Robust and Resource-Efficient Data-Free Knowledge Distillation by Generative Pseudo Replay

Kuluhan Binici[1, 2], **Shivam Aggarwal**[2], **Nam Trung Pham**[1], **Karianto Leman**[1], **Tulika Mitra**[2]

[1]Institute for Infocomm Research, A*STAR, Singapore
[2]School of Computing, National University of Singapore

Abstract

Data-Free Knowledge Distillation (KD) allows knowledge transfer from a trained neural network (teacher) to a more compact one (student) in the absence of original training data. Existing works use a validation set to monitor the accuracy of the student over real data and report the highest performance throughout the entire process. However, validation data may not be available at distillation time either, making it infeasible to record the student snapshot that achieved the peak accuracy. Therefore, a practical data-free KD method should be robust and ideally provide monotonically increasing student accuracy during distillation. This is challenging because the student experiences knowledge degradation due to the distribution shift of the synthetic data. A straightforward approach to overcome this issue is to store and rehearse the generated samples periodically, which increases the memory footprint and creates privacy concerns. We propose to model the distribution of the previously observed synthetic samples with a generative network. In particular, we design a Variational Autoencoder (VAE) with a training objective that is customized to learn the synthetic data representations optimally. The student is rehearsed by the generative pseudo replay technique, with samples produced by the VAE. Hence knowledge degradation can be prevented without storing any samples. Experiments on image classification benchmarks show that our method optimizes the expected value of the distilled model accuracy while eliminating the large memory overhead incurred by the sample-storing methods.

Introduction

Recently there is a surging interest to deploy neural networks on the edge devices. Most of these devices have strict resource (e.g., memory or power) constraints that are incompatible with the high computational demand of the neural networks. Knowledge Distillation (KD) (Hinton, Vinyals, and Dean 2015) is a well-studied approach to deploy pre-trained neural networks on resource-constrained devices by making them more compact. KD needs access to the original dataset that was used to train the teacher model. This is challenging if the distillation is performed by a party other than the model developers as they may not have access to the original training dataset, either due to privacy issues or the extremely large dataset size making its relocation infeasible.

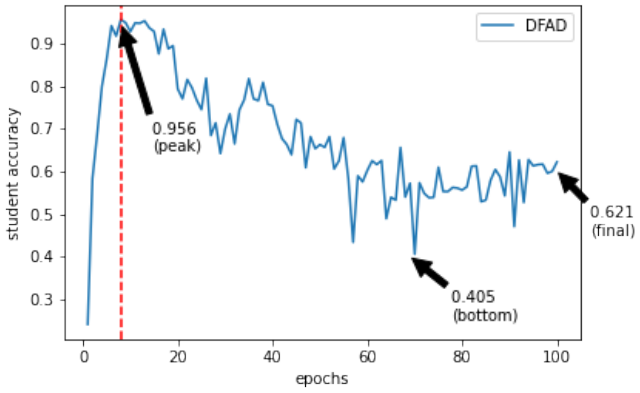

Figure 1: Example of student accuracy degradation over distillation steps due to catastrophic forgetting. Red vertical line marks the epoch with peak accuracy on the validation dataset.

Addressing this issue, some works utilize alternative data that are publicly available (Addepalli et al. 2020; Nayak, Mopuri, and Chakraborty 2021), while others propose to use synthetic data. (Zhang et al. 2021; Ma et al. 2020) The second approach is called Data-Free KD and completely eliminates the need for any real data. The decoupling of the KD from the original dataset allows distillation to be performed in a much wider spectrum of scenarios in comparison to the data-dependent approaches. Typically data-free KD approaches contain multiple rounds (epochs) of synthetic data samples generation and knowledge transfer. In each round, the synthetic samples are generated to close the current information gap between the teacher and the student. As the student is trained and the gap gets smaller, the synthetic sample distribution changes as well. Therefore, if the previously generated synthetic samples are not periodically rehearsed to the student, the information acquired in earlier distillation epochs might be lost, causing accuracy degradation over time. This is known as catastrophic forgetting (French 1999) Such accuracy degradation is not desirable in practice, especially if the validation dataset is also not available. The absence of the validation dataset prevents the

user from monitoring the student accuracy over time and choosing the distilled model with the peak accuracy. For example, Figure 1 shows the CIFAR10 validation accuracy of the student model distilled by an existing work, DFAD (Fang et al. 2019), over 100 epochs. The peak accuracy is achieved at the 8^{th} epoch and the accuracy degrades subsequently. As the users do not have access to the validation dataset, they cannot evaluate the student accuracy and find the peak. Hence, the final accuracy will be a function of the arbitrary termination epoch chosen by the user. This suggests that an ideal data-free KD method should be robust and either sustain high accuracy or monotonically increase it over time so that the distillation can be safely terminated after a pre-determined sufficient number of epochs. Recent research (Yin et al. 2020; Fang et al. 2021) propose to store the generated samples over epochs and continuously expose the student to the entire collection at each distillation step. However, with this approach, the memory overhead and the wall-clock time per distillation step increase substantially. If the dataset is complex in terms of the number of classes and the samples it contains, the required number of distillation steps to achieve an accurate student model can be quite large. For instance, for the Imagenet dataset, the DeepInversion method (Yin et al. 2020) stores 140K synthetic samples that occupy around 17 GB and takes 2.8K NVIDIA V100 GPU-hours to complete distillation. Moreover, storing the generated samples may violate the privacy considerations of the original dataset as they can leak information related to the original samples (Li and Zhang 2020). In summary, the current data-free KD methods are either not robust in terms of student accuracy throughout the distillation process or are resource inefficient by storing the generated samples and can have potential privacy concerns. This trade-off casts them less appealing to the end-users.

In this work, we aim to democratize data-free KD by proposing a novel method that preserves student accuracy by performing replay without storing any data samples. We achieve this by designing a Variational Auto-Encoder (VAE) (Kingma and Welling 2013) to model the cumulative distribution of all the generated samples. We establish that the vanilla VAE optimization objective is not suitable to model the synthetic data distribution and propose a synthetic data-aware reconstruction loss. The modeled distribution is used to infer "memory samples" that represent those generated in earlier steps and rehearse the student with these samples in conjunction with the newly generated samples to further bridge the information gap between the teacher and the student. Hence, our approach has a constant and very limited memory overhead of a few megabytes to store only the VAE parameters and offers accurate distillation. Experimental results show that our method achieves up to 26.8% increase in average student accuracy compared to methods without replay on image classification benchmarks. Moreover, compared to methods that replay stored samples, ours can reduce the memory footprint by gigabytes. Our concrete contributions can be summarized as follows.

- A novel data-free KD framework containing two specialized generators to allow the student network acquire new knowledge while retaining the prior information.
- Enabling VAEs to operate with synthetic data, produced for knowledge distillation, by modifying the reconstruction loss.
- Extensive experimental evaluation of our approach in comparison with the state-of-the-art.

Related Work

Knowledge Distillation (KD)

KD (Hinton, Vinyals, and Dean 2015) can be defined as transferring the learnt prediction behavior of a complex neural network model to a relatively smaller one. In literature, the complex network is often referred as the "teacher" and the compact network is referred as the "student". Throughout the distillation process, the student is trained with the guidance of the ground truth labels as well as the teacher's responses to the training data. The inclusion of teacher's responses in the training objective consolidates the information provided to the student about the data-label relationships. Thus the student network can achieve much higher accuracy than when it is trained with the supervision of only the ground truth labels. This is viewed as a form of compression, as the compact student model approximates the teacher by carrying more information than it could have learnt from data on its own.

Data-Free KD

The concept of performing knowledge distillation using synthetic data samples in place of a real dataset is called data-free KD. Unavailability of real data with ground truth labels limits the student training to be guided only by the teacher outputs (softmax logits, activation maps, etc.). One quality that the synthetic samples are expected to have is that their distribution should match that of the original data (Yoo et al. 2019; Nayak et al. 2019; Chen et al. 2019; Haroush et al. 2020). Another one is that they should be optimal to close the information gap between the teacher and the student (Micaelli and Storkey 2019; Fang et al. 2019). Producing samples that satisfy both, generally yields the best distillation results (Yin et al. 2020; Fang et al. 2021; Binici et al. 2021). During distillation, the synthetic data distribution is updated at each epoch to achieve the above-mentioned qualities. Such distribution shift could cause the student model to lose the information over epochs and experience accuracy degradation (Binici et al. 2021). This could be avoided by periodically rehearsing the student with samples from all the distributions it has previously observed. We can group the existing data-free KD methods based on whether they employ such practice or not as *replay-based methods* and *replay-free methods*.

The replay-based methods typically store a collection of generated synthetic samples over epochs. CMI (Fang et al. 2021) is a recent work that utilizes the model inversion technique to generate the synthetic samples and store them all in memory. Each time a newly generated batch is added, the student is trained with the entire collection of samples. This approach suffers from high memory utilization and requires a significant amount of time to achieve high-quality distillation. (Binici et al. 2021) proposed to use a fixed-sized

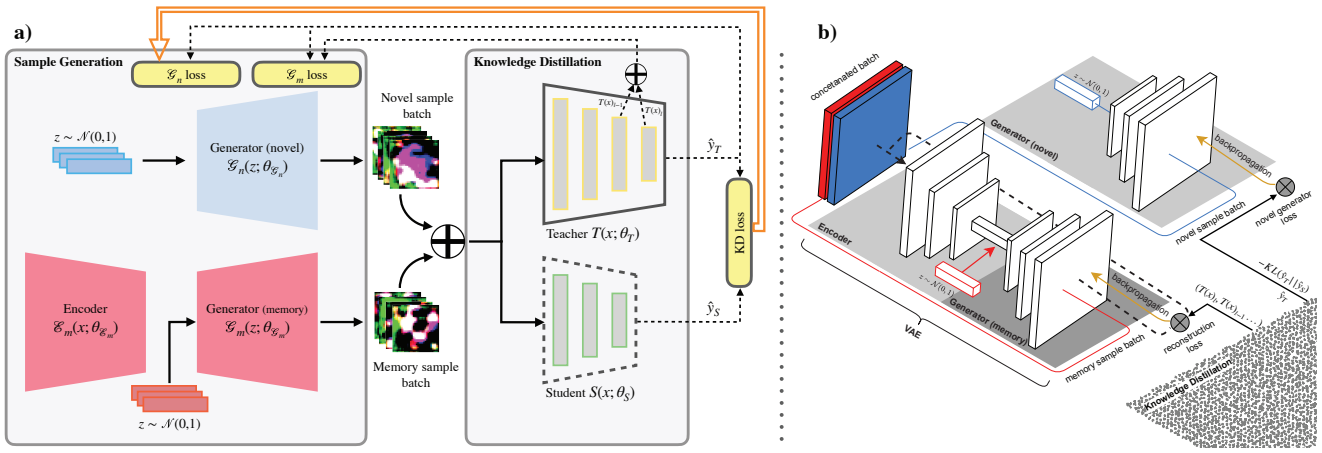

Figure 2: Method overview. (a) Proposed PRE-DFKD framework. The student and the generators are trained alternately. First the generators are fixed and the student is trained by a combination of novel and memory samples. In the next stage, student is fixed and the generators are trained by the learning signals received from the KD. (b) Training process of VAE. First, the memory generator is frozen and a batch of memory samples are inferred. Later the memory batch is combined with a novel sample batch to train the encoder-decoder pair.

memory buffer instead of storing all synthetic samples. Although the memory overhead is constant, this approach does not completely eliminate the possibility of catastrophic forgetting as the student is rehearsed only with a subset of previously observed samples.

The replay-free approaches train the student only with the newly generated samples at any distillation epoch. DAFL (Chen et al. 2019) introduced three novel loss terms that motivate the generated samples to be categorically diverse and be classified with high confidence by the teacher. DFAD (Fang et al. 2019) targeted generating samples that would cause maximum information gain to the student when learned. Since these works do not contain any replay mechanism, the student accuracy often fluctuates and degrades over epochs.

Replay in Continual Learning

Continual Learning (CL) methods focus on incrementally learning from a non-stationary dataset, with the goal of mitigating catastrophic forgetting (Goodfellow et al. 2015), i.e., the inability of the model to preserve the past knowledge while learning from the recent data. The CL methods (Mai et al. 2021) can be broadly classified into three major approaches: regularization-based, architecture-based, and replay-based (either generative or from a stored buffer), with the latter being the most effective and the focus of this paper. Vanilla replay-based techniques (Rebuffi et al. 2017) maintain a buffer of raw samples from the past tasks and replay them periodically with the new data in the learning process. However, these methods become infeasible when raw samples cannot be stored due to privacy or memory constraints. Moreover, they cannot perform lifelong learning due to the inability to scale well with the growing data stream. The generative replay strategies (Shin et al. 2017; Wu et al. 2018) mitigate these shortcomings by training a deep generative model to create pseudo-samples that mimic the data distribution from past experiences. Our work leverages generative pseudo replay. In particular, we adopt a VAE-based strategy (Shin et al. 2017) to preserve the knowledge of the previously generated synthetic samples. Unlike existing continual learning works, our method focuses on the generation of synthetic samples in a data-free environment. We propose a modified optimization objective to preserve the categorical information in reconstructed samples and mitigate catastrophic forgetting in a data-free knowledge distillation scenario.

Proposed PRE-DFKD Approach

Our data-free KD approach is called *Pseudo Replay Enhanced Data-Free Knowledge Distillation* (PRE-DFKD). Figure 2(a) provides an overview of PRE-DFKD. It consists of *data generation* and *knowledge distillation*. In data generation, randomly sampled latent variables are transformed by generative models to produce synthetic samples. Later during knowledge distillation, the student model is trained to categorize these samples similar to the teacher model. These two stages repeat alternately until the target number of steps is reached. We use two generative models in our framework. The first one (*novel sample generator)* produces samples that bring novel information to the student. The second generator (*memory sample generator*) is responsible for re-exposing the student to the information acquired earlier.

Novel Sample Generation

We define novel samples as those that the student classifies differently from the teacher. When the student is trained with these novel samples, it better approximates the teacher. We condition our novel sample generator by including the distance between student and teacher predictions in the optimization objective. To quantify such distance, we used Jensen-Shannon (JS) divergence (see Eq. 3) (Yin et al. 2020) as it is shown to provide better performing student models compared to other alternatives (e.g. Kullback–Leibler, L1 norm) (Binici et al. 2021). Moreover, to ensure that the novel sample distribution is similar to the original one, we also include the loss terms introduced by (Chen et al. 2019) (see Eq. 2). The first two of these are *predictive entropy* and *activation* loss terms, which are minimized when the synthetic data induce the teacher to output high valued activation maps and low entropy prediction vectors. The third term, *categorical entropy* loss, sustains class balance in the generated batches by maximizing the categorical entropy of synthetic sample distribution. Eq. 1 shows the complete optimization objective of the novel sample generator.

$$\theta_n^* := \arg\min_{\theta} \left(\mathcal{L}_{\phi}^{(T)} + \alpha \mathcal{L}_{\delta} \right) \tag{1}$$

$$\mathcal{L}_{\phi}^{(T)} = \frac{1}{n} \sum_{i} \left(\lambda_1 t_T^i log(\hat{y}_T^i) - \lambda_2 \left\| f_T^i \right\|_1 \right) - \lambda_3 \mathcal{H}(p(\hat{y}_T)) \tag{2}$$

$$\mathcal{L}_{\delta} = 1 - JS(\hat{y}_T \parallel \hat{y}_S); \quad z \sim \mathcal{N}(0,1) \tag{3}$$

where θ_n^*, represents the optimal values of the generator parameters denoted by θ_n. Moreover, $\hat{y}_T = \mathcal{T}(\theta(z))$ and f_T are the softmax outputs and the activation maps at the last FC layer of the teacher $\mathcal{T}$ for the generated batch $\theta(z)$ respectively, while $t_T^i = argmax(\hat{y}_T^i)$. $\mathcal{H}(p(\hat{y}_T))$ denotes the entropy of class label distribution in generated batch. $\lambda_i (i = 1, 2, 3)$ and α coefficients adjust the weighted contribution of each loss term.

Memory Sample Generation

As mentioned earlier, the distribution of novel samples shifts over distillation steps. To prevent such shift from causing the student to lose earlier learned information, we replay the samples from earlier distributions. These memory samples are inferred from a VAE that models the past synthetic data distributions. Our memory sample generator is the decoder part of the VAE and is trained jointly with the encoder. To train the pair, we use a combination of novel samples and memory samples. The inclusion of memory samples in the training process is to prevent the generator itself from experiencing catastrophic forgetting. The process is described in Algorithm 1 and visualized in Figure 2(b).

Incompatibility of Vanilla VAE loss for Synthetic Data Representation Learning After conducting experiments with vanilla VAE for pseudo replay in data-free KD, we observed that the teacher's predictions of the memory samples were of low confidence (14%). However, the average prediction confidence was significantly higher for novel samples (72%). This indicated that the modelled distribution was not accurate. We posited that the training loss of the vanilla VAE is incompatible with the properties of synthetic samples. The reconstruction term in the loss function ($\mathcal{L}_{rec} = \|x - D(E(x))\|_2^2$) is optimized when the decoded samples $D(E(x))$ match the inputs x in pixel space. If the L2 distance between the input and target image pixels is low, the loss term yields a small value. This assumes that small differences in pixel values do not impact the information content significantly. While this assumption holds for real images, it might not be accurate for synthetic ones. The synthetic samples are generated for the sole purpose of knowledge transfer. They are not visually realistic (see Figure 3) yet still achieve their goal. Therefore, the assumption that two images that are visually similar represent the same content may not hold for synthetic samples. To test this hypothesis, we inject random Gaussian noise to both real and synthetic samples and record the percentage of samples that preserved their class labels. The experimental details are given in the appendix. The effect of pixel-wise perturbations on class labels is visualized in Figure 3. It can be seen that the class information of synthetic samples got affected more significantly than real samples. This suggests that even if a reconstructed sample yields a small reconstruction loss with very similar pixel values to those of the synthetic input, its categorical information could be completely different.

Algorithm 1 Memory Sample Generator Training

INPUT: Novel sample generator $\mathcal{G}_n(z; \theta_{\mathcal{G}_n})$, memory sample generator $\mathcal{G}_m(z; \theta_{\mathcal{G}_m})$, encoder $\mathcal{E}_m(x; \theta_{\mathcal{E}_m})$, batch size b, latent vector dimension n.

Train with novel samples
sample B vectors ($z \sim \mathcal{N}(0,1)$)
$x_n \leftarrow \mathcal{G}_n(z)$
$\hat{z}_\mu, \hat{z}_\sigma \leftarrow \mathcal{E}_m(x_n)$
$z_n \sim \mathcal{N}(\hat{z}_\mu, \hat{z}_\sigma)$
$\hat{x}_n \leftarrow \mathcal{G}_m(z_n)$

Rehearse by reconstructing old samples
sample B vectors ($z_m \sim \mathcal{N}(0,1)$)
$x_m \leftarrow \mathcal{G}_m(z_m)$
$\bar{z}_\mu, \bar{z}_\sigma \leftarrow \mathcal{E}_m(x_m)$
$\bar{z}_m \sim \mathcal{N}(\bar{z}_\mu, \bar{z}_\sigma)$
$\hat{x}_m \leftarrow \mathcal{G}_m(\bar{z}_m)$

Calculate & backpropagate loss
$\mathcal{L}_{VAE} \leftarrow \mathcal{L}_{rec}(x_m^{(i)}, \hat{x}_m^{(i)}, x_n^{(i)}, \hat{x}_n^{(i)}) + \gamma \mathcal{L}_{KLD}(\bar{z}_\mu, \bar{z}_\sigma)$
$\theta_{\mathcal{G}_m}, \theta_{\mathcal{E}_m} \leftarrow optimizer.step(backward(\mathcal{L}_{VAE}), \theta_{\mathcal{G}_m}, \theta_{\mathcal{E}_m})$

Synthetic Data-Aware VAE Reconstruction Loss To ensure that the categorical information is preserved, we modify the VAE reconstruction loss. Our synthetic data-aware loss constrains the reconstructed samples to cause similar feature responses in the last layers of the teacher. Typically, these layers are considered to represent content-related information in Convolutional Neural Networks (CNNs) (Jing et al. 2019). To impose the mentioned constraint, we add the term in Eq. 4 to the VAE optimization objective.

$$\sum_{l \in L} |T(x)_l - T(D(E(x)))_l|_1 \tag{4}$$

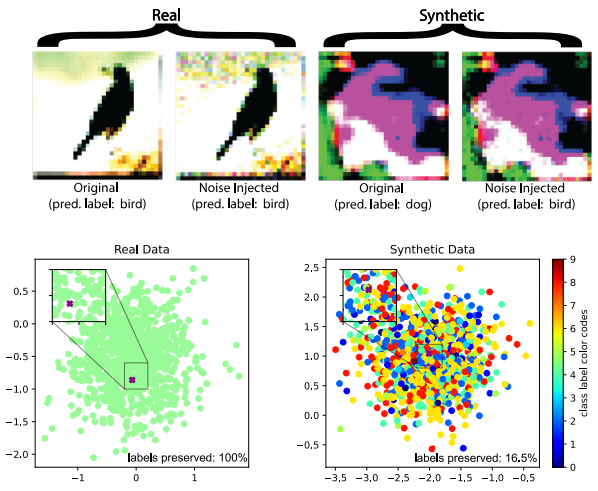

Figure 3: Example real and synthetic CIFAR10 samples are given before and after noise injection in the first row. In the second row, the projection of real and synthetic samples injected with noise on 2-dimensional planes are shown. Colors represent the class labels assigned to these samples by the teacher model. The original samples are marked with purple crosses.

The encoder and decoder networks are represented by D and E respectively. Moreover, the set L denotes the selected set of teacher network (T) layers. The total training loss we use can be denoted as,

$$\mathcal{L}_{VAE} = \mathcal{L}_{rec} + \mathcal{L}_{KLD} \tag{5}$$

$$\mathcal{L}_{rec} = |x - D(E(x))|_1 + \sum_{l \in L} |T(x)_l - T(D(E(x)))_l|_1 \tag{6}$$

$$\mathcal{L}_{KLD} = (\mathcal{N}(\mu_z, \sigma_z) \parallel \mathcal{N}(0,1)) \tag{7}$$

where x is the input sample batch; μ_z and σ_z are the mean and variance of the latent vector ($z = E(x)$) distribution.

Sustaining Class Balance in Inferred Memory Batches Once we accomplish training a VAE with synthetic data, another remaining issue is to ensure that the inferred memory samples are evenly distributed into classes. Typically, latent vectors that are sampled from standard Gaussian distribution are fed to the generator to produce new images. However, our task requires the generated batch to be diverse in terms of the categorical contents of the samples. This is to guarantee that the student learns to approximate the teacher for samples from any category. Therefore, to provide such class balance, we freeze the memory sample generator and tune the input latent vectors.

$$z_m^* := \arg \max_{z \sim \mathcal{N}(0,1)} (p(\hat{y}_N) log(p(\hat{y}_N))) \tag{8}$$

The procedure we used to solve the optimization objective described in Eq. 8 is given in algorithm 2. Tuning the input latent variables can deviate their distribution from standard normal, which would cause the generator to produce noise. To prevent such deviation, we use the Kullback–Leibler (KL) divergence as a regularization term. Moreover, to sample the most representative images (within each class) from the modeled distribution, we also include the predictive entropy loss in the regularization term ($\mathcal{R}$).

Algorithm 2 Memory Sample Generator Inference

INPUT: memory sample generator $\mathcal{G}_m(z; \theta_{\mathcal{G}_m})$, teacher $T(x; \theta_T)$.
OUTPUT: memory batch x_m.

Tune latent variables
sample B vectors ($z_m \sim \mathcal{N}(0,1)$)
$x_m \leftarrow \mathcal{G}_m(z_m)$
$\hat{y}_\mathcal{T} \leftarrow \mathcal{T}(x_m)$
$\mathcal{R} \leftarrow \frac{1}{n} \sum_i \left(t_T^i log(\hat{y}_T^i)\right) + KL(\mathcal{N}(\mu_z, \sigma_z) \parallel \mathcal{N}(0,1))$
$\mathcal{L}_z \leftarrow -\mathcal{H}(p(\hat{y}_\mathcal{T})) + \mathcal{R}$
$z_m \leftarrow optimizer.step(backward(\mathcal{L}_z), z)$

Infer memory samples
$x_m \leftarrow \mathcal{G}_m(z_m)$

Knowledge Distillation

Knowledge transfer happens by minimizing the distance between teacher and student predictions against combined batches of novel and memory samples. The minimization problem that results in optimal student model parameters is

$$\theta_S^* := \arg\min_{\theta_S} \|S((x_n, x_m); \theta_S) - T((x_n, x_m); \theta_T)\|_1 \tag{9}$$

θ_S^* in Eq. 9 denotes the optimal student model parameters; x_n and x_m are novel and memory samples, respectively.

Experimental Evaluation

We demonstrate the effectiveness of PRE-DFKD in improving the expected student accuracy and reducing resource utilization on several image classification benchmarks. We provide a comparison with methods that store samples for replay and those that do not employ replay. Our baselines from the first category are CMI (Fang et al. 2021), and the memory bank approach (Binici et al. 2021) that we refer to as MB-DFKD. For the baselines that do not use replay, we select DAFL (Chen et al. 2019) and DFAD (Fang et al. 2019) methods. For a fair comparison, we used the implementations of CMI, DAFL, and DFAD available from the authors' GitHub pages.

Datasets: We use four datasets with different complexities and sizes. The simplest is MNIST (LeCun et al. 1998) that contains 32×32 grayscale images from ten classes. CIFAR10 (Krizhevsky, Hinton et al. 2009) contains RGB images from ten classes ($3 \times 32 \times 32$). CIFAR100 (Krizhevsky, Hinton et al. 2009) contains hundred different categories while the samples have the same dimensions as those in CIFAR10. Lastly, the most complex dataset we use is Tiny ImageNet (Deng et al. 2009) that contains 64×64 RGB samples from 200 classes.

	MNIST $\mathcal{T}$:LeNet5 (98.9%) $\mathcal{S}$:LeNet-half			CIFAR10 $\mathcal{T}$:ResNet34 (95.4%) $\mathcal{S}$:ResNet18			CIFAR100 $\mathcal{T}$:ResNet34 (77.9%) $\mathcal{S}$:ResNet18			Tiny ImageNet $\mathcal{T}$:ResNet34 (71.2%) $\mathcal{S}$:ResNet18		
Method	μ	σ^2	acc_{max}	μ	σ^2	acc_{max}	μ	σ^2	acc_{max}	μ	σ^2	acc_{max}
Train with data	98.7	0.5	98.9	89.0	8.1	95.2	71.3	8.1	77.1	60.2	8.8	64.9
DAFL	87.3	6.6	98.2	62.6	17.1	92.0	52.5	12.8	74.5	39.5	**10.3**	52.2
DFAD	63.5	6.8	98.3	86.1	12.3	93.3	54.9	12.9	67.7	-	-	-
CMI	memory: N.A.			memory: 250 MB			memory: 500MB			memory: 2.7 GB		
	-	-	-	82.4	16.6	**94.8**	55.2	24.1	77.0	-	-	-
MB-DFKD	memory: 6.7 MB			memory: 20 MB			memory: 20 MB			memory: 20MB		
	88.6	3.2	98.3	83.3	16.4	92.4	64.4	18.3	75.4	45.7	11.5	53.5
PRE-DFKD (ours)	memory: 2.1 MB			memory: 2.1 MB			memory: 2.1 MB			memory: 2.1 MB		
	90.3	**1.9**	**98.3**	**87.4**	**10.3**	94.1	**70.2**	**11.1**	**77.1**	**46.3**	11.0	**54.2**

Table 1: Student accuracy results for Data-Free KD on four image classification benchmarks. The T and S values denote the teacher-student pairs. The values for μ and σ^2 represent mean and variance $(\%)$ of the averaged student validation accuracy (over 4 runs), over epochs. acc_{max} is the maximum recorded accuracy at any epoch of any distillation run.

Implementation Details We run each method for 200, 200, 400, and 500 epochs for MNIST, CIFAR10, CIFAR100, and Tiny ImageNet, respectively. To evaluate each dataset and method pair, we conduct four runs. For MNIST, we select LeNet5 (LeCun et al. 1998) and LeNet5-half as the teacher-student pair. For the remaining datasets, we use ResNet34 (He et al. 2016) as the teacher and ResNet18 as the student. While benchmarking the baseline methods, we use the hyper-parameter configurations from the papers or GitHub pages where available. Additional experimental details are given in the appendix. We note that CMI failed for MNIST and Tiny ImageNet datasets. The original implementation required the teacher models to contain Batch Normalization layers and as LeNet does not contain any, we could not test it on MNIST. For Tiny ImageNet, although we searched extensively for proper hyper-parameter configuration, none of our trials yielded comparable results. Similarly, all our experiments with DFAD on Tiny ImageNet were unsuccessful.

Evaluation Metrics: If we consider the termination step of the distillation as a random variable $ts \in \mathbb{R}$, then our goal is

$$\max_{ts} \mathbb{E}[acc_{ts}] - \sigma^2[acc_{ts}] \tag{10}$$

where acc_{ts} stands for student accuracy at epoch ts. To report results, first, we average the acc_{ts} across all runs at the corresponding epoch (ts). These averaged accuracy series can be denoted as $\mathbb{E}_i[acc_{ts}^{(i)}]$ where $acc_{ts}^{(i)}$ is the student accuracy observed in the i^{th} run at epoch ts. We report the mean and variance values of $\mathbb{E}_i[acc_{ts}^{(i)}]$. The mean ($\mu$) corresponds to $\mathbb{E}_{ts}[\mathbb{E}_i[acc_{ts}^{(i)}]]$ and the variance (σ^2) corresponds to $\sigma^2[\mathbb{E}_i[acc_{ts}^{(i)}]]$. Additionally, we include the peak accuracy recorded across all runs (acc_{max}) to demonstrate that our method achieves state-of-the-art performance when the validation set is available.

Improved Distillation Quality: Table 1 provides a summary of the results across four runs. The accuracies of the pre-trained teachers are given next to $\mathcal{T}$ in Table 1 followed by the student accuracy with different methods. The first row *Train with data* shows the student accuracy when trained with the original dataset. The remaining rows report the accuracy for different baseline data-free KD methods and our approach (PRE-DFKD). As expected, the *Train with data* student accuracy is lower than the teacher accuracy and the data-free methods have lower student accuracy than *Train with data* student accuracy. PRE-DFKD achieves student accuracy that is better than almost all the baseline data-free KD approaches and approximates the student accuracy trained with the original dataset closer for all datasets. When compared with baselines that do not contain replay (DAFL and DFAD), PRE-DFKD is more robust for validation set-agnostic distillation with higher expectation and lower variance. When compared with CMI and MB-DFKD that store generated samples for replay, PRE-DFKD typically is at least similar and more often better. It might be counter-intuitive that although CMI stores and replays all generated samples, PRE-DFKD outperforms it. This is because the accuracy gain rate of the student models distilled via CMI is relatively low (see Figure 4) causing lower expected value. We believe the gain rate is low because the novel samples constitute only a small proportion when compared to the replay samples. Lastly, MB-DFKD performing lower than PRE-DFKD suggests that limited amount of stored samples in MB-DFKD is not as effective as modeling the entire distribution of earlier observed samples in PRE-DFKD.

Furthermore, to visualize the effect of pseudo replay on student accuracy throughout distillation, we plot the student accuracy curves (see Figure 4) for the MNIST and CIFAR100 benchmarks. For all methods, the observed accuracy series ($acc_{ts}^{(i)}$) across different runs on MNIST benchmark, had relatively higher variation than on other benchmarks. Therefore, for MNIST, we plot the accuracy curves with mean and variances across runs are indicated. For CIFAR100, we only display the mean accuracy curve (μ) across four runs for each baseline.

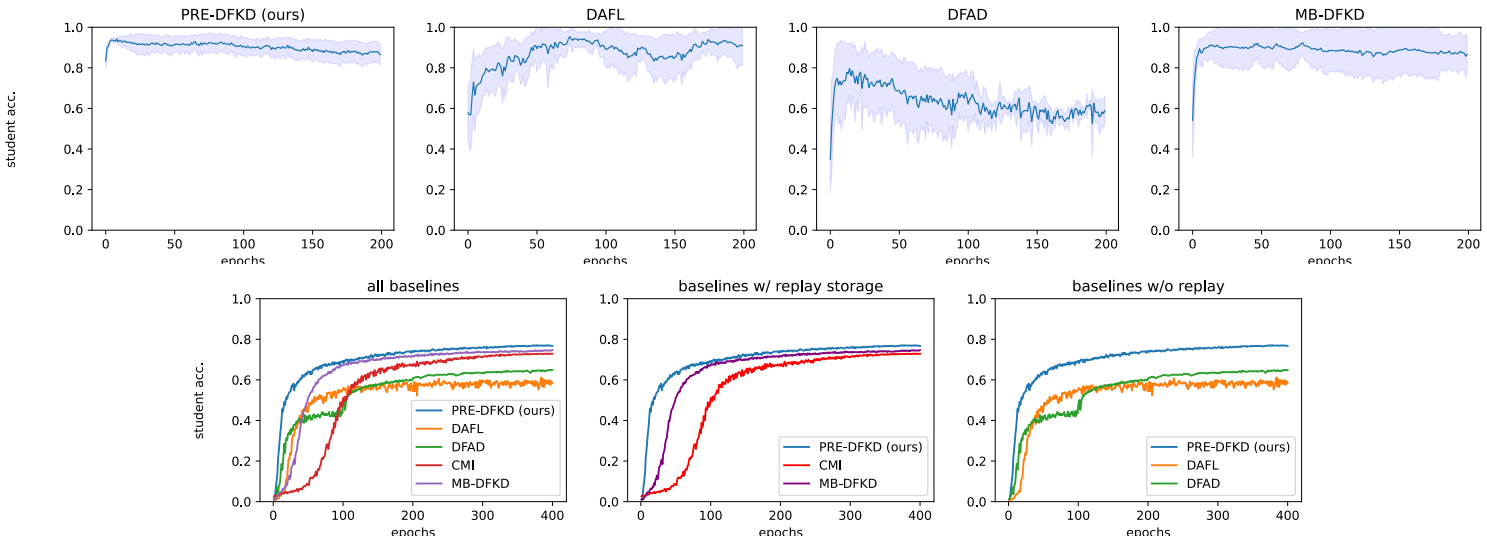

Figure 4: Visualized student accuracy curves. The first row contains MNIST results with mean and variance values over runs. The second row contains student accuracies averaged over runs for CIFAR100.

Reduced Memory Footprint: We next compare PRE-DFKD with baselines using replay in terms of memory footprint. The additional memory required for the replay are noted in Table 1. Although CMI failed in Tiny ImageNet experiments, we report the memory it utilized after running for the same number of epochs (500) as other methods. The Table 1 shows that PRE-DFKD significantly reduces the memory footprint for replay from hundreds of megabytes and even gigabytes to a constant 2MB, that is the memory occupied by the VAE parameters. Such reduction becomes more significant compared to CMI as the number of distillation steps increases since more batches of samples are generated and stored. The calculated memory footprint of CMI for different choices of distillation steps ts and synthetic data dimensions are marked in Table 2. As the VAE architecture is independent of the selection of teacher-student pairs and distillation steps, the memory overhead does not scale based on them. Although MB-DFKD maintains fixed-sized storage, the required storage for optimal distillation largely depends on the dataset. If the storage is small relative to the number of distillation steps, the effectiveness of replay diminishes.

pixel dim. \ ts	2000	4000	8000	16000
32	2.5 GB	5 GB	10 GB	20 GB
64	10 GB	20 GB	40 GB	80 GB
128	40 GB	80 GB	160 GB	320 GB

Table 2: CMI memory footprint for various pixel dimensions and distillation epochs

Ablation Study: We now validate that the pseudo replay is effective regardless of the novel sample synthesis strategy. To do this, we couple our memory generator to the DFAD baseline method. In Figure 5, DFAD distilled student accuracy is plotted with and without the inclusion of our memory generator. We name the VAE-enhanced baseline PRE-DFAD. For a fair comparison, we fix the random seeds for all the runs. The plots indicate that pseudo replay succeeded in preventing accuracy degradation caused by catastrophic forgetting even for DFAD. Additionally, we plot the PRE-DFKD results where the proposed class-balanced memory sample inference and synthetic data-aware reconstruction loss are omitted. This way, we examine the individual contributions of these techniques. In Figure 5, it is apparent that the exclusion of any of these impaired the student accuracy progressions.

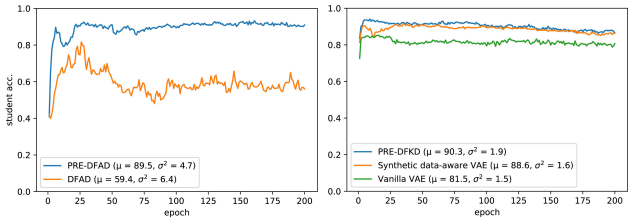

Figure 5: Ablation experiment results on MNIST.

Conclusion

In this work, we propose a data-free KD method that prevents the distilled model accuracy from degrading over time by generative pseudo replay. While our method is more robust than existing approaches in the absence of validation data, it also significantly reduces the memory footprint and improves privacy preservation. We believe these contributions provide a step towards allowing a wider audience to benefit from data-free KD, i.e., democratize it. In the future, our approach can be further improved by exploring ways to reduce the impact of dataset-specific hyper-parameter selection on distillation performance.

Acknowledgement

This work was supported by the A*STAR Computational Resource Centre through the use of its high-performance computing facilities and by the National Research Foundation, Singapore under its Competitive Research Programme Award NRF-CRP23-2019-0003.

References

Addepalli, S.; Nayak, G. K.; Chakraborty, A.; and Radhakrishnan, V. B. 2020. DeGAN: Data-Enriching gan for retrieving representative samples from a trained classifier. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3130–3137.

Binici, K.; Pham, N. T.; Mitra, T.; and Leman, K. 2021. Preventing Catastrophic Forgetting and Distribution Mismatch in Knowledge Distillation via Synthetic Data. *arXiv preprint arXiv:2108.05698* .

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *ICCV*, 3514–3522.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Fang, G.; Song, J.; Shen, C.; Wang, X.; Chen, D.; and Song, M. 2019. Data-Free Adversarial Distillation. *arXiv preprint arXiv:1912.11006* .

Fang, G.; Song, J.; Wang, X.; Shen, C.; Wang, X.; and Song, M. 2021. Contrastive Model Inversion for Data-Free Knowledge Distillation. *arXiv preprint arXiv:2105.08584* .

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* 3(4): 128–135.

Goodfellow, I. J.; Mirza, M.; Xiao, D.; Courville, A.; and Bengio, Y. 2015. An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks.

Haroush, M.; Hubara, I.; Hoffer, E.; and Soudry, D. 2020. The knowledge within: Methods for data-free model compression. In *CVPR*, 8494–8502.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Jing, Y.; Yang, Y.; Feng, Z.; Ye, J.; Yu, Y.; and Song, M. 2019. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics* 26(11): 3365–3385.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* .

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images .

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11): 2278–2324.

Li, Z.; and Zhang, Y. 2020. Membership Leakage in Label-Only Exposures. *arXiv preprint arXiv:2007.15528* .

Ma, X.; Shen, Y.; Fang, G.; Chen, C.; Jia, C.; and Lu, W. 2020. Adversarial Self-Supervised Data-Free Distillation for Text Classification. *arXiv preprint arXiv:2010.04883* .

Mai, Z.; Li, R.; Jeong, J.; Quispe, D.; Kim, H.; and Sanner, S. 2021. Online Continual Learning in Image Classification: An Empirical Survey.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot knowledge transfer via adversarial belief matching. In *NIPS*, 9551–9561.

Nayak, G. K.; Mopuri, K. R.; and Chakraborty, A. 2021. Effectiveness of arbitrary transfer sets for data-free knowledge distillation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1430–1438.

Nayak, G. K.; Mopuri, K. R.; Shaj, V.; Radhakrishnan, V. B.; and Chakraborty, A. 2019. Zero-shot knowledge distillation in deep networks. In *International Conference on Machine Learning*, 4743–4751. PMLR.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 5533–5542.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual Learning with Deep Generative Replay.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; Zhang, Z.; and Fu, Y. 2018. Incremental Classifier Learning with Generative Adversarial Networks. *CoRR* abs/1802.00853. URL http://arxiv.org/abs/1802.00853.

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via DeepInversion. In *CVPR*, 8715–8724.

Yoo, J.; Cho, M.; Kim, T.; and Kang, U. 2019. Knowledge extraction with no observable data. In *NIPS*, 2705–2714.

Zhang, Y.; Chen, H.; Chen, X.; Deng, Y.; Xu, C.; and Wang, Y. 2021. Data-Free Knowledge Distillation for Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7852–7861.

Appendix

Additional Implementation Details

This section contains additional details related to the experimental setup. For all datasets,

- Both generators are updated with Adam optimizer.
- Input vectors of the memory generator are tuned with learning rate same as lr_{G^n}.
- λ_1, λ_2, λ_3 and α coefficients are set as 1, 5, 0.1 and 1 respectively
- PyTorch framework is used.
- Experiments are done on Linux version 4.18.0-147.el8.x86_64

MNIST: In MNIST experiments, we set the learning rates of the student (lr_S) and the novel sample generator (lr_{G^n}) as 0.004 and 0.2. We use Adam optimizer to update the student. The learning rate for the memory sample generator (lr_{G^m}) is $\frac{1}{40}$ times lr_{G^n}. We set the novel and memory sample batch sizes as 512 and 256 respectively. We use 100-dimensional latent vectors as inputs to the generators. The whole distillation process of 200 epochs took 1.3 hours to complete on 2 NVIDIA V100 GPUs.

CIFAR10: In CIFAR10 experiments, we set (lr_S) and lr_{G^n} as 0.1 and 0.02 respectively. We use SGD optimizer to update the student. lr_{G^m} is $\frac{1}{10}$ times lr_{G^n}. We set the novel and memory sample batch sizes as 1024 and 128 respectively. We used 1000-dimensional latent vectors as inputs to the generators. The whole distillation process of 200 epochs took 11.6 hours to complete on 2 NVIDIA V100 GPUs.

CIFAR100: In CIFAR100 experiments, we set (lr_S) and lr_{G^n} as 0.1 and 0.02 respectively. We use SGD optimizer to update the student. lr_{G^m} is $\frac{1}{10}$ times lr_{G^n}. We set the novel and memory sample batch sizes as 1024 and 128 respectively. We used 1000-dimensional latent vectors as inputs to the generators. The whole distillation process of 400 epochs took 24 hours to complete on 2 NVIDIA V100 GPUs.

Tiny ImageNet: In Tiny ImageNet experiments, we set (lr_S) and lr_{G^n} as 0.1 and 0.02 respectively. We use SGD optimizer to update the student. lr_{G^m} is $\frac{1}{10}$ times lr_{G^n}. We set the novel and memory sample batch sizes as 1500 and 187 respectively. We used 2000-dimensional latent vectors as inputs to the generators. The whole distillation process of 500 epochs took 29 hours to complete on 2 NVIDIA V100 GPUs.

Synthetic Sample Class Label Sensitivity Against Pixel-Space Perturbations

The reconstructed outputs of a VAE could be viewed as $X_s + \epsilon$ for the source image X_s. To measure the degree that the introduced error (ϵ) affects the content information of synthetic samples, we simulate the reconstruction by injecting noise to synthetic novel samples and observe the class labels that the teacher assigns to these noisy versions. Here we assume that ϵ comes from Gaussian distribution and sample the injected noise from the Guassian distribution with 0 mean and 0.2 variance ($\mathcal{N}(0, 0.2)$). We sample one image from the real CIFAR10 training set and one synthetic image from a generator that is optimized to replicate CIFAR10 images. Later we sample 1000 ϵ values from $\mathcal{N}(0, 0.2)$ and perturb both images with them. This can be viewed as producing samples from the neighborhoods of the original images in pixel space. All 1000 of the noisy images obtained from the original real image had the same class label. Whereas, the original class label was preserved for only 165 images out of 1000 obtained from the synthetic sample. This suggests that the content information of synthetic samples gets significantly more affected by the reconstruction error than the real samples.