Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Multi-exit self-distillation with appropriate teachers*

Wujie SUN†, Defang CHEN, Can WANG†‡, Deshi YE, Yan FENG, Chun CHEN

College of Computer Science and Technology, Zhejiang University, Hangzhou 310000, China

†E-mail: sunwujie@zju.edu.cn; wcan@zju.edu.cn

Received Dec. 16, 2022; Revision accepted July 4, 2023; Crosschecked Feb. 22, 2024

Abstract: Multi-exit architecture allows early-stop inference to reduce computational cost, which can be used in resource-constrained circumstances. Recent works combine the multi-exit architecture with self-distillation to simultaneously achieve high efficiency and decent performance at different network depths. However, existing methods mainly transfer knowledge from deep exits or a single ensemble to guide all exits, without considering that inappropriate learning gaps between students and teachers may degrade the model performance, especially in shallow exits. To address this issue, we propose Multi-exit self-distillation with Appropriate TEachers (MATE) to provide diverse and appropriate teacher knowledge for each exit. In MATE, multiple ensemble teachers are obtained from all exits with different trainable weights. Each exit subsequently receives knowledge from all teachers, while focusing mainly on its primary teacher to keep an appropriate gap for efficient knowledge transfer. In this way, MATE achieves diversity in knowledge distillation while ensuring learning efficiency. Experimental results on CIFAR-100, TinyImageNet, and three fine-grained datasets demonstrate that MATE consistently outperforms state-of-the-art multi-exit self-distillation methods with various network architectures.

Key words: Multi-exit architecture; Knowledge distillation; Learning gap
https://doi.org/10.1631/FITEE.2200644 **CLC number:** TP181

1 Introduction

In recent years, the number of mobile and edge devices has been increasing rapidly, and the need to deploy small and compact deep neural networks (DNNs) on these devices is becoming more and more urgent due to considerations such as data privacy and computational resources (Schwartz et al., 2020). Thanks to the development of multi-exit architectures (Teerapittayanon et al., 2016; Huang et al., 2018), devices can now store the whole model but use only part of it during inference based on its operational status and available resources. This provides an opportunity to achieve a good trade-off between efficiency and accuracy, and inspires us to further improve the inference accuracy of each exit given the constrained efficiency. A feasible approach to this goal is knowledge distillation (Hinton et al., 2015).

Traditionally, knowledge distillation (Ba and Caruana, 2014; Hinton et al., 2015; Chen et al., 2022) improves the training of a small student model by transferring knowledge from a large pre-trained teacher model. Because the teacher pre-training involves substantial resources, online knowledge distillation (Lan et al., 2018; Zhang Y et al., 2018; Anil et al., 2020; Chen et al., 2020) was proposed as training a group of student models simultaneously to transfer group knowledge into each student model. Online knowledge distillation essentially leverages virtual teachers with superior learning capability formed by the student ensembles. To further reduce training cost, self-distillation (Furlanello et al., 2018; Xu and Liu, 2019; Zhang LF et al., 2019; Ji et al., 2021) uses the same network for both the

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (No. U1866602) and the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study, China (No. SN-ZJU-SIAS-001)
ORCID: Wujie SUN, https://orcid.org/0000-0001-7739-3517; Can WANG, https://orcid.org/0000-0002-5890-4307

teacher and student models by distilling knowledge from the network itself. Recently, multi-exit architecture has been employed in self-distillation by distilling knowledge from different exits to improve the model capability (Phuong and Lampert, 2019; Zhang LF et al., 2019; Lee and Lee, 2021). An additional benefit of multi-exit self-distillation is that the inference accuracy of each exit can be improved without incurring extra computational cost, as illustrated in Fig. 1.

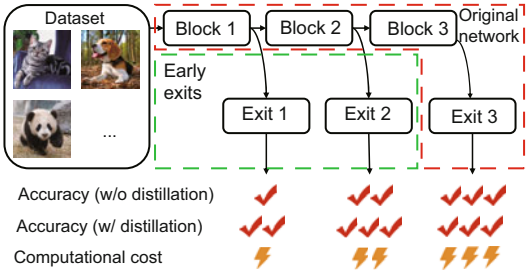

Fig. 1 Multi-exit self-distillation enabling adaptive inference with high accuracy without incurring extra computational cost (w/o: without; w/: with)

However, existing multi-exit self-distillation works (Phuong and Lampert, 2019; Sun et al., 2019; Zhang LF et al., 2019; Lee and Lee, 2021) fail to address two issues in training, which hinders further performance improvements. First, existing works (Phuong and Lampert, 2019; Zhang LF et al., 2019) have not effectively used the knowledge of multiple exits. They attempt only to transfer knowledge from deep exits into shallow ones, leaving deep exits less efficiently trained. However, predictions from shallow exits can serve as regularization on deep exits to improve the model learning capability (Yuan et al., 2020). Model learning can also benefit from diversity of knowledge by learning from different exits (Chen et al., 2020). Therefore, leveraging knowledge from multiple exits is expected to effectively improve model training. More importantly, all existing methods (Phuong and Lampert, 2019; Sun et al., 2019; Zhang LF et al., 2019; Lee and Lee, 2021) ignore the learning capacity variations among different exits; that is, models of different sizes and complexities exhibit different capabilities in capturing knowledge patterns of different granularities. Existing studies (Mirzadeh et al., 2020; Shi et al., 2021) have shown that an inappropriate learning gap, or mismatch in learning capabilities between teachers and students, will negatively impact the knowledge transfer between them. By limiting the learning gap between the teacher and student, these works successfully improve student performance in traditional knowledge distillation. However, learning gap is rarely considered in online knowledge distillation and self-distillation.

When we focus on acquiring better ensemble teacher in multi-exit self-distillation, the learning gap between students and teachers can become large (especially for shallow exits), and lead to performance degradation. However, when we focus on narrowing the learning gap, we may not be able to acquire valuable teacher knowledge. To address these issues, we propose a novel method called Multi-exit self-distillation with Appropriate TEachers (MATE). We provide students with an equal number of teachers which are obtained by different weighted combinations of exits' logits. Each student is required to learn mainly from its primary teacher whose knowledge is generally more appropriate for student learning. To prevent students from becoming overly focused on their primary teacher and failing to capture the rest valuable knowledge, students are asked to acquire some knowledge from other teachers as well. We use a neural network to calculate the weights for composing the teachers, and generate diverse and appropriate knowledge for each student by using a novel loss function. Our contributions are summarized as follows:

1. We propose a multi-exit self-distillation method that boosts model performance using its intrinsic knowledge and achieves adaptive inference.

2. We stress the importance of providing diverse and appropriate teacher knowledge for different exits, which is ignored in previous works.

3. Experimental results on CIFAR-100, TinyImageNet, and fine-grained datasets demonstrate that our method consistently outperforms other methods with various network architectures.

2 Related works

2.1 Online knowledge distillation

Traditional knowledge distillation uses a pre-trained teacher model to help the training of a student model (Ba and Caruana, 2014; Hinton et al., 2015; Ahn et al., 2019; Chen et al., 2021, 2022; Tian et al., 2022). However, pre-trained teacher

models are not always available due to privacy and resource constraints. Therefore, online knowledge distillation manages to use multiple individual student models with the same architecture for training (Zhang Y et al., 2018; Anil et al., 2020). Because such network-based methods increase computational costs, branch-based methods share the shallow model blocks to further reduce training costs (Lan et al., 2018; Chen et al., 2020). Generally, a weighted ensemble of student logits is viewed as the teacher, and each student learns from it to get consistent knowledge (Lan et al., 2018); alternatively, each student can choose to acquire different knowledge from other students to better increase the peer diversity (Zhang Y et al., 2018; Anil et al., 2020; Chen et al., 2020). Although online knowledge distillation is free from the constraints of pre-trained teachers, using such methods still requires enormous computational resources.

2.2 Self-distillation

Self-distillation explores a model's intrinsic knowledge to improve performance (Yang et al., 2019a, 2019b; Zhang LF et al., 2019; Yuan et al., 2020). With the help of results in different training phases, old predictions can be used to guide new ones (Furlanello et al., 2018; Deng X and Zhang, 2021). Data augmentation (Xu and Liu, 2019) or the correlation between samples (Yun et al., 2020; Ge et al., 2021) can be used to achieve self-distillation. By combining the multi-exit architecture (Teerapittayanon et al., 2016; Huang et al., 2018) with self-distillation, model performance can be boosted and adaptive inference can be achieved. Be your own teacher (BYOT) (Zhang LF et al., 2019) and distillation-based training (DBT) (Phuong and Lampert, 2019) use knowledge from the deepest exit to guide shallower exits. The above methods leave the deep exit without efficient guidance. Therefore, in deeply-supervised knowledge synergy (DKS) (Sun et al., 2019), each exit needs to draw on knowledge from other exits, whereas in exit-ensemble distillation (EED) (Lee and Lee, 2021), the average of exits' logits acts as the teacher to guide all exits. However, all methods ignore the learning capacity variations (Mirzadeh et al., 2020) among exits, which may result in some exits not being efficiently improved.

2.3 Learning gaps in knowledge distillation

In recent years, scholars have found that better teachers do not always teach better students (Mirzadeh et al., 2020). Therefore, attempts have been made to control the learning gap between teacher and student models, but mostly in traditional knowledge distillation. For example, given ResNet50 and ResNet18 (He et al., 2016) as the teacher and student respectively, some works (Mirzadeh et al., 2020; Son et al., 2021) create teacher assistants which are shallower than the teacher but deeper than the student to narrow the learning gap between each distillation pair. However, such methods consume a large number of computational resources. Instead of using the knowledge from a pre-trained teacher, a teacher's knowledge from its different training epochs can be used to guide the student at different training stages (Jin et al., 2019), which is easier to learn. By training teacher and student simultaneously and limiting the gap between them (Shi et al., 2021), model performance can also be further improved. However, these works have studied only the learning gap problem in traditional knowledge distillation, ignoring the possibility of this happening in online knowledge distillation and self-distillation. As a result, these methods are not suitable for multi-exit self-distillation.

3 Methodology

Because our MATE is the integrated framework of multi-exit architecture and self-distillation, we will first briefly introduce multi-exit architecture and existing self-distillation methods in Section 3.1, and then describe our method in Section 3.2.

3.1 Preliminary

3.1.1 Multi-exit architecture

As shown in Fig. 1, a multi-exit architecture attaches multiple exits at different depths to achieve adaptive inference with different computational resources. Because the last exit is already included in the original network, only those early exits are newly added. Following the previous work (Zhang LF et al., 2019), each early exit is designed to include multiple convolutional layers, batch normalization (BN) layers, and activation layers, so that the output feature

of each block is resized to the same dimensions with the final output feature of the original network. After that, a classifier is used to generate the logits $\boldsymbol{z}^i$, and the prediction $\boldsymbol{p}^i$ is calculated as

$$p_k^i = \sigma(z_k^i) = \frac{\mathrm{e}^{z_k^i}}{\sum_{j=1}^{K} \mathrm{e}^{z_j^i}}, \tag{1}$$

where i denotes the exit index, k denotes the class index, K denotes the number of classes, and $\sigma(\cdot)$ is the Softmax function.

For BranchyNet (Teerapittayanon et al., 2016), the training target is achieved by minimizing the loss function between $\boldsymbol{p}^i$ and label $\boldsymbol{y}$:

$$L = \sum_{i=1}^{M} L_{\mathrm{CE}}(\boldsymbol{p}^i, \boldsymbol{y}), \tag{2}$$

where L_{CE} denotes the cross-entropy loss, and M is the number of exits and often equals 4 for popular convolutional neural networks (CNNs) such as VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2016), and DenseNet (Huang et al., 2017). BranchyNet does not involve any distillation method, and can be viewed as the baseline for multi-exit self-distillation methods.

3.1.2 Multi-exit self-distillation

Because knowledge distillation is a powerful tool, attempts have been made to combine it and multi-exit architecture to improve model performance and achieve adaptive inference (Phuong and Lampert, 2019; Sun et al., 2019; Zhang LF et al., 2019; Lee and Lee, 2021). In this study, we consider only logit distillation and summarize the loss function of existing multi-exit self-distillation methods as

$$\begin{aligned} L = &\sum_{i=1}^{M} L_{\mathrm{CE}}(\boldsymbol{p}^i, \boldsymbol{y}) \\ &+ \sum_{i=1}^{M} L_{\mathrm{KL}}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^i}{T}\right)_{\dagger}\right), \end{aligned} \tag{3}$$

where $\hat{\boldsymbol{z}}^i$ denotes the i^{th} teacher logits (each method uses different $\hat{\boldsymbol{z}}^i$), and L_{KL} is the Kullback–Leibler (KL) divergence. The stop-gradient operation is represented by "†." Logits $\boldsymbol{z}$ are divided by temperature T for better distillation (Hinton et al., 2015). We set T to 3 for all methods because it is a common setting in knowledge distillation (Hinton et al., 2015; Lan et al., 2018; Chen et al., 2020).

The main difference in existing multi-exit self-distillation methods is the way that knowledge is transferred among exits. We illustrate three major styles of multi-exit knowledge transfer in Fig. 2: (1) learning from the deepest exit; (2) learning from all other exits; (3) learning from an ensemble of all exits.

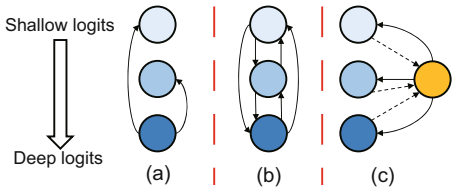

Fig. 2 Comparison of various knowledge transfer approaches: (a) BYOT and DBT; (b) DKS; (c) EED (The solid lines indicate the knowledge transfer and the dashed lines indicate the knowledge ensemble)

BYOT (Zhang LF et al., 2019) and DBT (Phuong and Lampert, 2019) use knowledge from the deepest exit to guide the training of the shallower exits. However, the deepest exit in these methods tends to be insufficiently trained. In practice, the deepest exit is not necessarily the best exit. As the simple experiment in Fig. 3 shows, the accuracy of a shallower exit could be better than that of the deepest exit; i.e., exit 4 exhibits the lowest performance among all the exits in early training stages. Even in late training stages, exit 3 can outperform exit 4, as shown later in Section 4.4.3, when training on the dataset CUB-200-2011. Meanwhile, transferring knowledge only from the deepest exit will leave the useful knowledge from other exits unexploited, leading to an inferior model performance. To leverage knowledge from different exits, DKS (Sun et al., 2019) enables mutual learning between different exits. EED (Lee and Lee, 2021) forms an ensemble teacher from all exits in hope that the ensemble will boast superior performance. Although knowledge transferred from different exits improves model training, the learning capacity variations among exits are ignored in these methods, hurting the accuracy improvement.

3.2 Multi-exit self-distillation with appropriate teachers

To improve knowledge transfer among different exits, MATE exploits knowledge from all exits while respecting the learning gap between teachers and students. The framework of MATE is shown in

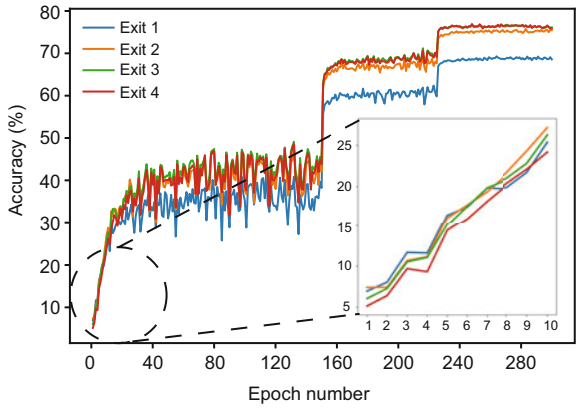

Fig. 3 Top-1 test accuracy curve during training with CIFAR-100 and VGG16 (BranchyNet (Teerapittayanon et al., 2016) is used to remove the impact of distillation on the accuracy)

Fig. 4. The i^{th} teacher acts as the primary teacher for the i^{th} exit, and also acts as one of the secondary teachers for other exits. Here, the term "primary" suggests that this teacher imparts knowledge more fittingly than others, serving as the main source of learning for its corresponding student. Instead of using the fixed weights as in existing methods, we use a weight network to calculate the ensemble weights as shown in Fig. 4c. In this learning framework, we propose a novel loss function to achieve two-way learning and narrow the distillation gap.

3.2.1 Weight network

The weight network takes the resized flattened features $\boldsymbol{F}^i \in \mathbb{R}^C$ from all exits as the input, and outputs the ensemble weights for obtaining the teachers. i and C are the exit index and number of channels, respectively. Because the parameters of the weight network are constantly updated, different teachers can be generated using different ensemble weights even if the inputs are constant, allowing students to learn more diverse knowledge. One commonly used technique to compute the ensemble weights is to use a neural network called gate (Lan et al., 2018), which includes one fully connected (FC) layer, one BN layer, and one rectified linear unit (ReLU) layer. However, only one single ensemble teacher can be obtained using one gate. As we have stressed, because exits show various learning capacities, a single teacher does not have the capability to provide all students with appropriate knowledge. Multiple gate networks can be used, but gate mechanism usually costs more training time than self-attention (Vaswani et al., 2017) in our experiments. Therefore, we choose to use self-attention as the weight network to generate multiple teachers.

Specifically, we use two networks called query $\boldsymbol{\theta}_{\text{Q}}$ and key $\boldsymbol{\theta}_{\text{K}}$, each consisting of a single FC layer, to map features into another dimensional space. Assuming that the ensemble weight matrix is represented as $\boldsymbol{w} \in \mathbb{R}^{M\times M}$, the ensemble weight of the i^{th} exit at the j^{th} teacher is calculated as

$$w_{ji} = \frac{\mathrm{e}^{\boldsymbol{\theta}_{\text{Q}}(\boldsymbol{F}^j)\boldsymbol{\theta}_{\text{K}}^{\text{T}}(\boldsymbol{F}^i)}}{\sum_{m=1}^{M}\mathrm{e}^{\boldsymbol{\theta}_{\text{Q}}(\boldsymbol{F}^j)\boldsymbol{\theta}_{\text{K}}^{\text{T}}(\boldsymbol{F}^m)}}, \tag{4}$$

$$\begin{cases} \boldsymbol{w} = \text{Weight network}(\boldsymbol{F}), \\ \displaystyle\sum_{i=1}^{M} w_{ji} = 1, \\ \hat{\boldsymbol{z}}^j = \displaystyle\sum_{i=1}^{M} w_{ji}\boldsymbol{z}^i. \end{cases} \tag{5}$$

Note that features $\boldsymbol{F}$ and logits $\boldsymbol{z}$ are detached at this point to avoid affecting the overall network during back-propagation. The loss to update the weight network will be discussed in the next subsection.

3.2.2 Loss function

To achieve our goal, we carefully design the loss function. First, each exit should accept knowledge from all teachers, while learning mainly from its primary teacher, which can be represented as

$$\begin{aligned} L_{\text{exit-}i} =& L_{\text{CE}}(\boldsymbol{p}^i, \boldsymbol{y}) \\ &+ \sum_{j=1}^{M} L_{\text{KL}}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^j}{T}\right)_{\dagger}\right) \\ &+ \alpha_i L_{\text{KL}}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^i}{T}\right)_{\dagger}\right). \end{aligned} \tag{6}$$

We call this student loss, and it is used to update the overall network (excluding the weight network). α_i is used to adjust the extent to which the student learns from its primary teacher. When $\alpha_i = 0$, each student learns equally from M teachers. When α_i gets larger, the student tends to focus more on its primary teacher.

In addition, the teacher should provide knowledge that is more appropriate for its primary student. To achieve this, the knowledge provided by the i^{th} teacher should contain more knowledge from the i^{th}

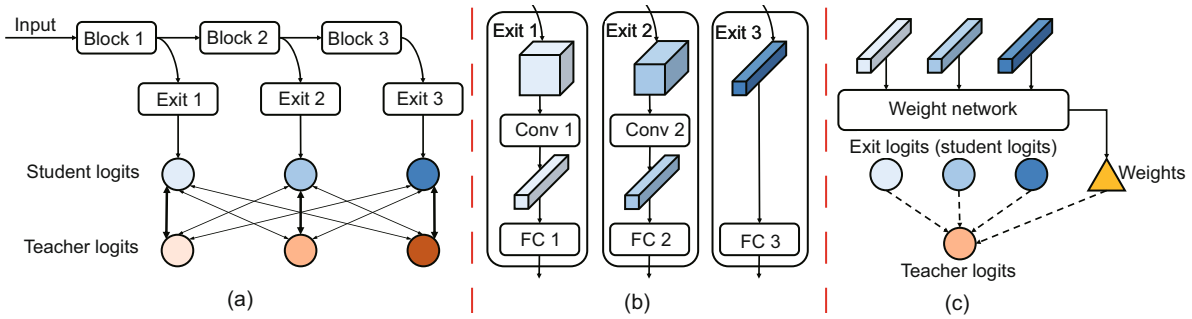

Fig. 4 Framework of MATE: (a) the overall framework, where two-way learning is required between students and teachers; students gain knowledge from teachers, and teachers dynamically adjust ensemble weights based on students' output; (b) exit architecture, where each exit resizes the block's output feature to match the dimensions of the last block's output feature, which is then inputted to the fully connected classifier to generate logits; (c) weight network, where a weight network based on self-attention is used to obtain teacher logits. It takes resized features as the input, and outputs the weights. Teacher logits are computed using weights and logits from all exits. Cuboids indicate the features and circles indicate the logits

student and its adjacent exits. If so, teachers need to learn from students' exit logits to generate appropriate knowledge. To achieve this, we use the following loss function:

$$\begin{aligned} L_{\text{teacher-}i} = &\sum_{j=1}^{M} L_{\text{KL}}\left(\sigma\left(\frac{\boldsymbol{z}^j}{T}\right)_{\dagger}, \sigma\left(\frac{\hat{\boldsymbol{z}}^i}{T}\right)\right) \\ &+ \alpha_i' L_{\text{KL}}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right)_{\dagger}, \sigma\left(\frac{\hat{\boldsymbol{z}}^i}{T}\right)\right). \end{aligned} \tag{7}$$

We call this teacher loss, and it is used to update the weight network. When $\alpha_i' = 0$, each teacher will not bias towards any student. When $\alpha_i' > 0$, the teacher's ensemble is more similar to its primary student, thus facilitating the student learning by reducing the gap.

We can combine Eqs. (6) and (7) to form an overall loss function:

$$\begin{aligned} L = &\sum_{i=1}^{M} L_{\text{CE}}(\boldsymbol{p}^i, \boldsymbol{y}) \\ &+ \sum_{i=1}^{M}\sum_{j=1}^{M} L_{\text{KL}}^{*}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^j}{T}\right)\right) \\ &+ \alpha \sum_{i=1}^{M} L_{\text{KL}}^{*}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^i}{T}\right)\right), \end{aligned} \tag{8}$$

where $L_{\text{KL}}^{*}(A, B)$ represents the two-way learning of A and B, which means that neither A nor B is fixed during training. We use a single α to reduce the workload of tuning the hyper-parameters. We find that it achieves satisfactory results.

The parameter α plays a pivotal role in the MATE framework. It effectively manages the equilibrium between efficient learning (which is characterized by an appropriate learning gap) and the diversity of available knowledge sources. When α is too small, the gaps cannot be effectively narrowed. When α is far too large, the i^{th} teacher's logits will be overly similar to the i^{th} exit, causing the model performance drop because students can learn little from the teachers.

4 Experiments

We evaluated multiple methods with five different types of CNNs: VGG (Simonyan and Zisserman, 2015), ResNet (He et al., 2016), ResNeXt (Xie et al., 2017), WideResNet (WRN) (Zagoruyko and Komodakis, 2017), and DenseNet (Huang et al., 2017) on five image classification datasets, including CIFAR-100 (Krizhevsky and Hinton, 2009), TinyImageNet (Le and Yang, 2015), CUB-200-2011 (Wah et al., 2011), Stanford Dogs (Khosla et al., 2011), and FGVC-Aircraft (Maji et al., 2013). We also conducted distillation with MSDNet (Huang et al., 2018) in budget mode, which is shown in Section 4.7.6.

4.1 Datasets

1. CIFAR-100. CIFAR-100 consists of 32 × 32 RGB images of 100 classes. The training set contains 50 000 images and the test set contains 10 000 images.

2. TinyImageNet. TinyImageNet is an image classification dataset extracted from ImageNet (Deng J et al., 2009) with 200 classes. Each class contains 500 training images, 50 validation images, and 50 test images. We use the validation images to test the model performance.

3. Fine-grained datasets. CUB-200-2011, Stanford Dogs, and FGVC-Aircraft are fine-grained image classification datasets, containing 11 788 images of 200 bird species, 20 580 images of 120 dog breeds, and 10 000 images of 100 aircraft model variants, respectively.

4.2 Compared methods

We compare MATE with the following methods: BranchyNet (Teerapittayanon et al., 2016), BYOT (Zhang LF et al., 2019), DKS (Sun et al., 2019), and EED (Lee and Lee, 2021). BranchyNet is used to demonstrate the effectiveness of distillation, and the latter three are state-of-the-art multi-exit self-distillation methods. In addition, a baseline method is compared, where the model is trained from scratch without multi-exit architecture or knowledge distillation. For MATE, if not specified, α is set to 3.

The floating point operations (FLOPs) for different models and exits are reported in Table 1. Because the inference FLOPs are the same for all multi-exit methods, we provide only the accuracy comparison. These experimental results with mean±STD are based on three runs (here, STD is short for standard deviation). We use bold texts to denote the best results and underlined texts to indicate the second-best results.

4.3 Implementation details

Following the standard training procedure and data augmentation, we trained all networks using stochastic gradient descent (SGD) with momentum 0.9 and weight decay 5e-4 on all datasets. The batch size was set to 128 for all models trained on CIFAR-100 and TinyImageNet. Because the input dimension of fine-grained data is higher, we set the batch size to 32 when training on fine-grained datasets due to memory restrictions. The training epoch was set to 200 and 300 when training on fine-grained datasets and other datasets, respectively. The initial learning rate was 0.1 in all cases, and the learning rate was divided by 10 at one-half and three-quarters of the training epochs.

The input data size of CIFAR-100 and TinyImageNet was 32×32 and the input data size of fine-grained datasets was 224×224.

When implementing ResNet, WideResNet, and DenseNet, for 32×32 input, the first 7×7 convolutional kernel of stride 2 and padding 3 was changed to a 3×3 convolutional kernel of stride 1 and padding 1, and the first max pooling layer was removed, which is commonly used in practice. The rest of the original network's architecture is consistent with the implementation in torchvision.

4.4 Results

4.4.1 CIFAR-100

We summarize the top-1 test accuracy of different models and methods on CIFAR-100 in Table 2. The results from all exits are reported when the last exit achieves the optimal test accuracy.

It can be seen from Table 2 that MATE achieves considerable improvements in all cases, especially for the shallow exits. This indicates the necessity of providing exits with appropriate knowledge, because shallow exits probably cannot absorb complex knowledge well due to the performance gap between shallow students and teachers. As MATE can enhance the performance of the shallow exits, the highly performing low-level blocks are able to extract more discriminative features, further aiding in the training of higher-level blocks.

As shown in Table 2, due to the lack of effective guidance for the last exit, BYOT's exit 4 failed to achieve sub-optimal accuracy in all cases. Because knowledge from other exits was used to guide current exits, DKS achieved at least one sub-optimal accuracy at all exits. EED did not achieve a sub-optimal accuracy at exit 1, possibly due to the complexity of the ensemble knowledge, which results in the inability of shallow exits to learn effectively. This indicates that selecting a better teacher to guide all exits is not enough, emphasizing the importance of selecting appropriate teachers for each exit.

In some cases, such as when the model is WRN-14-4, shallower exits can outperform deeper exits, indicating that overly complex models tend to overfit during training. In such cases, using the deepest exit to guide shallower exits may not be as effective, and MATE can mitigate this issue.

Table 1 Comparison of floating point operations (FLOPs) on different models and exits

Input size	Model	FLOPs			
		Exit 1	Exit 2	Exit 3	Exit 4
32×32	VGG16	99.19M	192.79M	286.39M	314.18M
	ResNet18	163.19M	294.05M	425.02M	556.03M
	ResNet50	374.76M	659.48M	1087.25M	1300.99M
	ResNeXt50	364.14M	668.10M	1116.04M	1348.20M
224×224	ResNet18	614.68M	1015.44M	1416.52M	1817.76M

Table 2 Top-1 test accuracy comparison on CIFAR-100

Model	Exit No.	Accuracy (%)					
		Baseline	BranchyNet	BYOT	DKS	EED	MATE
VGG16	1	–	68.87±0.42	69.43±0.27	69.63±0.40	69.50±0.33	**70.79±0.08**
	2	–	75.26±0.25	75.21±0.26	75.44±0.15	75.49±0.20	**75.92±0.17**
	3	–	76.85±0.07	76.45±0.06	76.75±0.30	77.28±0.12	**77.60±0.17**
	4	75.51±0.21	76.86±0.07	76.60±0.12	76.86±0.16	77.23±0.21	**77.60±0.16**
ResNet18	1	–	74.91±0.34	75.21±0.27	75.13±0.17	75.12±0.14	**76.18±0.20**
	2	–	77.45±0.22	77.91±0.52	77.84±0.07	78.18±0.13	**78.88±0.32**
	3	–	80.52±0.04	80.58±0.29	80.87±0.05	80.72±0.14	**81.33±0.32**
	4	79.82±0.11	81.57±0.06	81.42±0.16	81.74±0.26	81.65±0.24	**82.19±0.40**
ResNeXt50	1	–	79.24±0.38	80.67±0.28	80.23±0.48	80.19±0.09	**81.72±0.16**
	2	–	80.93±0.35	82.02±0.42	82.19±0.33	81.69±0.20	**82.86±0.15**
	3	–	83.53±0.52	83.91±0.36	84.28±0.04	84.15±0.21	**84.57±0.25**
	4	82.28±0.20	83.43±0.45	83.53±0.21	84.21±0.17	84.01±0.06	**84.66±0.13**
WRN-14-4	1	–	78.69±0.45	80.05±0.16	79.80±0.48	79.75±0.53	**80.74±0.23**
	2	–	78.89±0.24	79.78±0.15	79.95±0.16	79.96±0.05	**80.74±0.22**
	3	–	78.76±0.08	79.17±0.17	79.52±0.17	79.44±0.16	**80.34±0.28**
	4	78.99±0.18	79.36±0.16	79.58±0.08	80.35±0.08	80.04±0.19	**81.25±0.06**
DenseNet121	1	–	73.64±0.81	75.13±0.67	74.85±0.18	74.63±0.15	**75.85±0.39**
	2	–	78.38±0.69	79.45±0.30	79.13±0.41	79.03±0.47	**79.91±0.29**
	3	–	82.27±0.23	82.79±0.45	82.29±0.28	82.83±0.17	**82.92±0.12**
	4	82.24±0.24	83.00±0.16	83.07±0.20	83.23±0.47	83.46±0.32	**83.59±0.22**
WRN-50-2	1	–	78.96±1.03	80.69±0.26	79.66±0.53	79.85±0.42	**81.16±0.55**
	2	–	80.66±0.41	82.12±0.33	81.60±0.57	81.58±0.39	**82.13±0.80**
	3	–	83.26±0.26	84.06±0.13	83.99±0.20	**84.22±0.43**	84.16±0.63
	4	82.08±0.09	83.77±0.16	84.15±0.28	84.25±0.23	84.49±0.35	**84.52±0.33**

Bold texts denote the best results, and underlined texts indicate the second-best results

4.4.2 TinyImageNet

As shown in Table 3, MATE still achieves the optimal accuracy in all cases on this more challenging dataset. Similar to Table 2, the baseline method consistently performs worst in Table 3, and BranchyNet performs worst among all multi-exit methods due to the absence of knowledge distillation.

4.4.3 Fine-grained datasets

As shown in Fig. 5, MATE outperforms other methods overall. Though BYOT outperforms MATE at exit 3 on CUB-200-2011, the margin is small and there are still large gaps between these two methods at other exits. Because the baseline method has only one exit and performs much worse than other methods, we do not plot it in Fig. 5. The accuracy of the baseline method's final exit on CUB-200-2011, Stanford Dogs, FGVC-Aircraft is 57.64%, 63.78%, and 73.68%, respectively. Note that using the multi-exit architecture greatly improves the final model performance compared to the baseline, indicating its effectiveness.

Table 3 Top-1 test accuracy comparison on TinyImageNet

Model	Exit No.	Accuracy (%)					
		Baseline	BranchyNet	BYOT	DKS	EED	MATE
VGG16	1	–	46.25±0.29	46.75±0.53	46.52±0.25	46.37±0.50	**47.64±0.10**
	2	–	50.91±0.26	50.77±0.38	51.15±0.06	50.97±0.25	**51.70±0.51**
	3	–	52.33±0.35	52.30±0.74	52.77±0.15	52.47±0.24	**53.29±0.45**
	4	51.11±0.12	52.30±0.42	52.36±0.44	52.84±0.09	52.51±0.37	**53.37±0.35**
ResNet18	1	–	51.10±0.40	51.93±0.24	51.56±0.30	52.13±0.57	**52.45±0.16**
	2	–	54.43±0.50	55.08±0.56	54.96±0.31	54.84±0.37	**55.30±0.42**
	3	–	58.16±0.38	58.88±0.51	58.65±0.46	58.60±0.20	**58.92±0.19**
	4	57.11±0.25	59.21±0.13	59.55±0.31	60.19±0.26	59.80±0.27	**60.23±0.10**
ResNeXt50	1	–	55.66±0.71	57.28±0.33	57.30±0.19	56.93±0.44	**58.47±0.27**
	2	–	58.06±0.41	59.17±0.08	58.86±0.09	58.39±0.32	**60.38±0.16**
	3	–	61.33±0.40	62.47±0.33	61.99±0.38	61.63±0.12	**62.66±0.38**
	4	60.13±0.27	61.56±0.41	62.12±0.17	62.32±0.19	62.07±0.08	**63.11±0.10**
WRN-14-4	1	–	55.74±0.26	56.08±0.35	55.66±0.23	56.15±0.29	**57.62±0.62**
	2	–	55.28±0.14	55.70±0.15	55.74±0.11	56.05±0.51	**57.50±0.34**
	3	–	54.73±0.39	54.87±0.05	55.09±0.28	55.48±0.32	**56.82±0.57**
	4	54.14±0.27	55.39±0.22	55.07±0.21	55.93±0.12	56.16±0.39	**57.67±0.19**
DenseNet121	1	–	52.01±0.77	53.35±0.08	53.15±0.40	52.57±0.50	**54.61±0.85**
	2	–	56.85±0.23	57.77±0.77	57.27±0.97	56.92±0.12	**58.48±0.31**
	3	–	60.65±0.27	61.17±0.20	61.39±0.55	61.06±0.09	**62.02±0.38**
	4	59.51±0.24	61.20±0.42	61.01±0.26	61.91±0.44	61.69±0.24	**62.84±0.17**
WRN-50-2	1	–	55.71±0.42	57.77±0.22	57.09±0.48	56.83±0.55	**58.60±0.28**
	2	–	57.60±0.69	59.23±0.50	58.92±0.61	58.13±0.24	**59.54±0.29**
	3	–	61.81±0.16	62.84±0.32	62.88±0.23	62.07±0.32	**62.90±0.22**
	4	59.69±0.37	61.82±0.34	62.43±0.22	62.81±0.24	62.36±0.31	**63.10±0.33**

Bold texts denote the best results, and underlined texts indicate the second-best results

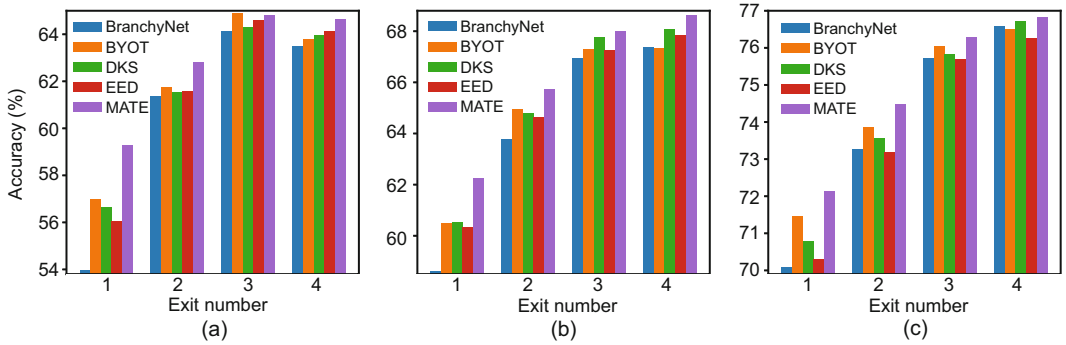

Fig. 5 Top-1 test accuracy comparison on the fine-grained datasets with ResNet18: (a) CUB-200-2011; (b) Stanford Dogs; (c) FGVC-Aircraft

4.5 Learning gaps

Though the optimal learning gaps are hard to determine (Mirzadeh et al., 2020), some principles should be satisfied: the knowledge provided by the teachers should be superior to what the students had already mastered. Furthermore, the learning gap should be stable. In our work, we use the top-1 test accuracy to measure the knowledge superiority, and view the test accuracy difference between the teacher and the student as the learning gap. Note that a positive difference value means that the teacher's knowledge is superior to that of the student and a smaller standard deviation means a more stable learning gap. We calculated the mean and standard deviation based on 300 epochs, and the results are shown in Table 4. As we can see, BYOT fails to provide good knowledge in exit 3, which might explain

why it has poor performance in Table 2. MATE has the most stable learning gaps in all four exits among compared methods to better improve the performance. The trends in the learning gap across training epochs are shown in Fig. 6.

4.6 Ensemble weights

To provide a better understanding of the ensemble weights for each teacher in MATE, we show the weights during training on CIFAR-100 with VGG16 in Fig. 7. Weights at training epochs 1, 11, 51, and 300 are reported. We also check the ensemble weights when training a ResNet-like architecture, and find that its changing trend is similar to that of the VGG architecture. It can be seen that each teacher tends to collect knowledge from its nearby exits at all training epochs reported due to the self-attention mechanism. This ensures that the knowledge that students are required to learn aligns closely with what they have already mastered, and providing such within-capability knowledge will help learning. It can also be observed that the poor-performance exit's primary teacher is composed of more "self-knowledge" (larger diagonal weights), indicating a more urgent need to narrow the learning gap. Because the performance of shallow exits is better than that of deep exits at the early training stage, diagonal weights of shallow exits are smaller than those of the deepest exit at epoch 1 compared to other epochs.

Table 4 Learning gap comparison on CIFAR-100 with VGG16

Exit No.	Accuracy difference (%)			
	BYOT	DKS	EED	MATE
1	5.91±2.47	6.83±2.13	7.49±1.98	5.67±**1.57**
2	0.50±1.19	1.78±1.00	2.18±0.85	1.40±**0.60**
3	−0.59±0.61	0.36±0.76	0.71±0.56	0.48±**0.45**
4	–	1.01±0.97	1.27±0.76	0.94±**0.71**

Bold texts denote the best results of standard deviation

4.7 Ablation study

4.7.1 Impact of α

We show the accuracy of exits 1 and 4 of VGG16 and ResNeXt50 when using different α on CIFAR-100. As shown in Fig. 8, MATE achieves quite good results when $\alpha = 3$, and we adopt it as the default setting in our experiments. Additionally, the final

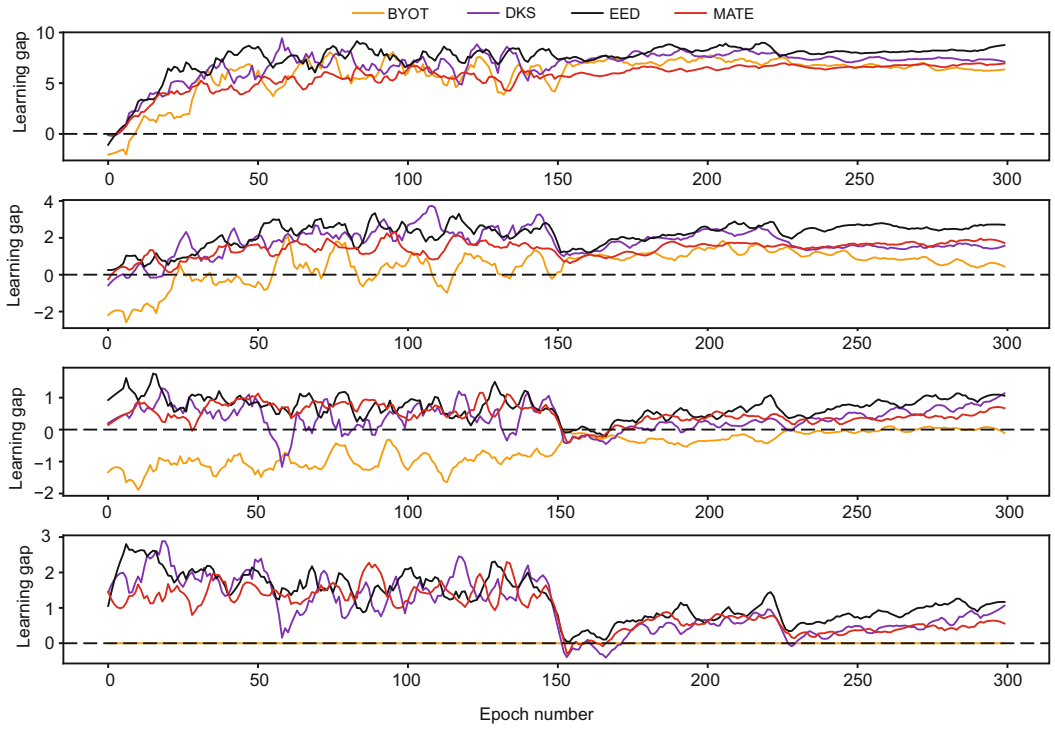

Fig. 6 Trends in the learning gap across training epochs with CIFAR-100 and VGG16 (Sub-figures from top to bottom indicate the deepening of the network. The data are smoothed using Savitzky–Golay filter (Schafer, 2011))

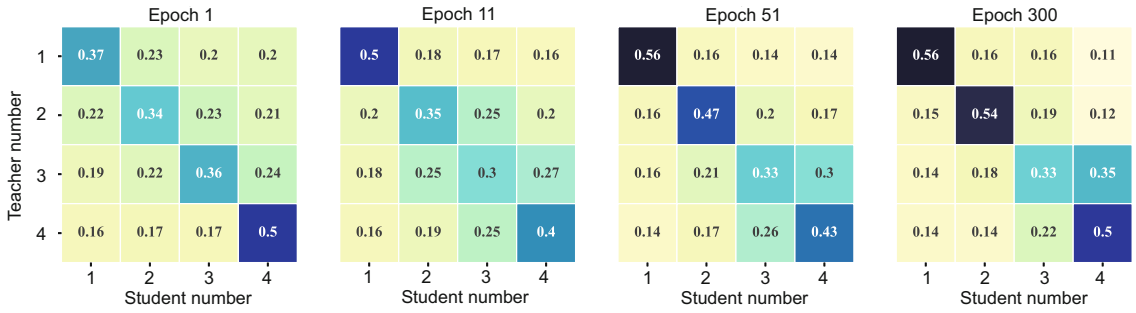

Fig. 7 Heat map of ensemble weights during training (Darker colors indicate larger weights)

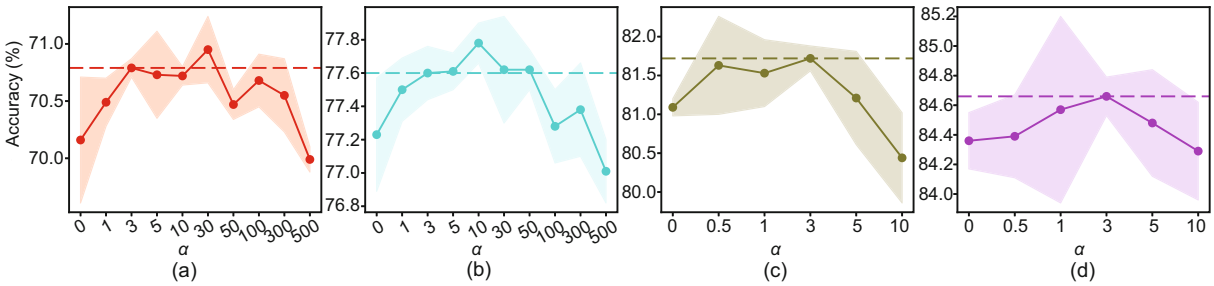

Fig. 8 Impact of α on top-1 test accuracy with different network architectures and exits on CIFAR-100: (a) VGG16-exit 1; (b) VGG16-exit 4; (c) ResNeXt50-exit 1; (d) ResNeXt50-exit 4 (The accuracy values for $\alpha = 3$ are marked with dashed lines)

performance degrades whether α is too small or too large, and the optimal α is different for different exits and models. It can be seen that the optimal α of VGG16 (30 for exit 1 and 10 for exit 4) is greater compared to that of ResNeXt50 (3 for exits 1 and 4). This may be because the performance difference between exits of VGG16 is larger compared to that of ResNeXt50 (see the results in Table 2). Therefore, the need for smaller learning gap in the distillation is greater for VGG16, which is achieved by using a larger α.

4.7.2 Impact of diverse α

As shown in Fig. 8, the optimal α varies among different exits. Therefore, we modify Eq. (8) to the following equation to see whether diverse α can achieve better results:

$$\begin{aligned} L = &\sum_{i=1}^{M} L_{\mathrm{CE}}(\boldsymbol{p}^i, \boldsymbol{y}) \\ &+ \sum_{i=1}^{M}\sum_{j=1}^{M} L_{\mathrm{KL}}^{*}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^j}{T}\right)\right) \\ &+ \sum_{i=1}^{M} \alpha_i L_{\mathrm{KL}}^{*}\left(\sigma\left(\frac{\boldsymbol{z}^i}{T}\right), \sigma\left(\frac{\hat{\boldsymbol{z}}^i}{T}\right)\right), \end{aligned} \tag{9}$$

where each exit has its unique α_i. We train MATE by setting $\alpha_1 = 30$, $\alpha_2 = 5$, and $\alpha_3 = \alpha_4 = 10$ because they are the optimal values we obtained for exits on CIFAR-100 with VGG16. Table 5 shows that simply applying diverse α fails to boost the performance, and the results are even worse than those when α is consistently set to 3. This indicates that model training is a complex process that requires exit synergy. What is more, it is easier and more time-saving to tune a single α than to provide the most appropriate α for each exit.

4.7.3 Impact of diverse teachers

To prevent student from overly focusing on its primary teacher, we ask each exit to learn from all

Table 5 Impact of diverse α on top-1 test accuracy on CIFAR-100 with VGG16 using MATE

Exit No.	Accuracy (%)		
	Optimal	$\alpha = 3$	Diverse α
1	**70.95±0.29**	70.79±0.08	70.71±0.18
2	**76.09±0.25**	75.92±0.17	75.88±0.37
3	**77.71±0.21**	77.60±0.17	77.43±0.35
4	**77.78±0.12**	77.60±0.16	77.48±0.34

Bold texts denote the best results, and underlined texts indicate the second-best results

teachers with a variety of knowledge to improve the model performance. To verify its usefulness, we let students learn only from their primary teachers by modifying Eq. (6). This will cause significant performance degradation and the results in Table 6 confirm the importance of providing diverse teachers.

4.7.4 Impact of distillation temperature

We further experimented on more distillation temperatures to show the superiority of MATE. As shown in Fig. 9, MATE still outperforms other algorithms, even we do not tune other hyper-parameters for temperature variations.

4.7.5 Impact of weight network

Results when using the same α but applying different ensemble mechanisms can be quite different. Due to the nature of the self-attention mechanism, for more similar features, it tends to output larger weights. However, this is not the case when using the gate mechanism. We compared these two mechanisms on CIFAR-100 with VGG16, and the results are shown in Fig. 10. As can be seen, there is no significant difference in the optimal accuracy between the gate and self-attention mechanisms. However, it is noteworthy that the value of α at which self-attention reaches its optimal accuracy is usually lower than that for the gate mechanism. This is because the corresponding teacher logits for each exit tend to be closer to the logits of the current exit when using self-attention and thus a smaller α is enough.

Table 6 Impact of diverse teachers (DTs) on top-1 test accuracy on CIFAR-100 with VGG16 using MATE

Exit No.	Accuracy (%)	
	With DT	Without DT
1	**70.79±0.08**	70.37±0.48
2	**75.92±0.17**	75.68±0.17
3	**77.60±0.17**	77.25±0.19
4	**77.60±0.16**	77.37±0.11

Bold texts denote the better results

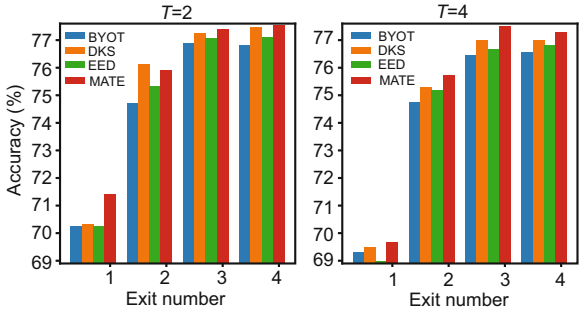

Fig. 9 Ablation study on temperature T with VGG16 on CIFAR-100

4.7.6 Budget mode

To better demonstrate the superiority of our method, we conducted distillation in MSDNet (Huang et al., 2018) with 5 exits and 15 layers. Following the settings in DBT (Phuong and Lampert, 2019), we report the top-5 test accuracy. In budget mode with dynamic evaluation, comparison results on CIFAR-100 are shown in Table 7. Under different FLOPs, MATE achieves the best results.

Table 7 Top-5 test accuracy comparison in budget mode

FLOPs	Accuracy (%)				
	MSDNet	BYOT	DKS	EED	MATE
∼6.9M	86.37	<u>87.02</u>	86.67	86.84	**87.42**
∼8.3M	88.07	88.31	<u>88.38</u>	88.28	**88.86**
∼10.5M	89.42	89.57	<u>89.71</u>	89.67	**90.13**
∼13.1M	90.41	<u>90.78</u>	90.53	90.75	**91.07**
∼16.1M	91.51	<u>91.69</u>	91.30	91.60	**91.96**
∼19.3M	92.14	92.36	92.09	<u>92.49</u>	**92.66**

Bold texts denote the best results, and underlined texts indicate the second-best results

5 Limitations and future work

Although no extra inference time is introduced, training with MATE is more time-consuming and costs more memory space compared to other methods. Therefore, MATE may fail to run when applied to a huge model with many exits due to memory constraints. In addition, it may take significant computational resources and time to search the optimal α, which is even aggravated when diverse α is used. It is a challenge to automatically determine the appropriate α based on the performance of the model during training. We consider it as our future work.

6 Conclusions

Using self-distillation in the training of multi-exit architecture can improve the performance of each exit, which is particularly useful for resource-constrained circumstances. Existing methods use mainly the knowledge from deep exits or a single

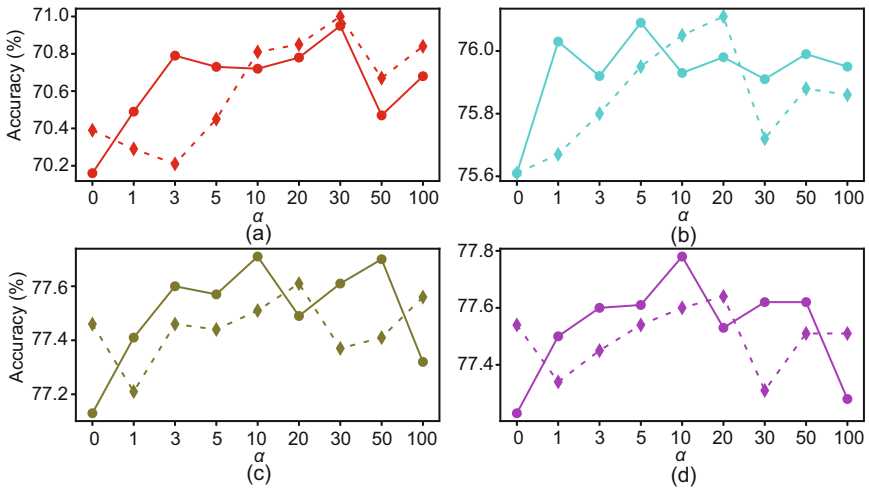

Fig. 10 Top-1 test accuracy comparison on CIFAR-100 with VGG16 based on different α: (a) exit 1; (b) exit 2; (c) exit 3; (d) exit 4 (The results of the self-attention mechanism and gate mechanism are represented by solid lines with • and dotted lines with ⧫, respectively)

ensemble to guide all exits, and ignore the fact that shallow exit performance may not be significantly improved due to the learning gap between the teacher and the student. In this paper, we propose Multi-exit self-distillation with Appropriate TEachers (MATE) to provide diverse and appropriate knowledge for each exit. We highlight the necessity of controlling the learning gap between students and teachers. Experimental results show that our method consistently achieves better performance than state-of-the-art methods with various network architectures on multiple datasets.

Contributors

Wujie SUN designed the research, processed the data, and drafted the paper. Defang CHEN, Can WANG, Deshi YE, Yan FENG, and Chun CHEN helped organize the paper. All the authors revised and finalized the paper.

Acknowledgements

The authors would like to thank the advanced computing resources provided by the Supercomputing Center of Hangzhou City University, China.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are openly available. The data based on CIFAR-100 are available from `https://www.cs.toronto.edu/~kriz/cifar.html`. The data based on TinyImageNet are available from `http://cs231n.stanford.edu/tiny-imagenet-200.zip`. The data based on CUB-200-2011 are available from `https://www.vision.caltech.edu/datasets/cub_200_2011/`. The data based on Stanford Dogs are available from `http://vision.stanford.edu/aditya86/ImageNetDogs/`. The data based on FGVC-Aircraft are available from `https://www.robots.ox.ac.uk/~vgg/data/fgvc-aircraft/`.

References

Ahn S, Hu SX, Damianou A, et al., 2019. Variational information distillation for knowledge transfer. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9155-9163. https://doi.org/10.1109/CVPR.2019.00938

Anil R, Pereyra G, Passos A, et al., 2020 Large scale distributed neural network training through online distillation. https://arxiv.org/abs/1804.03235

Ba LJ, Caruana R, 2014. Do deep nets really need to be deep? Proc 27th Int Conf on Neural Information Processing Systems, p.2654-2662.

Chen DF, Mei JP, Wang C, et al., 2020. Online knowledge distillation with diverse peers. *Proc AAAI Conf Artif Intell*, 34(4):3430-3437. https://doi.org/10.1609/aaai.v34i04.5746

Chen DF, Mei JP, Zhang Y, et al., 2021. Cross-layer distillation with semantic calibration. *Proc AAAI Conf Artif Intell*, 35(8):7028-7036. https://doi.org/10.1609/aaai.v35i8.16865

Chen DF, Mei JP, Zhang HL, et al., 2022. Knowledge distillation with the reused teacher classifier. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11923-11932. https://doi.org/10.1109/CVPR52688.2022.01163

Deng J, Dong W, Socher R, et al., 2009. ImageNet: a large-scale hierarchical image database. IEEE Conf on Computer Vision and Pattern Recognition, p.248-255. https://doi.org/10.1109/CVPR.2009.5206848

Deng X, Zhang ZF, 2021. Learning with retrospection. *Proc AAAI Conf Artif Intell*, 35(8):7201-7209. https://doi.org/10.1609/aaai.v35i8.16885

Furlanello T, Lipton Z, Tschannen M, et al., 2018. Born again neural networks. Proc 35th Int Conf on Machine Learning, p.1607-1616.

Ge YX, Zhang X, Choi CL, et al., 2021. Self-distillation with batch knowledge ensembling improves ImageNet classification. https://arxiv.org/abs/2104.13298

He KM, Zhang XY, Ren SQ, et al., 2016. Deep residual learning for image recognition. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.770-778. https://doi.org/10.1109/CVPR.2016.90

Hinton G, Vinyals O, Dean J, 2015. Distilling the knowledge in a neural network. https://arxiv.org/abs/1503.02531

Huang G, Liu Z, Van Der Maaten L, et al., 2017. Densely connected convolutional networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.2261-2269. https://doi.org/10.1109/CVPR.2017.243

Huang G, Chen DL, Li TH, et al., 2018. Multi-scale dense networks for resource efficient image classification. https://arxiv.org/abs/1703.09844

Ji M, Shin S, Hwang S, et al., 2021. Refine myself by teaching myself: feature refinement via self-knowledge distillation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10659-10668. https://doi.org/10.1109/CVPR46437.2021.01052

Jin X, Peng BY, Wu YC, et al., 2019. Knowledge distillation via route constrained optimization. IEEE/CVF Int Conf on Computer Vision, p.1345-1354. https://doi.org/10.1109/ICCV.2019.00143

Khosla A, Jayadevaprakash N, Yao BP, et al., 2011. Novel Dataset for Fine-Grained Image Categorization: Stanford Dogs. http://vision.stanford.edu/aditya86/ImageNetDogs/ [Accessed on Dec. 30, 2021].

Krizhevsky A, Hinton G, 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report, Computer Science Department, University of Toronto, Canada.

Lan X, Zhu XT, Gong SG, 2018. Knowledge distillation by on-the-fly native ensemble. https://arxiv.org/abs/1806.04606

Le Y, Yang X, 2015. Tiny ImageNet Visual Recognition Challenge. `http://cs231n.stanford.edu/tiny-imagenet-200.zip` [Accessed on Dec. 30, 2021].

Lee H, Lee JS, 2021. Students are the best teacher: exit-ensemble distillation with multi-exits. https://arxiv.org/abs/2104.00299

Maji S, Rahtu E, Kannala J, et al., 2013. Fine-grained visual classification of aircraft. https://arxiv.org/abs/1306.5151

Mirzadeh SI, Farajtabar M, Li A, et al., 2020. Improved knowledge distillation via teacher assistant. *Proc AAAI Conf Artif Intell*, 34(4):5191-5198. https://doi.org/10.1609/aaai.v34i04.5963

Phuong M, Lampert C, 2019. Distillation-based training for multi-exit architectures. Proc IEEE/CVF Int Conf on Computer Vision, p.1355-1364. https://doi.org/10.1109/ICCV.2019.00144

Schafer RW, 2011. What is a Savitzky–Golay filter? *IEEE Signal Process Mag*, 28(4):111-117. https://doi.org/10.1109/MSP.2011.941097

Schwartz R, Dodge J, Smith NA, et al., 2020. Green AI. *Commun ACM*, 63(12):54-63. https://doi.org/10.1145/3381831

Shi WX, Song YX, Zhou H, et al., 2021. Follow your path: a progressive method for knowledge distillation. https://arxiv.org/abs/2107.09305

Simonyan K, Zisserman A, 2015. Very deep convolutional networks for large-scale image recognition. https://arxiv.org/abs/1409.1556

Son W, Na J, Choi J, et al., 2021. Densely guided knowledge distillation using multiple teacher assistants. Proc IEEE/CVF Int Conf on Computer Vision, p.9375-9384. https://doi.org/10.1109/ICCV48922.2021.00926

Sun DW, Yao AB, Zhou AJ, et al., 2019. Deeply-supervised knowledge synergy. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.6990-6999. https://doi.org/10.1109/CVPR.2019.00716

Teerapittayanon S, McDanel B, Kung HT, 2016. BranchyNet: fast inference via early exiting from deep neural networks. 23rd Int Conf on Pattern Recognition, p.2464-2469. https://doi.org/10.1109/ICPR.2016.7900006

Tian YL, Krishnan D, Isola P, 2022. Contrastive representation distillation. https://arxiv.org/abs/1910.10699

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Wah C, Branson S, Welinder P, et al., 2011. The Caltech-UCSD Birds-200-2011 Dataset. `https://www.vision.caltech.edu/datasets/cub_200_2011/` [Accessed on Dec. 30, 2021].

Xie SN, Girshick R, Dollár P, et al., 2017. Aggregated residual transformations for deep neural networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5987-5995. https://doi.org/10.1109/CVPR.2017.634

Xu TB, Liu CL, 2019. Data-distortion guided self-distillation for deep neural networks. *Proc AAAI Conf Artif Intell*, 33(1):5565-5572. https://doi.org/10.1609/aaai.v33i01.33015565

Yang CL, Xie LX, Su C, et al., 2019a. Snapshot distillation: teacher-student optimization in one generation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2854-2863. https://doi.org/10.1109/CVPR.2019.00297

Yang CL, Xie LX, Qiao SY, et al., 2019b. Training deep neural networks in generations: a more tolerant teacher educates better students. *Proc AAAI Conf Artif Intell*, 33(1):5628-5635. https://doi.org/10.1609/aaai.v33i01.33015628

Yuan L, Tay FEH, Li GL, et al., 2020. Revisiting knowledge distillation via label smoothing regularization. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3902-3910. https://doi.org/10.1109/CVPR42600.2020.00396

Yun S, Park J, Lee K, et al., 2020. Regularizing class-wise predictions via self-knowledge distillation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.13873-13882. https://doi.org/10.1109/CVPR42600.2020.01389

Zagoruyko S, Komodakis N, 2017. Wide residual networks. https://arxiv.org/abs/1605.07146

Zhang LF, Song JB, Gao AN, et al., 2019. Be your own teacher: improve the performance of convolutional neural networks via self distillation. Proc IEEE/CVF Int Conf on Computer Vision, p.3712-3721. https://doi.org/10.1109/ICCV.2019.00381

Zhang Y, Xiang T, Hospedales TM, et al., 2018. Deep mutual learning. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4320-4328. https://doi.org/10.1109/CVPR.2018.00454