

# Model Conversion via Differentially Private Data-Free Distillation

Bochao Liu<sup>1,2</sup>, Pengju Wang<sup>1,2</sup>, Shikun Li<sup>1,2</sup>, Dan Zeng<sup>3</sup>, Shiming Ge<sup>1,2\*</sup>

<sup>1</sup>Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China

<sup>2</sup>School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

<sup>3</sup>School of Communication and Information Engineering, Shanghai University, Shanghai 200444, China

{liubochoao, wangpengju, lishikun, geshiming}@iie.ac.cn, dzeng@shu.edu.cn

## Abstract

While massive valuable deep models trained on large-scale data have been released to facilitate the artificial intelligence community, they may encounter attacks in deployment which leads to privacy leakage of training data. In this work, we propose a learning approach termed differentially private data-free distillation (DPDFD) for model conversion that can convert a pretrained model (teacher) into its privacy-preserving counterpart (student) via an intermediate generator without access to training data. The learning collaborates three parties in a unified way. First, massive synthetic data are generated with the generator. Then, they are fed into the teacher and student to compute differentially private gradients by normalizing the gradients and adding noise before performing descent. Finally, the student is updated with these differentially private gradients and the generator is updated by taking the student as a fixed discriminator in an alternate manner. In addition to a privacy-preserving student, the generator can generate synthetic data in a differentially private way for other downstream tasks. We theoretically prove that our approach can guarantee differential privacy and well convergence. Extensive experiments clearly demonstrate that our approach significantly outperform other differentially private generative approaches.

## 1 Introduction

The success of deep neural networks in a wide array of applications [Jia *et al.*, 2019a; Jia *et al.*, 2019b; Ye *et al.*, 2020] greatly owes to the open source of massive models. However, a major problem is that the training data of these models often contain a large amount of sensitive information that can be easily recovered with a few access to the models [Fredrikson *et al.*, 2015; Yang *et al.*, 2019]. *How to protect such private information while maintaining the*

*model performance* has attracted a lot of attentions. Differential privacy (DP) [Dwork *et al.*, 2006] is a common technique for protecting privacy. [Abadi *et al.*, 2016] guaranteed that the model was differentially private regarding the training data by clipping gradients and adding Gaussian noise to gradients. However, the model accuracy decreases severely when the privacy requirements increase so that the model can't be applied directly to the case where only the pretrained models are given. [Papernot *et al.*, 2017; Papernot *et al.*, 2018] proposed a semi-supervised learning framework called PATE to reduce the impact of the DP noise by leveraging the noisy aggregation of multiple teacher models trained directly on the private data. It is possible to train a privacy-preserving student model using PATE framework given only sensitive teacher models, but it is difficult to find a suitable unlabeled public dataset for the distillation process.

In the meantime, an independent line of research concerning model compression shows that some data-free knowledge distillation approaches (DFKD) [Chen *et al.*, 2019; Zhu *et al.*, 2021; Choi *et al.*, 2020] could achieve similar performance by vanilla training with only a teacher model. The data used for the distillation process is generated by a generator, which could potentially be a remedy for the above problem of the suitable public dataset. The generators of such methods mainly learn the data distribution rather than the image details, which intuitively also provides a degree of protection of privacy. [Ge *et al.*, 2023] has showed that: It is possible to leverage the power of data-free knowledge distillation to train a privacy-preserving student model that is not necessary to access to the original dataset. But there is still a long way to take advantage of this intuition.

Inspired by the above observations, in this paper we propose a model conversion approach with *Differentially Private Data-Free Distillation* (DPDFD) to facilitate the model releasing by distilling a pretrained model as teacher into a differentially private student. Specifically, our DPDFD combines DFKD and DP, which applies DFKD to distill private knowledge and a DP mechanism  $\mathcal{A}_{C,\sigma}$  to guarantee privacy. The objective is to enable an effective conversion that achieves strong privacy protection with minimum accuracy loss when only private models (teachers) are given. As shown in Fig. 1, we first generate massive synthetic data with a generator. Then, we feed the synthetic data into the teacher model and student model to compute the loss. Differentially

\*Shiming Ge is the corresponding author (geshiming@iie.ac.cn). Supplementary material is available at: <https://arxiv.org/abs/2304.12528>

private gradients are calculated by applying DP mechanism  $\mathcal{A}_{C,\sigma}$ . Finally, we update the student with these gradients and update the generator by taking the student as a fixed discriminator. In particular, we achieve DP by performing normalization on the gradients of student outputs and adding Gaussian noise to gradients during student learning. The reason for performing normalization instead of clipping is that it will retain the relative size information of the gradients and achieve better performance with a smaller norm bound. The reason for adding Gaussian noise to the gradients of student outputs is that it has a lower dimension compared to other gradients. Both smaller norm bound and lower dimension gradients make it easier to balance the performance and privacy. In addition, the DP mechanism  $\mathcal{A}_{C,\sigma}$  also ensures differentially private training of the generator according to the post-processing mechanism. We can use the generator to generate data for other down stream tasks if needed. We also provide privacy and convergence analysis for our DPDFD in theory. Furthermore, DPDFD can be extended to multi-model case, which aggregates multiple sensitive teachers into a privacy-preserving student model.

In summary, our DPDFD can effectively convert a sensitive teacher model to a privacy-preserving student model through three key components. First, performing normalization instead of clipping which is usually used in other approaches retains information about the relative size of the gradients. Second, achieving DP by adding noise on the gradients of lower dimensional outputs makes it easier to balance performance and privacy. Third, synthetic data generated by the generator has a similar distribution with the training data of teacher model. In this way, we can convert the sensitive model to a privacy-preserving student with minimum accuracy loss.

Our major contributions are three folds: 1) we propose a model conversion approach to convert a pretrained sensitive model into a privacy-preserving model for secure releasing; 2) we provide privacy analysis and convergence analysis for our approach in theory; 3) we conduct extensive experiments and analysis to demonstrate the scalability and effectiveness of our approach.

## 2 Related Works

The approach we proposed in this paper aims to train privacy-preserving student models by distilling knowledge from given sensitive model(s) via a differentially private data-free distillation. Therefore, we briefly review the related works for two aspects, including differentially private learning and data-free knowledge distillation.

### 2.1 Differentially Private Learning

Differentially private learning aims to ensure the learning model is differentially private regarding the private data. [Abadi *et al.*, 2016] proposed a differentially private stochastic gradient descent (DPSGD) algorithm which achieved DP by clipping and adding noise to the gradients of all parameters during the training process. However, the model performance degrades severely with strong privacy requirements. [Papernot *et al.*, 2017] later proposed PATE which used semi-supervised learning to transfer the knowledge of the teacher

ensemble to the student by using a noisy aggregation. It can reduce the impact of noise on performance by increasing the number of teacher models. However, it is difficult to find an unlabeled public dataset that has similar distribution to the training data of teachers. Some works want to train differentially private generators to generate data that has similar distribution to private data while preserving privacy. [Xie *et al.*, 2018] applied DPSGD to the training process of Generative Adversarial Networks (GAN) [Goodfellow *et al.*, 2014] to get a differentially private generator. [Chen *et al.*, 2020] suggested that it is not necessary to clip and add noise to all gradients, but only needs to achieve DP in the back propagation process from discriminator to generator. [Cao *et al.*, 2021] applied differentially private optimal transmission theory to train generators. [Chen *et al.*, 2022] proposed an energy-guided network trained on sanitized data to indicate the direction of the true data distribution via Langevin Markov Chain Monte Carlo sampling method. In this paper, we make a better balance between performance and privacy by applying normalization instead of clipping and post-processing of DP.

### 2.2 Data-Free Knowledge Distillation

Data-free knowledge distillation is a class of approaches that aims to train a student model with a pretrained teacher model without access to original training data. It uses the information extracted from the teacher model to synthesize data used in the distillation process. [Srinivas and Babu, 2015] proposed to directly merge similar neurons in fully-connected layers, which cannot be applied to convolutional layers and networks because the detailed architectures and parameters information is unknown. [Lopes *et al.*, 2017] first proposed data-free knowledge distillation, using the distribution information of the original data to reconstruct the synthetic data used in the distillation process. [Deng and Zhang, 2021] used this approach to generate synthetic graph data for data-free knowledge distillation for graph classification tasks. [Zhu *et al.*, 2021] modified and applied it to federated settings. [Chen *et al.*, 2019] proposed a novel framework named DAFL for training the student model by exploiting GAN. It uses the teacher model as the discriminator to train a generator. [Choi *et al.*, 2020] proposed matching statistics from the batch normalization layers for generated data and the original data in the teacher. It can make the generated data closer to the original data. [Fang *et al.*, 2022] proposed FastDFKD that applied the idea of meta-learning to the training process to accelerate the efficiency of data synthesis. Inspired by these approaches, it is possible to convert a sensitive model into a privacy-preserving model without access to the original data.

## 3 Preliminaries

Here we will first provide some background knowledge about differential privacy. We then draw connections between the definitions and theorems we introduced here and our DP analysis on DPDFD later. The following definition explains how DP provides rigorous privacy guarantees clearly.

**Definition 1** (Differential Privacy). *A randomized mechanism  $\mathcal{A}$  with domain  $\mathcal{R}$  is  $(\epsilon, \delta)$ -differential privacy, if for all*

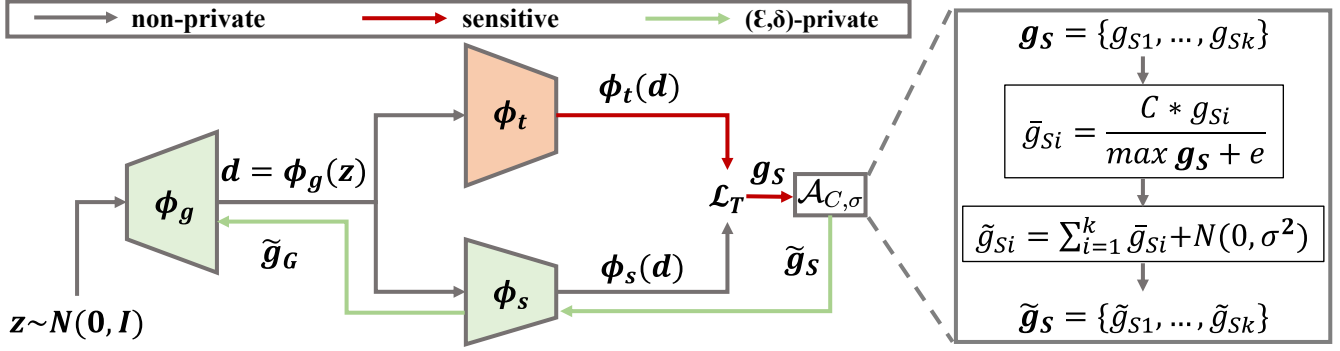


Figure 1: Overview of our differentially private data-free distillation approach. The approach learns to convert a pretrained model  $\phi_t$  into a privacy-preserving student  $\phi_s$  via an intermediate generator  $\phi_g$ . The learning is performed to collaborate three parties in a unified way. First, the generator generates massive data. Then, these data are fed into the teacher and student models to calculate the gradients  $g_s$ ; Finally, the student and generator are updated with differentially private gradients  $\tilde{g}_s$ , which are computed by applying DP mechanism  $\mathcal{A}_{C,\sigma}$  to  $g_s$ . Here,  $C$  is the norm bound and  $N(0, \sigma^2)$  is Gaussian noise with mean 0 and variance  $\sigma^2$ .

$\mathcal{O} \subseteq \mathcal{R}$  and any adjacent datasets  $D$  and  $D'$  :

$$\Pr[\mathcal{A}(D) \in \mathcal{O}] \leq e^\varepsilon \cdot \Pr[\mathcal{A}(D') \in \mathcal{O}] + \delta, \quad (1)$$

where adjacent datasets  $D$  and  $D'$  differ from each other with only one training example.  $\varepsilon$  is the privacy budget and the smaller it is the better, and  $\delta$  is the failure probability.

[Mironov, 2017] proposed a variant Rényi differential privacy (RDP), which computes privacy with Rényi divergence. We review the definition of RDP and its connection to DP.

**Definition 2** (Rényi Differential Privacy). *A randomized mechanism  $\mathcal{A}$  is  $(\lambda, \varepsilon)$ -RDP with  $\lambda > 1$  if for any adjacent datasets  $D$  and  $D'$  :*

$$D_\lambda(\mathcal{A}(D) || \mathcal{A}(D')) = \frac{1}{\lambda - 1} \log \mathbb{E}_{(x \sim \mathcal{A}(D))} \left[ \left( \frac{\Pr[\mathcal{A}(D) = x]}{\Pr[\mathcal{A}(D') = x]} \right)^{\lambda - 1} \right] \leq \varepsilon. \quad (2)$$

Different from DP, RDP has a more friendly composition theorem and can be applied to both data-independent and data-dependent settings. It supports a tighter composition of privacy budget which can be described as follows: For a sequence of mechanisms  $\mathcal{A}_1, \dots, \mathcal{A}_i, \dots, \mathcal{A}_k$ , where  $\mathcal{A}_i$  is  $(\lambda, \varepsilon_i)$ -RDP, the composition of them  $\mathcal{A}_1 \circ \dots \circ \mathcal{A}_i \dots \circ \mathcal{A}_k$  is  $(\lambda, \sum_i \varepsilon_i)$ -RDP.  $\mathcal{A}_i$  represents one query to the teacher model in our case. Moreover, the connection between RDP and DP can be described as follows:

**Theorem 1** (Convert RDP to DP). *A  $(\lambda, \varepsilon)$ -RDP mechanism  $\mathcal{A}$  also satisfies  $(\varepsilon + \log \frac{\lambda-1}{\lambda} - \frac{\log \delta + \log \lambda}{\lambda-1}, \delta)$ -DP.*

To provide DP guarantees, we exploit the post-processing [Dwork and Roth, 2014] described as follows:

**Theorem 2** (Post-processing). *If mechanism  $\mathcal{A}$  satisfies  $(\varepsilon, \delta)$ -DP, the composition of a data-independent function  $\mathcal{F}$  with  $\mathcal{A}$  is also  $(\varepsilon, \delta)$ -DP.*

## 4 Proposed Approach

### 4.1 Problem Formulation

Given a sensitive teacher model  $\phi_t$  with parameters  $\theta_t$ , the objective is converting it into a privacy-preserving student

model  $\phi_s$  with parameters  $\theta_s$  that does not reveal data privacy and has the ability to perform similarly to the teacher model. To achieve that, we introduce a differentially private data-free distillation approach. We first sample a batch of noise vectors  $\mathbf{z} = \{z_i\}_{i=1}^B$  and feed them into the generator  $\phi_g$  with parameters  $\theta_g$  to generate massive synthetic data  $D = \{d_i\}_{i=1}^B$ . Enter them into the teacher model and student model to calculate the loss  $\mathcal{L}_T(\phi_t(\theta_t; D), \phi_s(\theta_s; D))$  and then calculate updated outputs of student  $y_s$  to achieve DP with a differentially private mechanism  $\mathcal{A}_{C,\sigma}$ . Finally update the student by the loss  $\mathcal{L}_S(\phi_s(\theta_s; D), y_s)$ . Thus, the converting process can be formulated by minimizing an energy function  $\mathbb{E}$ :

$$\begin{aligned} \mathbb{E}(\theta_s; \theta_t) &= \mathbb{E}(\phi_s(\theta_s; D), y_s) \\ &= \mathbb{E}(\phi_s(\theta_s; D), \phi_s(\theta_s; D) - \gamma \cdot \mathcal{A}_{C,\sigma}(g)), \end{aligned} \quad (3)$$

where  $g = \frac{\partial \mathcal{L}_T(\phi_t(\theta_t; D), \phi_s(\theta_s; D))}{\partial \phi_s(\theta_s; D)}$  and  $\gamma$  is the learning rate. We suppress the risk of privacy leakage with DP mechanism  $\mathcal{A}_{C,\sigma}$  which normalizes the gradients with norm bound  $C$  and adds Gaussian noise with variance  $\sigma^2$ . We will introduce how the differentially private mechanism  $\mathcal{A}_{C,\sigma}$  protects privacy in detail in the following.

### 4.2 Our Solution: DPDFD

Student model distilled from a sensitive teacher model directly may lead to privacy leakage and another main problem is that we don't have access to the original dataset. Thus, we aim to convert the sensitive teacher model to a privacy-preserving student model while having similar performance to the sensitive model in a data-free manner.

Unlike [Abadi *et al.*, 2016] which requires clipping and adding Gaussian noise to gradients of all parameters, due to the post-processing of DP, we only need to perform normalization and add noise to the gradients of student's outputs to calculate new differentially private outputs. As shown in Fig. 1 and Alg. 1, we first sample a batch of noise vectors  $\mathbf{z} = \{z_i\}_{i=1}^B$  and feed them into the generator  $\phi_g$  with parameter  $\theta_g$  to obtain massive synthetic data  $D = \phi_g(\mathbf{z})$ . Then enter the synthetic data  $D$  into the teacher and student to compute the loss  $\mathcal{L}_T(\phi_t(\theta_t; D), \phi_s(\theta_s; D))$ . To get better results,

**Algorithm 1** DPDPFD

**Input:** Training iterations  $T$ , loss function  $\mathcal{L}_T, \mathcal{L}_S, \mathcal{L}_G$ , noise scale  $\sigma$ , sample size  $B$ , learning rate  $\gamma, \gamma_s, \gamma_g$ , gradient norm bound  $C$ , a positive stability constant  $e$

---

```

1: for  $t \in [T]$  do
2:   Sample  $B$  noise samples  $\mathbf{z} = \{z_i\}_{i=1}^B$ 
3:   Generate  $B$  synthetic samples  $D = \{\phi_g(\theta_g; z_i)\}_{i=1}^B$ 
4:   for each synthetic data  $d_i = \phi_g(\theta_g; z_i)$  do
5:     Compute loss  $\mathcal{L}_T(\phi_t(\theta_t; d_i), \phi_s(\theta_s; d_i))$ 
6:     Compute the gradient  $g_i = \frac{\partial \mathcal{L}_T}{\partial \phi_s(\theta_s; d_i)}$ 
7:     Normalize the gradient  $\bar{g}_i = \frac{C \cdot g_i}{\|g_i\|_2 + e}$ 
8:   end for
9:   Add noise  $\tilde{g} = \frac{1}{B} (\sum_{i=1}^B \bar{g}_i + \mathcal{N}(0, \sigma^2 C^2 I))$ 
10:  Compute differentially private outputs of student  $y_s = \phi_s(\theta_s; D) - \gamma \cdot \tilde{g}$ 
11:  Compute loss  $\mathcal{L}_S(\phi_s(\theta_s; D), y_s)$ 
12:  Update student  $\theta_s^{t+1} = \theta_s^t - \gamma_s \cdot \frac{\partial \mathcal{L}_S}{\partial \theta_s^t}$ 
13:  Compute loss  $\mathcal{L}_G(\phi_s(\theta_s; D))$ 
14:  Update generator  $\theta_g^{t+1} = \theta_g^t - \gamma_g \cdot \frac{\partial (\mathcal{L}_S + \mathcal{L}_G)}{\partial \theta_g^t}$ 
15: end for
16: return  $\theta_s$  and  $\theta_g$ 
    
```

---

we treat  $\text{argmax}(\phi_t(\theta_t; D))$  as the target labels and then calculate the distillation loss  $\mathcal{L}_T$  same as [Zhao *et al.*, 2022]. After that we achieve DP by the mechanism  $\mathcal{A}_{C, \sigma}$  which can be described as follows:

$$\mathcal{A}_{C, \sigma}(g) = \frac{1}{B} \left( \sum_{i=1}^B \frac{C \cdot g_i}{\|g_i\|_2 + e} + \mathcal{N}(0, \sigma^2 C^2 I) \right), \quad (4)$$

and generate new differentially private outputs  $y_s$ . Compared with clipping, normalization can achieve higher accuracy at smaller norm bound  $C$ . This is because when  $C$  is small, clipping makes the gradients lose their variability but normalization retains the relative size relationship of gradients. Another important point is that the smaller the  $C$ , the smaller the privacy budget consumed by each query to the teacher model, which is exactly what we want. So we choose a smaller  $C$  and normalization operation to get a lower privacy budget and better performance. Finally, we compute the loss  $\mathcal{L}_S(\phi_s(\theta_s; D), y_s)$  and update the student model with it. This loss function can take many forms, and we use cross-entropy loss in our experiments. In order to make the distribution of the synthetic data closer to the private data and balance the classes of the synthetic data, we add an additional loss  $\mathcal{L}_G$  when updating the generator. It can be formulated as:

$$\begin{aligned} \mathcal{L}_G(\phi_s(\theta_s; D)) = & \ell(\phi_s(\theta_s; D), \text{argmax}(\phi_s(\theta_s; D))) \\ & + \alpha \phi_s(\theta_s; D) \log(\phi_s(\theta_s; D)) \\ & + \beta \|\phi_s(\theta_s; D)\|_2, \end{aligned} \quad (5)$$

where  $\alpha$  and  $\beta$  are the tuning parameters to balance the effect of three terms, and we set both of them to 1. The first term  $\ell(\cdot)$  is a cross entropy function that measures the one-hot clas-

sification loss, which enforces the synthetic data having similar distribution as the private data. The second term is the information entropy loss to measure the class balance of generated data. The third term uses  $l_2$ -norm  $\|\cdot\|_2$  to measure the activation loss, since the features that are extracted by the student and correspond to the output before the fully-connected layer tend to receive higher activation value if input data are real rather than some random vectors, where  $\theta_s^- \subset \theta_s$  is the student's backbone parameters. We update the student and generator alternately in our experiments. In the nutshell, the trained student and generator satisfy DP because the training process can be seen as post-processing, given that  $y_s$  is differentially private results. Overall, DPDPFD can be formally proved DP in Theorem. 3.

**Theorem 3.** DPDPFD guarantees  $(\frac{2C^2 n B T \lambda}{\sigma^2} + \log \frac{\lambda-1}{\lambda} - \frac{\log \delta + \log \lambda}{\lambda-1}, \delta)$ -DP for all  $\lambda \geq 1$  and  $\delta \in (0, 1)$ .

### 4.3 Convergence Analysis

To analyze the convergence of our DPDPFD, we follow the standard assumptions in the SGD literature [Allen-Zhu, 2017; Bottou *et al.*, 2018; Ghadimi and Lan, 2013], with an additional assumption on the gradient noise. We assume that  $\mathcal{L}_T$  has a lower bound  $\mathcal{L}_*$  and  $\mathcal{L}_T$  is  $\kappa$ -smoothness, which can be described as follows:  $\forall x, y$ , there is a non-negative constant  $\kappa$  such that  $\mathcal{L}_T(x) - \mathcal{L}_T(y) \leq \nabla \mathcal{L}_T(x)^\top (x-y) + \frac{\kappa}{2} \|x-y\|^2$ . The additional assumption is that  $(g_r - g) \sim \mathcal{N}(0, \zeta^2)$ , where  $g_r$  is the ideal gradients of  $\mathcal{L}_T$  and  $g$  is the gradient we compute as an unbiased estimate of  $g_r$ . Then according to [Bu *et al.*, 2022], in our case we have:

$$\chi \leq \mathcal{F} \left( \sqrt{\frac{2(\mathcal{L}_0 - \mathcal{L}_*) + 2T\kappa\gamma^2 C^2(1 + \sigma^2 d)}{T\gamma C}}; \zeta, e \right), \quad (6)$$

where  $\chi$  is  $\min_{0 \leq t \leq T} \mathbb{E}(\|g_r^t\|)$ ,  $d$  is a constant number,  $\mathcal{L}_0$  is the initial loss and  $\mathcal{F}(\cdot)$  results only from the normalization operation same as [Bu *et al.*, 2022] and it won't affect the monotonicity of input variables. We simply set the learning rate  $\gamma \propto \frac{1}{\sqrt{T}}$  and the gradients will gradually tend to 0 as  $T$  increases.

### 4.4 Discussion

**Extension to Multi-Model Ensemble.** Our DPDPFD can be extended to the case of multiple sensitive models. Different from the case of a single teacher that reduces the impact of DP noise by averaging a batch of gradients, the case of multi-model achieves it by averaging the gradients from given sensitive models. In particular, given multiple teacher models  $\{\phi_t^j\}_{j=1}^n$ , we first sample noise vectors and generate massive synthetic data same as the case of a single teacher. For each data  $d_i$ , we enter it into multiple teacher models  $\{\phi_t^j(\theta_t^j; d_i)\}_{j=1}^n$  and student model  $\phi_s(\theta_s; d_i)$  and compute losses  $\{\mathcal{L}_T(\phi_t^j(\theta_t^j; d_i), \phi_s(\theta_s; d_i))\}_{j=1}^n$ . Then, we compute gradients  $g_{ij} = \left\{ \frac{\partial \mathcal{L}_T(\phi_t^j(\theta_t^j; d_i), \phi_s(\theta_s; d_i))}{\partial \phi_s(\theta_s; d_i)} \right\}_{j=1}^n$  with the losses. After that, we normalize them with norm bound  $C$  and add noise to them to get differentially private gradients

Dataset	Teacher	$\epsilon$	DP-GAN	PATE-GAN	G-PATE	GS-WGAN	DP-MERF	P3GM	DataLens	DPSH	DPGEN	DPDFD
MNIST	0.9921	1	0.4036	0.4168	0.5810	0.1432	0.6367	0.7369	0.7123	N/A	0.9046	<b>0.9512</b>
		10	0.8011	0.6667	0.8092	0.8075	0.6738	0.7981	0.8066	0.8320	0.9357	<b>0.9751</b>
FMNIST	0.9102	1	0.1053	0.4222	0.5567	0.1661	0.5862	0.7223	0.6478	N/A	0.8283	<b>0.8386</b>
		10	0.6098	0.6218	0.6934	0.6579	0.6162	0.7480	0.7061	0.7110	0.8784	<b>0.8988</b>
CelebA-G	0.9353	1	0.5330	0.6068	0.6702	0.5901	0.5936	0.5673	0.7058	N/A	0.6999	<b>0.7237</b>
		10	0.5211	0.6535	0.6897	0.6136	0.6082	0.5884	0.7287	0.7630	0.8835	<b>0.8992</b>
CelebA-H	0.8868	1	0.3447	0.3789	0.4985	0.4203	0.4413	0.4532	0.6061	N/A	0.6614	<b>0.7839</b>
		10	0.3920	0.3900	0.6217	0.5225	0.4489	0.4858	0.6224	N/A	0.8147	<b>0.8235</b>

Table 1: Accuracy comparisons with 9 explicit approaches under different privacy budget  $\epsilon$  ( $\delta = 10^{-5}$ ).

$\tilde{g}_i = \frac{1}{n} (\sum_{j=1}^n \frac{C \cdot g_{ij}}{\|g\|_2 + \epsilon} + \mathcal{N}(0, \sigma^2 C^2 I))$ . Next we can compute

an updated output of student  $y_s^i = \phi_s(\theta_s; d_i) - \gamma \cdot \tilde{g}_i$ . Finally, we update the student and generator same as Alg. 1. In this way, we can aggregate multiple sensitive teacher models into a privacy-preserving student model.

**Direct Training on Private Data.** The proposed DP mechanism  $\mathcal{A}_{C, \sigma}$  is a key building block of DPDFD, and can also be applied to situations where private data is available. Given a private dataset  $\{x, y\}$ , we input the data  $x$  into the model and calculate the cross-entropy loss  $\mathcal{L}(\phi(\theta; x), y)$ . After that, we run the same procedures to achieve DP and update the model as Alg. 1 except we don't need to update the generator. In this way, we can train a privacy-preserving model with direct access to private data.

## 5 Experiments

In this section, we present the experimental evaluation of DPDFD for converting the sensitive model to a differentially private model with high performance.

### 5.1 Experimental Setup

**Datasets.** We conduct our experiments on 7 image datasets, including MNIST [LeCun *et al.*, 1998], FashionMNIST(FMNIST) [Xiao *et al.*, 2017], CIFAR10 [Krizhevsky, 2009], CelebA [Liu *et al.*, 2015], PathMNIST [Yang *et al.*, 2021], COVIDx and ImageNet [Deng *et al.*, 2009]. We created CelebA-H and CelebA-G based on CelebA. CelebA-H and CelebA-G are classification datasets with hair color (black/blonde/brown) and gender as the label, respectively. COVIDx is a classification dataset for COVID.

**Baselines.** We perform comparisons with 14 state-of-the-art benchmarks, including 9 explicit approaches that training classifiers with generative data (DP-GAN [Xie *et al.*, 2018], GS-WGAN [Chen *et al.*, 2020], PATE-GAN [Jordon *et al.*, 2019], DP-MERF [Harder *et al.*, 2021], P3GM [Takagi *et al.*, 2021], DataLens [Wang *et al.*, 2021], G-PATE [Long *et al.*, 2021], DPSH [Cao *et al.*, 2021], DPGEN [Chen *et al.*, 2022]) and 5 implicit approaches that training classifiers with differentially private learning (DPSGD [Abadi *et al.*, 2016], TSADP [Papernot *et al.*, 2021], TOPAGG [Wang *et al.*, 2021], GM-DP [McMahan *et al.*, 2018], DGD [Ge *et al.*,

2023]). To make the comparisons fair, we take the results from original papers or run the official codes.

**Networks.** We adopt several popular network architectures as our teacher models, including AlexNet [Krizhevsky *et al.*, 2012], VGGNet [Simonyan and Zisserman, 2015], ResNet [He *et al.*, 2016], WideResnet [Zagoruyko and Komodakis, 2017], DenseNet [Huang *et al.*, 2016], MobileNet [Howard *et al.*, 2017], ShuffleNet [Zhang *et al.*, 2018], GoogleNet [Szegedy *et al.*, 2015] and ViT [Dosovitskiy *et al.*, 2020]. For VGGNet, we use 19-layer net with BN. For ResNet, we use 50-layer networks for ImageNet and 34-layer networks for others. For WideResnet, we use 50-layer networks as teacher. For DenseNet, we use 161-layer networks with growth rate equals 24. For ViT, we use the same architecture as [Dosovitskiy *et al.*, 2020]. For student model, we use 34-layer ResNet for ImageNet and the same network as DataLens for others.

### 5.2 Experimental Results

In this section, we compare DPDFD with 14 baselines and evaluate it on different networks to verify its effectiveness. We first compare the model performance of DPDFD and other state-of-the-art approaches. Then, we conduct experiments on ImageNet with different networks under different privacy budget. We show that our DPDFD is scalable and outperforms all baselines.

**Comparisons with 9 Explicit Baselines.** To demonstrate the effectiveness of our DPDFD, we conduct comparisons with 9 baselines under different privacy budget and report the results in Tab. 1. All approaches are under a low failure probability  $\delta = 10^{-5}$ . We can see that our DPDFD achieves the highest performance under the same condition of privacy budget. In particular, when  $\epsilon = 1$ , our DPDFD achieves an accuracy of 0.9512 on MNIST and 0.8386 on FMNIST, which remarkably reduces the accuracy drop by 0.0409 and 0.0716 respectively, while most of the other approaches fail to achieve an accuracy of 0.8000. It shows that our approach has the best privacy-preserving ability and minimal accuracy drop. Even for high dimensional datasets like CelebA whose dimensionality is about 16 times larger than MNIST, all approaches suffer from accuracy drop with respect to their counterparts under high privacy budget while our DPDFD still de-

Approach	$\epsilon$	MNIST	$\epsilon$	CIFAR10
DPSGD	2.0	0.9500	2.0	0.6623
TSADP	1.0	0.7991	7.5	0.6620
TOPAGG	1.0	0.9465	2.0	0.8518
GM-DP	1.0	0.9508	2.0	0.8597
DGD	1.0	0.7360	3.0	0.7365
DPDFD	1.0	<b>0.9512</b>	2.0	<b>0.8601</b>

Table 2: Accuracy comparisons with 5 implicit approaches under different privacy budget  $\epsilon$ .

Networks	Teacher	$\epsilon$		
		1	2	5
AlexNet	<b>0.5655</b>	0.2756	0.5124	0.5218
VGGNet	<b>0.7423</b>	0.3419	0.6381	0.6602
ResNet	<b>0.7602</b>	0.3823	0.6499	0.6781
WideResNet	<b>0.7848</b>	0.2982	0.5997	0.7117
DenseNet	<b>0.7765</b>	0.4552	0.6612	0.7029
MobileNet	<b>0.7186</b>	0.3830	0.6203	0.6424
ShuffleNet	<b>0.6953</b>	0.4964	0.6313	0.6652
GoogleNet	<b>0.6246</b>	0.2987	0.5717	0.5858
ViT	<b>0.8140</b>	0.3142	0.6880	0.7240

Table 3: Accuracy on ImageNet with different networks under different privacy budget  $\epsilon$  ( $\delta = 10^{-5}$ ).

livers the highest test accuracy. This demonstrates that our approach is also effective for high-dimensional datasets.

**Comparisons with 5 Implicit Baselines.** In addition, we conduct experimental comparisons with 5 implicit baselines on MNIST and CIFAR10. The results are shown in Tab. 2, where our DPDFD achieves the highest accuracy of 0.9512 and 0.8601 under the lowest privacy budget of 1.0 and 2.0. The main reason comes from that we use normalization instead of traditional clipping. In this way, the differentially private gradients obtained after  $\mathcal{A}_{C,\sigma}$  retain more information (relative size between gradients).

**Evaluation on Open Source Networks.** To demonstrate the scalability of our DPDFD, we apply it to convert several popular networks pretrained on ImageNet under different privacy budget  $\epsilon$ . These teacher models are taken directly from the PyTorch model zoo. The results are shown in Tab. 3. We find that the effect of network architectures is sometimes greater than the effect of public teacher model accuracy when  $\epsilon$  is small. In particular, when  $\epsilon = 1$ , the student model reaches the highest accuracy of 0.4964 when the teacher model is ShuffleNet, but the accuracy of teacher model is 0.1187 lower than the highest ViT. We guess this is because the simpler the teacher model is, the easier it is for the generator to learn the distribution of its training data,

Dataset	$\epsilon$	DPSGD	TOPAGG	GM-DP	DPDFD
MNIST	0.5	0.8103	0.9235	0.9331	<b>0.9377</b>
	0.7	0.8932	0.9382	0.9438	<b>0.9447</b>
	1.0	0.9247	0.9465	0.9508	<b>0.9762</b>
CIFAR10	2.0	0.6623	0.8518	0.8597	<b>0.8652</b>
	4.0	0.6884	0.8540	0.8663	<b>0.8708</b>
	8.0	0.7159	0.8562	0.8705	<b>0.8973</b>

Table 4: Accuracy comparisons with 3 DPSGD mechanisms on MNIST and CIFAR10 under different privacy budget  $\epsilon$ .

resulting in faster convergence of the student model. As  $\epsilon$  increases, the student model can learn more and more fully from the teacher model, so the effect of teacher performance dominates. The more complex the teacher model, the more information about the training data it contains. The student model learns more accurate predictions from the teacher and the generator learns a more similar data distribution, both of which lead to higher accuracy of the student model. We can see that at  $\epsilon = 5$ , the accuracy of the student is positively correlated with the accuracy of teacher model and it has approached that of the teachers, which proves that our DPDFD is not only scalable but also effective to a variety of popular network architectures.

**Evaluation of Training Directly with Private Data.** In this section, we conduct experiments to evaluate the application of our DP mechanism  $\mathcal{A}_{C,\sigma}$  to the case where private data can be accessed. We compare 3 state-of-the-art approaches which trained directly with private data (based on DPSGD mechanism): DPSGD, TOPAGG and GM-DP on MNIST and CIFAR10 under the same settings except for the DP mechanism. The results are shown in Tab. 4. We can see that our approach achieves the best performance on both the low-dimensional MNIST dataset and the high-dimensional CIFAR10 dataset. These results depend on two aspects. On one hand, we only need to compute a new differentially private output in the first step of backpropagation instead of adding noise to each gradient like other approaches. On the other hand, we use normalization instead of clipping, which retains more gradient information in the case of small norm bound  $C$ . So our approach can achieve better results.

### 5.3 Ablation Studies

After the promising performance is achieved, we further analyze the impact of each component in our approach, including normalization vs. clipping, norm bound, noise scale, and composition of loss function.

**Normalization vs. Clipping.** To study the effect of different operations on the gradients, we conduct experiments with 34-layer ResNet pretrained on MNIST and FMNIST as teachers and report the results in Fig. 2. Performing normalization is significantly better than clipping when  $C$  is small, and even clipping makes the student not converge. The advantage of normalization gradually decreases as  $C$  increases, and when it increases to a certain level (about  $10^{-3}$  as Fig. 2 shows),

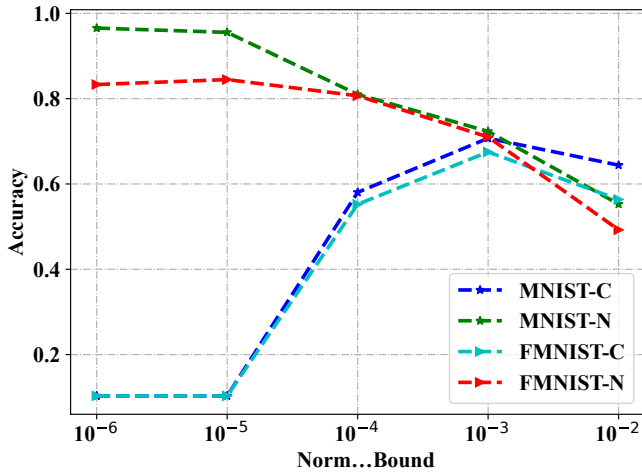


Figure 2: Student accuracy under different norm bound  $C$  and different operations on gradients (‘-C’ or ‘-N’) in the case  $\varepsilon = 0.6$  and  $\delta = 10^{-5}$  (‘-C’: Clipping, ‘-N’: Normalization)

Dataset	$\sigma$				
	20	50	100	200	500
<b>MNIST</b>	0.4159	0.8314	0.9822	0.9815	0.9751
<b>FMNIST</b>	0.5193	0.6578	0.8386	0.8053	0.8142

Table 5: Student accuracy under different noise scale  $\sigma$

clipping will be superior to normalization. This is because normalization retains the relative size information of the gradients while clipping retains the absolute size information.

**Norm Bound.** Norm bound is an important hyper-parameter of our DP mechanism  $\mathcal{A}_{C,\sigma}$ . We conduct experiments to investigate how norm bound  $C$  affects the performance of student model. The results are shown as the red and green lines in Fig. 2. As we see, the student performs better when  $C$  is small. This is because normalization preserves information about the relative size of the gradients. Although smaller  $C$  will be more affected by DP noise, it will allow more training epochs. So the student performs better when  $C$  is small.

**Noise Scale.** Like norm bound, noise scale is also an important hyper-parameter of our DP mechanism  $\mathcal{A}_{C,\sigma}$ . To study the effect of it, we conduct experiments and the results are shown in Tab. 5. We observe that the model performance gets better as noise scale  $\sigma$  increases from 20 to 100. This is because a larger noise scale consumes a smaller privacy budget per epoch, which allows the student model to learn more fully with a limited privacy budget. However, we find that there is a slightly worse decrease in model performance as noise scale increases from 100 to 500. This is because the gradients will be more broken with a larger noise scale, thus leading to a slightly worse performance. In practical applications, a trade-off should be made based on the actual situation.

**Composition of Loss Functions.** To check the effect of loss terms in the training generator, we investigate how each

Dataset	CE	IE	Norm	Accuracy
<b>MNIST</b>	✓	✓	✓	<b>0.9751</b>
	✗	✓	✓	0.9691
	✓	✗	✓	0.5463
	✓	✓	✗	0.9496
<b>FMNIST</b>	✓	✓	✓	<b>0.8988</b>
	✗	✓	✓	0.7649
	✓	✗	✓	0.5463
	✓	✓	✗	0.8806

Table 6: Impact of each term in  $\mathcal{L}_G$ . The test accuracy of student models trained on synthetic data under  $\varepsilon=10$  is reported. CE: Cross Entropy; IE: Information Entropy; Norm:  $l_2$ -Normalization.

component of loss function contributes to the performance of student with 34-layer ResNet as the teacher model. We evaluate how they impact the performance by adding or removing each component and the results are shown in Tab. 6. We note that the information entropy loss term is the most important component based on the results. This is because removing it will result in an imbalance in the classes of the synthetic data. Cross entropy loss contributes differently for different datasets, with 1% improvement on the MNIST dataset and 13% improvement on the FMNIST dataset.  $l_2$ -normalization is also a useful term though less critical, contributing to the 2%-3% of the performance improvement as shown in Tab. 6. In summary, all three compositions have a positive effect on the performance of the converted model.

## 6 Conclusion

Public pretrained models in model zoos may pose the risk of privacy leakage. To facilitate model deployment, we proposed a differentially private data-free distillation approach to convert sensitive teacher models into privacy-preserving student models. We train a generator for approximating the private dataset without the training data and student networks can be learned effectively through the knowledge distillation scheme. In addition, we perform normalization on the gradients of student outputs and add Gaussian noise to them to guarantee privacy. We also provide privacy analysis and convergence analysis for DPDFD. Extensive experiments are conducted to show the effectiveness of our approach. In the future, we will explore the approach in more real-world applications like federated learning on medical images.

## Acknowledgements

This work was partially supported by grants from the National Key Research and Development Plan (2020AAA0140001), and the Beijing Natural Science Foundation (19L2040).

## References

- [Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, et al. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016.
- [Allen-Zhu, 2017] Z Natasha Allen-Zhu. Natasha 2: Faster non-convex optimization than *sgd*. *arXiv:1708.08694*, 2017.
- [Bottou *et al.*, 2018] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM Review*, 60(2):223–311, 2018.
- [Bu *et al.*, 2022] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. Automatic clipping: Differentially private deep learning made easier and stronger. *arXiv:2206.07136*, 2022.
- [Cao *et al.*, 2021] Tianshi Cao, Alex Bie, Arash Vahdat, Sanja Fidler, and Karsten Kreis. Don’t generate me: Training differentially private generative models with sinkhorn divergence. In *NeurIPS*, pages 12480–12492, 2021.
- [Chen *et al.*, 2019] Hanqing Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *ICCV*, pages 3514–3522, 2019.
- [Chen *et al.*, 2020] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. Gs-wgan: A gradient-sanitized approach for learning differentially private generators. In *NeurIPS*, pages 12673–12684, 2020.
- [Chen *et al.*, 2022] 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 *CVPR*, pages 8387–8396, 2022.
- [Choi *et al.*, 2020] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Lee Jungwon. Data-free network quantization with adversarial knowledge distillation. In *CVPR Workshop*, pages 710–711, 2020.
- [Deng and Zhang, 2021] Xiang Deng and Zhongfei Zhang. Graph-free knowledge distillation for graph neural networks. *arXiv:2105.07519*, 2021.
- [Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.
- [Dwork and Roth, 2014] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *FTCS*, pages 211–407, 2014.
- [Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
- [Fang *et al.*, 2022] Gongfan Fang, Kanya Mo, Xinchao Wang, Jie Song, Shitao Bei, Haofei Zhang, and Mingli Song. Up to 100x faster data-free knowledge distillation. In *AAAI*, pages 6597–6604, 2022.
- [Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, pages 1322–1333, 2015.
- [Ge *et al.*, 2023] Shiming Ge, Bochao Liu, Pengju Wang, Yong Li, and Dan Zeng. Learning privacy-preserving student networks via discriminative-generative distillation. *IEEE TIP*, 32:116–127, 2023.
- [Ghadimi and Lan, 2013] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for non-convex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- [Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.
- [Harder *et al.*, 2021] Frederik Harder, Kamil Adamczewski, and Mijung Park. Dp-merf: Differentially private mean embeddings with random features for practical privacy-preserving data generation. In *AISTATS*, pages 1819–1827, 2021.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [Howard *et al.*, 2017] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.
- [Huang *et al.*, 2016] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661, 2016.
- [Jia *et al.*, 2019a] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nezihe Merve Gurel, Bo Li, Ce Zhang, Costas J Spanos, and Dawn Song. Efficient task-specific data valuation for nearest neighbor algorithms. *Proc. VLDB Endow*, pages 1610–1623, 2019.
- [Jia *et al.*, 2019b] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *AISTATS*, pages 1167–1176, 2019.
- [Jordon *et al.*, 2019] James Jordon, Jinsung Yoon, and Michaela Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *ICLR*, 2019.
- [Krizhevsky *et al.*, 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.



- [Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [LeCun *et al.*, 1998] 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.
- [Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Tang Xiaoou. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.
- [Long *et al.*, 2021] Yunhui Long, Boxin Wang, Zhuolin Yang, Bhavya Kailkhura, Aston Zhang, Carl Gunter, and Bo Li. G-pate: Scalable differentially private data generator via private aggregation of teacher discriminators. In *NeurIPS*, pages 2965–2977, 2021.
- [Lopes *et al.*, 2017] Raphael Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv:1710.07535*, 2017.
- [McMahan *et al.*, 2018] H Brendan McMahan, Galen Andrew, Ulmar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. A general approach to adding differential privacy to iterative training procedures. *arXiv:1812.06210*, 2018.
- [Mironov, 2017] Ilya Mironov. Rényi differential privacy. In *CSF*, pages 263–275, 2017.
- [Papernot *et al.*, 2017] Nicolas Papernot, Martín Abadi, Ulmar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *ICLR*, 2017.
- [Papernot *et al.*, 2018] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *ICLR*, 2018.
- [Papernot *et al.*, 2021] Nicolas Papernot, Abhradeep Thakurta, Shuang Song, Steve Chien, and Úlfar Erlingsson. Tempered sigmoid activations for deep learning with differential privacy. In *AAAI*, pages 9312–9321, 2021.
- [Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [Srinivas and Babu, 2015] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv:1507.06149*, 2015.
- [Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.
- [Takagi *et al.*, 2021] Shun Takagi, Tsubasa Takahashi, Yang Cao, and Masatoshi Yoshikawa. P3gm: Private high-dimensional data release via privacy preserving phased generative model. In *ICDE*, pages 169–180, 2021.
- [Wang *et al.*, 2021] Boxin Wang, Fan Wu, Yunhui Long, Luka Rimanic, Ce Zhang, and Bo Li. Datalens: Scalable privacy preserving training via gradient compression and aggregation. In *CCS*, pages 2146–2168, 2021.
- [Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.
- [Xie *et al.*, 2018] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv:1802.06739*, 2018.
- [Yang *et al.*, 2019] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. Neural network inversion in adversarial setting via background knowledge alignment. In *CCS*, pages 225–240, 2019.
- [Yang *et al.*, 2021] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. *arXiv:2110.14795*, 2021.
- [Ye *et al.*, 2020] Yunan Ye, Hengzhi Pei, Boxin Wang, Pin-Yu Chen, Yada Zhu, Ju Xiao, and Bo Li. Reinforcement-learning based portfolio management with augmented asset movement prediction states. In *AAAI*, pages 1112–1119, 2020.
- [Zagoruyko and Komodakis, 2017] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *CVPR*, 2017.
- [Zhang *et al.*, 2018] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.
- [Zhao *et al.*, 2022] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *CVPR*, pages 11953–11962, 2022.
- [Zhu *et al.*, 2021] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *ICML*, pages 12878–12889, 2021.