Aligning Logits Generatively for Principled Black-Box Knowledge Distillation

Jing Ma*, Xiang Xiang*†
School of Artificial Intelligence and Automation,
Huazhong University of Science and Tech, Wuhan, China

Ke Wang, Yuchuan Wu, Yongbin Li
Alibaba Damo Academy
Hangzhou, China

Abstract

Black-Box Knowledge Distillation (B2KD) is a formulated problem for cloud-to-edge model compression with invisible data and models hosted on the server. B2KD faces challenges such as limited Internet exchange and edge-cloud disparity of data distributions. In this paper, we formalize a two-step workflow consisting of deprivatization and distillation, and theoretically provide a new optimization direction from logits to cell boundary different from direct logits alignment. With its guidance, we propose a new method Mapping-Emulation KD (MEKD) that distills a black-box cumbersome model into a lightweight one. Our method does not differentiate between treating soft or hard responses, and consists of: 1) deprivatization: emulating the inverse mapping of the teacher function with a generator, and 2) distillation: aligning low-dimensional logits of the teacher and student models by reducing the distance of high-dimensional image points. For different teacher-student pairs, our method yields inspiring distillation performance on various benchmarks, and outperforms the previous state-of-the-art approaches.

1. Introduction

Knowledge Distillation (KD) is a widely accepted approach to the problem of model compression and acceleration, which has received sustained attention from both the academic and industrial research communities [14, 16, 35, 44]. The goal of KD is to extract knowledge from a cumbersome model or an ensemble of models, known as the teacher, and use it as supervision to guide the training of lightweight models, known as the student [1, 5, 36]. In the application of KD, privacy protection has always been a very concerning issue for researchers and users, which not only refers to the privacy of user data but also includes the model copyright of cloud service providers.

Black-Box Knowledge Distillation (B2KD) is a problem posed in the process of cloud-to-edge model compression [34, 46, 49]. The cloud server hosts a teacher model whose internal structure and composition, connections between layers, model parameters, and gradients used for back-propagation are all invisible and unavailable to edge devices, as shown in Fig. 1. Due to resource limitations, the edge device can only host a lightweight student model. At the same time, low-quality and unlabeled local data cannot be used to train a reliable deep neural network. As a result, it must rely on sending query samples to the APIs of cloud servers for heavy inference [45].

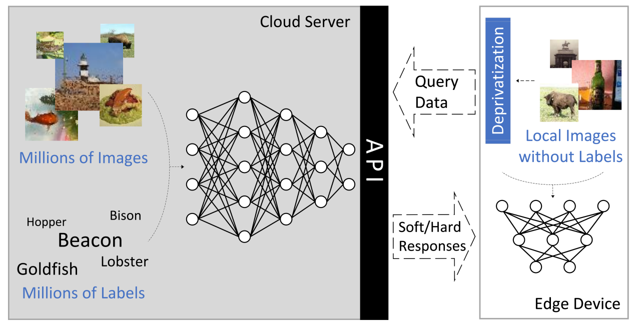


Figure 1. Schematic process of cloud-to-edge model compression. A cumbersome black-box model is deployed on a cloud server, trained with millions of samples and tags. The cloud server only provides APIs to receive query data and return inference responses of either soft or hard type. The edge device needs to distill a lightweight model using unlabeled local data.

In practice, B2KD faces some key challenges. (a) Cloud servers and edge devices should maintain limited data exchange due to Internet latency and bandwidth constraints, as well as charges for the amount of queried data or API usage time. (b) In some cases, for query samples, these APIs only provide indexes or semantic tags for the category with the highest probability (*i.e.*, hard responses), rather than probability vectors for all possible classes (*i.e.*, soft responses). (c) Because users refuse to send sensitive data to cloud servers, the distribution gap between local and cloud data is difficult to measure, making the distilled student model inaccurate in the application.

Adversarial learning has been shown to be effective in generating pseudo samples, which is widely used in data augmentation and low-shot learning [8, 49]. A well-trained

*Equal contribution, co-first author; also with Nat. Key Lab of MSIIPT.
†Correspondence to xex@hust.edu.cn; also with Peng Cheng Lab.

generator can overcome the mode collapse problem and align real and synthetic data distribution. In particular, we want to produce images relevant to training, whether or not they resemble real data [30]. Meanwhile, images generated to obtain high responses from the teacher model combine different patterns with highly generalized features instead of sample-specific idiosyncrasies [47]. Therefore, using a well-trained generator to synthesize pseudo images can automatically filter out privacy-related high-frequency information, this process is called **deprivatization**,

In this paper, we propose an approach to solve B2KD by *mapping emulation*. Our motivation is in accordance with the fact that it can drive alignment between low-dimensional logits by reducing the distance between two generated images in the high-dimensional space. In addition, we argue that an image contains a lot of fine-grained information, which can be treated as another type of knowledge to provide different gradient directions for updating the parameters of student model, as shown in Fig. 4. Combining image-level loss with coarse-grained logit-level loss can effectively improve the distillation effect. According to the Kolmogorov theorem [6, 24], a sufficiently complex neural network is capable of representing an arbitrary multivariate continuous function from any dimension to another. Thus, a well-trained generator can not only emulate the inverse mapping of the teacher function (Thm. 1) but also help update the logits of a student to converge to the logits of a teacher (Thm. 2), with reasonable generalizability (Thm. 3).

In practice, we derive using a generative adversarial network (GAN) for deprivatization and exploit it as an inverse mapping of the teacher function. The generator uses random variables as inputs that are sampled from a prior distribution with the same dimensionality as the logits. The well-trained generator is frozen and grafted behind the teacher and student model, whose output logits of the same examples are used as the inputs of the generator, as shown in Fig. 2. Experimental results show that MEKD can effectively protect the privacy of local data and models in the cloud, and it performs well under either soft or hard responses. At the same time, MEKD has robust results in the case of limited query samples and out-of-domain data.

Overall, the contributions of this paper are: **1)** We formalize the problem of B2KD and provide a two-step workflow of deprivatization and distillation. **2)** We theoretically provide a new optimization direction from logits to cell boundary different from direct logits alignment. **3)** We propose a new method of Mapping-Emulation Knowledge Distillation (MEKD). The improved experimental performance has demonstrated the effectiveness of our approach.

2. Related Work

Knowledge Distillation (KD). Hinton *et al.* [18] propose an original teacher-student architecture that uses the logits of the teacher model as the knowledge. Since then, some KD methods regard knowledge as final responses to input samples [3, 29, 53], some regard knowledge as features extracted from different layers of neural networks [22, 23, 38], and some regard knowledge as relations between such layers [9, 37, 52]. The purpose of defining different types of knowledge is to efficiently extract the underlying representation learned by the teacher model from the large-scale data. If we consider a network as a mapping function of input distribution to output, then different knowledge types help to approximate such a function. Based on the type of knowledge transferred, KD can be divided into response-based, feature-based, and relation-based [14]. The first two aim to derive the student to mimic the responses of the output layer or the feature maps of the hidden layers of the teacher, and the last approach uses the relationships between the teacher's different layers to guide the training of the student model. Feature-based and relation-based methods [23, 52], depending on the model utilized, may leak the information of structures and parameters through the intermediate layers' data. For example, we can reconstruct a ResNet [17] based on the feature dimensions of different layers, and calculate each neuron's parameter using specific images and their responses in the feature maps.

Black-Box Knowledge Distillation (B2KD). Response-based KD methods [3, 18, 53] have the natural property of hiding models. Hinton *et al.* [18] use Kullback-Leibler Divergence (KLD) between the softened logits of teacher and student models as the loss to align the output distribution, and Zhao *et al.* [53] decouple the KLD into two uncorrelated losses and combine them by weighted summation. These calculations do not take into account the details of the teacher model, which is exactly a black box. The recently proposed approaches for B2KD also address the issue of hiding the teacher model deployed in the cloud server [34, 46, 49]. Orekondy *et al.* [34] use a reinforcement learning approach to improve query sample efficiency. Wang *et al.* [46] blend mixup and activate learning to augment the few unlabeled images and choose hard examples for distillation. And Wang [49] proposes a decision-based black-box model and constructs the soft label for each training sample by computing its distances to the decision boundaries of the teacher model. These existing approaches partially address the challenges of cloud-to-edge black-box model distillation, but none of them take into account the privacy leak of user data when sending original local images to the cloud.

Generative Adversarial Networks (GANs) have the capacity to handle sharp estimated density functions and generate realistic-looking images efficiently. A typical GAN [13] comprises a discriminator distinguishing real images and generated images, and a generator synthesizing images to fool the discriminator. GANs are divided into architecture-variant and loss-variant. The former focuses on

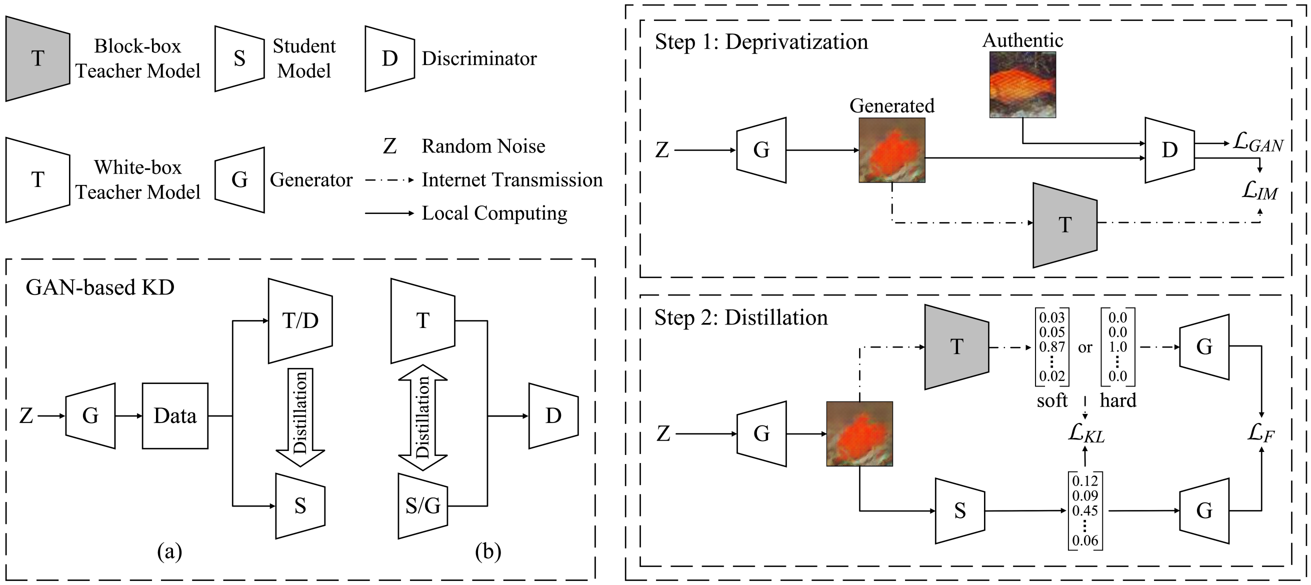


Figure 2. The overall framework of MEKD. Lower left: two architectures of GAN-based KD. Upper right: the process of deprivatization. GAN is used to synthetic high-response images to the teacher model within the distribution of data in edge devices. Lower right: the process of distillation with the frozen generator. The synthetic privacy-free images are query samples sent to the teacher model through the APIs of cloud servers. The student model is distilled by reducing the logit-level and image-level discrepancy.

network architectures [7, 39] or latent space [10, 31], *e.g.*, some specific architectures are proposed for specific tasks [21, 54]. The latter utilizes different loss types and regularization tools [2, 15] to enable more stable learning.

Adversarial Distillation (AD) exploits adversarial architecture to help the teacher and student model have a better understanding of the real data distribution [4, 14, 41, 48, 50]. The methods of AD can be divided into two types according to the generator-discriminator architecture, as shown in Fig. 2: (a) the generator is used to synthetic images to obey a real distribution, and these images are used to help distill models [8, 48]; (b) the teacher and student models are regarded as generators and another discriminator is drafted behind them to judge whether the distribution of features or logits is consistent [50, 51]. AD is also employed for low-shot knowledge distillation and received inspiring results [8]. Our method provides an alternative adversarial architecture, which utilizes a well-trained generator to guide the alignment between the outputs of models.

3. Theory for Mapping-Emulation KD

First, we propose two definitions. Def. 1 defines that two functions that map the same data distribution μ to the same latent distribution υ are equivalent. The ideal state of KD is to obtain a student function f_S that is equivalent to the teacher function f_T. Def. 2 defines that the mapping function of a generator G, which can map a prior distribution p to data manifold Σ and guarantee that the generated image distribution μ' is the same as the real image distribution μ, is considered to be the inverse mapping of the teacher function, *i.e.* $f_G = f_T^{-1}$. And we call it a *well-trained* generator. The mapping relationships are shown in Fig. 3.

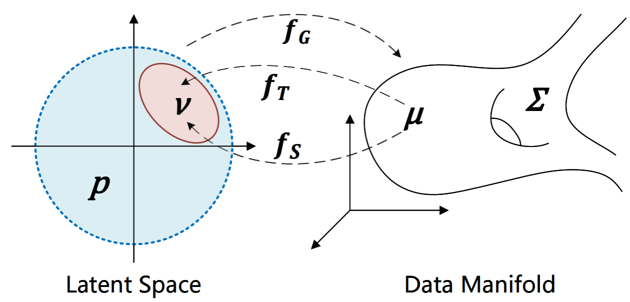


Figure 3. Mapping relationships of f_S, f_T, f_G. If f_S and f_T can map μ to the same distribution υ, then $f_S = f_T$, and if f_G can map the prior distribution p to μ, then $f_G = f_T^{-1}$.

Definition 1. (Function Equivalence) *Giving the student and teacher model f_S and f_T, for a data distribution $\mu \in \mathcal{X}$ in image space which is mapped to $\mathbb{P}_S \in \mathcal{Y}$ and $\mathbb{P}_T \in \mathcal{Y}$ in latent space. If the Wasserstein distance between $\mathbb{P}_S$ and $\mathbb{P}_T$ equals zero,*

$$W(\mathbb{P}_S, \mathbb{P}_T) = \inf_{\gamma \in \Pi(\mathbb{P}_S, \mathbb{P}_T)} \mathbb{E}_{(y_S, y_T)\sim\gamma}[\,\|y_S - y_T\|\,] = 0, \tag{1}$$

the student and teacher model are equivalent, i.e., $f_S = f_T$*, where $\Pi(\mathbb{P}_S, \mathbb{P}_T)$ is the set of all joint distributions $\gamma(y_S, y_T)$ whose marginals are $\mathbb{P}_S$ and $\mathbb{P}_T$, respectively.*

Definition 2. (Inverse Mapping) *Giving a prior distribution $p \in \mathbb{R}^C$, for a data distribution $\mu \in \mathbb{R}^n$, if the Wasserstein distance between generated distribution $\mu' = (f_G)_{\#}p$ and μ equals zero,*

$$W(\mu', \mu) = \inf_{\gamma \in \Pi(\mu', \mu)} \mathbb{E}_{(x', x)\sim\gamma}[\,\|x' - x\|\,] = 0, \tag{2}$$

then the generator $f_G : \mathbb{R}^C \to \mathbb{R}^n$ is the inverse mapping of the teacher function $f_T : \mathbb{R}^n \to \mathbb{R}^C$, denoted as $f_G = f_T^{-1}$, where $\Pi(\mu', \mu)$ is the set of all joint distributions $\gamma(x', x)$ whose marginals are respectively μ' and μ.

Fixing a decoding map f_G for a well-trained generator G, the latent space $\mathcal{Z}$ is partitioned as

$$\mathcal{D}(f_G) : \mathcal{Z} = \bigcup_{\alpha} U_\alpha, \tag{3}$$

where $\mathcal{D}(f_G)$ is called the decomposition induced by the decoding map f_G [28], and $\{U_\alpha\}$ are called cells. As shown in Fig. 4, f_G maps a cell decomposition in the latent space $\mathcal{D}(f_G)$ to a cell decomposition in the image space $\frac{1}{n}\sum_i \delta_{x^{(i)}}$. Each cell U_α is mapped to a sample $\delta_{x^{(i)}}$ by the decoding map f_G [27]. In another word, f_G pushes the prior distribution p to the exact empirical distribution,

$$(f_G)_{\#}p = \frac{1}{n}\sum_i \delta_{x^{(i)}}. \tag{4}$$

Theorem 1. (Empirical Approximation) *For any $0 < \epsilon < 1/2$ and any integer $m > 4$, let $g : \mathbb{R}^C \to \mathbb{R}^n$ be the mapping function of generator G with $n \leq \frac{20 \log m}{\epsilon^2}$. For two sets $V_S = \{y_S : y_S \in \mathbb{P}_S\}$ and $V_T = \{y_T : y_T \in \mathbb{P}_T\}$, both of which have m points in $\mathbb{R}^C$, if the empirical Wasserstein distance between $g(V_S)$ and $g(V_T)$ equals zero,*

$$\hat{W}(g(V_S), g(V_T)) = \frac{1}{m}\sum_{i=1}^{m} \|g(y_S^i) - g(y_T^i)\| = 0, \tag{5}$$

then $W(\mathbb{P}_S, \mathbb{P}_T) = 0$.

Thm. 1 (see Appendix for proof) provides a method to approximate the expected Wasserstein distance $W(\mathbb{P}_S, \mathbb{P}_T)$ using the empirical Wasserstein distance $\hat{W}(g(V_S), g(V_T))$. By reducing the distance between points $g(y_S^i)$ and $g(y_T^i)$ in high-dimensional space, an optimization direction $\nabla\mathcal{L}_F$ different from $\nabla\mathcal{L}_{KL}$ is produced for logits y_S^i and y_T^i in low-dimensional space. The gradient update causes y_S^i to move towards the boundary of the cell in which y_T^i resides, as shown in Fig. 4.

Theorem 2. (Optimization Direction) *Let $\mu \in \mathcal{X}$ be any distribution. f_S, f_T, f_G are the mapping functions of the student, teacher, and generator, respectively. f_S is parameterized by $\theta_S \in \Theta_S$. Then, when*

$$\min_{\theta_S \in \Theta_S} \mathbb{E}_{x\sim\mu}\left[\|f_G \circ f_S(x), f_G \circ f_T(x)\|\right] \to 0, \tag{6}$$

it holds that $f_S \to f_T$, and we have

$$\begin{aligned}\nabla_{\theta_S}\mathbb{E}_{x\sim\mu}[f_S(x)] &= \nabla_{\theta_S} W(\mathbb{P}_S, \mathbb{P}_T)\\ &= \mathbb{E}_{x\sim\mu}[\nabla_{\theta_S}\|f_G \circ f_S(x) - f_G \circ f_T(x)\|].\end{aligned} \tag{7}$$

Thus, to achieve $f_S \to f_T$, it is sufficient to optimize $\mathbb{E}_{x\sim\mu}\left[\|f_G \circ f_S(x), f_G \circ f_T(x)\|\right]$ in the parameter space Θ_S. The global gradient of parameter θ_S can be replaced by the gradient calculated on the empirical distance of high-dimensional image points, refer to Appendix for proof.

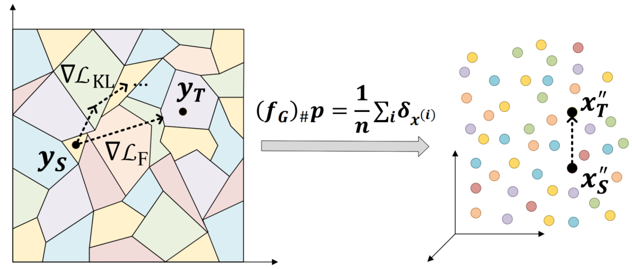


Figure 4. Cell U_α in the latent space is mapped via f_G to an exact image $x^{(i)}$ of the same color. The move of point x'_S to x'_T causes the logits y_S to align with y_T from a direction different from $\mathcal{L}_{KL}$.

Theorem 3. (Generalization Bound) *Let $H \subseteq \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ be a hypothesis set for C-way classification task. For any $0 < \epsilon < 1/2$ and a sample S of size $m > 4$ drawn according to μ, let $g : \mathbb{R}^C \to \mathbb{R}^n$ be a mapping function of generator G with $n \leq \frac{20 \log m}{\epsilon^2}$. Fix $\rho > 0$, for any $1 > \delta > 0$, with probability at least $1 - \delta$, the following holds for all $h \in H$,*

$$R(h) \leq \hat{R}_\rho(h) + \frac{2C^2}{\rho(1-\epsilon)}\sqrt{\frac{r^2\Lambda^2}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}. \tag{8}$$

For any $x \in \mathcal{X}$, the $\Lambda \geq 0$ and $(\sum_{y=1}^{C}\|h(x,y)\|^p)^{1/p} \leq \Lambda$ for any $p \geq 1$, and the $r > 0$ for $K(x,x) \leq r^2$ where kernel $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is positive definite symmetric.

Thm. 3 (see Appendix for proof) gives the generalization bound of aligning low-dimensional logits by reducing the distance of high-dimensional image points, which guarantees generalizability to the unseen samples.

4. Algorithm of Mapping-Emulation KD

Hinton *et al.* [18] propose a simple but effective KD method that uses the softened logits of the teacher model as a supervision to guide student training. They use the Kullback-Leibler Divergence (KLD) to measure the discrepancy between the logits of the two models, where the student model is trained to minimize the gap in the hope of achieving the same output. The loss is defined as

$$\begin{aligned}\mathcal{L}_{KL} &= \mathcal{KL}[p(c|\mathbf{x}_i;\theta_T)||p(c|\mathbf{x}_i;\theta_S)]\\ &= \frac{1}{N}\sum_i^N\sum_c^C p(c|\mathbf{x}_i;\theta_T)\log\left[\frac{p(c|\mathbf{x}_i;\theta_T)}{p(c|\mathbf{x}_i;\theta_S)}\right],\end{aligned} \tag{9}$$

where i is the sample index and N is the number of samples. Regardless of the method used, the essence of KD is to learn the mapping function of the teacher model from input to output, *i.e.*, f_T. However, it is hard to deduce the mapping function from the existing parameters of the teacher model. One can only guess the mapping process by using the responses to the input samples of different network layers or the relations between features and treat them as *knowledge* to guide the training of the student model [52]. However, in the black-box KD problem, the internal responses or relations between layers of the teacher model are not available, which makes effective distillation more challenging.

Deprivatization. For a C-way classification problem, we first train a GAN using the random noise variable z sampled from the prior distribution p in latent space $\mathcal{Y}$ as input. Note that the dimensionality of z is the same as the output logits of the teacher model, *i.e.* $|z| = C$. The generator G uses noise z to synthesize images, and the discriminator D minimizes the Wasserstein distance between the generated μ' and the real distribution μ. The synthetic *privacy-free* images are simultaneously sent to the cloud server for inference responses, which can be soft (probability vectors for all possible classes) or hard (indexes or semantic tags for the category with the highest probability). We expect the synthetic images to match the high responses of the teacher model so that they can maximize the containment of patterns in real data. We adopt the information maximization (IM) loss [20, 42], which is formulated as

$$\mathcal{L}_{IM} = -\frac{1}{m}\sum_{i=1}^{m} \hat{y}_t^{(i)} \log\left(D\left(G\left(z^{(i)}\right)\right)\right), \tag{10}$$

where $\hat{y}_t^{(i)} = \max_{c\in C} T\left(G\left(z^{(i)}\right)\right)$ for $0.0 \leq \hat{y}_t^{(i)} \leq 1.0$ in soft responses and $\hat{y}_t^{(i)} = 1.0$ in hard responses.

Suppose the discriminator is capable of completely blurring the discrepancy between synthetic and genuine images. In this case, the resulting generator represents a function from the latent space to the image space, defined as $f_G : \mathcal{Y} \rightarrow \mathcal{X}$, with an inverse mapping of the teacher function. Note that the generator and the discriminator are trained simultaneously: we adjust parameters for the generator to minimize $\log(1 - D(G(z)))$ and adjust parameters for the discriminator to minimize $\log D(x)$. And their loss functions are

$$\mathcal{L}_D = -\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right], \tag{11}$$

$$\mathcal{L}_G = -\frac{1}{m}\sum_{i=1}^{m}\log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right). \tag{12}$$

We introduce a trade-off hyperparameter α to balance $\mathcal{L}_{GAN}$ and $\mathcal{L}_{IM}$, and all the losses in the first step of deprivatization constitute

$$\mathcal{L}_{Dp} = (\mathcal{L}_G + \mathcal{L}_D) + \alpha\mathcal{L}_{IM}, \tag{13}$$

Distillation. The well-trained generator G contains the knowledge that the teacher uses to make inferences. It is equivalent to a teacher assistant transferring the teacher's knowledge to the student. Fig. 2 illustrates the architecture of MEKD. We freeze the generator and graft it behind the teacher and student model in the same way, using the softened logits of both models as the generator input. A batch of synthetic images $X' = \{x'^{(i)} = f_G(z^{(i)})\}_{i=1}^{m}$ is fed into the embedded network to output high-dimensional points in the same image space, simultaneously. The distance between the output high-dimensional points from the logits of the teacher model $X''_T = f_G \circ f_T(X')$ and the others from the student $X''_S = f_G \circ f_S(X')$ are measured by the distance measurement formula $\mathcal{L}_F = \mathbb{d}(X''_S, X''_T)$. We minimize the distance $\mathcal{L}_F$ to drive the student model to mimic the output logits of the teacher model, and use $\mathcal{L}_1$-norm ($F = 1$) of X''_S and X''_T as the loss function to distill the student,

$$\mathcal{L}_{Dt} = \frac{1}{m}\sum_{i=1}^{m}\left\|G\left(S\left(x'^{(i)}\right)/\tau\right) - G\left(T\left(x'^{(i)}\right)/\tau\right)\right\|_F + \beta\frac{1}{m}\sum_{i=1}^{m} T\left(x'^{(i)}\right)\log\left(\frac{T\left(x'^{(i)}\right)}{S\left(x'^{(i)}\right)}\right), \tag{14}$$

where query sample $x'^{(i)} = G(z^{(i)})$ is generated from noise $z^{(i)}$ and temperature τ is used to soften the output logits. Through the experiments, we found that $\mathcal{L}_2$-norm has a similar effect with $\mathcal{L}_1$-norm, refer to Tab. 5.

We also add logit-level knowledge (Eqn. 9) to induce distillation and use a hyperparameter β to balance these two losses. Unlike most KD methods, we do not use cross-entropy loss with ground-truth labels, due to its unavailability in edge devices. An algorithm is summarized in Alg. 1.

Algorithm 1 MEKD optimization algorithm.

Input: Pre-trained teacher $T(x;\theta_T)$ deployed in the cloud server, random initialized student $S(x;\theta_S)$ and local dataset X hosted in the edge device.

Output: An optimized student $S(x;\theta_S)$ on dataset X.

```
▷ Step 1: Deprivatization
Initialize a generator G(z; θ_G) and a discriminator D(x; θ_D), and ensure the dimensionality of z equals to the category count C.
repeat
    Sample a batch of noises 𝒵 from a prior distribution p and synthetic images 𝒳′ = G(𝒵).
    The 𝒳′ is sent to T in cloud to get soft or hard inference responses 𝒴̂′_t = T(𝒳′).
    Sample a batch of examples 𝒳 from dataset X.
    Update the discriminator D to distinguish 𝒳 and 𝒳′ using ℒ_D from Eqn. 11.
    Update the generator G to fool the discriminator D using ℒ_G + αℒ_IM from Eqn. 10 and Eqn. 12.
until converge
▷ Step 2: Distillation
Initialize the student S and freeze the generator G.
repeat
    Sample a batch of noises 𝒵 from a prior distribution p and synthetic images 𝒳′ = G(𝒵).
    The 𝒳′ is sent to T in cloud to get soft or hard inference responses 𝒴̂′_t = T(𝒳′).
    Update the student S using ℒ_Dt from Eqn. 14.
until converge
```

Method	Data Size	MNIST		CIFAR-10		CIFAR-100		Tiny ImageNet	
Teacher	50K∼100K	ResNet32	VGG13	ResNet56	ResNet56	VGG13	ResNet56	ResNet110	ResNet110
		99.50	99.52	94.15	94.15	74.68	72.06	60.71	60.71
Student	50K∼100K	ResNet8	VGG11	ResNet8	VGG11	VGG11	VGG11	ResNet32	MobileNet
		99.24	99.41	87.74	91.81	69.12	69.12	55.47	56.07
KD [18]	50K∼100K	99.33	99.44	86.58	82.25	70.88	67.97	54.14	57.85
ML [3]	50K∼100K	99.49	99.40	87.89	91.91	67.78	70.18	56.56	60.07
AL [48]	50K∼100K	99.37	99.26	87.25	91.97	69.92	71.13	46.02	51.29
DKD [53]	50K∼100K	99.33	99.43	86.61	92.42	67.32	70.10	55.99	59.43
DAFL [8]	0K	96.42	97.00	60.67	66.03	43.78	48.32	38.44	40.93
KN [34]	10K	98.61	98.81	80.62	82.41	57.83	55.64	48.92	50.22
AM [46]	10K	99.33	**99.47**	74.89	74.26	62.17	63.20	47.72	51.54
DB3KD [49]	10K	98.94	99.16	78.47	85.84	63.48	62.76	47.95	50.49
MEKD (soft)	10K	**99.40**	99.43	**85.36**	**87.27**	**64.76**	64.83	**50.87**	**54.93**
MEKD (hard)	10K	99.40	99.45	84.45	87.25	64.72	**65.32**	49.89	54.71

Table 1. Top-1 classification accuracy (%) of the student model on MNIST, CIFAR-10, CIFAR-100 and Tiny ImageNet.

5. Experiments

5.1. Experiment Setup

In this section, we compare our method with response-based KD and black-box KD methods in an *unsupervised* environment. Experimental results show that when the cross-entropy loss based on ground-truth labels is removed, the distillation performance of these methods decreases.

Datasets Setup. We conduct experiments on MNIST [26], CIFAR [25], Tiny ImageNet [11], and ImageNet-1K [11], all of which are widely used for image classification. While training B2KD methods, we randomly select $10K$ images ($100K$ for ImageNet-1K) from the training set, and all images in the test set (val set for ImageNet) are used as the benchmark to calculate accuracy. For other approaches, except DAFL [8] based on zero-shot learning, we use the whole training set. We mainly use top-1 classification accuracy as an evaluation metric to assess the distillation effect. To make a fair and intuitive comparison, we follow the same setup as previous B2KD methods in our *main experiments*. However, we find that the original settings in the B2KD experiments do not represent the challenges raised in practical applications, so we add *extended experiments* in Sec. 5.4 to illustrate the practicability of our proposed method.

Implementation. See also the project page[1]. We use ResNet [17], VGG [43] and MobileNet [19] as the backbone, and adopt standard data augmentation techniques (random crop and horizontal flip) and an SGD optimizer in all experiments. We consistently train the teacher and student model for 350 epochs, except for 12 epochs for MNIST, and we adopt a multi-step LR scheduler following the paper [22]. After training the teacher, we train a DC-GAN [39] with Gaussian noise in the same dimension as the category counts. The output logits of teacher or student for samples in the same class follow a Gaussian distribution, and the logits center is the mean of the Gaussian. Since the conversion between different Gaussian distributions is a linear process, using Gaussian as the prior distribution p provides a smooth dual space for the student's logits update.

Competing Methods. In order to verify the effectiveness of our method, we compare several methods of response-based KD and black-box KD. We select KD [18] proposed by Hinton *et al.* and ML [3] proposed by Ba and Caruana as the baselines, and we also compare the recently published DKD [53] based on decoupled KLD. For the two GAN-based KD frameworks summarized in Sec. 2, we choose AL [48] and DAFL [8] as comparison methods. Meanwhile, we compete with some black-box KD methods such as KN [34], AM [46] and DB3KD [49]. Of these methods, DB3KD and MEKD(hard) only utilize hard responses, while the other methods are based on soft responses.

5.2. Performance Evaluation

On MNIST, CIFAR, and Tiny ImageNet, we use ResNet32/56/110 and VGG13 as the teacher model and use ResNet8/32, VGG11, and MobileNet as the student model. We compare the top-1 classification accuracy (ACC) of different teacher-student pairs, the results are shown in Tab. 1.

On relatively easy tasks, such as MNIST and CIFAR-10,

T - S Pairs	KD (soft)	AL (soft)	AM (soft)	DB3KD (hard)	MEKD (soft)
RN50 - RN34	52.08	53.50	56.92	58.61	**59.89**
RX101 - RX50	54.90	50.88	55.64	59.90	**61.21**

Table 2. Top-1 classification accuracy (%) of the student model on ImageNet-1K with data size $100K$. We use the pre-trained ResNet50 (76.13%) and ResNeXt101 (79.32%) as teachers.

[1] https://github.com/HAIV-Lab/MEKD

our proposed method has a small gap compared to response-baed KD methods that use the full training set. This makes sense in the applications of cloud-to-edge model compression because edge devices do not have a lot of capacity to store more than ten thousand pieces of data.

CIFAR-100 and Tiny ImageNet are more challenging. These tasks contain far more patterns than MNIST and CIFAR-10, and data distributions are so complex that it is difficult for a generator to capture all the patterns. However, as long as the mode collapse problem can be mitigated, it is possible to synthesize complex samples beneficial to distillation, so we exploit DCGAN [39] as our generator. DCGAN has a more stable training process and is more suitable for generating RGB images than a fully-connected GAN [39]. Experimental results show that MEKD can obtain an accuracy improvement of $5\% \sim 10\%$ compared to other B2KD methods, and the accuracy of MEKD with soft or hard responses is similar, with a difference of less than 1%.

We also conduct experiments on large-scale datasets and sophisticated networks. On ImageNet-1K, we use two teacher-student (T-S) pairs of ResNet50 (RN50) - ResNet34 (RN34) and ResNeXt101 (RX101) - ResNeXt50 (RX50). All methods are trained using a subset of $100K$ samples. The experimental results are shown in Tab. 2.

Uniformly, we set the number of query samples to $50K$ on CIFAR and MNIST, $300K$ on ImageNet, and discuss the performance impact of limited query samples in Sec. 5.4.

5.3. Ablation Study

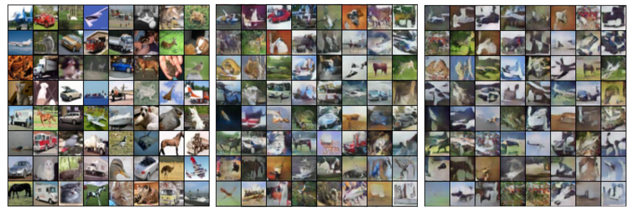
We choose an effective T-S pair [32] of ResNet56 - MobileNet for ablation studies unless otherwise stated.

Ablation Study of Data Size. We explore the performance with different data sizes, the results are shown in Tab. 3. In general, B2KD methods have higher robustness to small data sizes than traditional KD methods, and in which MEKD achieves the highest distillation performance.

Ablation Study of Deprivatization. The α is a hyperparameter to balance $\mathcal{L}_{GAN}$ and $\mathcal{L}_{IM}$. The $\mathcal{L}_{IM}$ is used to maximize the responses of the teacher to the generated samples. Therefore, the training of the generator with or without $\mathcal{L}_{IM}$ will affect the quality of synthetic images. Fig. 5 (a) shows real images of CIFAR-10. Fig. 5 (b) shows synthetic

Data Size	0.1K	1K	10K	50K (full)
KD [18]	16.74	31.25	70.90	90.43
AL [48]	12.97	32.05	68.61	90.54
AM [46]	48.31	62.05	73.65	86.33
DB3KD [49]	43.05	64.28	81.67	92.46
MEKD (soft)	49.04	69.84	86.85	93.48
MEKD (hard)	47.12	68.66	86.53	93.09

Table 3. Ablation study of data size on CIFAR-10. We use the T-S pair of ResNet56 - MobileNet, and the full training set is $50K$.



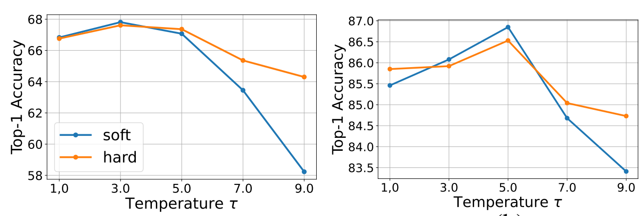
(a) (b) (c)

Figure 5. Real images of CIFAR-10 (a) and synthetic images using MEKD with $\mathcal{L}_{IM}$ (b) and without $\mathcal{L}_{IM}$ (c).

images with $\alpha = 0.5$ and Fig. 5 (c) shows synthetic images without $\mathcal{L}_{IM}$ (*i.e.* $\alpha = 0$), both using the same noise vectors. The teacher of ResNet56 responds from $0.72 \sim 0.96$ to the synthetic image in Fig. 5 (b) and from $0.41 \sim 0.87$ to the one in Fig. 5 (c). The effect of α is also reported in Tab. 4, which reflects that the utilization of $\mathcal{L}_{IM}$ can improve the performance of model distillation.

Ablation Study of Distillation. In Eqn. 14, the β is a trade-off hyperparameter to balance $\mathcal{L}_F$ and $\mathcal{L}_{KL}$, which provide different gradient directions for θ_S. As shown in Tab. 4, the distillation performance can be improved by introducing KLD as an additional loss function.

The temperature τ is another important hyperparameter for MEKD since it softens the output logits of both the teacher and student models. The results are shown in Fig. 6. Its validity comes from the fact that softened logits can increase the probability of being sampled in a standard normal distribution. Since GANs use a standard Gaussian distribution as input, samples generated from out-of-distribution noises with low-sampling probability are usually fuzzy and incorporate few patterns [40], which are meaningless for distillation. Meanwhile, a high value of τ reduces the discrepancy between softened logits, and $\mathcal{L}_F = 0$ when they locate in the same cell. It reduces the performance of distillation, especially for challenging tasks, such as ImageNet.



(a) (b)

Figure 6. Ablation study of temperature τ on CIFAR-100 (a) and CIFAR-10 (b). We use the T-S pair of ResNet56 - MobileNet.

α	Response Type Soft	Response Type Hard	β	Response Type Soft	Response Type Hard
0.0	60.87	61.31	0.0	56.28	56.23
0.1	66.06	66.86	0.1	64.13	65.60
0.5	67.07	67.36	0.5	66.79	67.01
1.0	67.01	67.11	1.0	67.07	67.36

Table 4. Ablation study of hyperparamete α and β on CIFAR-100. We use the T-S pair of ResNet56 - MobileNet.

Dataset	Method	Model	ACC ($\mathcal{L}_1/\mathcal{L}_2$)
CIFAR-10	MEKD (soft)	MobileNet	86.85 / 86.63
	MEKD (hard)	MobileNet	86.53 / 86.88
CIFAR-100	MEKD (soft)	MobileNet	67.07 / 66.95
	MEKD (hard)	MobileNet	67.36 / 66.94

Table 5. Ablation study of different $\mathcal{L}_F$. We use ResNet56 as the teacher model. ACC: top-1 classification accuracy (%).

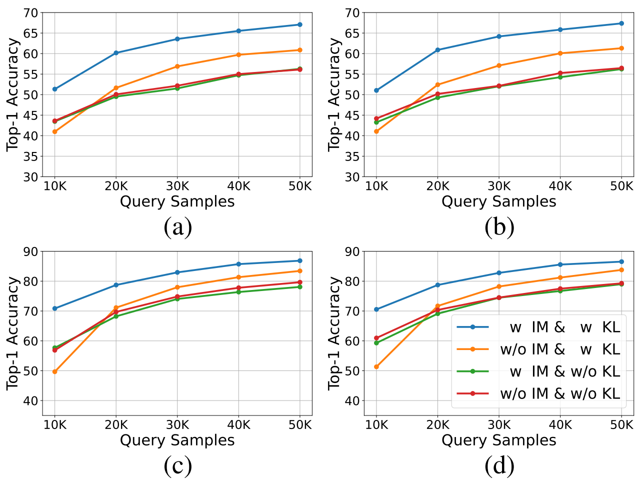


Figure 7. Curve of top-1 classification accuracy on the datasets of CIFAR-100 (a,b) and CIFAR-10 (c,d). Using MEKD with soft (a,c) or hard (b,d) responses with or without $\mathcal{L}_{IM}$ and $\mathcal{L}_{KL}$. We use ResNet56 as the teacher and use MobileNet as the student.

Ablation Study of Different $\mathcal{L}_F$. In Eqn. 14, we use $\mathcal{L}_F$ to calculate the distance between generated samples X''_S and X''_T. From the analysis of experimental results, as shown in Tab. 5, we argue that the effect on distillation is similar whether F equals 1 or 2. The reason is that $\mathcal{L}_F$ is used to measure the distance between logits of the student and the boundary of cells, in which logits of the teacher reside, and different $\mathcal{L}_F$ represent similar gradient directions.

5.4. Extended Experiments

In the real-world application of cloud-to-edge model compression, there are some restrictions, such as the limitation of Internet data exchange and the domain shift in practical scenarios. We conduct additional experiments to explore the effect of MEKD under these constraints.

MEKD with Limited Query Samples. We distill a student MobileNet on CIFAR-10 and CIFAR-100 with a total query sample size ranging from $10K$ to $50K$ with an interval of $10K$. We report the ACC of MEKD with or without $\mathcal{L}_{IM}$ and $\mathcal{L}_{KL}$. The curves in Fig. 7 show that with more query samples sent to the cloud server, the student model in the edge device can be trained more fully. We can also analyze from the curves that $\mathcal{L}_{IM}$ does not seem to be that useful without using KLD as an additional distillation loss function, and it gives a big boost to the overall MEKD due to the extra gradient direction of the mapping emulation.

Teacher / Student	ResNet56 74.37 MobileNet	VGG13 79.86 MobileNet	Data Exchange
KD [18]	76.27	80.67	$\sim$ 175 MB
ML [3]	76.78	81.90	$\sim$ 175 MB
AL [48]	77.09	80.98	$\sim$ 175 MB
DKD [53]	75.47	80.64	$\sim$ 175 MB
DAFL [8]	69.20	67.07	$\sim$ 28.4 GB
KN [34]	79.65	83.37	$\sim$ 145 MB
AM [46]	84.05	86.70	$\sim$ 11.6 GB
DB3KD [49]	90.15	91.14	$\sim$ 20.8 GB
MEKD (soft)	86.45	88.65	$\sim$ 120 MB
MEKD (hard)	86.77	89.21	$\sim$ 120 MB

Table 6. Top-1 classification accuracy (%) of methods on SVHN. The teacher models are trained on Syn. Digits with vanilla supervised learning, and achieve the top-1 classification accuracy of 99.56% for ResNet56 and 99.52% for VGG13 on Syn.Digits.

MEKD with Out-of-Domain Data. We train a teacher (ResNet56 or VGG13) with vanilla supervised learning on Syn. Digits [12], which contains about $500K$ software-synthesized images. We distill a student (MobileNet) on SVHN [33] consisting only of real-shooting photographs. Tab. 6 shows the ACC on the test set of SVHN. MEKD outperforms most methods in the task of out-of-domain distillation, while DB3KD achieves higher performance due to the use of robust labels [49]. However, DB3KD leads to a very high data exchange cost between the server and client, since it requires multiple queries to find a mixed image located in the decision boundary to compute robust labels. In contrast, the data exchange cost of MEKD is much lower.

6. Conclusion

In this paper, we provide a two-step workflow of deprivatization and distillation for B2KD. Different from aligning logits directly, we theoretically provide a new optimization direction from logits to cell boundaries, and propose a new method of MEKD. Taking a generator as an inverse mapping of the teacher function does not leak information about the internal structure or parameters of the teacher, because it has a completely different network structure.

Limitation. A well-trained generator is critical in MEKD, and GANs are known to suffer from mode collapse, especially for challenging tasks. We alleviate this problem with DCGAN. Although the parameter size and structural limitations of the model prevent the student from fully mimicking the function of the teacher, MEKD can still improve distillation performance compared with other B2KD methods.

Acknowledgement. This research was supported by Natural Science Fund of Hubei Province (Grant # 2022CFB823), Alibaba Innovation Research program under Grant Contract # CRAQ7WHZ11220001-20978282, and HUST Independent Innovation Research Fund (Grant # 2021XXJS096).

References

[1] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7350–7357, 2020.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

[3] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in Neural Information Processing Systems*, 27, 2014.

[4] Vasileios Belagiannis, Azade Farshad, and Fabio Galasso. Adversarial network compression. In *Proceedings of the European Conference on Computer Vision Workshops*, pages 0–0, 2018.

[5] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4183–4192, 2020.

[6] Jürgen Braun and Michael Griebel. On a constructive proof of kolmogorov's superposition theorem. *Constructive Approximation*, 30(3):653–675, 2009.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[8] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3514–3522, 2019.

[9] Hanting Chen, Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Learning student networks via feature embedding. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):25–35, 2020.

[10] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in Neural Information Processing Systems*, 29, 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[12] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[14] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[15] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

[16] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33:14068–14080, 2020.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[18] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint:1503.02531*, 2(7), 2015.

[19] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[20] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017.

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[22] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in Neural Information Processing Systems*, 31, 2018.

[23] Nikos Komodakis and Sergey Zagoruyko. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[24] Mario Köppen. On the training of a kolmogorov network. In *International Conference on Artificial Neural Networks*, pages 474–479. Springer, 2002.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Unvieristy of Toronto: Technical Report*, 2009.

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[27] Na Lei, Kehua Su, Li Cui, Shing-Tung Yau, and Xianfeng Gu. A geometric view of optimal transportation and generative model. *Computer Aided Geometric Design*, 68:1–21, 2019.

[28] Na Lei, Dongsheng An, Yang Guo, Kehua Su, Shixia Liu, Zhongxuan Luo, Shing-Tung Yau, and Xianfeng Gu. A geometric understanding of deep learning. *Engineering*, 6(3): 361–374, 2020.

[29] Zhong Meng, Jinyu Li, Yong Zhao, and Yifan Gong. Conditional teacher-student learning. In *Proceedings of the IEEE*

International Conference on Acoustics, Speech and Signal Processing, pages 6445–6449, 2019.
[30] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. *Advances in Neural Information Processing Systems*, 32, 2019.
[31] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint:1411.1784*, 2014.
[32] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, pages 5191–5198, 2020.
[33] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
[34] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963, 2019.
[35] Kaan Ozkara, Navjot Singh, Deepesh Data, and Suhas Diggavi. Quped: Quantized personalization via distillation with applications to federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.
[36] Boxiao Pan, Haoye Cai, De-An Huang, Kuan-Hui Lee, Adrien Gaidon, Ehsan Adeli, and Juan Carlos Niebles. Spatio-temporal graph for video captioning with knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10870–10879, 2020.
[37] Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2020.
[38] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13657–13665, 2021.
[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint:1511.06434*, 2015.
[40] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.
[41] Zhiqiang Shen, Zhankui He, and Xiangyang Xue. Meal: Multi-model ensemble via adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4886–4893, 2019.
[42] Yuan Shi and Fei Sha. Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In *ICML*, 2012.
[43] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[44] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34, 2021.
[45] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618, 2016.
[46] Dongdong Wang, Yandong Li, Liqiang Wang, and Boqing Gong. Neural networks are more productive teachers than human raters: Active mixup for data-efficient knowledge distillation from a blackbox model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1498–1507, 2020.
[47] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020.
[48] Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. Adversarial learning of portable student networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
[49] Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. In *International Conference on Machine Learning*, pages 10675–10685. PMLR, 2021.
[50] Zheng Xu, Yen-Chang Hsu, and Jiawei Huang. Training shallow and thin networks for acceleration via knowledge distillation with conditional adversarial networks. *arXiv preprint:1709.00513*, 2017.
[51] Jingwen Ye, Yixin Ji, Xinchao Wang, Xin Gao, and Mingli Song. Data-free knowledge amalgamation via group-stack dual-gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12516–12525, 2020.
[52] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.
[53] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11953–11962, 2022.
[54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.