Adversarial Variational Knowledge Distillation *

Xuan Tang[1] and Tong Lin(✉)[1,2]

[1] The Key Laboratory of Machine Perception (MOE), School of EECS, Peking University
{xuantang,lintong}@pku.edu.cn
[2] Peng Cheng Laboratory, Shenzhen, China

Abstract. Knowledge Distillation (KD) is one of the most popular and effective techniques for model compression and knowledge transfer. However, most existing KD approaches are heavily relying on the labeled training data, which is usually unavailable due to privacy concerns. Thus, data-free KD focus on restoring the training data with Generative Adversarial Networks (GANs) by either catering the pre-trained teacher or fooling the student. In this paper we introduce Adversarial Variational Knowledge Distillation (AVKD), a framework that formulates the restoring process as Variational Autoencoders (VAEs). Different from vanilla VAEs, AVKD is specified by a pre-trained teacher model $p(y|x)$ of the visible labels y given the latent x, a prior $p(x)$ over the latent variables and an approximate generative model $q(x|y)$. In practice, we refer the prior $p(x)$ as an alternate unlabeled data distribution from other related domains. Similar to Adversarial Variational Bayes (AVB), we estimate the KL-divergence term between $p(x)$ and $q(x|y)$ by introducing a discriminator network. Although the original training data are unavailable, we argue that the prior data drawn from other related domains can be easily obtained to learn the knowledge distillation efficiently. Extensive experiments testify that our method outperforms the state-of-the-art algorithms in the absence of the original training data, with performance approaching the case where the original training data are provided.

Keywords: Data-free Knowledge Distillation · Variational Autoencoders · Generative Adversarial Networks

1 Introduction

Knowledge Distillation (KD) [8] is a machine learning approach that transfers knowledge from a larger capacity teacher model (or ensembles) into a more compact student model. Given a pre-trained teacher model, a student aims to learn knowledge from the teacher on the training data. Through distillation, one hopes to obtain a student model that not only inherits better performance from the teacher, but is also more efficient in the inference stage. In recent years, the

* This work was supported by NSFC Tianyuan Fund for Mathematics (No. 12026606), and National Key RD Program of China (No. 2018AAA0100300).

knowledge distillation community has made great achievements with respect to model architecture and several application domains [19–21, 23].

In spite of the significant progress, classical distillation methods heavily rely on sufficient training data, which is often unavailable due to privacy, confidentiality, property, size or transience. Hence, it is necessary to explore data-free knowledge distillation algorithms which are independent of the original training data.

Most existing data-free approaches concentrate on modeling the data distribution via GANs but are insufficient in theoretical derivations. In this work, we formulate the data generation process from the perspective of Variational Autoencoders (VAEs) [11].

Contrary to vanilla VAEs, we regard the input feature (e.g. images) as the latent variables instead of visible variables while the output labels are viewed as observations (i.e. visible variables). Here, we denote x as the feature and y as the labels, respectively. As a result, we first propose a principled framework named **Variational Knowledge Distillation** (VKD), which is composed of a classification model $p(y|x)$ of the visible variables y given the latent variables x, a prior $p(x)$ over the latent variables and an approximate generative model $q(x|y)$. Specifically, the classification model $p(y|x)$ is implemented with a well-trained teacher model $p_\theta(y|x)$ with fixed parameters θ. Taking the prior $p(x)$ as Gaussian (e.g. $\mathcal{N}(0, \mathbf{I})$) will cause poor performance in experiments since the generative model $q(x|y)$ can only produce ill-conditioned images that are harmful for distillation.

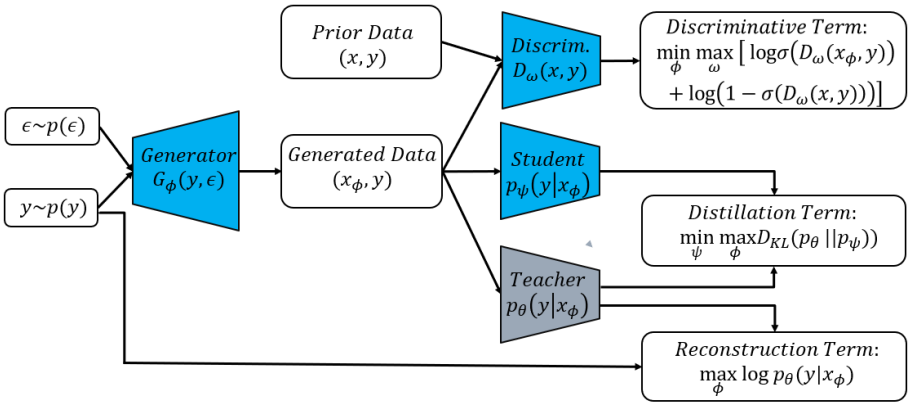

Fig. 1. Overview of our proposed Adversarial Variational Knowledge Distillation (AVKD) framework.

To address this problem, we employ some unlabeled data that can be easily obtained from other related domains as the prior. However, computing the Kullback–Leibler divergence term between $q(x|y)$ and $p(x)$ is intractable since

the unlabeled data have no explicit analytic expressions. To this end, we borrow the idea from Adversarial Variational Bayes (AVB) [15] and propose a refined framework called **Adversarial Variational Knowledge Distillation** (AVKD), which can estimate the Kullback–Leibler divergence based on a new discriminator network. The purpose of the discriminator is to determine if an example (x, y) is drawn from $p(y)p(x)$ or from $p(y)q(x|y)$. Here $p(y)$ denotes the real distribution of labels which are categories in a typical classification task. The AVKD framework is shown in Figure 1.

Experiments in section 4 on various image classification datasets shows that the student networks trained with our framework outperform existing state-of-the-art methods in the absence of the original training data, with performance approaching the settings where the original training data are provided.

2 Related Works

2.1 Knowledge Distillation

Knowledge Distillation is a popular model compression technique that transfers the knowledge of a large-capacity model or an ensemble of models into a small one. Buciluǎ et al. [3] first proposed the idea of training a network with another network's outputs on a large-scale unlabeled dataset. Ba et al. [2] later trained shallow neural networks to mimic deep neural networks via regressing logits (outputs of a neural network before the softmax activation) with mean square error (MSE). Hinton et al. [8] further trained student neural networks with "soft targets" produced by the teacher and first proposed the concept of *Knowledge Distillation*.

2.2 Data-free Knowledge Distillation

The data-free knowledge distillation, i.e. optimizing the student model without the original training data, becomes more challenging than vanilla knowledge distillation. Most existing approaches focus on synthetic images generation. Lopes et al. [13] first utilized the "meta-data" (e.g. means and standard deviation of activations from each layer) stored the teacher networks to generate fake samples that can produce similar activations. Following-up works [17, 22] intended to use less "meta-data" or design different training objectives. A similar strategy [5, 7, 14] is to exploit the means and variances statistics stored in the batch normalization layers [9] of neural networks. However, these "meta-data" or batch normalization statistics are not always provided by the teacher networks, indicating that related approaches would failed sometimes. Another strategy [1, 4] proposed to train a generator network for data generation, which enables the teacher network to produce predictions with high confidence, significantly relying on large batch size and training steps to produce large amount of diverse images. Micaelli et al. [16] developed a new generation scheme via adversarial learning. Specifically, a generator is employed to produce samples that maximize

the knowledge distillation loss [8] between the teacher and student networks. Similarly, Fang et al. [6] replaced the knowledge distillation loss with mean average error (MAE) and achieved better performance. These two methods aim to search hard samples for the student throughout the whole input space, which are easily to fall into a suboptimal solution due to the high dimensions of the searching space.

3 Method

3.1 Problem Formulation

Knowledge Distillation In typical knowledge distillation form of classification tasks, given a pre-trained teacher model $p_\theta(y|x)$ parameterized by θ, a student model $p_\psi(y|x)$ parameterized by ψ aims to solve the following problem:

$$\min_{\psi} E_{p_{data}(x,y)} \left[D_{KL}(p_\theta(y|x) \parallel p_\psi(y|x)) \right], \tag{3.1}$$

where D_{KL} refers to the Kullback–Leibler divergence that evaluates the discrepancy between the distributions produced by the teacher and student models. Here $p_{data}(x, y)$ denotes joint distribution of the original training samples and labels. Note that the D_{KL} term is independent with labels y, so we can reformulate problem (3.1) as

$$\min_{\psi} E_{p_{data}(x)} \left[D_{KL}(p_\theta(y|x) \parallel p_\psi(y|x)) \right]. \tag{3.2}$$

Data-free Knowledge Distillation Optimization (3.1) can be also rewritten as

$$\min_{\psi} E_{p_{data}(y)}[E_{p_{data}(x|y)}[D_{KL}(p_\theta(y|x) \parallel p_\psi(y|x)))]], \tag{3.3}$$

where $p_{data}(y)$ is a categorical distribution in classification tasks. However, optimizing either (3.2) or (3.3) requires the knowledge of the data distribution $p_{data}(x)$ or $p_{data}(x|y)$, which is unavailable in the data-free setting. Most existing data-free approaches concentrate on modeling the distribution $p_{data}(x)$ or $p_{data}(x|y)$ directly via GANs without any derivations.

In this work, we focus on approximating $p_{data}(x|y)$ with generative model $q_\phi(x|y)$ parameterized by ϕ. Then, the data-free knowledge distillation problem can be reformulated as

$$\min_{\psi} E_{p_{data}(y)}[E_{q_\phi(x|y)}[D_{KL}(p_\theta(y|x) \parallel p_\psi(y|x)))]]. \tag{3.4}$$

Thus in the rest of this paper, we concentrate on modeling and learning the generative model $q_\phi(x|y)$.

3.2 Variational Knowledge Distillation (VKD)

Given a pre-trained teacher $p_\theta(y|x)$, our goal is to approximate the true generative model $p_\theta(x|y)$ via a approximate parametric model $q_\phi(x|y)$. In this work, we consider x as the latent variables of features (such as images) and y as the visible variables of labels. As a result, following VAEs, it can be shown as

$$\log p_\theta(y) \geq -D_{KL}(q_\phi(x|y) \parallel p(x)) + E_{q_\phi(x|y)}\left[\log p_\theta(y|x)\right], \tag{3.5}$$

where $p(x)$ denotes the prior distribution over the latent variables. The right hand side of (3.5) is called the **evidence lower bound** (ELBO).

When performing maximum-likelihood training, the goal of VAEs is to optimize the marginal log-likelihood

$$E_{p_{data}(y)}\left[\log p_\theta(y)\right]. \tag{3.6}$$

However, computing $\log p_\theta(y)$ requires marginalizing out x in $p_\theta(y, x)$ which is usually intractable. Instead, VAEs use inequality (3.5) to rephrase the intractable problem of optimizing (3.6) into optimizing the ELBO:

$$\max_\theta \max_\phi E_{p_{data}(y)}\left[-D_{KL}(q_\phi(x|y) \parallel p(x)) + E_{q_\phi(x|y)}\left[\log p_\theta(y|x)\right]\right]. \tag{3.7}$$

In the knowledge distillation setting, the teacher model $p_\theta(y|x)$ has been trained well on the original data distribution $p_{data}(x, y)$. Consequently, we fix the weight θ of teacher while optimizing (3.7), then the training objective becomes

$$\max_\phi E_{p_{data}(y)}[-D_{KL}(q_\phi(x|y) \parallel p(x)) + E_{q_\phi(x|y)}\left[\log p_\theta(y|x)\right]]. \tag{3.8}$$

Note that the term $D_{KL}(q_\phi(x|y) \parallel p(x))$ is an expectation on $q_\phi(x|y)$ according to its definition, then we can rewrite the optimization problem in (3.8) as

$$\max_\phi E_{p_{data}(y)}[E_{q_\phi(x|y)}[\log p(x) - \log q_\phi(x|y) + \log p_\theta(y|x)]]. \tag{3.9}$$

When we have an explicit representation of $q_\phi(x|y)$ and $p(x)$ such as Gaussian, (3.9) can be optimized using the reparameterization trick [11, 18] and Stochastic Gradient Descent (SGD).

3.3 Adversarial Variational Knowledge Distillation (AVKD)

One significant drawback of VKD, however, is that samples drawn from the $q_\phi(x|y)$ are ill-formed since it is almost impossible to find a perfect explicit expression for $p(x)$ to model real data such as images. Thus, the performance provided by the student model trained with $q_\phi(x|y)$ might be very poor. To this end, we replace the prior $p(x)$ with large amount unlabeled data in real scenarios from other related domains. For ease of description, we term the unlabeled data as *prior data* in this work. Using the *prior data* the student model can produce much better performance and experiments details are described in section 4.

However, it is intractable to optimize (3.9) when only given *prior data* sampled from an implicit prior $p(x)$. Following the Adversarial Variational Bayes (AVB) [15], we therefore introduce an auxiliary discriminative network to rephrase the maximum-likelihood-problem as a two-player game, as described in the following.

The main idea of AVB is to implicitly representing the term in (3.9)

$$\log p(x) - \log q_\phi(x|y) \tag{3.10}$$

as the optimal value of an additional real-valued discriminator network $D_\omega(x, y)$ parameterized by ω.

Specifically, when given $q_\phi(x|y)$, the objective of the discriminator is

$$\max_\omega E_{p_{data}(y)}[E_{q_\phi(x|y)} \log \sigma(D_\omega(x, y)) + E_{p(x)} \log(1 - \sigma(D_\omega(x, y)))], \tag{3.11}$$

where $\sigma(t) := 1/(1 + e^{-t})$ is the sigmoid function. As shown in literature [15], it turns out that the optimal is given as

$$D^*(x, y) = \log q_\phi(x|y) - \log p(x). \tag{3.12}$$

Here, $D^*(x, y)$ denotes the function that maximizes (3.11) and the right hand side of (3.12) is the negative of (3.10). The detail proof can be found in [15].

We can rewrite the objective in (3.8) as

$$\max_\phi E_{p_{data}(y)}[E_{q_\phi(x|y)}[-D^*(x, y) + \log p_\theta(y|x)]]. \tag{3.13}$$

Optimizing (3.13) requires to calculate the gradient w.r.t ϕ, which is difficult since we have defined $D^*(x, y)$ as the solution of problem (3.11) that itself depends on ϕ. Fortunately, the literature [15] has proved that

$$E_{q_\phi(x|y)}[\nabla_\phi D^*(x, y)] = 0, \tag{3.14}$$

which indicates that it is unnecessary to take the gradient w.r.t the explicit occurrence of ϕ in $D^*(x, y)$.

Adversarial training Inspired by the idea of [6, 16], we introduce an auxiliary objective for the generative model $q_\phi(x|y)$ that maximizes the discrepancy between the teacher and student model, resulting in a two-player game between the generative model and the student model. With adversarial training, the goal of $q_\phi(x|y)$ changes to maximize objective in (3.13) and adversarial objective jointly:

$$\begin{aligned} \max_\phi E_{p_{data}(y)}[E_{q_\phi(x|y)}[&- D^*(x, y) + \log p_\theta(y|x) \\ &+ D_{KL}(p_\theta(y|x) \parallel p_\psi(y|x))]]. \end{aligned} \tag{3.15}$$

The objective of the generator consists of three terms:

- The $-D^*(x, y)$ is the discriminative term that allows the generator to produce similar samples to *prior data* by fooling the discriminator $D(x, y)$.
- The second term $\log p_\theta(y|x)$ is called the reconstruction log-likelihood for the input labels y.
- The adversarial term $D_{KL}(p_\theta(y|x) \parallel p_\psi(y|x))$ encourages the generator to generate hard training samples for the student.

Training objectives Using the reparameterization trick [11, 18], we can rewrite the objective of the generative model $q_\phi(x|y)$ in (3.15) as

$$\begin{aligned}\max_{\phi} E_{p_{data}(y)}[E_{p(\epsilon)}[&- D^*(G_\phi(y,\epsilon),y) + \log p_\theta(y|G_\phi(y,\epsilon)) \\ &+ D_{KL}(p_\theta(y|G_\phi(y,\epsilon)) \parallel p_\psi(y|G_\phi(y,\epsilon)))]],\end{aligned} \tag{3.16}$$

where $p(\epsilon)$ is a Gaussian and $G_\phi(y,\epsilon)$ is a learnable generator network parameterized by ϕ. Similarly, the objective of the discriminator model in (3.11) and the objective of the student model in (3.4) can be rewritten in the form

$$\begin{aligned}\max_{\omega} E_{p_{data}(y)}[E_{p(\epsilon)} &\log \sigma(D_\omega(G_\phi(y,\epsilon),y)) \\ &+ E_{p(x)}\log(1-\sigma(D_\omega(x,y)))]\end{aligned} \tag{3.17}$$

and

$$\min_{\psi} E_{p_{data}(y)}[E_{p(\epsilon)}[D_{KL}(p_\theta(y|G_\phi(y,\epsilon)) \parallel p_\psi(y|G_\phi(y,\epsilon)))]], \tag{3.18}$$

respectively.

Note that applying SGD on the objective of (3.16) requires keep $D^*(x,y)$ optimal which might be very time-consuming. Therefore, we treat the optimization problems in (3.16) and (3.17) as a two-player game following AVB [15]. An overview of our proposed AVKD is shown in Figure 1.

3.4 Algorithm

In practice, we applying SGD jointly to (3.16), (3.18) and (3.17), see Algorithm 1. Here, η and m denote the learning rate and batch size, respectively. Note that we apply n SGD-updates to the student model at each iteration to distill the teacher model more efficiently.

4 Experiments

4.1 Experiments Setup

CIFAR The CIFAR10 dataset [12] consists of 50K training and 10K testing RGB images with resolution 32×32 of 10 categories. For the CIFAR10 classification task, we use the CIFAR100 [12] and Tiny-ImageNet datasets as the *prior data* respectively. Since the CIFAR100 dataset contains two coarse classes (i.e. vehicles1 and vehicles2) related with two fine-grained classes (i.e. automobile and truck) of CIFAR10, we therefore remove the 10 fine-grained classes of vehicles1 and vehicles2 in CIFAR100, resulting in a modified dataset termed as CIFAR90. Thus, we train the AVKD with various *prior data* such as CIFAR90, CIFAR100 and Tiny-ImageNet of 32×32 image size, respectively.

Similar to CIFAR10, CIFAR100 [12] consists of 50K training and 10K testing RGB images except that all images are distributed over 100 categories. In this experiment, the CIFAR10 and Tiny-ImageNet of 32×32 image size are employed as *prior data* respectively.

Algorithm 1: Adversarial Variational Knowledge Distillation (AVKD)

for *1,2,...,N* **do**
  Sample $\{x^{(1)}, ..., x^{(m)}\}$ from *prior data*
  Sample $\{y^{(1)}, ..., y^{(m)}\}$ from $p_{data}(y)$
  Sample $\{\epsilon^{(1)}, ..., \epsilon^{(m)}\}$ from $\mathcal{N}(0, \mathbf{I})$

  Compute ϕ-gradient for generator (eq. 3.16):
  $g_\phi \leftarrow \frac{1}{m} \sum_{k=1}^{k=m} \nabla_\phi [-D^*(G_\phi(y^{(k)}, \epsilon^{(k)}), y^{(k)}) + \log p_\theta(y^{(k)}|G_\phi(y^{(k)}, \epsilon^{(k)}))$
    $+ D_{KL}(p_\theta(y^{(k)}|G_\phi(y^{(k)}, \epsilon^{(k)})) \parallel p_\psi(y^{(k)}|G_\phi(y^{(k)}, \epsilon^{(k)})))]$
  Update parameters of generator:
  $\phi \leftarrow \phi + \eta \times g_\phi$

  Compute ω-gradient for discriminator (eq. 3.17):
  $g_\omega \leftarrow \frac{1}{m} \sum_{k=1}^{k=m} \nabla_\omega [\log \sigma(D_\omega(G_\phi(y^{(k)}, \epsilon^{(k)}), y^{(k)}))$
    $+ \log(1 - \sigma(D_\omega(x^{(k)}, y^{(k)})))]$
  Update parameters of discriminator:
  $\omega \leftarrow \omega + \eta \times g_\omega$

  for *1,2,...,n* **do**
    Sample $\{y^{(1)}, ..., y^{(m)}\}$ from $p_{data}(y)$
    Sample $\{\epsilon^{(1)}, ..., \epsilon^{(m)}\}$ from $\mathcal{N}(0, \mathbf{I})$
    Compute ψ-gradient for student (eq. 3.18):
    $g_\psi \leftarrow \frac{1}{m} \sum_{k=1}^{k=m} \nabla_\psi [D_{KL}(p_\theta(y|G_\phi(y^{(k)}, \epsilon^{(k)})) \parallel p_\psi(y|G_\phi(y^{(k)}, \epsilon^{(k)})))]$
    Update parameters of student:
    $\psi \leftarrow \psi - \eta \times g_\psi$
  end
end

Tiny-ImageNet The Tiny-ImageNet dataset is a modified subset of the original ImageNet dataset. Here, there are 200 different classes instead of 1000 classes of ImageNet, with 100K training examples and 10K validation examples. The resolution of the images is resized to 64×64 pixels, which is different from the ImageNet. Since the Tiny-ImageNet also contains 10K testing images without labels, we can only report the test accuracies on the validation images. We explore two target resolutions: 32×32 matching that of CIFAR10 and CIFAR100, and full resolution 64×64. For 32×32 image size we therefore choose training sets of CIFAR10 and CIFAR100 as the *prior data* respectively. For resolution of 64×64, we take the Caltech101[1] and Caltech256[2] as the *prior data*, respectively.

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/
[2] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

4.2 Implementation details

To verify the robustness of our method, we conduct all experiments with same hyperparameters. Note that the $p_{data}(y)$ is set to discrete uniform distribution since we suppose every dataset is of category balance. We train our proposed method with 100K iterations and batch size of 256 for all experiments, and we update the student network with 5 times at each iteration so that it can learn more sufficiently from the current generative model. To make a fair comparison, we follow the setting used in the previous literature [4, 6] with a pre-trained ResNet34 as the teacher network and ResNet18 as the student. The generator architecture is modified from the one used in [4, 6] and the discriminator is constructed with three convolutional layers. We optimize the generator network with Adam [10] with an initial learning rate of 10^{-3} that is divided by 10 at the 20K-th and 40K-th iteration respectively. The student and discriminator are trained by the Nesterov Accelerated Gradient (NAG) optimizer with momentum 0.9 and weight decay 5×10^{-4}. The discriminator is trained with constant learning rate of 2×10^{-4}. The initial learning rate of the student is 0.1 and decayed by 0.985 every 200 iterations. For each experiment we run it three times and report the mean accuracy.

4.3 Results

CIFAR10 Table 1 summarized test accuracies of student models trained by different methods on CIFAR10 dataset. Trained in fully supervised setting, the teacher (ResNet34) and the student networks (ResNet18) achieve accuraies of 95.53% and 93.92%, respectively. The student network trained with knowledge distillation [8] using the original training data (CIFAR10) achieves a +0.38% improvement over the one trained from scratch. In the data-free setting, using Gaussian noise as inputs results in poor performance which is only slightly better than a random guess (around 10%). For CIFAR10, we employ CIFAR90, CIFAR100 and TinyImageNet 32×32 as the *prior data*, respectively. Training with CIFAR100 achieves the highest accuracy of **93.50%** since CIFAR100 is more diverse than CIFAR90 and more revalant to CIFAR10 than TinyImageNet.

We also compare our methods with DAFL [4] and DFAD [6] using their released codes with batch size of 256. Note that our method is only slightly better than other data-free algorithms, as one reason is that the 10-category classification task with resolution 32×32 is too simple to differentiate these methods.

CIFAR100 Results on CIFAR100 obtained by different methods are also listed in Table 1. It can be found that our proposed method outperforms others with considerable improvements better than the case in CIFAR10 experiments. Specifically, while training with CIFAR10 as *prior data*, our method exceeds the DAFL [4], DFAD [6] and DeGAN [1] with **10.53%**, **4.18%** and **6.63%**, respectively. While training with TinyImagenet 32×32, our method also outperforms other state-of-the-art methods with considerable improvements. Note that the

Table 1. Test accuracies on CIFAR10 and CIFAR100. In our experiments, we employ the ResNet34 as the teacher and ResNet18 as the student model, respectively.

		CIFAR10		CIFAR100	
Model	Method	Prior Data	Accuracy	Prior Data	Accuracy
ResNet34	Supervised Training	N/A	95.53%	N/A	77.58%
ResNet18	Supervised Training	N/A	93.92%	N/A	76.51%
ResNet18	KD[8]	N/A	94.30%	N/A	76.89%
ResNet18	Gaussian Noise	N/A	11.43%	N/A	1.23%
ResNet18	DAFL[4]	N/A	88.41%	N/A	61.35%
ResNet18	DFAD[6]	N/A	93.30%	N/A	67.70%
ResNet18	DeGAN[1]	N/A	N/A	CIFAR90	65.25%
ResNet18	Ours	CIFAR90	**93.41%**	CIFAR10	**71.88%**
		CIFAR100	**93.50%**	TinyImageNet	**70.26%**
		TinyImageNet	93.02%	N/A	N/A

DeGAN also takes CIFAR90 as an alternative of the original training data but achieves lower test accuracy than ours.

Tiny-ImageNet For Tiny-ImageNet classification task, we conduct experiments on two target resolutions of 32×32 and 64×64. In order to compare our method with DAFL [4] and DFAD [6], we modified their released codes to run it on Tiny-ImageNet with the two different resolutions and all hyperparameters remain unchanged. Note that we don't report the results of DeGAN [1] since the authors didn't release their codes.

The results of Tiny-ImageNet 32×32 can be found in table 2. Our method achieves accuracies of **46.96%** and **50.22%** with CIFAR10 and CIFAR100 as the *prior data* respectively, outperforming all other data-free approaches by a large margin. We can also observe that the proposed method trained with more diverse dataset (i.e. CIFAR100) gains higher accuracy than CIFAR10 with improvement of **+3.26%** that is considerable for a 200-category classification task.

For results of Tiny-ImageNet 64×64, as shown in table 2, our proposed method exceeds DFAD [6] with better improvements than the case of 32×32 resolution.

Visualization Results We plot the reconstructed images of CIFAR10 by our proposed AVKD trained with TinyImagenet in Figure 2. Images of same category are plotted in one row, where images on the left side are sampled from the true dataset while on another side are the generated images. It can be found that the generative model in the proposed AVKD can learn the key features of different categories from the pretrained teacher and the *prior data*, hence these synthesized samples can transfer relevant knowledge about the original training data. These results, therefore, testify the effectiveness our proposed method from another perspective.

Table 2. Test accuracies on TinyImageNet of two resolutions. We employ ResNet34 as the teacher and ResNet18 as the student model, respectively.

		TinyImageNet32 × 32		TinyImageNet64 × 64	
Model	Method	Prior Data	Accuracy	Prior Data	Accuracy
ResNet34	Supervised Training	N/A	57.68%	N/A	61.49%
ResNet18	Supervised Training	N/A	54.30%	N/A	58.30%
ResNet18	KD[8]	N/A	55.06%	N/A	58.98%
ResNet18	Random Noise	N/A	0.57%	N/A	0.54%
ResNet18	DAFL[4]	N/A	26.32%	N/A	13.24%
ResNet18	DFAD[6]	N/A	29.52%	N/A	15.92%
ResNet18	Ours	CIFAR10	**46.96%**	Caltech101	**46.11%**
		CIFAR100	**50.22%**	Caltech256	**46.25%**

Fig. 2. Visualization of generated images. Images on the left side are sampled from the CIFAR10 dataset while on the right side are generated by our proposed AVKD.

5 Conclusion

In this work, we introduce the Adversarial Variational Knowledge Distillation (AVKD), a framework that can distil a well-trained large-capacity teacher model into a compact student model in the absence of original training data on which the teacher are trained. Since the original training data is unavailable, we treat the data (i.e. images in this work) as latent variables and learn the generative model $q_\phi(x|y)$ to model the original training data. By employing the unlabeld prior data, experiments have shown that our method outperforms other data-free KD algorithms on various images classification tasks. Furthermore, we found that our method can exceed other methods with larger margin for more difficult tasks, indicating the effectiveness of the proposed method.

References

1. Addepalli, S., Nayak, G.K., Chakraborty, A., Babu, R.V.: Degan: Data-enriching gan for retrieving representative samples from a trained classifier (2019)
2. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: NeurIPS (2014)
3. Buciluǎ, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: SIGKDD (2006)
4. Chen, H., Wang, Y., Xu, C., Yang, Z., Liu, C., Shi, B., Xu, C., Xu, C., Tian, Q.: Data-free learning of student networks. In: ICCV (2019)
5. Choi, Y., Choi, J., El-Khamy, M., Lee, J.: Data-free network quantization with adversarial knowledge distillation. arXiv preprint arXiv:2005.04136 (2020)
6. Fang, G., Song, J., Shen, C., Wang, X., Chen, D., Song, M.: Data-free adversarial distillation. arXiv preprint arXiv:1912.11006 (2019)
7. Haroush, M., Hubara, I., Hoffer, E., Soudry, D.: The knowledge within: Methods for data-free model compression. In: CVPR (2020)
8. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
9. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
10. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
11. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2014)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012)
13. Lopes, R.G., Fenu, S., Starner, T.: Data-free knowledge distillation for deep neural networks. arXiv preprint arXiv:1710.07535 (2017)
14. Luo, L., Sandler, M., Lin, Z., Zhmoginov, A., Howard, A.: Large-scale generative data-free distillation. arXiv preprint arXiv:2012.05578 (2020)
15. Mescheder, L., Nowozin, S., Geiger, A.: Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. arXiv preprint arXiv:1701.04722 (2018)
16. Micaelli, P., Storkey, A.J.: Zero-shot knowledge transfer via adversarial belief matching. In: NeurIPS (2019)
17. Nayak, G.K., Mopuri, K.R., Shaj, V., Babu, R.V., Chakraborty, A.: Zero-shot knowledge distillation in deep networks. arXiv preprint arXiv:1905.08114 (2019)

Adversarial Variational Knowledge Distillation 13

18. Rezende, D.J., Mohamed, S., Wierstra, D.: Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082 (2014)
19. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
20. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: CVPR (2019)
21. Wang, X., Zhang, R., Sun, Y., Qi, J.: Kdgan: Knowledge distillation with generative adversarial networks. In: NeurIPS (2018)
22. Yin, H., Molchanov, P., Li, Z., Alvarez, J.M., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. arXiv preprint arXiv:1912.08795 (2020)
23. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: ICLR (2017)