Progressively Knowledge Distillation via Re-parameterizing Diffusion Reverse Process

Xufeng Yao, Fanbin Lu, Yuechen Zhang, Xinyun Zhang, Wenqian Zhao, Bei Yu

Department of Computer Science & Engineering, The Chinese University of Hong Kong
{xfyao,fblu21,yczhang21,xyzhang21,wqzhao,byu}@cse.cuhk.edu.hk

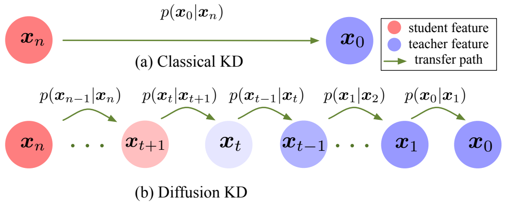
Abstract

Knowledge distillation aims at transferring knowledge from the teacher model to the student one by aligning their distributions. Feature-level distillation often uses $\mathcal{L}_2$ distance or its variants as the loss function, based on the assumption that outputs follow normal distributions. This poses a significant challenge when distribution gaps are substantial since this loss function ignores the variance term. To address the problem, we propose to decompose the transfer objective into small parts and optimize it progressively. This process is inspired by diffusion models from which the noise distribution is mapped to the target distribution step by step. However, directly employing diffusion models is impractical in the distillation scenario due to its heavy reverse process. To overcome this challenge, we adopt the structural re-parameterization technique to generate multiple student features to approximate the teacher features sequentially. The multiple student features are combined linearly in inference time without extra cost. We present extensive experiments performed on various transfer scenarios, such as CNN-to-CNN and Transformer-to-CNN, that validate the effectiveness of our approach. Code is available at https://github.com/yaoxufeng/KDiffusion.

Introduction

The revolutionary advancement of Deep Neural Network (DNN) models has exhibited immense success in various domains of computer vision. However, their remarkable triumph comes at the cost of significant computation and memory consumption, presenting a formidable challenge in deploying these models in resource-limited industrial applications. To address this, recent research suggests Knowledge Distillation as a promising resolution wherein knowledge from a large model (i.e., teacher) can be efficiently transferred to a lightweight model (i.e., student).

In the seminal work by Hinton *et al.* (Hinton, Vinyals, and Dean 2015), the concept of Knowledge Distillation (KD) was introduced and the transfer objective was achieved by minimizing the KL-Divergence (Kullback and Leibler 1951) between softened outputs (logits) of the teacher and student. On the other hand, many efforts have focused on enhancing the effectiveness of feature-level distillation. For instance, FitNet (Romero et al. 2014) leverages intermediate features to facilitate knowledge transfer, while CRD (Tian, Krishnan, and Isola 2020) utilizes a contrastive objective for distillation by maximizing mutual information between representations. Other approaches such as multi-level feature distillation (Ahn et al. 2019; Chen et al. 2021b) have also demonstrated promising results when applied to similar architectures (e.g., CNN-to-CNN) in recent studies.


$p(\boldsymbol{x}_0|\boldsymbol{x}_n)$
$\boldsymbol{x}_n$
$\boldsymbol{x}_0$
student feature
teacher feature
transfer path
$p(\boldsymbol{x}_{n-1}|\boldsymbol{x}_n)$
$p(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$
$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$
$p(\boldsymbol{x}_1|\boldsymbol{x}_2)$
$p(\boldsymbol{x}_0|\boldsymbol{x}_1)$
$\boldsymbol{x}_n$
$\boldsymbol{x}_{t+1}$
$\boldsymbol{x}_t$
$\boldsymbol{x}_{t-1}$
$\boldsymbol{x}_1$
$\boldsymbol{x}_0$

(a) Classical KD

(b) Diffusion KD

Figure 1: (a) Conventional feature-level distillation directly predicts teacher by student. (b) Our proposed diffusion KD decouples the objective into multiple timesteps and transfer step by step.

Our New Finding. However, it is observed that the distillation performance may be disrupted in the presence of significant distribution gaps. As outlined in Table 1, some conventional feature-level distillation techniques, such as CRD (Tian, Krishnan, and Isola 2020), exhibit inferior performance when confronted with the difficult Transformer-to-CNN scenario. Such methods yield only marginal distillation improvement, and may even introduce negative transfer effects.

Feature-level KD mainly uses $\mathcal{L}_2$ distance as the loss function. From the perspective of maximum likelihood estimation, this loss function is based on the assumption that the outputs conform to the normal distribution, and the objective is to predict the corresponding $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$. We can construct the density function as $p(\boldsymbol{x}^T|\boldsymbol{x}^S) = \frac{1}{\sqrt{2\pi}\hat{\sigma}} \exp\left(-\frac{1}{2}\frac{(\boldsymbol{x}^T-\hat{\boldsymbol{\mu}})^2}{\hat{\boldsymbol{\sigma}}^2}\right)$. The loss function can, therefore, be expressed as follows:

$$\mathcal{L}_{trans} = -\log p(\boldsymbol{x}^T|\boldsymbol{x}^S) \propto \log \hat{\boldsymbol{\sigma}} + \frac{(\boldsymbol{x}^T - \hat{\boldsymbol{\mu}})^2}{2\widehat{\boldsymbol{\sigma}^2}}. \quad (1)$$

In the standard $\mathcal{L}_2$ loss paradigm, variance is treated as a constant value. This assumption may pose a significant challenge when confronting large distribution gaps. Also, since

Teacher	Swin	Swin	Swin
	94.48%	94.48%	94.48%
Student	MobileNetV2	ResNet18	ShuffleNetV2
	84.04%	84.42%	76.86%
CRD	83.72%	84.26%	77.88%
	-0.32	-0.16	+1.02

Table 1: Top-1 accuracies of teacher and student networks on ImageNet100. Please refer to experiments for more details.

the $\mathcal{L}_2$ distance is isotropic, it may lead to noise amplification during training in the absence of appropriate constraints. Nonetheless, scrutinizing true distributions is a demanding task, particularly when both teacher and student features are sampled from complex distributions. In this work, we propose a solution that utilizes diffusion techniques to address this issue.

Motivation and Our Solution. The objective of KD can be formulated as matching student and teacher distributions. As a generative techinique, diffusion models have been used to map noise distribution to the target one by employing a DNN model to approximate the reverse process in each step. A significant advantage of utilizing diffusion techniques in KD is the ability to break down the transfer objective into smaller parts and transfer knowledge gradually. Additionally, diffusion models can model the distribution without loss of dimension compared with GAN (Goodfellow et al. 2020) or VAE (Kingma and Welling 2013), which is also crucial in the distillation scenario where there may be a high-dimensional optimization problem due to the large distribution gaps. As shown in Figure 1, classical feature-level distillation directly optimizes $-\log p(\boldsymbol{x_0}|\boldsymbol{x_n})$, where $\boldsymbol{x_0}$ and $\boldsymbol{x_n}$ represent teacher and student features, respectively. However, this approach may lead to instability during training, as previously explained. In contrast, our method progressively approximates intermediate features constructed by diffusion process, enabling us to optimize middle steps with low variance, and transfer knowledge with greater safety.

Although diffusion is known to be a practical approach to map the student distribution to the teacher one, it requires the involvement of multiple student features in the training process. To overcome this constraint, we apply recent advancements in the field of structural-reparameterization techniques, as presented in (Ding et al. 2021), to generate numerous student features. Throughout the training phase, we make use of these generated student features to approximate the corresponding teacher target features sequentially. In the inferece stage, we can efficiently merge all student features without additional inference costs by leveraging their linear properties. Our contributions are summarized as follows.

- We identify a limitation of classical KD methods when faced with large distribution gaps, and propose a diffusion-based framework that overcomes this challenge.
- We introduce a new challenging Transformer-to-CNN setting and benchmark ten different distillation methods on this task. Through our experiments, we demonstrate the effectiveness of our proposed algorithm compared to other state-of-the-art methods.
- We evaluate our approach on a range of computer vision tasks and achieve competitive results across multiple domains.

Related Work

Knowledge Distillation. As a model compression technique (Buciluǎ, Caruana, and Niculescu-Mizil 2006), knowledge distillation aims to transfer the knowledge from a teacher model to a student model by aligning their distributions. Conventional KD (Hinton, Vinyals, and Dean 2015) minimizes the KL divergence between teacher and student logits-level output. Extensive methods focus on feature-level distillation since intermediate features contain rich information. For instance, FitNet (Romero et al. 2014) uses $\mathcal{L}_2$ loss to minimize the distance between teacher and student's middle features. OFD (Heo et al. 2019) introduces partial L2 loss instead to prevent negative transfer. (Yim et al. 2017; Zagoruyko and Komodakis 2016) either use gram matrix or attention map to maximize the correlation. CRD (Tian, Krishnan, and Isola 2020) proposes a contrastive-based transfer objective where the cosine similarity between normalized representations is proportional to $\mathcal{L}_2$ distance. Additionally, methods like Review (Chen et al. 2021b), AFD (Ji, Heo, and Park 2021), and SemCKD (Chen et al. 2021a) guide single-level student features to learn multi-level teacher features. However, these methodologies overlook the large distribution issue.

Incorporating generative techniques to emulate teacher distributions has also been explored within the KD framework. For example, KDGAN (Wang et al. 2018) deploys GANs to ensure the student classifier aligns with the genuine teacher distribution. VID (Ahn et al. 2019) employs a variational information maximization approach to maximize mutual information between student and teacher outputs. These strategies, lacking a multi-step optimization approach, may pose optimization challenges.

Diffusion Models. Denoising diffusion models have exhibited the ability to produce high-quality samples across many domains (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; Rombach et al. 2022; Ramesh et al. 2022). As a family of generative models, diffusion models are proficient in establishing the connection between a noise distribution and the target one. Among the most imperative advantages conferred by diffusion models is the capacity to construct intermediate steps between the source and target, thus simplifying the complex and high-dimensional optimization problem into smaller parts, which can be subsequently solved progressively. Therefore, the adoption of diffusion techniques is bound to be effective in the distillation of large discrepancies between the teacher and student distributions.

Method

In this study, our objective is to enhance distillation performance by progressively transferring knowledge through approximating the diffusion reverse process. To address this, we first provide a comprehensive review of the general formulation of transfer learning. We then motivate the diffusion process in knowledge distillation. Meanwhile we review diffusion process and examine the limitations of directly using diffusion models in KD. Then we introduce proposed alogrithm which distills knowledge via re-parameterizing diffusion reverse process. To further enhance the reversal process, we incorporate our algorithm with target-guided and shuffle sampling strategies.

General Formulation of Transfer Learning

Generally, the objective of transfer learning is to align the teacher and student distributions. We define P and Q are corresponding distributions, then the conventional KL divergence between teacher and student distributions can be defined as :

$$\mathrm{KL}(P||Q) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}^T)\log(\frac{p(\boldsymbol{x}^T)}{q(\boldsymbol{x}^S)}), \tag{2}$$

where $\boldsymbol{x}^T$ and $\boldsymbol{x}^S$ are teacher and student feature outputs, respectively. If two distributions are equivalent, the KL divergence are zero. In deep transfer learning, we often use a small neural network to predict teacher feature outputs $\boldsymbol{x}^T$ by student feature $\boldsymbol{x}^S$. With regard to the maximum likelihood estimation approach, the transfer objective can be defined as $-\log(q_\theta(\boldsymbol{x}^T|\boldsymbol{x}^S))$. Normally we use $\mathcal{L}_2$ loss (or the variants of $\mathcal{L}_2$ loss) between teacher and predicted teacher feature outputs based on the assumption that outputs are Gaussian.

In the context of large distribution gaps, modeling $-(\log q_\theta(\boldsymbol{x}^T|\boldsymbol{x}^S)$ presents a significant challenge. In this work, we address this issue by decomposing the problem into small pieces and solving it progressively. By assuming the Markov chain for the intermediate steps between teacher and student, the transfer objective can be reformulated as:

$$-\log\left(q_\theta(\boldsymbol{x}_0^T|\boldsymbol{x}_1^T)\cdots q_\theta(\boldsymbol{x}_{t-1}^T|\boldsymbol{x}_t^T)\cdots q_\theta(\boldsymbol{x}_{n-1}^T|\boldsymbol{x}_n^S)\right), \tag{3}$$

where $\boldsymbol{x}_0^T$ and $\boldsymbol{x}_n^S$ are original teacher and student feature outputs. $\boldsymbol{x}_1^T, \boldsymbol{x}_2^T, \cdots, \boldsymbol{x}_n^T$ denote intermediate features between $\boldsymbol{x}_0^T$ and $\boldsymbol{x}_n^S$. Instead of directly predicting $\boldsymbol{x}_0^T$ by $\boldsymbol{x}_n^S$, which may lead to negative transfer, we can optimize the intermediate steps (e.g., $-\log q_\theta(\boldsymbol{x}_{t-1}^T|\boldsymbol{x}_t^T)$) and safely transfer the knowledge.

In order to enhance the optimization of intermediate processes, it is imperative to construct intermediate features and their corresponding probability density functions. To accomplish this task, we employ diffusion techniques that enable us to effectively address the problem at hand.

Review Diffusion Process

Assume we have a series student features $\boldsymbol{x}_0^S, \boldsymbol{x}_1^S \cdots \boldsymbol{x}_n^S$ which are sampled independently from the standard Normal distribution. Given the teacher target features $\boldsymbol{x}_0^T$ and student source features $\boldsymbol{x}_t^S$, the diffusion forward process can be given by:

$$\boldsymbol{x}_t^T = \alpha_t \boldsymbol{x}_{t-1}^T + \beta_t \boldsymbol{x}_t^S = \hat{\alpha}_t \boldsymbol{x}_0^T + \hat{\beta}_t \boldsymbol{x}_0^S, \tag{4}$$

where $\hat{\alpha}_t = \alpha_1 \cdots \alpha_t$ and $\hat{\beta}_t = \sqrt{1-\hat{\alpha_t^2}}$. We use the default setting by taking $\alpha_t^2 + \beta_t^2 = 1$. Since any $\boldsymbol{x}^S$ are sampled from the normal distribution, we can write down the density function of any intermediate features $\boldsymbol{x}_t^T$ by:

$$q(\boldsymbol{x}_t^T|\boldsymbol{x}_0^T) := \mathcal{N}(\boldsymbol{x}_t^T; \hat{\alpha}_t \boldsymbol{x}_0^T, \hat{\beta}_t^2 \mathbf{I}). \tag{5}$$

In the reverse process, we endeavor to recover the previous step $\boldsymbol{x}_{t-1}^T$ by current step $\boldsymbol{x}_t^T$ sequentially, which is often achieved by using a neural network (e.g., UNet (Ronneberger, Fischer, and Brox 2015)). If the student features are from noise distributions, the learning process can be regarded as a denoising process.

Assume we have a well-trained diffusion model u_θ, $\boldsymbol{x}_{t-1}^T$ can be recovered by:

$$\boldsymbol{x}_{t-1}^T = \frac{1}{\sqrt{\alpha_t}}(\boldsymbol{x}_t^T - \frac{1-\alpha_t}{\sqrt{1-\hat{\alpha}_t}}\mu_\theta(\boldsymbol{x}_t^T, t)) + \sigma_t \boldsymbol{x}_t^S. \tag{6}$$

We can establish a basic distillation pipeline based on the diffusion models. In the training process, we can train a diffusion model to establish a connection between teacher and student features. In the inference stage, we can recover teacher features from student features via trained diffusion models.

However, this basic design has several drawbacks. Firstly, the reverse of diffusion process is time-consuming and the introduction of a diffusion model in the inference stage is quite costly. Secondly, the diffusion theories rely on sampling multiple student features. Unfortunately, we only have one student feature $\boldsymbol{x}_n^S$ in our basic design setting.

Knowledge Transfer via Re-parameterizing Diffusion Reverse Process

Structural Re-parameterization. In order to overcome the issues of limited sampling and heavy reverse process, we propose the utilization of structural re-parameterization techniques (Ding et al. 2021) for generating more feature samples. Structural re-parameterization leverages the linear properties of a set of linear modules $f_0, f_1, \cdots, f_n$ which can produce diverse outputs with a common input, i.e., $f_0(x), f_1(x), ..., f_n(x)$. The combination of these modules can be expressed as follows:

$$\alpha_1 f_0(x) + \cdots + \alpha_n f_n(x) = (\alpha_1 f_0 + \cdots + \alpha_n f_n)(x). \tag{7}$$

Since both MLP and convolution operations in neural networks contain linear functions, structural re-parameterization techniques can be employed to produce an arbitrary number of feature outputs without additional inference cost. It is worth noting that we only generate multiple student feature outputs from the last layer of each stage to enable a fair comparison in our experiments. Please refer to the appendix for further details due to the page limit.

Contructing the diffusion forward process. We follow the same setting in (Kingma. et al. 2013; Ahn et al. 2019) that assumes feature outputs follow normal distributions. In this work, given multiple student features after a batch normalization (Ioffe and Szegedy 2015) layer, we define they

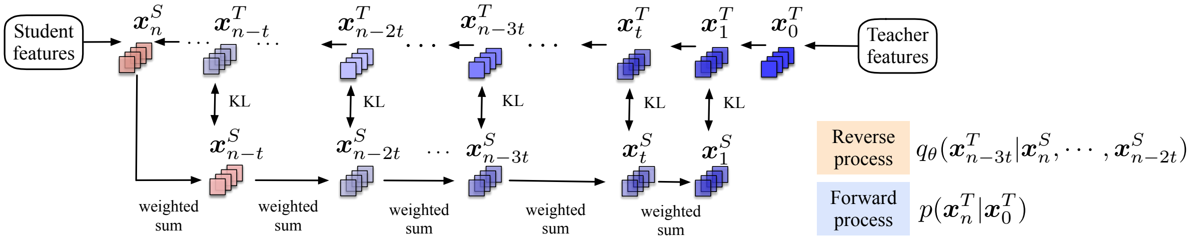


Figure 2: Proposed knowledge transfer via re-parameterizing diffusion reverse progress.

follow a complex normal distribution $\mathcal{N}(0, \sigma_S^2)$. We can obtain the probability distributions of each intermediate features $\boldsymbol{x}_t^T$ by:

$$q(\boldsymbol{x}_t^T|\boldsymbol{x}_0^T) := \mathcal{N}(\boldsymbol{x}_t^T; \hat{\alpha_t}\boldsymbol{x}_0^T, \hat{\beta}_t^2\sigma_S^2). \tag{8}$$

Formulating the diffusion reverse process. The objective of our algorithm is to leverage multiple student features to approximate the diffusion reverse process, while ensuring that all student features can be linearly integrated at the final stage. Despite the fact that we can generate numerous student features without incurring additional inference cost, the process of training a large student network can be quite expensive. Hence, we opt to create m ($m \ll n$) student features and subsequently recover the teacher features using m steps. Assuming that the duration for each reverse step is t ($t \approx \frac{n}{m}$), the objective in timestep $\{n-t\}$ is to recover $\boldsymbol{x}_{n-t}^T$ using $\boldsymbol{x}_n^T$.

Since we do not know the probability distribution of $q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T)$, similiar to (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020), we introduce $q(\boldsymbol{x}_0^T)$ to achieve the density function:

$$q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T, \boldsymbol{x}_0^T) = \frac{q(\boldsymbol{x}_n^T|\boldsymbol{x}_{n-t}^T)q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_0^T)}{q(\boldsymbol{x}_n^T|\boldsymbol{x}_0^T)}. \tag{9}$$

Equation (9) is also Gaussian, so the density function can be given as Equation (10).

$$\begin{aligned} &q(\boldsymbol{x}_{n-t}^T|\boldsymbol{x}_n^T, \boldsymbol{x}_0^T) := \mathcal{N}(\boldsymbol{x}_{n-t}^T; u(\boldsymbol{x}_n^T) + v(\boldsymbol{x}_0^T), w(\sigma_S^2)),\\ &\text{where} \quad u(\boldsymbol{x}_n^T) = \frac{\hat{\beta_{n-t}^2}\hat{\alpha_{n2t}}}{\hat{\beta_n^2}}\boldsymbol{x}_n^T, v(\boldsymbol{x}_0^T) = \frac{\hat{\beta_{n2t}^2}\hat{\alpha_{n-t}}}{\hat{\beta_n^2}}\boldsymbol{x}_0^T\\ &w(\sigma_S^2) = \frac{\hat{\beta_{n2t}^2}\hat{\beta_{n-t}^2}}{\hat{\beta_n^2}}\sigma_S^2, \hat{\alpha_{n2t}} = \frac{\hat{\alpha_n}}{\hat{\alpha_{n-t}}}, \hat{\beta_{n2t}^2} = 1 - \hat{\alpha_{n2t}^2}. \end{aligned} \tag{10}$$

Upon observation of the reverse process, there are two parts in reverse process that needs to predict, i.e., $u(\boldsymbol{x}_n^T)$ and $v(\boldsymbol{x}_0^T)$.

Establishing transfer objective on intermediate steps. We take one intermediate step $\boldsymbol{x}_{n-3t}^T$ as an example. To reverse $\boldsymbol{x}_{n-3t}^T$, we need to predict $\boldsymbol{x}_{n-2t}^T$ and $\boldsymbol{x}_0^T$. Since $\boldsymbol{x}_{n-2t}^T$ is given by the previous step, we first consider $u(\boldsymbol{x}_0^T)$. In diffusion models this term is predicted by $u_\theta(\boldsymbol{x}_{n-2t}^T)$ where u_θ parameterizes a neural network. In our setting, we do not want to introduce a diffusion model, then we use the current step of student features $\boldsymbol{x}_{n-2t}^S$ to predict this term directly. As $\boldsymbol{x}_{n-2t}^T$ is predicted by the previous steps of student features, the transfer objective of the intermediate step can be defined as:

$$D_{KL}(p(\boldsymbol{x}_{n-3t}^T|\boldsymbol{x}_{n-2t}^T, \boldsymbol{x}_0^T)||q_\theta(\boldsymbol{x}_{n-3t}^T|\boldsymbol{x}_{n-2t}^S \cdots \boldsymbol{x}_n^S)). \tag{11}$$

As shown in Figure 2, in each intermediate steps $\{n-3t\}$, we use $\{\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_{n-2t}^S\}$ student features to predict it, where $\boldsymbol{x}_{n-2t}^S$ is used to predict corresponding $\boldsymbol{x}_0^T$ part and the rest are used to predict $\boldsymbol{x}_{n-2t}^T$ parts (which is predicted by the previous step). Equation (11) consists of a log variance term and mean term, since both $p(\boldsymbol{x}_{n-3t}^T|\boldsymbol{x}_{n-2t}^T, \boldsymbol{x}_0^T)$ and $q_\theta(\boldsymbol{x}_{n-3t}^T|\boldsymbol{x}_{n-2t}^S \cdots \boldsymbol{x}_n^S)$'s variance term is based on σ_S^2 because of re-parameterization trick (Kingma and Welling 2013), the log variance term can be ignored. Then we can optimize them by directly minimizing $\mathcal{L}_2$ distance between the corresponding mean value, the middle step loss $\mathcal{L}_{middle}$ can be given by:

$$\left\|(u(\boldsymbol{x}_{n-2t}^T) + v(\boldsymbol{x}_0^T)) - (u(f(\boldsymbol{x}_n^S, \boldsymbol{x}_{n-t}^S)) + v(\boldsymbol{x}_{n-2t}^S))\right\|^2,$$

where f is the determined combination rules for last step, u and v are discussed previously.

Other training Strategies

Target guided diffusion training. As a form of generative models, pure diffusion models do not take task information (e.g., classification, detection, segmentation) into consideration. However, it's natural to combine task information to further improve performance. Inspired by class guided diffusion (Dhariwal and Nichol 2021), which offers a practical solution on conditional diffusion that considers class information (i.e., y), we can introduce y into our formulation:

$$\begin{aligned} \log p(\boldsymbol{x}_0^T|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S, y) &= \log p(\boldsymbol{x}_0^T|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S)\\ +(\log p(y|\boldsymbol{x}_0^T) &- \log p(y|\boldsymbol{x}_n^S, \cdots, \boldsymbol{x}_1^S)), \end{aligned} \tag{12}$$

where the first term can be included in the $\mathcal{L}_{middle}$. The second term measures the distance between the target predicted by the teacher and student accordingly. Here y can be any target information such as class label or next-layer prediction. Assume the weights of next teacher layer is $\boldsymbol{w}_t$, for $\boldsymbol{x}_0^T$ and predicted $\hat{\boldsymbol{x}_0^T}$, we simply use $\mathcal{L}_2$ loss, that is:

$$\mathcal{L}_{guided} = \left\|\boldsymbol{x}_0^T\boldsymbol{w}_t - \hat{\boldsymbol{x}_0^T}\boldsymbol{w}_t\right\|^2. \tag{13}$$

Note that we also observe that some previous works such as (Yang et al. 2021) construct the similar formulation with different motivations.

Shuffle sampling strategy. One issue is that if we strictly follow diffusion weights rule, the last step of student features will dominate large weights such that other features are not fully stimulated to learn target features. We resolve this problem by introducing the shuffle sampling strategy. For each training iteration, we randomly shuffle all student features such that all student features are forced to learn target features from different timesteps. In the inference stage, all student features gain similar abilities, allowing us to assign uniform weights for all student features. The setting of uniform weights is not trivial, since we assume all student features are from the same complex normal distribution, the density function of uniformly weighted of all student features is:

$$p(\frac{1}{m}(\boldsymbol{x}_n^S + \cdots + \boldsymbol{x}_1^S)) = \mathcal{N}(0, \frac{1}{m}\sigma_S^2). \qquad (14)$$

Then we implicitly reduce the variance of the whole prediction. However, we acknowledge that directly using uniform weights for training may not be practical, and thus provide ablation in the experiments to validate our approach.

Whole loss function The whole loss function of our framework is defined as:

$$\mathcal{L} = \mathcal{L}_{ce} + \alpha \sum_{i=1}^{m} \mathcal{L}_{middle} + \beta \mathcal{L}_{guided}, \qquad (15)$$

where $\mathcal{L}_{ce}$ is the conventional cross-entropy loss. m represents the number of re-parameterizing student features. α and β are corresponding weight factor.

Experimental Results

We experiment with different settings varying architectures and datasets, including: **CIFAR-100** (Krizhevsky, Hinton et al. 2009) which consists 32 × 32 images with 100 categories. Training and validation sets are composed of 50k and 10k images. **ImageNet1k** (Deng et al. 2009) which contains over 1280k images with 1000 categories. **ImageNet-100** is a subset of ImageNet which contains roughly 120k images. The training and validation splitting rule is introduced in (Wang and Isola 2020). Our implementation is mainly based on the DKD (Zhao et al. 2022) Review (Chen et al. 2021b) and CRD (Tian, Krishnan, and Isola 2020) with the default training and testing setting.

Main Results

Results on CIFAR-100. Table 2 presents a summary of the results obtained on CIFAR-100 by our proposed teacher and student models with different architecture styles. Previous methods have been categorized into various groups based on the features they utilize. Specifically, methods in the Single Layer group utilize only one layer of feature information. Among them, FitNet (Romero et al. 2014), PKT (Passalis and Tefas 2018), and RKD (Park et al. 2019) utilize middle features, whereas CRD (Tian, Krishnan, and Isola 2020) utilizes representation features, which correspond to the output features of penultimate layers. Our proposed method also has the capability to utilize only single-layer feature information. In this work, we adopt representation features in our single-layer implementation, and our method outperforms all previous methods in the Single-Layer group.

Moreover, we conduct experiments on the Multiple-Layer group and observe that our proposed method achieves competitive results compared to other methods. Since our method mainly distills knowledge at the timestep-level, it is compatible with other methods that use multiple-layer feature information. Furthermore, we perform an ablation study on our proposed method using a simple average strategy. The latter approach utilizes the $\mathcal{L}_2$ distance between the teacher feature and the summation of all student features (i.e., 10) with average weights. Our observations indicate that simply using average weights without other strategies cannot fully stimulate the ability of all student features.

Results on ImageNet-100. We investigate the performance of our proposed method on a larger dataset, ImageNet-100, which is a subset of ImageNet-1k. Following the splitting rule introduced in (Wang and Isola 2020), we conduct experiments in a challenging setting where the teacher is swin-transformer (Liu et al. 2021), and the students belong to different tiny CNN architectures. The challenges in this setting arise from two aspects. Firstly, the architecture gaps are significant. Secondly, the performance gaps are also substantial, given that we utilize an ImageNet-1k pre-trained model.

Table 3 presents the results obtained on ImageNet-100, where we observe some interesting phenomena. Specifically, some conventional feature-level KD methods that have proven effective in the CNN-to-CNN scenario failed to produce similar impressive results in this setting. On the other hand, some logits-level methods, such as KD (Hinton, Vinyals, and Dean 2015) and DKD (Zhao et al. 2022), achieve stable improvements as in the CNN-to-CNN scenario. However, our proposed method exhibits consistently prominent advantages, outperforming all previous methods on all architectures.

Results on ImageNet-1k. We also conduct experiments on ImageNet to verify our method. Top-1 and top-5 accuracies of image classification are reported in Table 4. Kdiffusion^1 indicates for single layer distillation and Kdiffusion^{2}represents multi-layer representation. Our method achieves a consistent improvement, particularly on ResNet50-to-MobileNetV2 setting, highlighting the efficay of our approach in addressing large distribution gaps.

Ablation Studies

We analyze the effectiveness of our method on various aspects. First we ablate structural reparameterization to show that the proposed algorithm is the main reason for performance improvement. Then we ablate the number of structural reparameterization student features. We also provide ablation studies on different stage reparameterization.

Ablation: Influence on structural reparameterization. Structural reparameterization is an effective technique to improve model performance without additional inference cost. By augmenting more features in each layer (Ding et al. 2021), it can boost the performance by a large margin. In this work, we leverage this technique solely in the final layer of each stage, in order to ensure a fair comparison

Distillation Manner	Teacher	ResNet32x4	WRN40-2	VGG13	ResNet50	ResNet32x4
	Acc	79.42	75.61	74.64	79.34	79.42
	Student	ShuffleNetV1	ShuffleNetV1	MobileNetV2	MobileNetV2	ShuffleNetV2
	Acc	70.50	70.50	64.6	64.6	71.82
Logits	KD	74.07	74.83	67.37	67.35	74.45
Logits	DKD	76.45	76.70	69.71	70.35	77.07
Single Layer	FitNet	73.59	73.73	64.14	63.16	73.54
Single Layer	PKT	74.10	73.89	67.13	66.52	74.69
Single Layer	RKD	72.28	72.21	64.52	64.43	73.21
Single Layer	CRD	75.11	76.05	69.73	69.11	75.65
Multiple Layers	AT	71.73	73.32	59.40	58.58	72.73
Multiple Layers	VID	73.38	73.61	65.56	67.57	73.40
Multiple Layers	OFD	75.98	75.85	69.48	69.04	76.82
Multiple Layers	Review	77.45	77.14	70.37	69.89	77.78
Single Layer	Avgerage	75.01	75.32	66.45	67.56	75.46
Single Layer	Kdiffusion	76.62	75.83	69.14	69.20	76.87
Multiple Layer	Kdiffusion	77.90	76.83	69.91	69.95	77.34
+ Target Guide	Kdiffusion	**78.14**	**77.26**	**70.49**	**71.14**	**77.84**

Table 2: Results on CIFAR-100 with the teacher and student having different architectures.

Distillation Manner	Teacher	Swin	Swin	Swin	Swin	Swin
	Acc	94.48	94.48	94.48	94.48	94.48
	Student	MobileNetV2	MobileNetV3	ResNet18	ShuffleNetV1	ShuffleNetV2
	Acc	84.04	84.98	84.42	74.74	76.86
Logits	KD	85.00	86.76	85.12	77.30	79.18
Logits	DKD	85.38	86.86	85.50	77.28	80.02
Single Layer	FitNet	84.86	86.44	85.46	76.58	78.58
Single Layer	PKT	84.32	86.84	85.36	76.72	78.86
Single Layer	SP	85.02	85.90	85.20	76.96	78.86
Single Layer	RKD	78.68	85.06	84.82	76.90	77.48
Single Layer	CRD	83.72	84.94	84.26	73.20	77.88
Multiple Layers	AT	84.70	85.86	85.23	77.26	76.74
Multiple Layers	VID	85.42	86.46	85.12	77.56	79.46
Multiple Layers	Review	84.94	86.94	85.22	76.88	79.92
Single Layer	Kdiffusion	85.88	87.48	86.18	77.90	80.54
Multiple Layer	Kdiffusion	**86.20**	**87.88**	**86.30**	**78.04**	**80.68**

Table 3: Results on ImageNet-100 with the teacher and student having different architectures.

with other results. Figure 3 reveals that in the absence of teacher supervision, the student model's performance suffers despite the incorporation of additional student features. It is worth noting that a careless design of such features may lead to training difficulties and consequently, degraded performance. Our findings strongly support the effectiveness of our approach.

Ablation: Number of re-parameterizing student features. To study the potential impact of increasing the number of structural reparameterization student features supervised by teacher features, we perform an ablation study. As presented in Table 5, indicate that increasing the number of student features leads to improved performance. Despite this, we have to consider practical limitations such as memory constraints, which impose an upper limit on the number of student features that can be utilized. Nevertheless, it would be of interest to explore the upper bound on performance with an increased number of student features.

	Student Stage				Acc
	1	2	3	4	
Feature Num					64.60
				✓	64.22
			✓	✓	63.32
		✓	✓	✓	62.31
	✓	✓	✓	✓	61.96

Figure 3: Ablation study on the influence of the structural reparameterization. The student model is MobileNet-V2 and the baseline performance without any extra student features is 64.60. We re-parameterize 10 student features as default.

Ablation: Re-parameterizing on different stages. Table Table 6 summarizes results on structural reparameterization at different stages. These results conclude that the proposed method performs better on deeper stages that contain

Setting		Teacher	Student	KD	AT	OFD	CRD	Review	DKD	Kdiffusion[1]	Kdiffusion[2]
(a)	Top-1	76.16	68.87	68.58	69.56	71.25	71.37	72.56	72.05	73.48	**73.62**
	Top-5	92.86	88.76	88.98	89.33	90.34	90.41	91.00	91.05	91.62	**91.82**
(b)	Top-1	73.31	69.75	70.66	70.69	70.81	71.17	71.61	71.70	71.68	**72.04**
	Top-5	91.42	89.07	89.88	90.01	89.98	90.13	90.51	90.41	90.48	**90.53**

Table 4: Results on ImageNet. (a) MobileNet as student, ResNet50 as teacher. (b) ResNet18 as student, ResNet34 as teacher.

Teacher	Student	Baseline	Feature Numbers			
			2	4	8	16
Res32x4	Sf1	70.50	74.85	75.96	76.28	76.80
Res50	Mv2	64.60	67.87	68.46	68.91	70.16

Table 5: Ablation study on different number of student features, we use the outputs of penultimate layers (i.e., the final stage outputs), Sf1 and Mv2 represent shufflenet-v1 and mobilenet-v2, respectively.

Teacher	Student	Baseline	Student Stage			
			1	2	3	4
Res32x4	Sf1	70.50	72.78	74.44	77.36	76.62
Res32x4	Sf2	71.82	73.15	75.28	77.32	76.87
Res50	Mv2	64.60	64.69	68.71	69.82	69.20

Table 6: Ablation on different stages.

more channels. From a structural reparameterization perspective, larger channels provide a larger solution space that could be further improved.

More Analysis

Comparison to CRD. Contrastive representation distillation (Tian, Krishnan, and Isola 2020) is a widely used feature-level method for representation-level distillation. However, CRD faces a challenging high-dimension optimization problem and maps both student and teacher representations to a low dimension (e.g., 256 or 128) to optimize the transfer objective smoothly. This approach unavoidably results in information loss, making it difficult to apply the same strategy to the middle feature-level. In contrast, our approach leverages diffusion techniques to optimize the feature in high-dimension, enabling the student model to learn potentially more useful features. Moreover, our method can perform middle feature-level alignment without requiring any additional design. High-dimension feature optimization presents a double-edged sword, offering more supervision information but also potentially introducing negative transfer. Our work presents a promising solution that encourages further exploration in this direction, offering valuable insights and paving the way for future research.

Comparison to Review. Feature-level distillation involves a crucial trade-off that necessitates the inclusion of a bridge module to enable the mapping of the student model to the teacher model in the same dimension. While the student model abandons the bridge module during inference to circumvent any additional inference costs, the design of the bridge module presents a challenging task. An overly powerful bridge module can hinder the acquisition of teacher knowledge, whereas an excessively weak one can lead to unstable training or negative transfer. Therefore, designing an ideal bridge module is a critical consideration for effective feature-level distillation. Review (Chen et al. 2021b) is a classical work that dedicatedly designs two bridge modules that utilize multiple layers of features. As shown in Figure 4, we can observe that our method outperforms Review (Chen et al. 2021b) on both performance and training parameters. Besides this work provides a potential direction for solving the bridge module problem. With more student features, we can safely use a relatively weak bridge module since more student features can reduce the transfer variance.

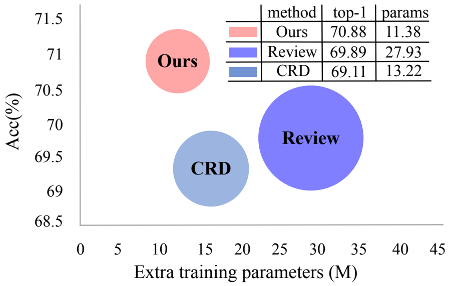


Figure 4: Extra training parameters vs. accuracy on CIFAR-100. We set ResNet50 as the teacher and MobileNetV2 as the student. The table shows the accuracy and number of extra parameters of each method.

Discussion and Conclusion

This paper presents a novel point of view of knowledge distillation with large distribution gaps between teacher and student models. This setting is a potential research direction when the teacher models become larger and larger and other model compression techniques such as model pruning and post-quantization can not afford the retraining cost. In this study, we reveal that the classical $\mathcal{L}_2$ loss may incur negative transfer when confronting large distribution gaps. To solve the problem, we present a novel transfer method based on re-parameterizing the diffusion reverse process. The insight is the reverse target can be decoupled into two parts, then different re-parameterizing features can take charge of different parts and combine linearly in the end. This method does not require sophisticated bridge modules to align distributions, instead naturally combining extra training parameters without additional inference cost. Experiments have demonstrated consistent improvements in various tasks.

References

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *CVPR*, 9163–9171.

Buciluǎ, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *SIGKDD*, 535–541.

Chen, D.; Mei, J.-P.; Zhang, Y.; Wang, C.; Wang, Z.; Feng, Y.; and Chen, C. 2021a. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7028–7036.

Chen, P.; Liu, S.; Zhao, H.; and Jia, J. 2021b. Distilling knowledge via knowledge review. In *CVPR*, 5008–5017.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255. Ieee.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34: 8780–8794.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13733–13742.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; and Choi, J. Y. 2019. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1921–1930.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. In *NeurIPS*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.

Ji, M.; Heo, B.; and Park, S. 2021. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 7945–7952.

Kingma.; et al. 2013. Auto-encoding variational bayes. *arXiv*.

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.

Kullback, S.; and Leibler, R. A. 1951. On information and sufficiency. volume 22, 79–86. JSTOR.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *CVPR*, 3967–3976.

Passalis, N.; and Tefas, A. 2018. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, 268–284.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 234–241. Springer.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive representation distillation. In *ICLR*.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 9929–9939. PMLR.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2018. KDGAN: Knowledge Distillation with Generative Adversarial Networks. In *NeurIPS*, 783–794.

Yang, J.; Martinez, B.; Bulat, A.; Tzimiropoulos, G.; et al. 2021. Knowledge distillation via softmax regression representation learning. International Conference on Learning Representations (ICLR).

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 4133–4141.

Zagoruyko, S.; and Komodakis, N. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; and Liang, J. 2022. Decoupled Knowledge Distillation. In *CVPR*, 11953–11962.