in evaluating privacy-preserving machine learning [6, 31, 9, 10]. We resize all images to 32x32 in our evaluation.

Since our attacks are all under the assumption that the attacker has an auxiliary dataset that shares similar distribution with the training data, we split each dataset into four disjoint parts: shadow training set, shadow testing set, target training set, and target testing set. Additionally, we allocate 90% of the data originally used for testing as public data for the algorithms in the ensemble category.

Model Architectures. We focus on four model architectures, including ResNet20 [20], VGG16 [21], InceptionNet [22], and a simple three convolution layer network as SimpleCNN. Batch normalization makes each sample’s normalized value depend on its peers in a batch, making it hard to restrict a single data contribution to the output. To adapt differential privacy, we replace all batch normalization [57] with group normalization [58]. We regard the models trained with the same hyperparameters without DP as the baseline to evaluate utility loss. Table 2 shows the performance of the baseline model across datasets, including testing accuracy and tailored AUC against black-box/white-box MIAs.

We use MLPs for black-box and white-box model architecture for the attack implementation as in [59, 40]. A detailed description of the model architecture can be found in Appendix E.

Hyperparameters. We use Rényi DP to accumulate the overall privacy budget and precompute the required noise scale (σ in DP-SGD) numerically [6, 60]. We keep $\delta = 10^{-5}$ and use different privacy budgets: $\epsilon = \{0.2, 0.3, 0.4, 0.5, 1, 2, 4, 8, 100, 1000\}$. All algorithms’ clipping threshold C are fixed to 4 unless the algorithm has special clipping strategies.

We use the hyperparameters obtained by grid search on DP-SGD if the original paper does not mention the setting. While searching hyperparameters, we refer to the guides of recent studies on hyperparameter settings for private training [61, 42]. For simplicity, we ignore the privacy leakage caused by hyperparameter tuning in our experiment [62]. For the attack models, we follow the settings in [59], where the

batch size is 64, the epoch is 50, the optimizer is Adam, and the learning rate is 10^{-5} . Appendix A shows the detailed hyperparameter settings.

Metrics. Following previous studies [16, 59, 19, 18], we use accuracy ACC to evaluate the models’ utility and the area under ROC curve (AUC) to evaluate the defense ability of the model. In MIAs, AUC lower than 0.5 indicates that the inference attack performs worse than a random guess and tends to infer non-members as members. Thus we set the lower bound of AUC to 0.5 for analysis convenience, indicating that AUC=0.5 implies no privacy leakage. We process the AUC metric as follows:

$$\widetilde{\text{AUC}} = \max(\text{AUC}, 0.5),$$

We name $\widetilde{\text{AUC}}$ as tailored AUC, which is always between 0.5 and 1.

To compare the performance of DPML algorithms and non-private algorithms more directly, we define proportional metric **utility loss** and **privacy leakage**, respectively:

$$\text{Utility Loss} = 1 - \frac{\text{ACC}_{M_{pri}}}{\text{ACC}_{M_{base}}}, \quad (6)$$

$$\text{Privacy Leakage} = \frac{\widetilde{\text{AUC}}_{M_{pri}} - 0.5}{\widetilde{\text{AUC}}_{M_{base}} - 0.5}, \quad (7)$$

where M_{pri} presents a private model trained by a DPML algorithm and M_{base} presents a non-private model trained by vanilla SGD with the same settings as M_{pri} . The utility loss denotes the percentage loss in accuracy of the DP model on the same test set relative to the normal model. The private leakage denotes the proportion of privacy models’ privacy leakage compared to the normal model.

5.2 Evaluation on Utility Loss

Overview. Table 3 reports an overview of algorithms’ utility loss across model architectures, datasets, and privacy budgets. Due to space limitations, we only show part of the experimental results. The rest results can be found in Appendix D (Table 9, which shows the similar trend as Table 3.). The experimental results for GEP on InceptionNet and VGG are unavailable due to memory limit. For brevity, we use a $\langle \text{Alg}, \text{Model}, \text{Dataset}, \epsilon \rangle$ tuple to denote the *Model* trained with *Alg* on *Dataset* in the case of privacy budget ϵ . For instance, $\langle \text{RGP}, \text{ResNet}, \text{MNIST}, 0.2 \rangle$ indicates the ResNet model trained by RGP with a privacy budget of 0.2 on MNIST.

We observe that the utility loss decreases with increasing privacy budget for all algorithms, which intuitively shows that the noise scale hurts the model’s utility. However, the utility loss varies widely across algorithms for the same privacy budget. We analyze improved DPML algorithms’ utility loss across four categories in the following. *NonPrivate* in figures denotes the model trained by normal SGD without DP.

Data Preparation. Initially, in [15], the classifier was trained on private data in order to label the synthetic data, and

Table 3: Overview of algorithms’ utility loss on different model architectures, datasets, and privacy budgets. For each privacy budget, we bold the value with the best performance (with the smallest value of utility loss). The experimental results of GEP on VGG are unavailable due to memory limits.

| | | SimpleCNN | | | | | ResNet | | | | | VGG | | | | |
|-----------|-----------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 |
| MNIST | Hand-DP | 89.19 ± 1.04 | 18.98 ± 1.67 | 8.46 ± 0.35 | 4.19 ± 0.17 | 3.73 ± 0.14 | 24.38 ± 1.93 | 11.58 ± 0.76 | 5.73 ± 0.46 | 2.88 ± 0.50 | 2.01 ± 0.68 | 88.63 ± 1.25 | 88.89 ± 1.23 | 90.46 ± 0.16 | 7.98 ± 0.15 | 3.80 ± 0.78 |
| | PrivSet | 77.26 ± 5.89 | 47.54 ± 9.23 | 30.86 ± 2.26 | 19.72 ± 2.28 | 17.58 ± 2.60 | 89.70 ± 0.20 | 82.24 ± 2.70 | 57.01 ± 6.11 | 20.83 ± 3.60 | 17.86 ± 2.67 | 81.36 ± 11.11 | 58.61 ± 5.03 | 51.32 ± 22.10 | 52.40 ± 27.64 | 66.52 ± 18.65 |
| | DPGEN | 60.99 ± 8.05 | 88.26 ± 0.83 | 14.95 ± 0.58 | 2.48 ± 0.43 | 2.70 ± 0.26 | 70.75 ± 4.84 | 84.07 ± 3.96 | 73.73 ± 3.77 | 3.64 ± 0.36 | 3.79 ± 0.25 | 90.08 ± 0.04 | 90.08 ± 0.04 | 58.28 ± 30.47 | 1.76 ± 0.14 | 2.20 ± 0.28 |
| | TanhAct | 85.19 ± 2.58 | 18.30 ± 1.29 | 3.13 ± 0.39 | 1.74 ± 0.14 | 1.74 ± 0.16 | 38.16 ± 3.27 | 18.61 ± 3.42 | 7.05 ± 0.54 | 3.89 ± 0.44 | 3.12 ± 0.19 | 90.21 ± 0.04 | 89.59 ± 0.12 | 90.15 ± 0.31 | 6.14 ± 0.08 | 1.91 ± 0.03 |
| | FocalLoss | 87.99 ± 1.97 | 29.11 ± 2.59 | 7.44 ± 0.40 | 3.32 ± 0.07 | 2.40 ± 0.01 | 43.09 ± 4.17 | 10.99 ± 1.70 | 6.32 ± 0.85 | 2.72 ± 0.12 | 1.78 ± 0.12 | 82.10 ± 1.77 | 88.16 ± 0.01 | 88.91 ± 0.39 | 11.46 ± 11.64 | 3.66 ± 0.52 |
| | DP-SGD | 89.61 ± 1.12 | 21.98 ± 0.19 | 7.50 ± 0.42 | 3.58 ± 0.21 | 3.04 ± 0.23 | 28.17 ± 3.48 | 11.33 ± 1.17 | 5.38 ± 0.85 | 2.88 ± 0.35 | 2.24 ± 0.36 | 88.79 ± 1.02 | 88.74 ± 0.47 | 90.56 ± 0.77 | 13.23 ± 4.42 | 3.55 ± 0.06 |
| | RGP | 36.65 ± 1.04 | 13.23 ± 0.78 | 10.17 ± 1.02 | 6.72 ± 0.09 | 6.37 ± 0.23 | 31.87 ± 2.62 | 21.24 ± 3.90 | 33.03 ± 7.12 | 34.06 ± 5.62 | 37.66 ± 8.97 | 90.30 ± 0.29 | 6.59 ± 0.95 | 3.86 ± 0.27 | 6.33 ± 2.24 | 4.78 ± 0.14 |
| | GEP | 90.30 ± 0.29 | 90.25 ± 0.34 | 14.37 ± 1.83 | 2.67 ± 0.52 | 1.52 ± 0.02 | 86.22 ± 2.16 | 17.61 ± 1.35 | 4.36 ± 0.22 | 1.00 ± 0.04 | 0.46 ± 0.26 | - | - | - | - | - |
| | AdpAlloc | 89.23 ± 0.81 | 18.79 ± 1.24 | 6.57 ± 0.20 | 3.59 ± 0.45 | 3.14 ± 0.29 | 24.26 ± 3.70 | 10.04 ± 2.13 | 4.91 ± 0.59 | 3.25 ± 0.57 | 2.48 ± 0.37 | 90.24 ± 0.22 | 89.00 ± 0.58 | 89.85 ± 1.04 | 6.44 ± 0.26 | 3.12 ± 0.04 |
| | AdpClip | 88.17 ± 4.04 | 75.55 ± 11.05 | 7.79 ± 0.37 | 8.00 ± 0.30 | 8.10 ± 0.28 | 59.66 ± 2.95 | 7.46 ± 0.53 | 4.85 ± 0.40 | 4.17 ± 0.35 | 4.21 ± 0.41 | 88.92 ± 0.22 | 88.13 ± 0.24 | 89.06 ± 0.75 | 14.00 ± 2.44 | 4.78 ± 0.15 |
| CIFAR-10 | PATE | 82.83 ± 3.94 | 71.81 ± 4.09 | 33.36 ± 12.33 | 11.89 ± 4.04 | 10.98 ± 2.78 | 90.86 ± 0.09 | 85.89 ± 6.08 | 30.74 ± 15.79 | 7.12 ± 3.66 | 10.38 ± 1.16 | 84.94 ± 3.47 | 76.17 ± 3.78 | 44.08 ± 23.93 | 32.15 ± 40.09 | 32.66 ± 39.74 |
| | Priv-kNN | 61.22 ± 2.13 | 34.97 ± 3.37 | 33.13 ± 0.62 | 34.77 ± 0.94 | 33.80 ± 1.56 | 25.03 ± 5.76 | 9.70 ± 0.87 | 8.43 ± 0.63 | 9.30 ± 0.74 | 9.91 ± 1.25 | 49.05 ± 1.62 | 17.80 ± 2.17 | 17.16 ± 0.94 | 16.29 ± 0.39 | 14.74 ± 1.43 |
| | Hand-DP | 90.06 ± 0.16 | 86.88 ± 4.04 | 48.67 ± 0.96 | 43.28 ± 0.74 | 44.14 ± 0.18 | 84.29 ± 2.26 | 58.34 ± 0.50 | 50.74 ± 0.58 | 41.95 ± 2.33 | 39.29 ± 3.11 | 90.03 ± 0.18 | 89.67 ± 0.07 | 89.86 ± 0.12 | 79.74 ± 7.71 | 37.58 ± 0.82 |
| | PrivSet | 88.85 ± 0.65 | 87.78 ± 1.28 | 86.78 ± 0.71 | 88.83 ± 0.94 | 89.43 ± 0.78 | 89.56 ± 0.83 | 89.28 ± 0.37 | 89.45 ± 1.18 | 87.91 ± 2.03 | 85.43 ± 2.43 | 89.77 ± 0.29 | 88.14 ± 0.78 | 89.07 ± 0.38 | 90.04 ± 0.26 | 87.64 ± 2.81 |
| | DPGEN | 90.16 ± 0.11 | 89.86 ± 0.09 | 89.66 ± 0.70 | 69.98 ± 2.28 | 76.98 ± 2.26 | 90.00 ± 0.35 | 90.00 ± 0.16 | 90.59 ± 1.39 | 79.51 ± 1.01 | 83.38 ± 4.20 | 90.52 ± 0.30 | 89.72 ± 0.21 | 89.47 ± 0.29 | 87.24 ± 3.07 | 88.86 ± 1.43 |
| | TanhAct | 89.74 ± 0.64 | 69.95 ± 1.13 | 45.39 ± 0.92 | 32.93 ± 0.55 | 32.21 ± 0.21 | 82.55 ± 1.17 | 62.11 ± 0.33 | 55.52 ± 0.32 | 48.95 ± 1.17 | 49.28 ± 2.73 | 90.22 ± 0.00 | 90.07 ± 0.25 | 90.13 ± 0.10 | 64.46 ± 1.43 | 34.26 ± 0.28 |
| | FocalLoss | 89.88 ± 0.08 | 88.17 ± 2.46 | 52.42 ± 0.47 | 38.55 ± 0.79 | 38.47 ± 0.90 | 84.36 ± 2.43 | 62.12 ± 0.53 | 52.06 ± 0.43 | 40.65 ± 2.12 | 39.00 ± 2.98 | 90.17 ± 0.26 | 89.75 ± 0.15 | 89.85 ± 0.23 | 66.36 ± 6.19 | 36.60 ± 0.33 |
| | DP-SGD | 89.80 ± 0.30 | 89.13 ± 1.30 | 48.79 ± 0.24 | 40.03 ± 0.93 | 40.48 ± 0.86 | 81.92 ± 2.61 | 58.32 ± 0.44 | 49.57 ± 1.76 | 41.17 ± 2.67 | 38.66 ± 3.80 | 90.20 ± 0.56 | 89.38 ± 0.49 | 89.73 ± 0.09 | 89.81 ± 0.12 | 35.15 ± 0.35 |
| | RGP | 90.15 ± 0.02 | 61.91 ± 1.32 | 58.52 ± 1.34 | 54.28 ± 0.68 | 54.41 ± 0.92 | 74.48 ± 0.54 | 65.24 ± 0.88 | 67.27 ± 2.00 | 66.38 ± 0.93 | 66.56 ± 0.82 | 90.16 ± 0.01 | 81.87 ± 4.19 | 53.66 ± 1.25 | 53.49 ± 0.09 | 54.37 ± 0.49 |
| | GEP | 90.16 ± 0.00 | 90.16 ± 0.01 | 90.16 ± 0.00 | 35.11 ± 0.20 | 31.90 ± 0.24 | 88.68 ± 2.19 | 85.19 ± 0.20 | 46.72 ± 0.73 | 30.45 ± 0.36 | 26.64 ± 0.93 | - | - | - | - | - |
| CIFAR-100 | AdpAlloc | 90.04 ± 0.30 | 89.89 ± 0.18 | 47.97 ± 0.57 | 38.49 ± 0.47 | 39.16 ± 0.88 | 80.04 ± 2.19 | 57.88 ± 0.86 | 48.86 ± 1.03 | 43.83 ± 1.52 | 42.22 ± 2.19 | 90.06 ± 0.05 | 89.57 ± 0.05 | 89.99 ± 0.08 | 51.42 ± 0.58 | 35.46 ± 0.42 |
| | AdpClip | 89.71 ± 0.23 | 89.79 ± 0.26 | 64.50 ± 2.33 | 35.64 ± 0.82 | 34.12 ± 0.42 | 86.57 ± 1.43 | 64.08 ± 0.77 | 48.05 ± 1.15 | 37.17 ± 1.27 | 33.55 ± 1.98 | 89.70 ± 0.34 | 89.86 ± 0.68 | 90.21 ± 0.19 | 89.69 ± 0.01 | 44.47 ± 0.07 |
| | PATE | 90.19 ± 1.30 | 91.70 ± 1.44 | 89.25 ± 0.53 | 83.30 ± 2.42 | 83.06 ± 0.54 | 88.34 ± 0.41 | 87.60 ± 1.13 | 85.99 ± 2.08 | 82.05 ± 1.18 | 83.50 ± 1.43 | 90.05 ± 1.23 | 91.60 ± 0.49 | 91.06 ± 1.11 | 89.92 ± 1.56 | 90.02 ± 2.79 |
| | Priv-kNN | 89.52 ± 0.45 | 89.38 ± 0.12 | 88.94 ± 0.17 | 90.19 ± 0.05 | 90.29 ± 0.40 | 87.77 ± 1.70 | 81.35 ± 1.56 | 77.38 ± 1.27 | 74.96 ± 0.34 | 74.43 ± 1.27 | 89.85 ± 1.27 | 87.26 ± 2.55 | 85.42 ± 2.04 | 84.81 ± 1.32 | 84.41 ± 1.56 |

then the labeled dataset was used to train the target model. This is similar to labeling public data through teacher ensemble in [9], which will consume additional privacy budgets. However, [15] does not count this part. In our implementation, we use data that does not overlap with private data to train the labeling model.

Figure 3a illustrates the accuracy comparison between algorithms in the data preparation category and vanilla DP-SGD. The plot shows that Hand-DP outperforms DPGEN and PrivSet in low privacy budget generally. Hand-DP’s accuracy is equivalent to vanilla DP-SGD and has a slight advantage on VGG. The performance of DPGEN and PrivSet is highly relative to the quality of synthetic data. When manually inspecting the generated data, we observe that there exist images with wrong labels and many similar, even identical images (e.g., mode collapse). More effort on hyperparameter tuning and manual data filtering for DP synthetic algorithms can improve the performance.

Moreover, Tramer *et al.* propose using the non-learned handcrafted feature to train a linear model with DP-SGD [12]. Thus, we perform the same experiment for Hand-DP on simple MLP. The experiment results on CIFAR-10 are shown as Table 7 in Appendix D. Comparing other model architectures, we observe that the simple MLP only has an advantage when the privacy budget is relatively small (e.g., $\epsilon < 0.5$). Thus, we exclude the MLP in subsequent experiments to maintain uniformity with other algorithms.

Model Design. Figure 3b illustrates the performance of algorithms in the model design category and vanilla DP-SGD.

In general, TanhAct outperforms vanilla DP-SGD and FocalLoss on SimpleCNN and VGG. However, neither TanhAct nor FocalLoss performs better than vanilla DP-SGD on ResNet and InceptionNet, TanhAct’s performance is even much worse than vanilla DP-SGD on ResNet. [11] shows that TanhAct has a better utility-privacy trade-off on their models, whose architecture is similar to SimpleCNN. The difference among the architectures is that ResNet and InceptionNet both have GroupNorm layers while the others do not.

To figure out the impact of the GroupNorm layer and ac-

tivation function, we add the GroupNorm layer before the activation function of the SimpleCNN and evaluate the performance of the vanilla DP-SGD (DP-SGD with ReLU) and TanhAct (DP-SGD with Tanh) respectively (in Figure 4). We observe that the GroupNorm layer improves the accuracy of the model overall. However, the improvement gap shrinks as the privacy budget increases when using Tanh as an activation function, e.g. DP-SGD (Tanh) w/o GroupNorm outperforms DP-SGD (Tanh) with GroupNorm when the privacy budget is greater than 10. The connection between the activation function and the normalization layer needs further exploration.

Model Training. Figure 3c illustrates the accuracy comparison of algorithms in the model training category and vanilla DP-SGD. When the privacy budget is large, the accuracy of GEP exceeds the baseline in some settings (e.g. $\langle \text{GEP}, \text{ResNet}, \text{CIFAR-10}, 1000 \rangle$) because of leveraging public data.

When the privacy budget is small, RGP is the only algorithm in this category to achieve acceptable performance on VGG. Model parameter dimensionality reduction is an effective technique to solve large models’ inability to adapt to DP. Nevertheless, there is a significant performance degradation when the privacy budget is large for RGP. We suspect the reason is that reparameterization not only reduces the noise scale in private training but also leads to information loss in the gradient. When the noise scale is small, the information loss caused by reparametrization is higher than the mitigation effect on noise perturbation.

Table 8 in Appendix D reports the accuracy of RGP (w/o DP) and vanilla SGD, and the difference between them is whether using reparameterization. We train models by using RGP (w/o DP) and vanilla SGD, respectively, and the difference between them is whether using reparameterization. Overall, the accuracy of RGP (w/o DP) is lower than that of SGD under the same settings across all datasets and model architectures. The results can be found in Table 8 in Appendix D. The results echo our previous speculation that reparameterization reduces noise scale in private training but

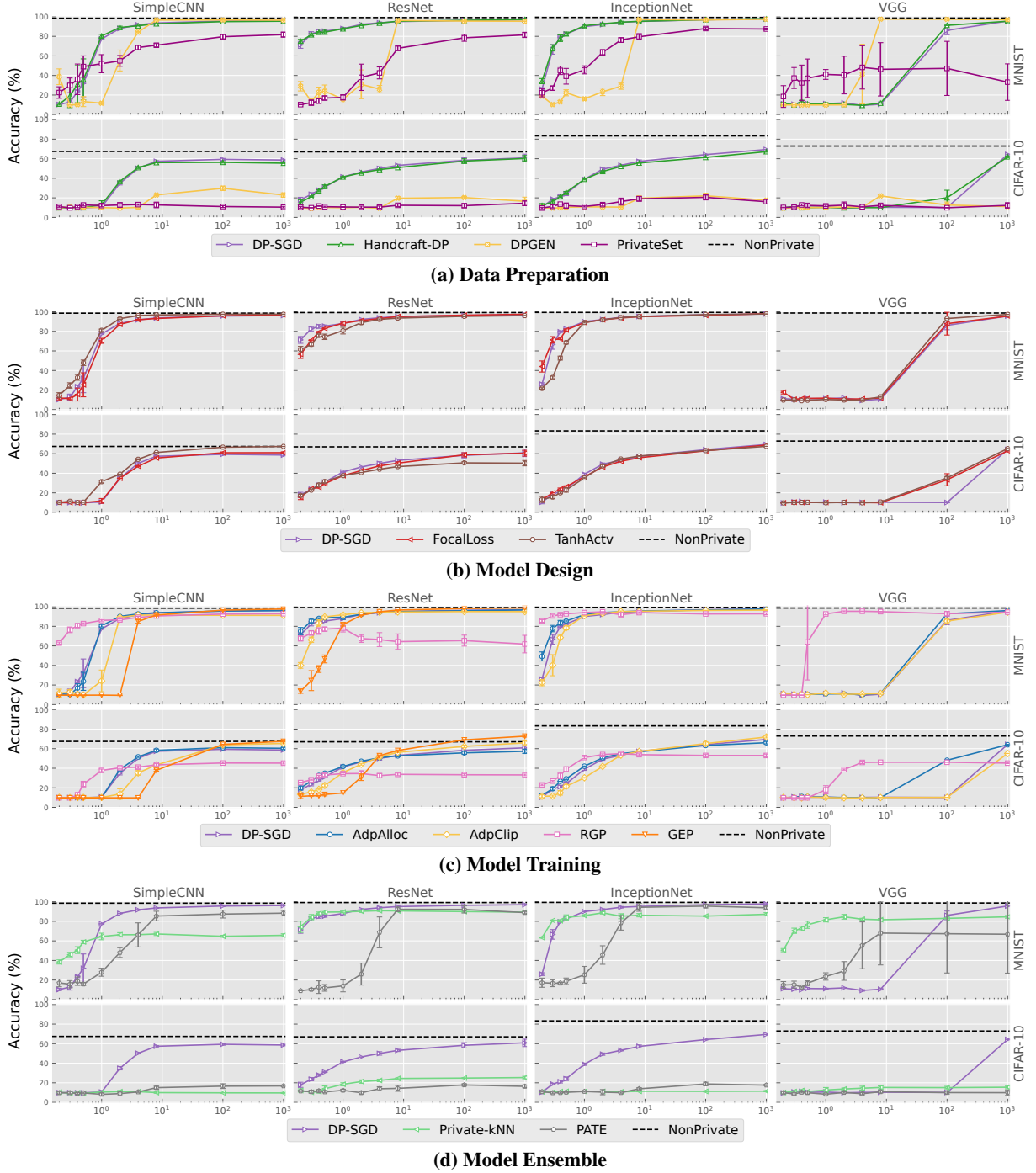


Figure 3: Accuracy comparison of the DPML algorithms in four categories, where the x-axis represents privacy budgets.

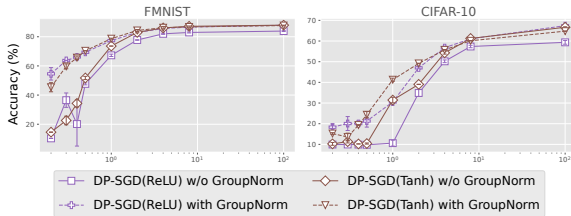


Figure 4: Accuracy of SimpleCNN models with/without GroupNorm layer trained by DP-SGD with ReLU and Tanh activation function across varies privacy budget.

impairs performance in non-private settings.

Model Ensemble. Figure 3d illustrates the accuracy of the algorithms in the model ensemble category and vanilla DP-

SGD.

Note that Priv-kNN and PATE use noise screening technique [10, 38], which ignores the data with low confidence in teacher ensembles to improve the utility-privacy tradeoff. We do not use this technique in our implementation because the privacy budget is given in our settings and the noise scale is precomputed, which requires a fixed number of queries.

When implemented on VGG, Priv-kNN can preserve an equivalent performance as other models, whereas PATE's performance plunges to random guesses. A large number of teachers can impair the noise effect, while the amount of data allocated to each teacher model is too small for a large model such as VGG to converge. The results echo the introduction

Table 4: Overview of algorithms’ tailored AUC in black-box MIA on different model architectures and privacy budgets. In every setting, we bold the value with the best performance (with the smallest value). The experimental results for GEP on InceptionNet and VGG are unavailable due to memory limits.

| | | SimpleCNN | | | | | ResNet | | | | | VGG | | | | |
|----------|-----------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|------------------------|
| | | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 |
| CIFAR-10 | Hand-DP | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.01 | 0.53 \pm 0.00 | 0.53 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.52 \pm 0.00 | 0.53 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.53 \pm 0.01 |
| | PrivSet | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.50 \pm 0.00 |
| | DPGEN | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.01 |
| | TanhAct | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.53 \pm 0.00 | 0.53 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.52 \pm 0.00 | 0.55 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.54 \pm 0.00 |
| | FocalLoss | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.51 \pm 0.00 | 0.52 \pm 0.00 | 0.53 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.52 \pm 0.00 | 0.52 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.53 \pm 0.00 |
| | DP-SGD | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.01 | 0.53 \pm 0.00 | 0.53 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.52 \pm 0.00 | 0.53 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.53 \pm 0.01 |
| | RGP | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 |
| | GEP | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.52 \pm 0.00 | 0.57 \pm 0.00 | 0.51 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.53 \pm 0.00 | 0.54 \pm 0.00 | - | - | - | - |
| | AdpAlloc | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.53 \pm 0.00 | 0.53 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.52 \pm 0.00 | 0.53 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.54 \pm 0.00 |
| | AdpClip | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.53 \pm 0.00 | 0.56 \pm 0.00 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.52 \pm 0.00 | 0.54 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.53 \pm 0.00 |
| | PATE | 0.50 \pm 0.00 | 0.52 \pm 0.01 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.51 \pm 0.00 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.01 | 0.51 \pm 0.01 | 0.51 \pm 0.01 | 0.50 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 |
| | Priv-kNN | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.50 \pm 0.01 | 0.50 \pm 0.00 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 | 0.51 \pm 0.01 | 0.50 \pm 0.00 | 0.50 \pm 0.00 |

in Section 3.5, PATE is hard to get a good trade-off on the number of teacher models.

When implemented on other model architectures, Priv-kNN outperforms PATE with a low privacy budget and vice versa with a high privacy budget. *PATE and Priv-kNN both show higher accuracy at some specific settings [9, 10]. However, they both fail to obtain a better utility-privacy trade-off than vanilla DP-SGD at most settings in our measurements. We suspect that semi-supervised training techniques introduce more randomness and require fine-grained hyperparameter tuning, which leads to a high standard deviation as our experimental results show.*

5.3 Evaluation on Defensive Capabilities

We report the tailored AUC of the black-box MIAs on CIFAR-10 in Table 4, and put the results of other datasets and more settings into Appendix D for having the similar trends (Table 10 for more datasets under black-box MIAs, Table 11 for white-box MIAs). Note that the tailored AUC of attacking the non-private model on the MNIST dataset is already very close to 0.5, so we omit the results on MNIST in this section. Generally, all algorithms’ tailored AUC is around 0.5, which means a strong defense against the MIA compared to the baseline results Table 2.

Figure 5 illustrates the privacy leakage of models trained by algorithms in a per-category manner. Compared to vanilla DP-SGD, the modification of RGP and FocalLoss change the feature of confidence vectors, resulting in training and testing data having a different distribution for the attack model. Thus, RGP and FocalLoss have a remarkable advantage over black-box and white-box attacks in general. Refer to Figure 5d and Figure 5a. We observe that PATE, Priv-kNN, DPGEN, and PrivSet remain nearly free of privacy leakage. It is because the target models do not access private data. PATE and Priv-kNN use the knowledge transferred from teacher ensemble, and DPGEN and PrivSet only access generated data.

Role of Sensitivity-bounding Techniques. To explore the role of sensitivity-bounding techniques in defending MIAs, we conduct attacks on a model trained with normal SGD and per-sample clipping to explore the impact of per-sample clipping on the defense. The results are shown in Table 5.

We observe that the per-sample clipping has a strong defense ability against MIAs with acceptable accuracy degradation compared to the non-private model. Moreover, the

Table 5: Impact of per-sample clipping on model utility and defense to attacks. The table reports the accuracy and the AUC of models on CIFAR-10 with different privacy guarantees. Inf indicates normal SGD; Inf (Clip) denotes normal SGD with per-sample clipping.

| | | 8 | 100 | 1000 | Inf (clip) | Inf |
|--------------|---------|-------|-------|-------|------------|-------|
| SimpleCNN | ACC (%) | 58.20 | 60.66 | 60.44 | 57.96 | 69.22 |
| | AUC | 0.52 | 0.52 | 0.53 | 0.52 | 0.78 |
| ResNet | ACC (%) | 53.80 | 61.50 | 65.90 | 57.42 | 69.70 |
| | AUC | 0.51 | 0.52 | 0.53 | 0.54 | 0.65 |
| InceptionNet | ACC (%) | 58.00 | 64.60 | 69.40 | 72.80 | 83.68 |
| | AUC | 0.51 | 0.51 | 0.52 | 0.58 | 0.71 |
| VGG | ACC (%) | 10.36 | 10.02 | 64.66 | 67.72 | 71.36 |
| | AUC | 0.50 | 0.50 | 0.52 | 0.59 | 0.78 |

defensive effects and accuracy degradation are model dependent. For example, *Inf (clip)* performs comparably to $\epsilon = 8$ on SimpleCNN, but when applied to other models, the performance is worse than when $\epsilon = 1000$.

We suspect the reason why the per-sample clipping technique can defend against MIAs is that it reduces the overfitting of the model. During the training process, applying gradient descent without clipping guides the model to the direction that overfits the training samples; while clipping the gradient makes the model move more conservatively and less overfit to the training samples. Note that the models trained by SGD with per-sample clipping have a defense ability against MIAs but do not satisfy the DP guarantee.

5.4 The Role of the Architecture

Architecture Complexity. According to baseline accuracy in Table 2, the model’s performance can be ordered as InceptionNet > VGG \approx ResNet > SimpleCNN.

Architecture versus Utility Loss. To figure out the impact of model architecture on algorithm performance, we illustrate the boxplot for the utility loss overall algorithms, network, and dataset jointly vary with the privacy budget as Figure 6a.

We observe that the utility loss is similar for ResNet and InceptionNet across different privacy budgets. When the privacy budget is small ($\epsilon \leq 1$), the performance of SimpleCNN and VGG is worse than that of ResNet and InceptionNet. As the noise amount becomes smaller ($\epsilon > 1$), the performance gap between SimpleCNN, ResNet, and InceptionNet narrows. The performance of VGG, the largest model in our assessment, is still poor unless perturbed noise is negligible ($\epsilon \geq 100$), while the privacy protection provided by DP is also meaningless. Further, we explored the test accuracy of

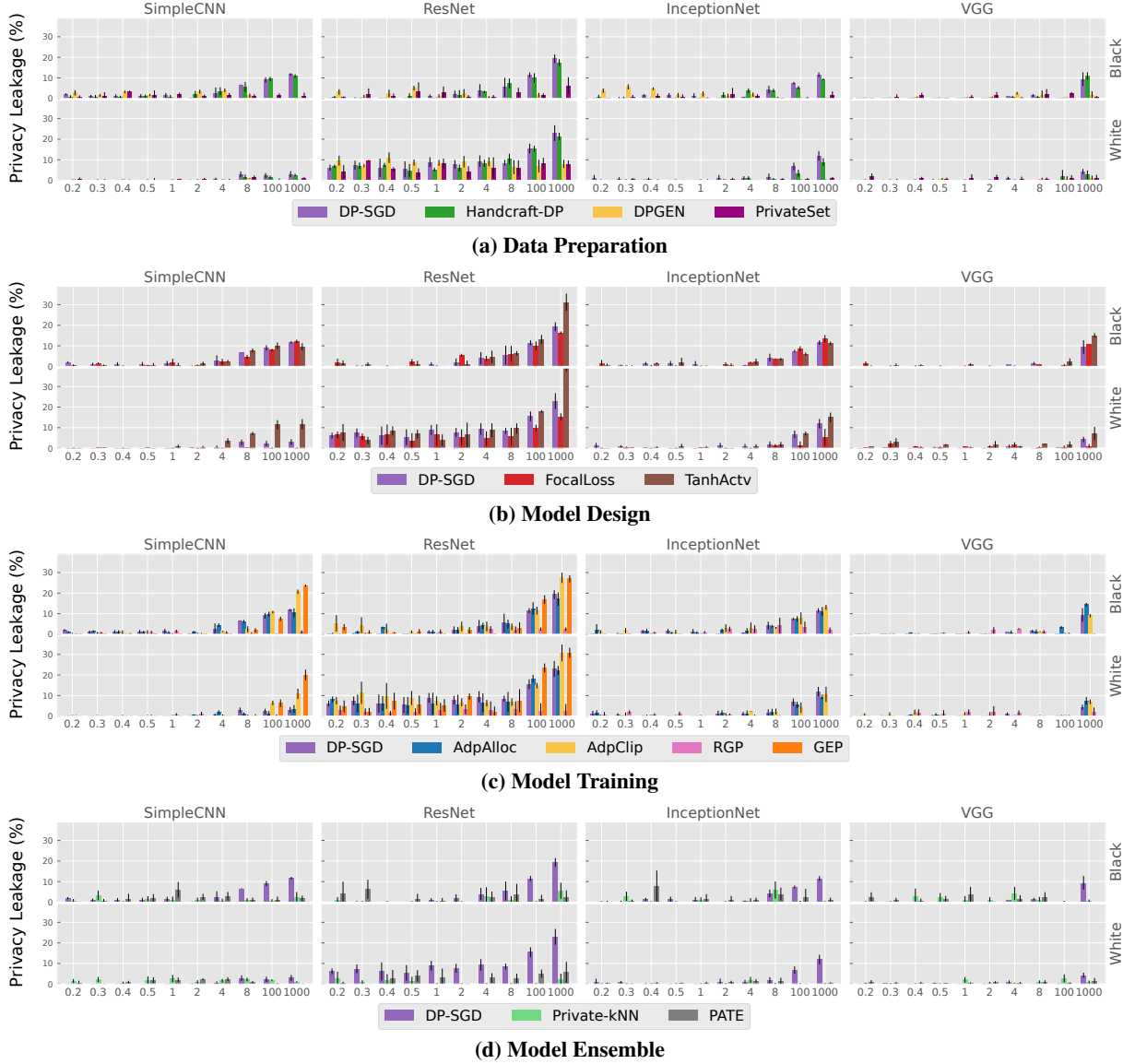


Figure 5: Privacy leakage (under MIA) of DPML algorithms in four categories when given different privacy budgets.

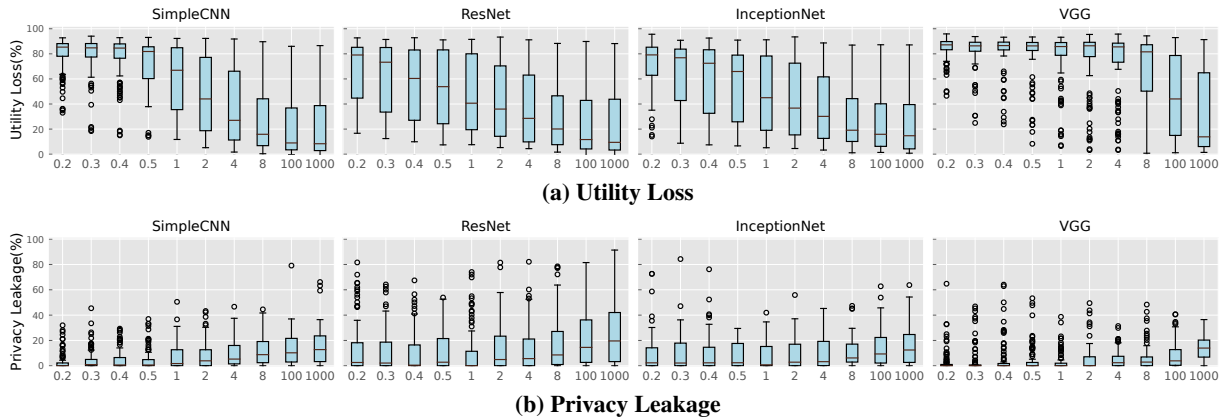
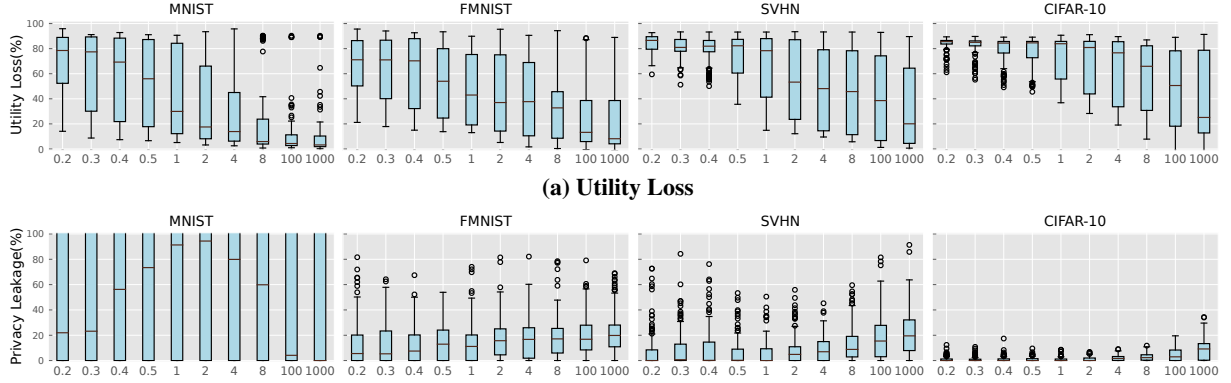


Figure 6: Boxplot of utility loss and privacy leakage on all DPML algorithms with various privacy budgets and four network architectures .

ResNet with different numbers of parameters under different privacy budgets. Due to space limitations, detailed results can be viewed at [Figure 9](#) in [Appendix D](#). Generally, the smaller the privacy budget and the more model parameters, the worse

the model accuracy when training with vanilla DP-SGD.

Architecture versus Privacy Leakage. We also present a boxplot for the privacy leakage of all algorithms on different network architectures across privacy budgets as [Figure 6b](#).



(b) Privacy Leakage. The tailored AUCs of MIAs on MNIST is around 0.5, whether with or without DP, which leads to privacy leakage close to 100%.

Figure 7: Boxplot of utility loss and privacy leakage on all DPML algorithms with various privacy budgets for different datasets.

We observe no strong correlation between privacy leakage and model architecture. VGG has the lowest privacy leakage because many algorithms fail to converge on VGG, leading to the following attack failure.

5.5 The Role of the Datasets

Dataset Complexity. As mentioned before, we resize all the samples in each dataset to 32×32 pixels. MNIST and FMNIST are simpler than SVHN and CIFAR10 as they only contain gray-scale images. When the number of channels is the same, MNIST and SVHN are easier than FMNIST and CIFAR10, respectively, because the contents of MNIST and FMNIST are digital numbers. The accuracy of baseline models in Table 2 shows the same conclusion.

Dataset versus Utility Loss. To explore the impact of the dataset on the DPML algorithm, we plot the relationship between dataset complexity and model utility loss in Figure 7a.

As shown in the plots, the algorithm’s performance on these datasets is correlated with the dataset complexity, with worse performance on the harder dataset. Even with a very large privacy budget ($\epsilon = 100$), nearly half of the private models had a utility loss of more than 30% on CIFAR10 compared to the non-private setting.

Dataset versus Privacy Leakage. We plot the relationship between dataset complexity and model privacy leakage in Figure 7b. We observe that more complex datasets lead to less privacy leakage. One reason is that a complex dataset is harder to converge under private settings, and attackers cannot obtain enough information to infer. Additionally, more complex datasets lead to better MIA performance [59] under non-private settings, leading to a smaller privacy leakage value. The tailored AUCs of MIAs on MNIST is around 0.5, whether with or without DP, which leads to privacy leakage close to 100%.

5.6 Comparison with Label DP

Label Differential Privacy (Label DP) is a variant of DP where the data labels are considered sensitive and must be protected. The definition of label differential privacy is:

Definition 5.1. (Label Differential Privacy). A randomized

training algorithm M taking a dataset as input is said to be (ϵ, δ) -label differentially private, if for any two training datasets D and D' that differ in the label of a single example,

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] + \delta.$$

If $\delta = 0$, then M is said to be ϵ -label differentially private (ϵ -LabelDP). Label DP and DP synthetic algorithms share similar paradigms but differ in generating synthetic datasets by satisfying Label DP instead of standard DP. Our evaluation covers two state-of-the-art Label DP algorithms: LP-MST [14] and ALIBI [63], to explore the difference between Label DP and standard DP algorithms. It is worth noting that the Label-DP satisfies bounded DP. We convert the privacy budget for equivalence while comparing it with other algorithms, and the figure shows the privacy budget in unbounded DP (e.g. $\langle \text{RGP}, \text{ResNet}, \text{CIFAR-10}, 1000 \rangle$ and $\langle \text{LP-MST}, \text{ResNet}, \text{CIFAR-10}, 2000 \rangle$ share the same horizontal coordinate, 1000). The concrete algorithm description can be found in Appendix C

Figure 8a illustrates the comparison of accuracy between Label DP algorithms and vanilla DP-SGD, TanhAct, RGP, Priv-kNN, and DPGEN. We notice that the accuracy of LP-MST and ALIBI can approach or even exceed baseline when the privacy budget is not very large, e.g. the accuracy of $\langle \text{LP-MST}, \text{ResNet}, \text{CIFAR-10}, 4 \rangle$ is 71.82 larger than the baseline of 66.56. There are two reasons behind this. One is that noise only affects labels. The training process gradually becomes the same as non-private training as the private budget increase. The other is that the techniques used to mitigate the effects of wrong labels usually also improve the model’s generalization, such as mixup [64] used in LP-MST [14].

Figure 8b illustrates the comparison of black-box MIA on Label DP algorithms and vanilla DP-SGD, TanhAct, RGP, Priv-kNN, and DPGEN with the metric of privacy leakage. We observe that Label DP algorithms have higher privacy leakage than standard DP algorithms, which is natural for Label DP because of no protection provided to data.

5.7 Takeaways

In the following, we summarize important insights obtained from our measurements and provide some actionable advice to future DPML practitioners.

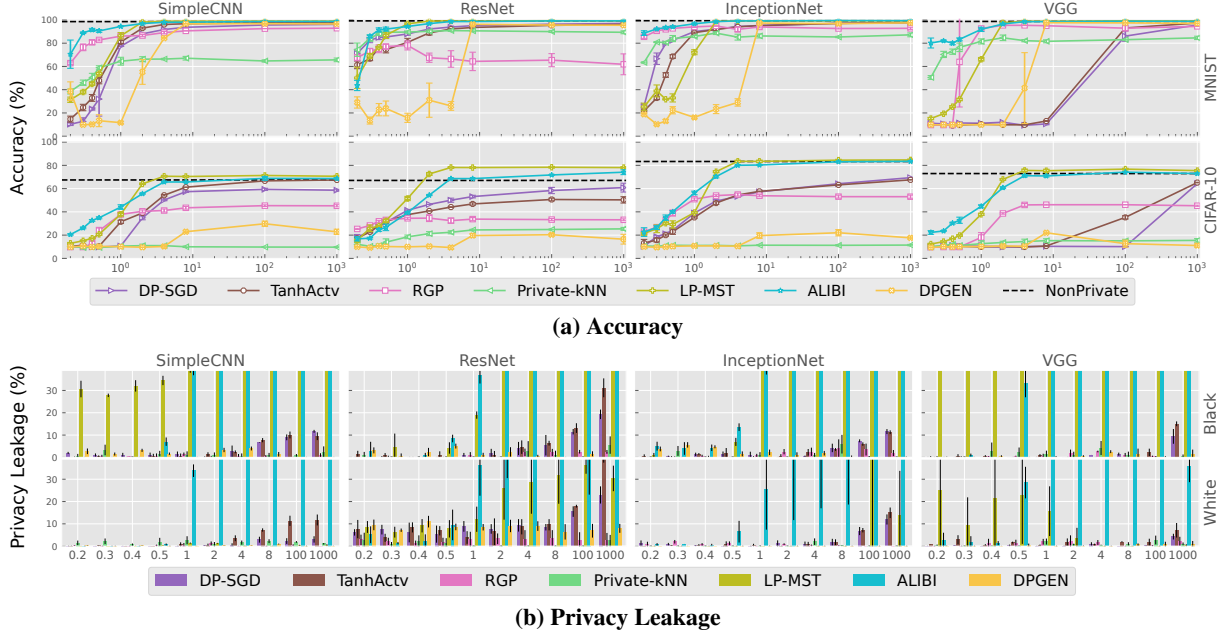


Figure 8: Comparison of Label DP algorithms (LP-MST, ALIBI) and vanilla DP-SGD, TanhAct, RGP, Priv-kNN, and DPGEN under different privacy budgets.

- Different improvement techniques can affect the privacy-utility trade-offs of the algorithm from different perspectives. Concretely, parameter dimension reduction in the model training category improves the performance of DPML on large models but impairs utility when the privacy budget is large. Thus, RGP is a good choice for those who want to provide a DP guarantee for large models. On the other hand, algorithms in the model ensemble category and DP synthetic algorithms can be used when stronger defense against MIAs is desired. However, more effort on manual data filtering for DP synthetic algorithms is needed for better utility.
- In general, the DPML algorithms provide an effective defense against practical MIAs in both black-box and white-box manner. The defense performance hardly decreases when the privacy budget increases. The reason is that sensitivity-bounding techniques such as gradient clipping play an important role in defense. More specifically, improved algorithms that do not directly access private data are better at defending against attacks, such as algorithms in the model ensemble category and DP synthetic algorithms. In addition, improved algorithms that affect the attack features of MIAs can achieve additional defensive capabilities. For instance, the confidence vector distribution of FocalLoss is different from that of shadow models, which causes FocalLoss to be more robust to attacks. All algorithms that provide the standard DP guarantee can defend MIAs effectively.
- Some model architecture design choices for non-private ML models are ineffective for private ML models. More specifically, a large model scale degrades utility for most DPML algorithms. In addition, using Tanh and GroupNorm can reduce the utility loss on vanilla DP-SGD. However, we also find that using both Tanh and GroupNorm has a negative effect. What model architectures are suitable for DPML is still a research question to be explored. When

applying DP to ML models, ResNet and InceptionNet are preferred architectures to attempt.

- In general, learning data distribution from more complex datasets is more difficult than that from easier datasets for all DP algorithms. Compared with the non-private setting, applying DP makes it even more difficult to learn from complex datasets. Leveraging external datasets (*e.g.*, pre-train on public dataset [6] and public data embedding [31]) can be helpful to improve the utility of the model on complex datasets. Therefore, designing DPML algorithms to better learn from complex datasets is an interesting future research direction.
- Label DP algorithms achieve better model utility than standard DP algorithms, which is expected since label DP algorithms loosen the constraint on adjacent datasets. However, the defense effectiveness of label DP algorithms is worse than that of standard DP algorithms since they only protect the privacy of the label instead of the privacy of the training sample. Label DP should only be used when the label is sensitive, not the data itself, and there is no need to defend against MIAs.

6 Discussion

In this section, we discuss several potential research directions to inspire interested readers to explore relevant domains.

Ensembled DPML Algorithms. As discussed in Section 3.1, the improved DPML algorithms in different phases of our taxonomy are independent of each other; thus, one interesting future work is to combine the improvements in different phases to achieve better performance. Shamsabadi *et al.* [8] take the first step and show that combining a hand-crafted feature extractor[12] in the data preparation phase and optimal loss function in the model design phase can

effectively improve the model utility. It would be exciting to follow our taxonomy and combine algorithms at different phases to achieve even better performance.

Extension to Other Domains. Our current measurement primarily focuses on image classification tasks, it would be interesting to leverage DPMLBench to measure the performance of DPML algorithms in other domains, such as natural language processing (NLP) and graph neural networks (GNN).

DPML Algorithms for Large Models. With the development of deep learning, the model scale increases rapidly, especially in the NLP field. For instance, the famous GPT-3 model contains 175B parameters [65]. However, our measurements show that most of the current DPML algorithms suffer from low model utilities. Furthermore, DP-SGD-based algorithms require calculating per-sample clipping of the gradients, which significantly increases the training time and memory consumption. Therefore, designing high-utility and efficient DPML algorithms for large models is of significant importance in the future.

7 Related Work

Differential Privacy. Differential privacy (DP) [34, 36] is a widely used rigorous mathematical definition to formalize and measure privacy guarantees based on a parameter called *privacy budget*. It has been adopted for a number of data analysis tasks, such as synthetic dataset generation [66, 67, 68, 69], marginal release [70], range query [71], and stream data analysis [72]. Some studies propose integrating DP with traditional machine learning algorithms, such as naive Bayes and Linear Support Vector Machine (SVM) [73, 74, 75]. Abadi *et al.* propose vanilla DP-SGD [6] as the first general DPML algorithm. Recent studies try to mitigate DP’s impairment on utility by proposing new algorithms [9, 12, 31, 10] or relax DP definition for specific scenarios [76, 14, 77].

Membership Inference Attacks. The adversary in MIAs aims to infer whether a given data sample is used to train the target model. Currently, the MIA is one of the critical methods to assess the privacy risk of ML models [3, 39, 40, 78, 79, 80]. According to the accessibility to the target model, the MIA can be categorized into black-box and white-box attacks. Shokri *et al.* [3] propose the first black-box MIA against ML models. They propose to train multiple shadow models to simulate the behavior of the target model and use shadow models to generate the data used to train the attack model. Salem *et al.* [39] simplify their method by using one shadow dataset and one shadow model. Nasr *et al.* [40] first propose white-box MIAs, where the adversary knows the internal parameters of the target model.

DPML Measurement. Several DPML measurement studies concentrate on different perspectives [16, 17, 18, 19]. Jayaraman *et al.* [16] analyzed the difference of privacy leakage of relaxed variants of differential privacy. They explore the difference in privacy leakage when using the same algorithm with different DP definitions. Iyengar *et al.* [17] evaluate several differentially private convex optimization algo-

ritms. The work of Zhao *et al.* [18] and Jarin *et al.* [19] analyze the performance of naive noise perturbation in different stages of the training pipeline.

ML-Doctor [59] also investigates the defenses and attacks against ML models. However, we have different objectives. ML-Doctor aims to evaluate the effectiveness of different types of defenses against attacks. For DPML, they only evaluate the vanilla DP-SGD, and their only conclusion is that DP-SGD can defend against MIAs while failing for other attacks without considering the impact on model utility. On the other hand, DPMLBench conducts more fine-grained taxonomy and evaluation on different DPML algorithms and aims to evaluate the trade-off between model utility, privacy guarantee, and defense effectiveness. This can better facilitate future research on DPML. As such, we obtained more insights on how to design proper DPML algorithms to trade off the above triangle, as stated in Section 5.7.

8 Conclusion

This paper establishes a taxonomy of improved DPML algorithms along the ML life cycle for four types: data preparation, model design, model training, and model ensemble. Based on taxonomy, we propose the first holistic measurement of improved DPML algorithms’ performance on utility and defense capability against MIAs on image classification tasks. Our extensive measurement study covers twelve DPML algorithms, two attacks, four model architectures, four datasets, and various privacy budget configurations. We also cover state-of-the-art label DP in the evaluation.

Among other things, we found that different improvement techniques can affect the privacy-utility trade-off of the algorithm from different perspectives. We also show that DP can effectively defend against MIAs and sensitivity-bounding techniques such as per-sample gradient clipping play an important role in defense. Moreover, some model architecture design choices for non-private ML models are ineffective for private ML models. In addition, label DP has less utility loss but is fragile to MIAs.

We implement a modular re-usable software, DPML-Bench, which contains all algorithms and attacks. DPML-Bench enables sensitive data owners to deploy DPML algorithms and serves as a benchmark tool for researchers and practitioners. Currently, while DPMLBench focuses on image classification models, we plan to extend other types of DP models, such as language models [81, 82], graph neural networks [83, 84, 85], and generative models [86, 44].

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China (NSFC) under No. 62302441, the Funding for Postdoctoral Scientific Research Projects in Zhejiang Province (ZJ2022072), and ZJU – DAS-Security Joint Research Institute of Frontier Technologies, the Helmholtz Association within the project “Trustworthy Federated Data Analytics” (TFDA) (No. ZT-I-OO1 4), and CISA-Stanford Center for Cybersecurity (FKZ:13N1S0762).

References

- [1] Guodong Guo and Na Zhang. A survey on deep learning based face recognition. *Computer vision and image understanding*, 189:102805, 2019.
- [2] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neuro-computing*, 300:17–33, 2018.
- [3] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.
- [4] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.
- [5] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.
- [6] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.
- [7] Da Yu, Huishuai Zhang, Wei Chen, Jian Yin, and Tie-Yan Liu. Large scale private learning via low-rank reparametrization. In *International Conference on Machine Learning*, pages 12208–12218. PMLR, 2021.
- [8] Ali Shahin Shamsabadi and Nicolas Papernot. Losing less: A loss for differentially private deep learning. 2021.
- [9] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *International Conference on Learning Representations*, 2017.
- [10] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. Private-knn: Practical differential privacy for computer vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020.
- [11] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9312–9321, 2021.
- [12] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2020.
- [13] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 332–349. IEEE, 2019.
- [14] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. Deep learning with label differential privacy. *Advances in Neural Information Processing Systems*, 34, 2021.
- [15] Jia-Wei Chen, Chia-Mu Yu, Ching-Chia Kao, Tzai-Wei Pang, and Chun-Shien Lu. Dpgen: Differentially private generative energy-guided network for natural image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8387–8396, 2022.
- [16] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, 2019.
- [17] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.
- [18] Benjamin Zi Hao Zhao, Mohamed Ali Kaafar, and Nicolas 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.
- [19] Ismat Jarin and Birhanu 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.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document

- recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [24] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [26] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [27] Xu Chu, Ihab F Ilyas, Sanjay Krishnan, and Jianan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206, 2016.
- [28] Teodor Fredriksson, David Issa Mattos, Jan Bosch, and Helena Holmström Olsson. Data labeling: An empirical investigation into industrial challenges and mitigation strategies. In *International Conference on Product-Focused Software Process Improvement*, pages 202–216. Springer, 2020.
- [29] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*, pages 372–378. IEEE, 2014.
- [30] Nikolay Chumerin and Marc M Van Hulle. Comparison of two feature extraction methods based on maximization of mutual information. In *2006 16th IEEE signal processing society workshop on machine learning for signal processing*, pages 343–348. IEEE, 2006.
- [31] Da Yu, Huishuai Zhang, Wei Chen, and Tie-Yan Liu. Do not let privacy overbill utility: Gradient embedding perturbation for private learning. In *International Conference on Learning Representations (ICLR)*, 2021.
- [32] Yingxue Zhou, Steven Wu, and Arindam Banerjee. Bypassing the ambient dimension: Private SGD with gradient subspace identification. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.
- [33] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research*, 11:169–198, 1999.
- [34] Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, 2008.
- [35] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.
- [36] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.
- [37] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*, pages 263–275. IEEE, 2017.
- [38] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.
- [39] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.
- [40] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.
- [41] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.
- [42] Alexey Kurakin, Steve Chien, Shuang Song, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328*, 2022.
- [43] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. *Advances in Neural Information Processing Systems*, 33:12673–12684, 2020.
- [44] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.
- [45] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. Private set generation with discriminative information. In *Neural Information Processing Systems (NeurIPS)*, 2022.
- [46] Anda Cheng, Jiaying Wang, Xi Sheryl Zhang, Qiang Chen, Peisong Wang, and Jian Cheng. Dpnas: Neural architecture search for deep learning with differential privacy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6358–6366, 2022.
- [47] Venkatadheeraj Pichapati, Ananda Theertha Suresh, Felix X Yu, Sashank J Reddi, and Sanjiv Kumar. Ada-clip: Adaptive clipping for private sgd. *arXiv preprint arXiv:1908.07643*, 2019.

- [48] Galen Andrew, Om Thakkar, Brendan McMahan, and Swaroop Ramaswamy. Differentially private learning with adaptive clipping. *Advances in Neural Information Processing Systems*, 34:17455–17466, 2021.
- [49] Edouard Oyallon and Stéphane Mallat. Deep roto-translation scattering for object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2865–2873, 2015.
- [50] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.
- [51] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and Fujie Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.
- [52] Laurent Younes. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999.
- [53] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.
- [54] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [55] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.
- [56] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. Opacus: User-friendly differential privacy library in pytorch. *arXiv preprint arXiv:2109.12298*, 2021.
- [57] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456, 2015.
- [58] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.
- [59] Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. MI-doctor: Holistic risk assessment of inference attacks against machine learning models. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 4525–4542, 2022.
- [60] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*, 2019.
- [61] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.
- [62] Nicolas Papernot and Thomas Steinke. Hyperparameter tuning with renyi differential privacy. In *International Conference on Learning Representations*, 2021.
- [63] Mani Malek Esmaeili, Ilya Mironov, Karthik Prasad, Igor Shilov, and Florian Tramer. Antipodes of label differential privacy: Pate and alibi. *Advances in Neural Information Processing Systems*, 34, 2021.
- [64] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.
- [65] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [66] Quan Yuan, Zhikun Zhang, Linkang Du, Min Chen, Peng Cheng, and Mingyang Sun. PrivGraph: Differentially Private Graph Data Publication by Exploiting Community Information. In *USENIX Security*, 2023.
- [67] Haiming Wang, Zhikun Zhang, Tianhao Wang, Shibo He, Michael Backes, Jiming Chen, and Yang Zhang. PrivTrace: Differentially Private Trajectory Synthesis by Adaptive Markov Model. In *USENIX Security*, 2023.
- [68] Zhikun Zhang, Tianhao Wang, Ninghui Li, Jean Honorio, Michael Backes, Shibo He, Jiming Chen, and Yang Zhang. PrivSyn: Differentially Private Data Synthesis. In *USENIX Security*, 2021.
- [69] Yuntao Du, Yujia Hu, Zhikun Zhang, Ziquan Fang, Lu Chen, Baihua Zheng, and Yunjun Gao. LDPTTrace: Locally Differentially Private Trajectory Synthesis. In *VLDB*, 2023.
- [70] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. CALM: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy. In *ACM CCS*, 2018.
- [71] Linkang Du, Zhikun Zhang, Shaojie Bai, Changchang Liu, Shouling Ji, Peng Cheng, and Jiming Chen. AHEAD: Adaptive Hierarchical Decomposition for Range Query under Local Differential Privacy. In *ACM CCS*, 2021.

[72] Tianhao Wang, Joann Qionga Chen, Zhikun Zhang, Dong Su, Yueqiang Cheng, Zhou Li, Ninghui Li, and Somesh Jha. Continuous Release of Data Streams under both Centralized and Local Differential Privacy. In *ACM CCS*, 2021.

[73] Jaideep Vaidya, Basit Shafiq, Anirban Basu, and Yuan Hong. Differentially private naive bayes classification. In *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 571–576. IEEE, 2013.

[74] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. *Advances in neural information processing systems*, 21, 2008.

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

[76] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. Privacy: Theory meets practice on the map. In *2008 IEEE 24th international conference on data engineering*, pages 277–286. IEEE, 2008.

[77] Cynthia Dwork. Differential privacy in new settings. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 174–183. SIAM, 2010.

[78] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When Machine Unlearning Jeopardize Privacy. In *ACM CCS*, 2021.

[79] Hai Huang, Zhikun Zhang, Yun Shen, Michael Backes, Qi Li, and Yang Zhang. On the Privacy Risks of Cell-Based NAS Architectures. In *ACM CCS*, 2022.

[80] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, and Yang Zhang. FACE-AUDITOR: Data Auditing in Facial Recognition Systems. In *USENIX Security*, 2023.

[81] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. In *International Conference on Learning Representations (ICLR)*, 2022.

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

[83] Zhikun Zhang, Min Chen, Michael Backes, Yun Shen, and Yang Zhang. Inference Attacks Against Graph Neural Networks. In *USENIX Security*, 2022.

[84] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. Graph Unlearning. In *ACM CCS*, 2022.

[85] Yun Shen, Yufei Han, Zhikun Zhang, Min Chen, Ting Yu, Michael Backes, Yang Zhang, and Gianluca Stringhini. Finding MNEMON: Reviving Memories of Node Embeddings. In *ACM CCS*, 2022.

[86] Chaoyang He, Keshav Balasubramanian, Emir Ceyani, Carl Yang, Han Xie, Lichao Sun, Lifang He, Liangwei Yang, Philip S Yu, Yu Rong, et al. Fedgraphnn: A federated learning benchmark system for graph neural networks. In *ICLR 2021 Workshop on Distributed and Private Machine Learning (DPML)*, 2021.

[87] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

A Hyperparameter Settings

Table 6 reports the detailed hyperparameter settings. Settings of DPGEN and PrivSet are for classifier training. We follow the author’s setting for generated algorithms.

Table 6: Detailed hyperparameter settings. Settings of DPGEN and PrivSet are for classifier training. We follow the author’s setting for generated algorithms.

| | Learning Rate | Batch Size | Epoch | Additional |
|----------------|---------------|------------|-------------------------------------|--|
| vanilla DP-SGD | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | |
| TanhAct | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | |
| AdpAlloc | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | ExpDecay k=0.01 |
| AdpClip | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | target.unclipped.quantile=0.7 clipbound_learning_rate=0.1 max_clipbound=10 min_clipbound=0.05 |
| FocalLoss | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | weight_decay=1e-4 |
| Handcrafted | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | |
| GEP | 0.1 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | num_groups=3 num_bases=1000 weight_decay=2e-4 aux_data_size=2000 |
| RGP | 0.1 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | width=1 rank=16 weight_decay=1e-4 |
| PATE | 0.001 | 200 | 500 | n_teacher=100 |
| Priv-kNN | 0.01 | 512 | 500 | iteration=2 sample_prob=0.15 |
| DPGEN | 0.01 | 1024 | 100 | |
| PrivSet | 0.01 | 10 | 300 | samples_per_class=10 |
| LP-MST | 0.01 | 256 | 200 | stage=2 |
| ALIBI | 0.01 | 256 | MNIST,FMNIST:60 SVHN,CIFAR-10:90 | post_process=mapwithprior |

B Dataset Description

- **MNIST** comprises 60000 training samples and 10000 test samples. Each sample is a 28x28 pixel gray handwritten numeral picture.
- **Fashion-MNIST (FMNIST)** has the same size, format, and train /test set division as the MNIST. It covers front images

Table 7: Test accuracy of 5 model architectures on CIFAR-10 when given various privacy budgets.

| | 0.2 | 0.4 | 1 | 4 | 100 | 1000 | Inf |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| MLP | 27.30 | 32.50 | 38.98 | 46.70 | 54.02 | 57.48 | 57.32 |
| SimpleCNN | 9.66 | 9.98 | 10.24 | 51.12 | 57.06 | 55.66 | 65.18 |
| ResNet | 14.78 | 28.04 | 42.02 | 49.36 | 60.80 | 64.58 | 72.16 |
| InceptionNet | 15.08 | 22.14 | 39.54 | 52.22 | 60.78 | 66.42 | 81.58 |
| VGG | 9.70 | 10.16 | 10.32 | 10.22 | 30.08 | 60.90 | 72.26 |

of products from 10 different clothing categories. It has 60000 training samples and 10000 test samples.

- **CIFAR-10** consists of 10 categories of real-world objects of color images, and the size of each picture is 32×32 . There are 50000 training images and 10000 test images in the dataset.
- **Street View House Number (SVHN)** is the house number extracted from the Google Street view image. It can be seen as a colorful and more realistic version of MNIST. It comprises 73257 training samples and 26032 test samples, which are 32×32 RGB images. We trim the testset size to 10000 while keeping distribution consistent with the original testset.

1. **Target Training Dataset** is regarded as private data and member samples while evaluating the performance of MIAs.
2. **Target Testing Dataset** is used to evaluate the utility performance of the model. It is also used to evaluate the performance of MIAs as non-member samples.
3. **Shadow Training Dataset** is used to train shadow models as auxiliary datasets of adversaries and then generate training data as members for attack models.
4. **Shadow Testing Dataset** is used to generate training data as non-members for attack models.

C Details of Label DP Algorithms

- LP-MST Ghazi *et al.* [14] introduced *RRWithPrior*, a Randomized Response (RR) [87] based algorithm, to perform label perturbation, to determine whether the label of each data sample is obtained by the RR mechanism or randomly generated. To mitigate the effects of mislabeling, LP-MST leverages a multi-stage training strategy.
- ALIBI Malek *et al.* [63] provide label DP guarantee by applying additive Laplace noise to a one-hot encoded label. To mitigate the effects of the perturbed label, they apply Bayesian post-processing to the output of the Laplace mechanism to mitigate the effect of mislabeling.

D Additional Results

Results of Hand-DP on MLP. Table 7 shows the results comparison of Hand-DP among 5 model architectures on CIFAR-10.

Impact of Model Parameter Amounts on Accuracy. Figure 9 shows the test accuracy of ResNet with the different

numbers of parameters trained by vanilla DP-SGD under different privacy budgets.

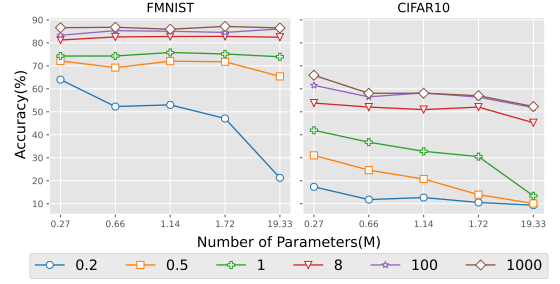


Figure 9: Accuracy of ResNet with the different number of parameters trained by vanilla DP-SGD under different ϵ levels.

Accuracy of RGP without DP. To figure out the effect of reparametrization in a non-private setting, Table 8 reports the accuracy comparison between RGP without DP and vanilla DP-SGD.

Table 8: Accuracy of RGP (w/o DP) and vanilla SGD. Other settings keep the same as Table 3.

| | SimpleCNN | | ResNet | | InceptionNet | | VGG | |
|----------|-------------|--------------|-------------|--------------|--------------|--------------|-------------|--------------|
| | RGP(w/o DP) | SGD | RGP(w/o DP) | SGD | RGP(w/o DP) | SGD | RGP(w/o DP) | SGD |
| MNIST | 95.50 | 98.42 | 97.78 | 99.24 | 99.04 | 99.18 | 98.56 | 98.68 |
| FMNIST | 85.70 | 88.04 | 86.62 | 88.60 | 90.76 | 91.70 | 88.72 | 90.48 |
| SVHN | 73.70 | 87.69 | 90.58 | 93.84 | 93.09 | 94.90 | 88.14 | 89.77 |
| CIFAR-10 | 50.40 | 69.22 | 59.94 | 68.16 | 75.74 | 83.68 | 68.32 | 71.36 |

Accuracy on FMNIST and SVHN. Figure 10 shows the accuracy of DPML algorithms on FMNIST and SVHN in terms of categories. As a supplementary to Figure 3.

Utility loss on FMNIST and SVHN. Table 9 shows the results of utility loss on FMNIST and SVHN as the supplement of Table 3.

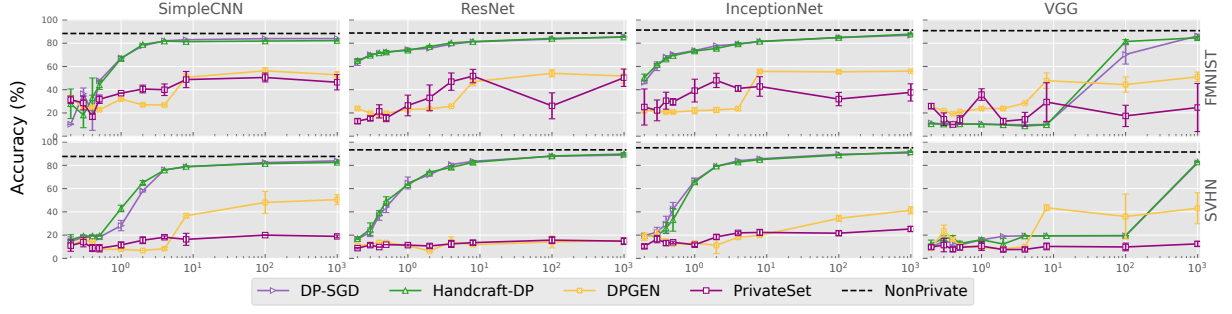
Extra Results of MIAs. Table 11 reports the tailored AUC in white-box style on all model architectures, datasets, and privacy budget. Table 10 reports the tailored AUC in black-box on FMNIST and SVHN.

E Model Architectures

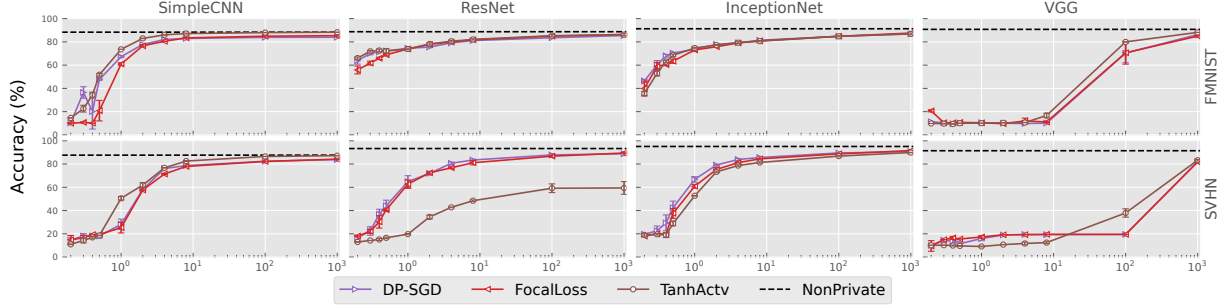
Target Model. Table 12, Table 13, Table 15 and Table 16 show target model architecture, respectively. For simplicity, the details of the block used in the network are shown in Table 14 and Table 17.

Attack Model. We present implementation details of attack models as follows:

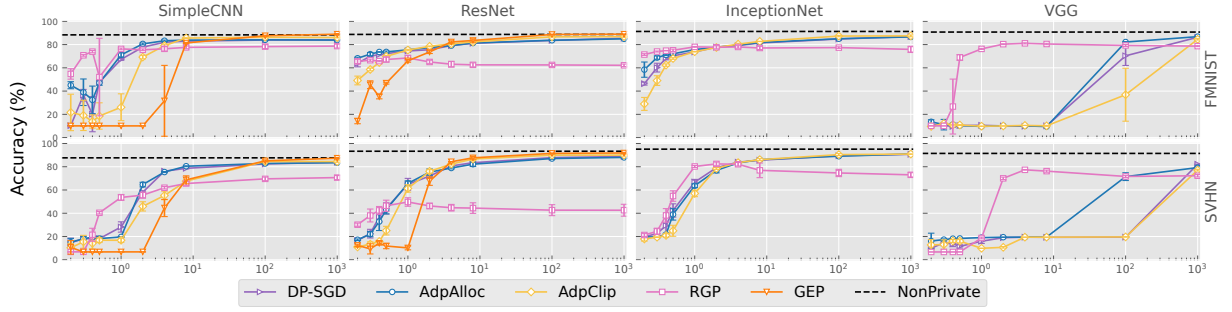
- **Black-Box.** We refer to the model architecture of Liu *et al.* [59]. The attack model receives two inputs: the target sample’s sorted posteriors and a binary indicator on whether the target sample is predicted correctly. The attack model consists of three MLPs (Multi-layer Perceptron). Two processes the input to extract features and concatenated output features are fed into the third MLP to obtain the final prediction.
- **White-Box.** We use a similar model architecture as the one used by Nasr *et al.* [40]. There are four inputs for this attack model, including the target model’s posteriors,



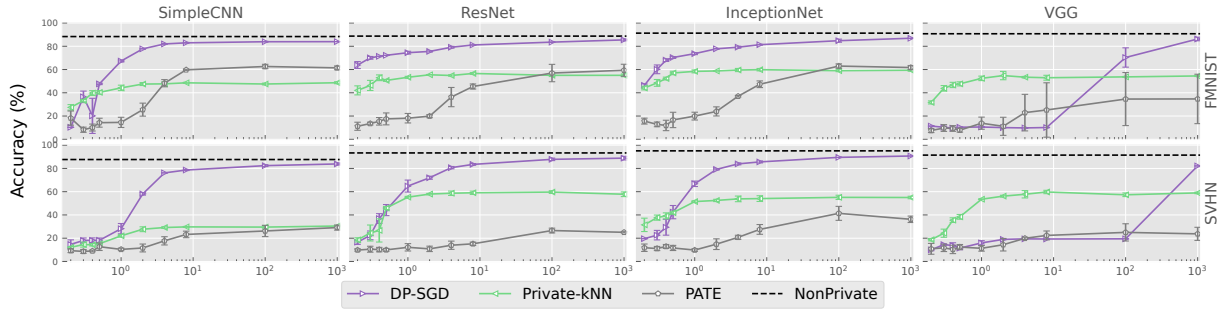
(a) Data Preparation



(b) Model Design



(c) Model Training



(d) Model Ensemble

Figure 10: Accuracy comparison on FMNIST and SVHN. As a supplementary to Figure 3.

classification loss, gradients of the parameters of the target model's last layer, and true labels in one-hot encoding. Each input is fed into a different neural network to extract the features respectively, and then the features are passed to the classifier after concatenation.

Table 9: Overview of algorithms’ utility loss on different model architectures, datasets, and privacy budget. For each privacy budget, we bold the value with the best performance (with the smallest value of utility loss). Supplement to Table 3.

| | | SimpleCNN | | | | | ResNet | | | | | InceptionNet | | | | | VGG | | | | |
|--------|-----------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|--------------------|--------------------|-------------------|
| | | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 |
| FMNIST | DPGEN | 66.83±3.18 | 67.80±0.84 | 72.94±0.77 | 43.30±2.40 | 46.79±2.73 | 75.84±1.15 | 76.69±2.18 | 73.98±0.31 | 45.47±3.09 | 47.87±6.09 | 77.64±2.13 | 77.92±2.63 | 76.09±1.60 | 44.21±1.41 | 43.57±0.23 | 74.17±0.26 | 76.04±1.32 | 71.35±0.31 | 55.22±6.61 | 48.50±4.07 |
| | PrivSet | 68.35±3.18 | 62.69±0.28 | 59.71±4.98 | 49.09±3.80 | 53.22±6.59 | 86.93±2.07 | 73.35±8.89 | 52.70±7.45 | 73.66±1.15 | 49.31±7.46 | 74.68±15.51 | 60.46±9.81 | 58.60±2.53 | 67.81±5.72 | 62.05±7.41 | 73.91±2.25 | 63.89±4.96 | 58.38±5.99 | 82.40±9.16 | 75.13±20.65 |
| | Hand-DP | 71.89±12.70 | 32.68±1.89 | 17.55±0.24 | 17.50±0.25 | 17.12±0.30 | 34.50±1.06 | 25.48±1.10 | 19.12±0.31 | 15.24±0.51 | 14.09±0.36 | 49.09±2.86 | 26.15±1.12 | 19.99±1.00 | 14.44±0.73 | 11.53±0.75 | 89.30±0.07 | 89.65±0.33 | 91.13±0.40 | 17.91±1.62 | 14.67±1.31 |
| | TanHAct | 84.79±1.74 | 26.01±1.07 | 12.85±0.29 | 10.99±0.32 | 11.00±0.08 | 33.38±0.99 | 25.50±0.71 | 18.79±0.76 | 14.00±0.90 | 13.14±0.85 | 64.14±1.88 | 24.70±0.67 | 20.01±0.73 | 14.52±0.48 | 12.42±0.79 | 90.08±0.09 | 89.81±0.06 | 89.31±0.19 | 19.38±0.20 | 11.11±0.04 |
| | FocalLoss | 89.94±0.47 | 38.54±1.12 | 18.85±0.04 | 14.44±0.65 | 13.95±0.43 | 43.65±3.44 | 25.48±0.61 | 19.46±0.65 | 14.37±0.20 | 12.88±0.35 | 59.75±6.38 | 26.32±1.17 | 20.31±1.31 | 14.48±0.95 | 11.72±0.30 | 79.11±0.91 | 89.53±0.47 | 87.89±2.39 | 28.78±9.79 | 14.35±0.71 |
| | DP-SGD | 77.79±17.00 | 31.26±1.65 | 17.24±0.62 | 15.72±0.56 | 15.83±0.58 | 35.65±2.99 | 24.92±1.21 | 20.21±0.77 | 15.76±0.28 | 13.83±0.82 | 53.24±1.26 | 25.75±0.84 | 20.15±0.87 | 14.51±1.33 | 12.40±0.15 | 88.56±0.19 | 89.34±0.25 | 90.15±0.14 | 29.09±8.31 | 13.02±1.54 |
| | AdpAlloc | 54.58±3.02 | 28.31±1.91 | 16.03±0.59 | 15.23±0.67 | 15.45±0.61 | 31.21±0.90 | 23.88±1.02 | 20.28±0.19 | 15.63±0.25 | 14.38±0.05 | 41.08±6.55 | 23.67±0.35 | 19.88±0.48 | 14.29±0.24 | 12.68±0.21 | 86.30±1.46 | 89.32±0.09 | 89.82±0.24 | 17.14±0.34 | 12.48±0.06 |
| | CEP | 89.85±0.05 | 89.85±0.05 | 68.11±0.30 | 11.69±0.46 | 10.35±0.54 | 85.69±2.27 | 33.40±1.20 | 17.09±0.99 | 10.60±0.31 | 10.33±0.20 | 70.79±5.57 | 25.55±0.19 | 19.02±0.82 | 11.97±0.58 | 11.81±0.78 | 90.34±0.37 | 90.13±0.15 | 89.43±0.03 | 62.83±22.77 | 14.81±0.57 |
| | RGP | 44.80±4.49 | 23.10±0.76 | 23.04±1.88 | 21.12±1.65 | 20.67±1.66 | 33.95±1.86 | 30.97±2.00 | 36.36±2.44 | 37.00±1.28 | 37.32±1.08 | 28.00±0.73 | 21.33±0.33 | 21.42±1.10 | 21.98±0.65 | 23.52±2.67 | 89.82±0.05 | 23.08±0.45 | 18.28±0.46 | 20.10±0.85 | 20.70±0.42 |
| | PATE | 81.62±6.09 | 55.35±4.39 | 51.37±3.09 | 36.89±1.88 | 38.07±1.62 | 88.71±3.48 | 81.72±4.08 | 63.40±7.45 | 12.57±7.48 | 10.15±5.16 | 84.24±2.54 | 79.90±3.35 | 62.70±0.67 | 36.52±1.90 | 37.80±1.75 | 92.41±1.76 | 86.05±5.23 | 65.12±22.83 | 62.50±21.25 | |
| SVHN | Priv-kNN | 72.31±2.36 | 55.44±2.34 | 50.44±1.77 | 52.08±0.75 | 50.97±0.83 | 57.59±2.89 | 46.23±0.81 | 44.69±0.54 | 44.55±1.04 | 44.55±1.20 | 81.12±3.22 | 86.86±1.87 | 81.70±1.76 | 63.54±2.70 | 58.32±1.30 | 68.04±1.31 | 47.17±1.93 | 46.16±0.65 | 45.90±0.38 | 45.09±0.49 |
| | DPGEN | 89.23±0.13 | 92.20±0.39 | 90.60±1.13 | 51.49±0.92 | 49.05±1.32 | 88.80±1.01 | 89.02±0.77 | 85.70±4.31 | 85.75±1.11 | 85.08±2.91 | 81.22±3.22 | 86.86±1.87 | 81.70±1.76 | 63.54±2.70 | 58.32±1.30 | 90.93±0.09 | 90.63±0.84 | 89.91±1.43 | 63.70±19.35 | 56.55±13.36 |
| | PrivSet | 88.96±4.87 | 88.36±2.55 | 81.66±1.12 | 78.83±0.29 | 80.94±1.15 | 91.14±0.62 | 88.93±0.74 | 87.49±3.02 | 84.20±2.91 | 85.58±2.90 | 89.60±1.67 | 88.21±1.45 | 77.93±1.06 | 78.17±1.98 | 74.50±2.11 | 92.26±1.85 | 89.34±3.79 | 92.26±1.48 | 90.11±3.31 | 87.44±2.08 |
| | Hand-DP | 83.78±4.33 | 56.72±2.80 | 23.46±0.46 | 17.85±0.20 | 16.77±0.75 | 82.99±0.76 | 36.15±0.51 | 21.09±1.58 | 11.23±0.75 | 9.56±1.07 | 80.24±0.11 | 33.78±0.96 | 16.57±0.83 | 10.44±0.28 | 7.71±0.51 | 87.61±3.42 | 83.85±0.70 | 80.64±0.18 | 80.24±0.14 | 16.50±0.56 |
| | TanHAct | 88.96±0.38 | 49.05±1.51 | 22.44±0.24 | 12.76±0.16 | 12.11±0.22 | 86.90±0.30 | 80.12±0.51 | 56.90±0.35 | 40.32±3.74 | 40.07±5.48 | 80.82±1.03 | 46.38±0.32 | 20.66±0.61 | 12.46±0.66 | 9.35±0.22 | 89.75±0.13 | 90.83±0.39 | 88.21±1.12 | 61.78±3.59 | 16.60±0.68 |
| | FocalLoss | 85.07±3.75 | 74.53±4.55 | 27.98±0.45 | 17.12±0.36 | 15.11±0.27 | 81.79±0.68 | 36.78±3.60 | 22.59±0.74 | 12.54±0.97 | 9.80±0.25 | 81.85±0.41 | 38.60±1.63 | 18.12±0.52 | 10.53±0.18 | 7.67±0.33 | 90.38±4.59 | 82.58±1.02 | 80.60±0.01 | 80.34±0.14 | 17.56±0.52 |
| | DP-SGD | 85.22±3.90 | 71.58±4.43 | 23.23±0.10 | 16.93±0.80 | 15.46±0.95 | 83.27±1.93 | 34.64±5.15 | 18.68±1.01 | 11.47±1.08 | 10.43±1.56 | 80.25±0.16 | 32.68±2.33 | 15.40±0.84 | 9.75±0.16 | 8.58±0.20 | 89.94±0.77 | 83.92±1.92 | 80.42±0.28 | 80.38±0.07 | 17.19±0.13 |
| | AdpAlloc | 84.76±3.26 | 80.19±0.13 | 23.95±0.48 | 16.51±0.46 | 15.83±0.65 | 82.66±0.68 | 33.80±2.21 | 20.45±0.81 | 12.23±0.73 | 11.25±0.73 | 80.19±0.12 | 35.80±1.66 | 16.02±0.91 | 10.22±0.38 | 8.31±0.23 | 87.14±6.73 | 80.77±0.26 | 80.40±0.05 | 27.93±4.34 | 20.09±1.53 |
| | AdpClip | 89.67±3.85 | 83.11±1.84 | 44.26±3.69 | 14.78±0.27 | 14.86±0.65 | 89.80±0.57 | 38.17±3.19 | 17.06±0.66 | 9.27±0.61 | 9.31±0.81 | 82.00±1.83 | 42.49±2.77 | 15.79±0.45 | 8.83±0.83 | 8.38±0.13 | 87.13±4.53 | 90.06±0.05 | 80.42±0.11 | 80.20±0.14 | 22.33±0.73 |
| | RGP | 89.25±5.81 | 93.17±0.01 | 55.13±7.38 | 14.25±0.83 | 11.93±0.66 | 87.56±1.05 | 89.73±1.65 | 15.02±0.24 | 7.77±0.38 | 7.44±0.57 | 78.54±1.42 | 19.12±0.88 | 17.20±1.51 | 24.74±3.14 | 26.37±1.83 | 93.24±0.04 | 81.11±2.47 | 82.03±0.26 | 27.76±0.91 | 27.16±1.08 |
| | PATE | 90.64±1.66 | 89.54±1.08 | 82.13±3.54 | 73.59±4.85 | 60.29±2.16 | 90.10±0.98 | 87.74±3.17 | 85.94±3.54 | 73.20±2.04 | 74.73±0.51 | 88.11±3.20 | 90.13±1.10 | 79.08±1.78 | 58.29±6.01 | 63.46±2.76 | 89.05±4.66 | 88.61±2.47 | 79.95±0.94 | 74.80±7.42 | 76.07±5.64 |
| | Priv-kNN | 87.78±0.71 | 77.79±0.12 | 70.58±0.65 | 70.21±1.42 | 69.29±0.55 | 81.30±1.82 | 44.26±0.67 | 48.96±1.87 | 39.84±1.14 | 41.74±0.24 | 68.28±3.75 | 48.09±0.69 | 45.75±2.42 | 44.43±1.83 | 44.55±1.20 | 81.14±1.03 | 46.04±0.37 | 41.76±3.15 | 42.19±1.55 | 40.58±0.65 |

Table 10: Overview of algorithms’ tailored AUC in black-box on either FMNIST and SVHN.

| | | SimpleCNN | | | | | ResNet | | | | | InceptionNet | | | | | VGG | | | | |
|--------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|--------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| | | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 | 0.2 | 1 | 4 | 100 | 1000 |
| FMNIST | Hand-DP | 0.50±0.00 | 0.51±0.01 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.01 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.01 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.01 | 0.51±0.00 |
| | PrivSet | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 |
| | DPGEN | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 | 0.52±0.00 | 0.52±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.01 | 0.51±0.00 |
| | TanHAct | 0.50±0.00 | 0.51±0.01 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.01 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 | 0.51±0.00 | 0.51±0.01 | 0.51±0.01 | 0.51±0.01 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.01 | 0.52±0.00 |
| | FocalLoss | 0.50±0.01 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.01 | 0.51±0.01 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.01 | 0.51±0.00 | 0.52±0.00 |
| | DP-SGD | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.01 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 |
| | RGP | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 |
| | CEP | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.53±0.01 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 |
| | AdpAlloc | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 |
| | AdpClip | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.52±0.00 |
| SVHN | Priv-kNN | 0.50±0.00 | 0.51±0.01 | 0.51±0.00 | 0.50±0.01 | 0.50±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.01 |
| | Hand-DP | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.53±0.00 | 0.53±0.00 | 0.50±0.00 | 0.50±0.00 | 0.51±0.00 | 0.52±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 | 0.50±0.00 | 0.50±0.00 | 0.52±0.00 | 0.50±0.00 | 0.53±0.00 |
| | PrivSet | 0.51±0.00 | 0.53±0.00 | 0.52±0.00 | 0.53±0.00 | 0.52±0.00 | 0.50±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.51±0.00 | 0.54±0.02 | 0.53±0.02 | 0.52±0.00 | 0.54±0.01 | 0.54±0.01 |
| | DPGEN | 0.50±0.00 | 0.50±0.00 | 0.52±0.00 | 0.50±0.00 | 0.50±0.00 | 0.54±0.00 | 0.50±0.00 | 0.51±0.00 | 0.51±0.00 | 0.52±0.00 | 0.50±0.00 | 0.51±0.00 | 0.52±0.00 | 0.52±0.00 | 0.52±0.00 | 0.50±0.00 | 0.50±0.00 | 0.52±0.00 | 0.53±0.00 | 0.51 |

Table 14: Details of ResBlock for ResNet. (*Shortcut perform identity mapping, and their outputs are added to the outputs of the stacked layers [20]*)

| Input Layer | <i>filters</i> |
|-------------|--|
| Conv2D | filters, kernel_size=3 |
| GroupNorm | num_groups=4, num_channels=filters, affine=False |
| Activation | ReLU |
| ResBlock 1 | <i>filters</i> |
| Conv2D | filters, kernel_size=3, stride=1 |
| GroupNorm | num_groups=4, num_channels=filters |
| Activation | ReLU |
| Conv2D | filters, kernel_size=3, stride=1 |
| GroupNorm | num_groups=4, num_channels=filters |
| Shortcut | |
| Activation | ReLU |
| ResBlock 2 | <i>filters, stride</i> |
| Conv2D | filters, kernel_size=3, stride |
| GroupNorm | num_groups=4, num_channels=filters |
| Activation | ReLU |
| Conv2D | filters, kernel_size=3, stride=1 |
| GroupNorm | num_groups=4, num_channels=filters |
| AvgPool2D | kernel_size=1, stride |
| GroupNorm | num_groups=4, num_channels=filters |
| Shortcut | |
| Activation | ReLU |

Table 15: InceptionNet architecture.

| Layer Type | Architecture |
|-------------------|-------------------------------------|
| InceptionBlock | filters=32, kernel_size=3, stride=1 |
| InceptionBlock | filters=32, kernel_size=3, stride=1 |
| MaxPool2D | kernel_size=2, stride=1 |
| | InceptionA |
| | InceptionB |
| | InceptionC |
| | InceptionD |
| | InceptionE |
| AdaptiveAvgPool2d | output_size=(1,1) |
| Dropout | p=0.5 |
| FC | 10 units |

Table 16: VGG architecture and the details of the VGGBlock.

| Layer Type | Architecture |
|------------|-------------------------|
| VGGBlock | filters=64 |
| MaxPool2D | kernel_size=2, stride=2 |
| VGGBlock | filters=128 |
| MaxPool2D | kernel_size=2, stride=2 |
| VGGBlock | filters=256 |
| VGGBlock | filters=256 |
| MaxPool2D | kernel_size=2, stride=2 |
| VGGBlock | filters=512 |
| VGGBlock | filters=512 |
| MaxPool2D | kernel_size=2, stride=2 |
| VGGBlock | filters=512 |
| VGGBlock | filters=512 |
| MaxPool2D | kernel_size=2, stride=2 |
| Flatten | |
| FC | 4096 units |
| Activation | ReLU |
| Dropout | p=0.5 |
| FC | 4096 units |
| Activation | ReLU |
| Dropout | p=0.5 |
| FC | 10 units |

| VGGBlock | <i>filters</i> |
|------------|-----------------------------------|
| Conv2D | filters, kernel_size=3, padding=1 |
| Activation | ReLU |

Table 17: Details of InceptionBlock for InceptionNet.

| InceptionBlock | | filters, kernel_size, padding | | | | | |
|----------------|-------------------------------------|------------------------------------|--------------------------------------|----------------|--------------------------------------|----------------|------------------------------------|
| Conv2D | | filters, kernel_size, padding | | | | | |
| GroupNorm | | num_groups=4, num_channels=filters | | | | | |
| Activation | | ReLU | | | | | |
| InceptionA | | | | | | | |
| InceptionBlock | filters=32, kernel_size=1 | InceptionBlock | filters=24, kernel_size=1 | InceptionBlock | filters=32, kernel_size=1 | AvgPool2D | kernel_size=3, stride=1, padding=1 |
| | | InceptionBlock | filters=32, kernel_size=5, padding=2 | InceptionBlock | filters=48, kernel_size=3, padding=1 | InceptionBlock | filters=16, kernel_size=1 |
| | | | | InceptionBlock | filters=48, kernel_size=3, padding=1 | | |
| Concat | | | | | | | |
| InceptionB | | | | | | | |
| InceptionBlock | filters=96, kernel_size=3, stride=2 | InceptionBlock | filters=32, kernel_size=1 | MaxPool2D | kernel_size=3, stride=2 | | |
| | | InceptionBlock | filters=48, kernel_size=3, padding=1 | | | | |
| | | InceptionBlock | filters=48, kernel_size=3, stride=2 | | | | |
| Concat | | | | | | | |
| InceptionC | | | | | | | |
| InceptionBlock | filters=48, kernel_size=1 | InceptionBlock | filters=48, kernel_size=1 | AvgPool2D | kernel_size=3, stride=1, padding=1 | | |
| | | InceptionBlock | filters=48, kernel_size=7, padding=3 | InceptionBlock | filters=48, kernel_size=1 | | |
| | | InceptionBlock | filters=48, kernel_size=7, padding=3 | | | | |
| Concat | | | | | | | |
| InceptionD | | | | | | | |
| InceptionBlock | filters=48, kernel_size=1 | InceptionBlock | filters=96, kernel_size=1 | MaxPool2D | kernel_size=3, stride=2 | | |
| InceptionBlock | filters=96, kernel_size=3, stride=2 | InceptionBlock | filters=96, kernel_size=7, padding=3 | | | | |
| | | InceptionBlock | filters=96, kernel_size=3, stride=2 | | | | |
| Concat | | | | | | | |
| InceptionE | | | | | | | |
| InceptionBlock | filters=80, kernel_size=1 | InceptionBlock | filters=96, kernel_size=1 | InceptionBlock | filters=112, kernel_size=1 | AvgPool2D | kernel_size=3, stride=1, padding=1 |
| | | InceptionBlock | filters=96, kernel_size=3, padding=1 | InceptionBlock | filters=96, kernel_size=3, padding=1 | InceptionBlock | filters=48, kernel_size=1 |
| | | | | InceptionBlock | filters=96, kernel_size=3, padding=1 | | |
| Concat | | | | | | | |