Decoupled Self-Knowledge Distillation Makes Differentially Private Deep Learning Stronger

Dengfeng Zhao[1,2], DaXuan Xue[1,2], Suyun Zhao[1,2], Cuiping Li[1,2], and Hong Chen[1,2] (🖂)

[1] Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, Beijing, China
[2] School of Information, Renmin University of China, Beijing, China
{zhaodengfengruc,xuedaxuan0726,zhaosuyun,licuiping,chong}@ruc.edu.cn

Abstract. To address the significant risks of privacy leakage, various differential privacy techniques have been incorporated into deep learning models. However, these privacy-preserving methods often result in noticeable performance degradation. To balance privacy and utility, this paper proposes the Differentially Private with Decoupled Self-Knowledge Distillation (DPDSD) method, which effectively transfers high-usability knowledge to privacy-preserving deep networks during training. Specifically, DPDSD employs a teacher-student module designed with differential privacy: intermediate checkpoints serve as the teacher network, focusing on acquiring high-usability knowledge, while the student network emphasizes privacy protection via differential privacy stochastic gradient descent. Moreover, we decouple the distillation loss into two components: the loss for the target class and the loss for the non-target classes. This process adaptively adjusts targets by merging ground-truth labels with intermediate checkpoint predictions, enabling the model to progressively enhance its informativeness throughout training. Simultaneously, through refining knowledge from the teacher, the student achieves better performance while ensuring data privacy. Finally, extensive experiments on three public datasets have demonstrated that DPDSD can effectively improve model performance in the case of ensuring rigorous data privacy.

Keywords: Deep learning · Differential privacy · Knowledge distillation.

1 Introduction

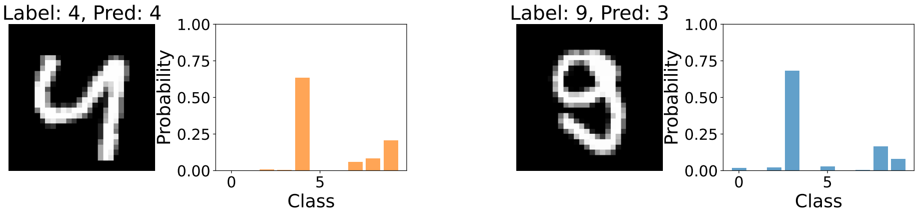
Deep learning (DL) has been successfully applied in computer vision and natural language processing tasks. However, recent studies [23, 24, 3] reveal that DL models are often over-parameterized and can inadvertently expose private information, enabling adversaries to re-identify individuals or reconstruct sensitive attributes like credit card details and medical records. Accordingly, DL models are exposed to huge privacy leakage risks. Differential privacy (DP)[7], is then introduced into DL as it can theoretically guarantee that an adversary cannot infer whether a specific user's data is included in the training of a model.

DL models with DP are effective against various attacks, including membership inference [23], property inference [11] and data extraction [4].

Unlike poor privacy of the DL models training directly on sensitive data, differentially private DL, injecting noise during the training phase, could protect the sensitive information from data. There are many such kinds of differentially private DL techniques. For example, Differential Privacy Stochastic Gradient Descent (DPSGD) [1] safeguards the privacy of individual data points by introducing Gaussian noise to the gradient during each iteration of the optimization process. Basically, these approaches demonstrate robust privacy preservation but result in catastrophic utility degradation. Consequently, most of those methods fail to balance utility and privacy. One feasible way may combine DP and knowledge transfer techniques. Private Aggregation of Teacher Ensembles (PATE) [19] is such a method that trains multiple independent teacher models on private data and then uses their noisy predictions on public data to train a differentially private student model. However, this method requires training numerous teacher models for voting and assumes that both public and private data come from the same distribution. Usually, such auxiliary data are hard to achieve. Accordingly, it is promising to propose a novel framework.

To better address the trade-off between privacy and utility, this paper proposes a Differential Privacy with Decoupled Self-knowledge Distilling (DPDSD) framework that introduces the privacy-preserving mechanisms into knowledge transfer techniques. Following the knowledge distillation paradigm, we introduce a general method to employ intermediate checkpoints as teacher model in differentially private deep learning. Specifically, the model performs a dual role: it serves as both the student and the teacher. This means that a student model becomes a teacher model itself, gradually employing its own accumulated knowledge to refine and enhance the training targets. During the training phase, the model employs the DPSGD algorithm to provide privacy safeguards. The proposed DPDSD focuses on two primary goals: predicting ground-truth labels and matching the output distribution of intermediate checkpoints. In this process, targets are adaptively adjusted by combining ground-truth labels with intermediate checkpoint predictions, enabling the model to gradually enhance their informativeness during training. Guided by the outputs from the teacher model, the student model with the privacy preserving mechanism is then refined.

Our DPDSD approach can be interpreted within a knowledge distillation framework that incorporates differential privacy, wherein the model progressively transforms from a student into its own teacher. The underlying principle is that the higher the confidence level of the intermediate checkpoint model on the training samples, the more reliable and valuable the knowledge it can provide. However, as illustrated in Figure 1b, the introduction of DP noise may cause intermediate checkpoints assign higher confidence to incorrect predictions, which could convey ambiguous knowledge and mislead the learning process of the student model. To mitigate the impact of incorrect predictions generated by the teacher model, inspired by the effectiveness of the decoupled method [30] , we reformulate the distillation loss in DPDSD into a combination of Target Class



(a) Correct prediction

(b) Incorrect prediction

Fig. 1: Prediction results of the teacher model on MNIST handwritten digit samples. The figure illustrates a correct prediction (a) and an incorrect prediction (b), with corresponding probability distributions.

Self-Knowledge Distillation (TCSD) loss and Non-Target Class Self-Knowledge Distillation (NCSD) loss. This facilitates the student model to gain richer and clearer semantic knowledge, which in turn enhances its generalization performance.

In summary, the main contribution of this paper are listed as follows.

- We propose a novel self-distillation framework that combines differential privacy and self-knowledge transfer techniques to balance privacy and utility in deep learning.
- A decoupled distillation loss is proposed and leveraged to enhance the performance of deep learning models with differential privacy.
- The use of intermediate checkpoint as teacher model exploits the post-processing property of differential privacy, ensuring no additional privacy cost and maintaining the privacy guarantees of the trained model.
- We conduct numerous experiments to demonstrate that DPDSD performs superiorly compared to three state-of-the-art learning frameworks with privacy protection techniques, given equal privacy cost.

2 Preliminary

2.1 Differential Privacy

Differential privacy [8] is a rigorous mathematical framework that formally defines data privacy. It ensures that the inclusion or exclusion of a single individual's data does not have a significant impact on the results of any analysis, thus providing quantifiable protection against the identification of individuals. This is achieved by ensuring that the probability distribution of the output results remains virtually unchanged whether or not any individual data is included in the dataset. We consider the (ϵ, δ)-DP [8] as follows:

Definition 1 ((ϵ, δ)-Differential Privacy). *A mechanism $\mathcal{A} : D^n \rightarrow R^d$ satisfies (ϵ, δ)-differential privacy if for any two adjacent datasets $D, D' \in D^n$ that* *differ by one element, and for all $S \subseteq R$, the following inequality holds:*

$$\Pr[\mathcal{A}(D) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{A}(D') \in S] + \delta \tag{1}$$

Here, ϵ is a non-negative parameter that controls the privacy loss of the algorithm. A smaller ϵ means better privacy, but possibly less accurate results. The parameter δ, ideally, should be smaller than the inverse of the size of the dataset to ensure that the probability of breaching privacy is negligible.

A highly attractive property of DP is that it is immunity to post-processing, meaning that any subsequent analysis performed on the outputs of a DP algorithm does not degrade its privacy guarantees.

Lemma 1 (Post-processing). *Let M be a mechanism satisfying (ϵ, δ)-differential privacy. If f is a function whose input is the output of M, then $f(M)$ also satisfies (ϵ, δ)-differential privacy.*

2.2 Deep Learning with Differential Privacy

To train deep learning models while preserving privacy, the most commonly used algorithm is DPSGD [1]. DPSGD modifies traditional SGD by incorporating noise into the gradient computations, thereby protecting the privacy of individual data points used in training. In DPSGD, gradients are clipped to a predefined norm threshold C before the addition of noise. Gradient clipping ensures that the influence of any single data point on the model update is bounded, thus limiting the sensitivity of the gradients. Mathematically, the gradient g_t at iteration t is clipped as follows:

$$\mathbf{g}_t \leftarrow \mathbf{g}_t \cdot \min\left(1, \frac{C}{\|\mathbf{g}_t\|_2}\right) \tag{2}$$

To achieve DP, noise is added to the clipped gradients. The noise is typically sampled from a Gaussian distribution with mean zero and standard deviation proportional to the clipping threshold and the privacy parameters. The noisy gradient $\tilde{\mathbf{g}}_t$ is computed as:

$$\tilde{\mathbf{g}}_t = \mathbf{g}_t + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}) \tag{3}$$

Here, σ is the noise multiplier that controls the trade-off between model utility and privacy.

At a high level, privacy loss is evaluated by analyzing the Gaussian Mechanism, utilizing privacy amplification through subsampling and the composition theorem across multiple iterations. This methodology enables the release of all intermediate checkpoints generated during the training process. To obtain tighter privacy bounds, it is essential to use techniques such as the Rényi Differential Privacy [18] or numerical composition algorithms [12].

2.3 Knowledge Distillation

Knowledge Distillation (KD) [13] is an advanced technique in deep learning for transferring knowledge from a larger, complex model (often called the "teacher") to a smaller model (often called the "student"). In KD, the student model learns not only from the traditional one-hot labels but also from the predictive probabilities (soft labels) generated by the teacher model. The benefit of using soft labels is intuitive: soft labels contain more knowledge about the relative differences between classes. This approach allows the student model to simulate the teacher's behavior, including class probability distributions and logits, resulting in improved performance even with its reduced size.

The goal of KD is to minimize the Kullback-Leibler (KL) divergence between the teacher and student model outputs. The KD loss function is:

$$L_{KD} = KL(p^t \parallel p^s) = \sum_{j=1}^{C} p_j^t \log\left(\frac{p_j^t}{p_j^s}\right), \tag{4}$$

where p_j^t and p_j^s represent the soft labels for the j-th class as predicted by the teacher and student models, respectively.

Self-KD [29] employs an identical neural network architecture for both the teacher and the student models. Several studies [10, 14] have attempted to utilize the student network itself as a teacher, which enhances its effectiveness by leveraging its own knowledge. Similar to the progressive self-knowledge distillation approach [15], our method employs the model's previous iteration as the teacher model for the current training iteration. This optimization allows for a more dynamic exchange of learned features and representations, potentially enriching the overall model performance.

To protect data privacy, some studies [9, 26] effectively combine knowledge distillation and differential privacy, but their primary focus on model compression limits broader applicability and often incurs additional privacy costs. Recent studies [19] have shown that public data and knowledge transfer techniques can be effectively utilized to enhance the utility of private training. Our DPDSD approach does not require external datasets for knowledge transfer to learn privacy-preserving student networks.

3 Method

In this section, we introduce our DPDSD framework, as illustrated in Figure 2. Both the teacher and student networks employ an identical neural network architecture. The student model, represented by θ_s, is trained under DP conditions. The teacher model, denoted by θ_t, is usually updated based on the previous student model, without directly receiving gradient updates. Given that the assessment of privacy loss in differentially private deep learning depends on composition theorems, which allow for the public release of intermediate training checkpoints, we sequentially apply DPDSD across multiple generations of

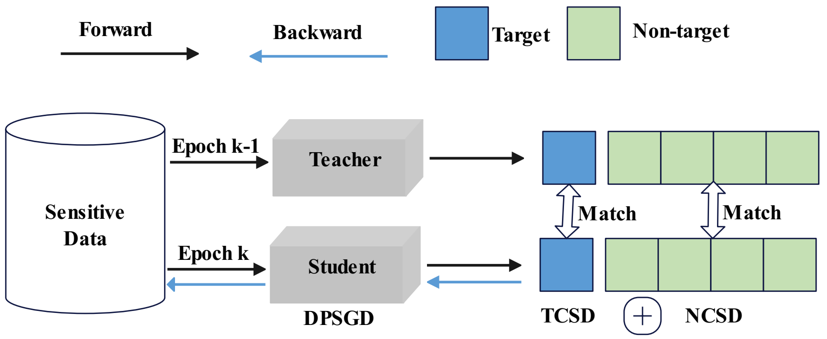


Fig. 2: An illustration of DPDSD. In each epoch K, the model trained with DPSGD by incorporating self-knowledge from the $k-1$-th model.

knowledge transfer. In this process, each k-th model is trained with DPSGD by incorporating knowledge from the $k-1$-th model. In DPDSD, we exploit intermediate checkpoints as auxiliary teacher models, effectively guiding subsequent model iterations without additional privacy costs, thereby enhancing the utility of the model.

3.1 Formulation

Suppose we have a private dataset, denoted as D, comprising labeled instances from C distinct classes. It's given by $D = \{(x_i, y_i)\}_{i=1}^{N}$ with size N. The labels for each instance are in the set Y, expressed as $Y = \{y_i\}_{i=1}^{N}$ with each y_i belonging to the set $\{1, 2, \ldots, C\}$.

For a classification task with C classes, the probability p_i for the i-th class given the logit z is defined as:

$$\hat{p}_i = \frac{e^{z_i/\tau}}{\sum_{c=1}^{C} e^{z_c/\tau}} \tag{5}$$

where τ is the temperature parameter. Higher values of τ produce a softer probability distribution, which alleviates over-confidence in neural networks.

The most commonly used loss function in multi-class classification model training is the cross-entropy (CE) metric, which contrasts the predicted label distribution with the true label distribution. For multi-class classification , the cross-entropy loss is expressed as:

$$L_{CE} = -\sum_{i=1}^{C} p_i \log(\hat{p}_i) \tag{6}$$

Here, p_i is the true label of the class i^{th} and $\hat{p}_i$ is the probability predicted by the model.

The Self-knowledge distillation aims to enhance model performance by progressively transferring knowledge accumulated during different training epochs. Within this framework, the model trained during one epoch becomes the teacher model for the next epoch, providing guidance to the student model in the subsequent training iteration. For simplicity, we denote the model from the $k-1$-th epoch as the teacher model T, and the model from the k-th epoch as the student model S.

Recent work [10] has observed that the gradient resulting from KD can be decomposed into two components: a residual knowledge term that captures information from the incorrect outputs, and a ground-truth component that corresponds to a rescaling of the original gradient derived from the true labels. Let t denote the target class, with T_i and S_i representing the soft labels from the teacher and student models for class i, respectively. Our distillation loss, consisting of target and non-target class losses, as detailed below.

$$\begin{aligned} L_{\mathrm{SD}} &= -\sum_{i=1}^{C} T_i \log(S_i) && (7) \\ &= -T_t \log(S_t) - \sum_{i \neq t}^{C} T_i \log(S_i). && (8) \end{aligned}$$

The effectiveness of the student network heavily relies on the quality of the teacher network. The more confident the teacher network is in the target class of a training sample, the more reliable and valuable the knowledge it provides. Due to the influence of noise, the teacher model may sometimes assign higher confidence to incorrect predictions. This can result in the student model learning misleading or ambiguous information, which may negatively affect its performance. To enable the student model to adaptively learn from the teacher model, we define γ as:

$$\gamma = T_t - \frac{1}{C}. \tag{9}$$

A larger γ indicates a higher level of trust in the teacher model. When $\gamma < 0$, it indicates that the teacher has been heavily affected by noise and is unable to provide valuable information.

Inspired by Decoupled knowledge distillation [30], we reformulated the slef-distillation loss into two parts: binary logit distillation for the target class and multi-category logit distillation for non-target classes. We term these components as Target Classification Self-Knowledge Distillation (TCSD) and Non-Target Classification Self-Knowledge Distillation (NCSD) for simplicity. To reduce the impact of noise, the TCSD and NCSD loss functions are formulated as follows:

$$L_{\mathrm{TCSD}} = -e^{\gamma} T_t \log(S_t) \tag{10}$$

$$L_{\mathrm{NCSD}} = -e^{\gamma} \sum_{i \neq t}^{C} T_i \log(S_i) \tag{11}$$

Our DPDSD method independently considers TCSD and NCSD within a decoupled framework. We introduce two hyper-parameters, α and β, as the respective weights for TCSD and NCSD. The DPDSD loss function is then expressed as follows.

$$L_{\mathrm{DPDSD}} = \alpha L_{\mathrm{TCSD}} + \beta L_{\mathrm{NCSD}}. \tag{12}$$

To enhance the performance of the student model, our training method combines CE loss and distillation loss. The total loss function for student model can be written as:

$$L_{\mathrm{total}} = L_{\mathrm{CE}} + L_{\mathrm{DPDSD}}. \tag{13}$$

Essentially, the CE loss maintains the foundational integrity of student learning, while the decoupled self-distillation loss bridges the experience gap between teacher and student, ensuring that the student captures the refined knowledge from the teacher.

3.2 Differentially Private with Decoupled Self-knowledge Distillation (DPDSD)

In DPSGD and its variants, it is typically assumed that each step of the iterative training process is public, enabling an adversary to exploit all intermediate checkpoints for potential attacks. This assumption facilitates the necessary differential privacy analysis of the overall mechanism. However, since only the final model is typically used for predictions, it prompts the question of whether utility can be improved by utilizing the intermediate models. recent studies [22, 21] suggest that utilizing intermediate checkpoints can enhance the utility of DP-trained models without compromising their privacy guarantees. These methods involve aggregating either the parameters of checkpoints or their outputs to enhance the accuracy of the final model. In this work, we primarily focus on post-processing techniques to improve differentially private training algorithms without the use of public data. We propose an alternative approach that uses intermediate checkpoints as auxiliary teacher networks, which can further improve prediction accuracy.

DPDSD adopts a self-distillation framework where a student model is trained using DPSGD with privacy guarantees. The teacher model is dynamically selected as the student model from the previous epoch, enabling continuous knowledge transfer throughout the training process. This evolving teacher-student paradigm eliminates the need for public data and instead relies on the intermediate models generated during training, making it well-suited for differentially private settings.

We initially train the student model using DPSGD to obtain a pre-trained model. This pre-trained model is then utilized to generate soft labels for its own further training. Contrasting with the standard approach in KD, our methodology features a dynamic, evolving teacher model rather than a static one. As training progresses, the teacher model is continuously updated. Among all the previous models suitable to be the teacher, we select the model from the $k-1$-th

Algorithm 1 Differentially Private Stochastic Gradient Descent with Decoupled Self-knowledge Distillation (DPDSD)

Input: Dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{n}$, learning rate η, noise scale σ, gradient clipping norm C, temperature τ, number of epochs K, batch size B

Output: Differentially private model parameters θ_S

```
Initialize student model parameters $\theta_S$ randomly.
for $k = 1$ to $K$ do
    Set teacher model parameters $\theta_T \leftarrow \theta_S^{k-1}$.
    for each mini-batch $\mathcal{B}$ in dataset $\mathcal{D}$ do
        Compute teacher logits: $p_T(x_i) = f(\theta_T, x_i, \tau),\ \forall x_i \in \mathcal{B}$.
        Compute student logits: $p_S(x_i) = f(\theta_S, x_i, \tau),\ \forall x_i \in \mathcal{B}$.
        Compute distillation loss:
            $\mathcal{L}_{DPDSD} = \alpha \mathcal{L}_{TCSD} + \beta \mathcal{L}_{NCSD}$
        Compute supervised loss: $\mathcal{L}_{CE}$.
        Compute total loss: $\mathcal{L}_{total} = \mathcal{L}_{CE} + \mathcal{L}_{DSD}$.
        Compute clipped gradients:
            $\tilde{g}_i = \frac{\nabla_\theta \mathcal{L}_{total}}{\max(1, \|\nabla_\theta \mathcal{L}_{total}\|_2 / C)}, \quad i \in \mathcal{B}$
        Add noise and update parameters:
            $\theta_S^k \leftarrow \theta_S^k - \eta \left( \frac{1}{B} \sum_{i \in \mathcal{B}} \tilde{g}_i + \mathcal{N}(0, \sigma^2 C^2 I) \right)$
    end for
end for
return $\theta_S^K$
```

epoch. This choice is based on its capacity to provide the most valuable information among the candidates. Because the predictions from the model at the $k-1$-th epoch are essential for training at the k-th epoch, we load the $k-1$-th epoch model into memory at the start of the k-th epoch. This ensures that the previous predictions, which are necessary for softening targets, are also computed during the forward passes. During each iteration, the teacher model produces soft labels, which are then used to guide the training of the student model with added differential privacy guarantees. Considering that DP noise may lead intermediate checkpoints to assign higher confidence to incorrect predictions, this could lead to ambiguous knowledge being conveyed and potentially mislead the student model's learning process. we use the TCSD loss to focus on the knowledge related to the target, while the NCSD loss considers the knowledge among non-target logits, facilitating the student model to adaptively learn from the teacher model. A comprehensive overview of the optimization procedure can be found in Algorithm 1.

3.3 Discussion of Privacy

In the privacy analysis of DPSGD and its variants, the (ϵ, δ) guarantee is derived from an analysis of the Gaussian Mechanism, which incorporates privacy amplification through subsampling and composition across multiple iterations. It is crucial to recognize that the privacy cost is incurred for all intermediate models generated during the training process. This implies that each intermediate model contributes to the overall privacy budget, regardless of whether only the final model is released.

The privacy guarantee of DPDSD is derived from the differential privacy properties of DPSGD. Since the student model is trained using DPSGD with gradient clipping and noise addition, the entire training process adheres to the (ϵ, δ)-differential privacy framework. The use of intermediate models as dynamic teachers does not incur additional privacy costs due to the post-processing property of differential privacy. This ensures that the privacy analysis of standard DPSGD remains valid for DPDSD.

Theorem 1. *There exist constants c_1 and c_2 such that, given the sampling probability $q = \frac{B}{N}$ and the number of iterations T, Algorithm 1 achieves (ϵ, δ)-differential privacy for any $\epsilon \leq c_1 q^2 T$ and $\delta > 0$ if the noise scale σ satisfies:*

$$\sigma \geq c_2 \frac{q\sqrt{T \log(1/\delta)}}{\epsilon}.$$

The model parameters produced by DPSGD satisfy privacy guarantees. Therefore, using the post-processing property of DP, the output of the DPDSD algorithm is differentially private.

4 Experiments

To verify the effectiveness of our proposed DPDSD approach, we conduct comprehensive experiments on three image classification datasets: MNIST [6], Fashion-MNIST [27], and CIFAR-10 [16].

We focused our assessment on measuring both utility and privacy, benchmarking our approach against three state-of-the-art techniques: DPSGD [1], DPKD [17] and DPEMA [5]. Methods such as PATE [19], which require access to additional public datasets, are excluded due to unavailability of such datasets in our problem setting. Similarly, approaches that focus on orthogonal aspects, such as feature engineering [25] or alternative model structures [20], are not included in our comparison.

The primary objective of these experiments is to assess the efficacy of DPDSD in maintaining model accuracy while ensuring differential privacy, and to establish its potential as a superior approach in privacy-preserving deep learning. Our experiments are implemented using the Opacus [28] repository, which is integrated with Pytorch.

4.1 Experiments Settings

Datasets. MNIST and Fashion-MNIST (FMNIST) are 10-class datasets, each containing 60K training images and 10K testing images. The datasets feature 28×28 grayscale images, with MNIST focusing on handwritten digits and FMNIST on fashion items. CIFAR-10 comprises 60K 32×32 color images across 10 classes, with 50K for training and 10K for testing, including categories such as airplanes, automobiles, and animals.
Models. we use the network architecture from opacus library. The model used for the MNIST and FMNIST dataset comprises two convolutional layers followed by two linear layers. For CIFAR-10 dataset, we employ a model with four convolutional layers and apply one fully connected layer.
Parameters. We use B to represent the batch size, C to represent the clip norm, η to represent the learning rate, τ to represent the temperature. A small clipping threshold often works best [2], we set C to 0.1 for all datasets. For MNIST, $B = 1200, \eta = 0.8, \tau = 2, \alpha = 0.1, \beta = 0.5$. In non-private training, the model achieved 0.992 accuracy on the MNIST dataset after 60 epochs. For FMNIST,$B = 1600, \eta = 3, \tau = 5, \alpha = 0.1, \beta = 0.3$. It reached 0.898 accuracy on the FMNIST dataset after 60 epochs. For the CIFAR-10 dataset, $B = 1000, \eta = 3, \tau = 5, \alpha = 0.1, \beta = 0.3$. the model attained 0.825 accuracy after 100 epochs.
Benchmarks. The DPSGD [1] algorithm is a basic method for integrating DP into DL using gradient perturbation to guarantee privacy. The DPKD algorithm [17] aims to achieve differential privacy protection through the technique of knowledge distillation. It employs a two-step process: first, the teacher model is trained with differential privacy guarantees, and then the knowledge from the teacher model is distilled into a student model. The DPEMA algorithm [5] aggregates the parameters from the intermediate checkpoints of DP-SGD and then uses the resulting aggregated parameters for inference.

4.2 Main Results

In order to demonstrate the trade-off between model accuracy and privacy, Table 1 shows how model accuracy varies with the associated privacy cost. As illustrated in Table 1, We observe that our method consistently achieves higher accuracy than other techniques under the same privacy budget. For instance, when evaluating the CIFAR-10 dataset under privacy level ($\epsilon = 3$), our DPDSD achieves an accuracy of 63.34%. In comparison, DPSGD registers at 59.99% (a decrease of 3.35%), DPKD at 56.19% (a decrease of 7.15%), and DPEMA at 61.62% (a decrease of 1.72%). These results highlight the primary advantage of the DPDSD framework. By decoupling self-knowledge distillation from the privacy-preserving training process, DPDSD utilizes historical model checkpoints as teachers, enabling more efficient knowledge transfer. Our approach enables the model to preserve more useful information while maintaining strict privacy guarantees, thereby enhancing model accuracy under the same privacy budget. Unlike traditional methods that require separate teacher models, which consume additional privacy budget, DPDSD ensures efficient knowledge transfer without any added privacy overhead.

To better illustrate the model's performance throughout the training process, we plotted the evolution of the testing accuracy over the number of epochs, as shown in Figure 3. As demonstrated, DPDSD not only attains higher accuracy compared to DPSGD, DPKD, and DPEMA, but also achieves this accuracy faster. This efficiency is driven by the use of intermediate checkpoints as auxiliary teacher models, which effectively guide the learning process and enable the model to converge faster, requiring fewer epochs to achieve a comparable level of accuracy. It is clear that our algorithm exhibits greater robustness to the impact of noise addition.

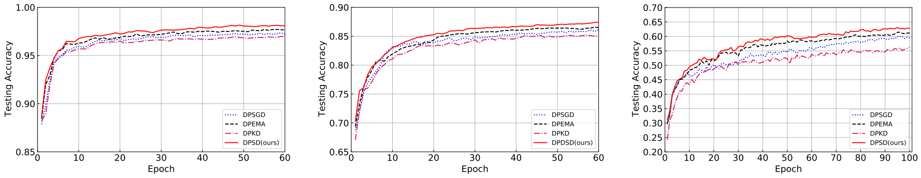
We now analyze how our DPDSD method achieves higher accuracy compared to each state-of-the-art technique. DPDSD outperforms DPSGD by leveraging the additional knowledge provided by intermediate checkpoints, which serve as auxiliary teacher models to guide the learning process more effectively. Unlike DPKD, our method does not require pre-training a teacher model on private datasets, thereby avoiding additional privacy costs. In contrast to DPEMA, DPDSD adaptively learns from intermediate checkpoints throughout the training process, enhancing both convergence speed and model utility.

Table 1: Performance Comparison of existing methods on MNIST, FashionMNIST, and CIFAR10 Datasets under different privacy budget $\epsilon = 1, 2, 3$ and $\delta = 10^{-5}$.

Dataset	**Method**	$\epsilon = 1$	$\epsilon = 2$	$\epsilon = 3$
MNIST	DPSGD	95.51%	96.46%	97.01%
	DPKD	94.31%	95.59%	96.88%
	DPEMA	96.63%	97.51%	97.93%
	DPDSD	**97.21%**	**97.95%**	**98.42%**
FashionMNIST	DPSGD	81.29%	85.28%	86.13%
	DPKD	79.58%	84.66%	85.85%
	DPEMA	82.54%	86.06%	86.91%
	DPDSD	**83.22%**	**86.53%**	**87.68%**
CIFAR10	DPSGD	48.74%	55.05%	59.99%
	DPKD	46.91%	52.58%	56.19%
	DPEMA	51.68%	58.51%	61.62%
	DPDSD	**52.36%**	**59.87%**	**63.34%**

4.3 Impact of the parameters

In this section, our focus is on evaluating how various parameters of the DPDSD algorithm affect its performance. These parameters include the learning rate η, the Batch size B, and the temperature τ. We choose the CIFAR10 dataset for our training.



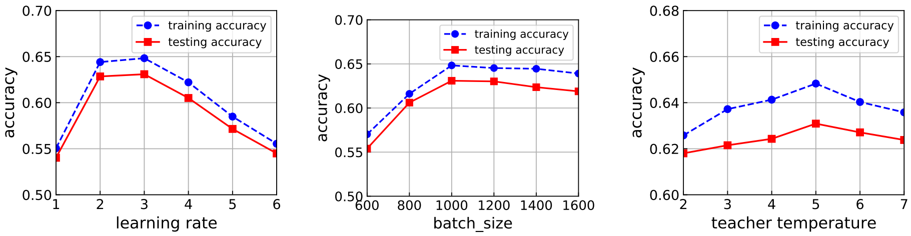
(a) MNIST (b) FMNIST (c) CIFAR10

Fig. 3: The comparative result of our algorithm with SOTA methods regarding the testing accuracy versus training epochs.

Learning rate. As illustrated in Fig. 4a, the model's accuracy remains stable within the learning rate range of [1, 6], reaching its peak at 3. Lower learning rates result in suboptimal performance, and higher learning rates lead to a decline in accuracy, suggesting that careful tuning of the learning rate is crucial for optimal model performance.

Batch size. The batch size B influences the sampling ratio. While a larger batch size can increase the sampling ratio and decrease the number of training steps, it's essential to strike a balance when selecting the lot size. Smaller B enable more epochs to run, whereas larger B make the added noise less significant in comparison. As illustrated in Fig. 4b, with an increase in the lot size, performance improves, reaching its peak at $B = 1000$, and then declines. This observation aligns with our earlier analysis, indicating that both excessively high and low sampling ratios can lead to a decrease in performance.

Temperature τ. The temperature τ affects the smoothness of the teacher model's output during the self-distillation process. We experimented with τ values ranging from 2 to 7, as illustrated in Fig. 4c. Higher values of τ result in a smoother output distribution, which facilitates effective knowledge transfer under differential privacy constraints. The optimal performance was observed at $\tau = 5$, as it provides a good balance between overly smooth outputs and insufficient smoothness.



(a) learning rate (b) variable lot size (c) variable temperture

Fig. 4: CIFAR10 accuracy when one parameter varies, and the others are fixed at reference values.

4.4 Ablation studies

Following the achievement of promising performance, we further investigated how each component of the loss function influences model performance on the FMNIST dataset. TCSD emphasizes the knowledge related to the target class, as its corresponding loss function focuses solely on binary probabilities associated with correct classification. In contrast, NCSD captures the relationships among non-target classes, leveraging the model's capacity to distinguish between these classes. We evaluated the impact of these components by systematically adding or removing them from the loss function, with the results presented in Table 2. The ablation results indicates that NCSD may be more crucial for enhancing model performance than TCSD.

Table 2: Impact of each term in L_{DPDSD}

CE	**TCSD**	**NCSD**	**Accuracy**
✓	✓	✓	**0.8693**
✓	×	✓	0.8642
✓	✓	×	0.8611

5 CONCLUSION

In this study, to balance utility and privacy, our proposed DPDSD designs a privacy-preserving framework by combining differential privacy and self-knowledge transfer techniques. On one hand, the student network leverages differential privacy techniques to achieve a robust theoretical privacy guarantee on sensitive information. On the other hand, the student network, distilling knowledge from the intermediate checkpoints, can effectively reduce the perturbation of the noise from differential privacy. Finally, through extensive experiments, we find that our DPDSD enables the student model to strike a promising trade-off between high utility and strong privacy protection. Future research may explore its scalability and potential improvements to increase its overall efficacy.

Acknowledgement. This study is supported by the National Key Research and Development Program of China (2023YFB4503600) and the National Natural Science Foundation of China (U23A20299, U24B20144, 62172424, 62276270, 62322214).

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
2. Bu, Z., Wang, Y.X., Zha, S., Karypis, G.: Automatic clipping: Differentially private deep learning made easier and stronger. Advances in Neural Information Processing Systems **36** (2024)
3. Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The secret sharer: Evaluating and testing unintended memorization in neural networks. In: 28th USENIX security symposium (USENIX security 19). pp. 267–284 (2019)
4. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2633–2650 (2021)
5. De, S., Berrada, L., Hayes, J., Smith, S.L., Balle, B.: Unlocking high-accuracy differentially private image classification through scale. arXiv preprint arXiv:2204.13650 (2022)
6. Deng, L.: The mnist database of handwritten digit images for machine learning research [best of the web]. IEEE signal processing magazine **29**(6), 141–142 (2012)
7. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3. pp. 265–284. Springer (2006)
8. Dwork, C., Roth, A., et al.: The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science **9**(3–4), 211–407 (2014)
9. Flemings, J., Annavaram, M.: Differentially private knowledge distillation via synthetic text generation. arXiv preprint arXiv:2403.00932 (2024)
10. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: International Conference on Machine Learning. pp. 1607–1616. PMLR (2018)
11. Ganju, K., Wang, Q., Yang, W., Gunter, C.A., Borisov, N.: Property inference attacks on fully connected neural networks using permutation invariant representations. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security. pp. 619–633 (2018)
12. Gopi, S., Lee, Y.T., Wutschitz, L.: Numerical composition of differential privacy. Advances in Neural Information Processing Systems **34**, 11631–11642 (2021)
13. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
14. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10664–10673 (2021)
15. Kim, K., Ji, B., Yoon, D., Hwang, S.: Self-knowledge distillation with progressive refinement of targets. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6567–6576 (2021)
16. Krizhevsky, A., Hinton, G.: Convolutional deep belief networks on cifar-10. Unpublished manuscript **40**(7), 1–9 (2010)

17. Mireshghallah, F., Backurs, A., Inan, H.A., Wutschitz, L., Kulkarni, J.: Differentially private model compression. Advances in Neural Information Processing Systems **35**, 29468–29483 (2022)
18. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th computer security foundations symposium (CSF). pp. 263–275. IEEE (2017)
19. Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., Talwar, K.: Semi-supervised knowledge transfer for deep learning from private training data. arXiv preprint arXiv:1610.05755 (2016)
20. Papernot, N., Thakurta, A., Song, S., Chien, S., Erlingsson, Ú.: Tempered sigmoid activations for deep learning with differential privacy. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 9312–9321. No. 10 (2021)
21. Rabanser, S., Thudi, A., Guha Thakurta, A., Dvijotham, K., Papernot, N.: Training private models that know what they don't know. Advances in Neural Information Processing Systems **36**, 53711–53727 (2023)
22. Shejwalkar, V., Ganesh, A., Mathews, R., Thakkar, O., Thakurta, A.: Recycling scraps: Improving private learning by leveraging intermediate checkpoints. arXiv preprint arXiv:2210.01864 (2022)
23. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
24. Song, C., Ristenpart, T., Shmatikov, V.: Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security. pp. 587–601 (2017)
25. Tramer, F., Boneh, D.: Differentially private learning needs better features (or much more data). arXiv preprint arXiv:2011.11660 (2020)
26. Wang, J., Bao, W., Sun, L., Zhu, X., Cao, B., Philip, S.Y.: Private model compression via knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 1190–1197 (2019)
27. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747 (2017)
28. Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., Nguyen, J., Ghosh, S., Bharadwaj, A., Zhao, J., Cormode, G., Mironov, I.: Opacus: User-friendly differential privacy library in PyTorch. arXiv preprint arXiv:2109.12298 (2021)
29. Yun, S., Park, J., Lee, K., Shin, J.: Regularizing class-wise predictions via self-knowledge distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13876–13885 (2020)
30. Zhao, B., Cui, Q., Song, R., Qiu, Y., Liang, J.: Decoupled knowledge distillation. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 11953–11962 (2022)