Interactive Knowledge Distillation for image classification

Shipeng Fu [a], Zhen Li [b], Zitao Liu [c], Xiaomin Yang [a],*

[a] College of Electronics and Information Engineering, Sichuan University, Chengdu 610064, China
[b] College of Computer Science, Nankai University, Tianjin 300350, China
[c] TAL Education Group, Beijing 100080, China

ABSTRACT

Knowledge distillation (KD) is a standard teacher-student learning framework to train a light-weight student network under the guidance of a well-trained, large teacher network. As an effective teaching strategy, interactive teaching has been widely employed at school to motivate students, in which teachers not only provide knowledge, but also give constructive feedback to students upon their responses, to improving their learning performance. In this work, we propose Interactive Knowledge Distillation (IAKD) to leverage the interactive teaching strategy for efficient knowledge distillation. In the distillation process, the interaction between the teacher network and the student one is implemented by swapping-in operations: randomly replacing the blocks in the student network with the corresponding blocks in the teacher network. In this way, we directly involve the teacher's powerful feature transformation ability for largely boosting the performance of the student network. Experiments with typical settings of teacher-student networks demonstrate that the student networks trained by our IAKD achieve better performance than those trained by conventional knowledge distillation methods on diverse image classification datasets.

© 2021 Elsevier B.V. All rights reserved.

1. Introduction

Over the past few years, deeper and deeper convolutional neural networks (CNNs) have demonstrated cutting edge performance on various computer vision tasks [1–5]. However, they usually undergo huge computational costs and memory consumption, and are difficult to be embedded into resource-constrained devices, *e.g.*, mobiles and UAVs. To reduce the resource consumption while maintaining good performance as much as possible, researchers leverage knowledge distillation techniques [6–11] to transfer informative knowledge from a cumbersome but well-trained teacher network into a light-weight but unskilled student network.

Most of existing knowledge distillation methods [12–17] encourage the student network to mimic the representation space of the teacher network for approaching its proficient performance. Those methods manually define various kinds of knowledge based on the teacher network responses, such as softened outputs [12], attention maps [13], flow of solution procedure [18], and relations [14]. These different kinds of knowledge facilitate the distillation process through additional distillation losses. If we regard the responses of the teacher network as the theorems in a textbook written by the teacher, manually defining those different kinds of knowledge

rephrases those theorems and further points out an easier way to make the student understand those theorems better. This kind of distillation process is viewed as *non-interactive* knowledge distillation, since the teacher network sets a goal for the student network to mimic, ignoring the interaction with the student network.

The *non-interactive* knowledge distillation methods undergo one major problem: since the feature transformation ability of the student network is less powerful than that of the teacher network, the knowledge that the student has learned through mimicking is impossible to be identical with the knowledge provided by the teacher, which may impede the knowledge distillation performance. This problem becomes more tough when introducing multi-connection knowledge [13,19]. Specifically, since the student can not perfectly mimic the teacher's knowledge even if in the shallow blocks, the imperfectly-mimicked knowledge is reused in the following blocks to imitate the corresponding teacher knowledge. Therefore, the gap between the knowledge learned by the student and the one provided by the teacher becomes larger, restricting the performance of knowledge distillation.

Based on the discussion above, we try to make the distillation process *interactive* and the teacher network truly involved in guiding the student network. Specifically, the student first extracts features on its own, then the teacher responds by improving the weak features with its powerful feature transformation ability, finally the student continues to extract deep features based on the
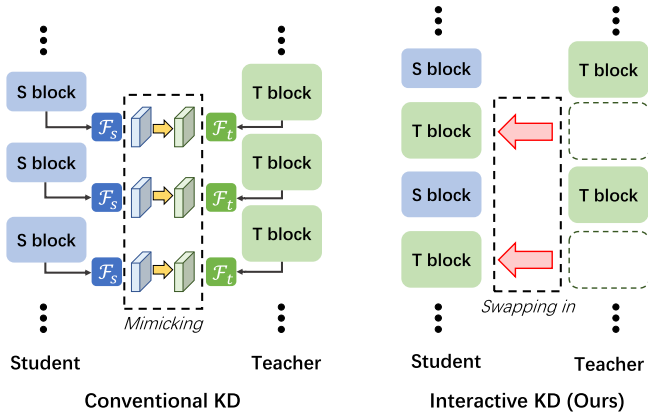
Fig. 1. Differences between our proposed interactive knowledge distillation method and conventional, non-interactive ones. "S block" denotes the student block, "T block" denotes the teacher block. $\mathscr{F}_s$ and $\mathscr{F}_t$ denote the definitions of the student and the teacher knowledge, respectively, and vary a lot from method to method.

improved features. Better features are critical for better classification performance and can provide a good foundation for the learning process of the student [20–22]. As shown in Fig. 1, instead of forcing the student to mimic the teacher's representation space, our interactive distillation directly involves the teacher's powerful feature transformation ability into the student. Consequently, the student can make the best use of the improved features to better exploit its potential on relevant tasks.

In this paper, we propose a simple yet effective knowledge distillation method named Interactive Knowledge Distillation (IAKD). In IAKD, we randomly swap in the blocks in the teacher network to replacing the blocks in the student network during the distillation phase. Each set of swapped-in teacher blocks responds to the output of previous student block, and provides feature maps to motivate the next student block, which are better than the feature maps that the replaced student block can provide. In addition, randomly swapping in the teacher blocks makes it possible for the teacher to guide the student in many different manners. Compared with other conventional knowledge distillation methods, our proposed method does not deliberately force the student's knowledge to be similar to the teacher's knowledge. Therefore, we do not need the additional distillation losses to drive the knowledge distillation process, resulting in no need of hyper-parameters to balancing the task-specific loss and distillation losses. Besides, the distillation process of our IAKD method is highly efficient. The reason is our IAKD discards the cumbersome knowledge transformation process and gets rid of the feature extraction of the entire teacher network, both of which commonly exist in conventional knowledge distillation methods. Experimental results demonstrate that our proposed method effectively boosts the performance of the student, which proves the student can actually benefit from the interaction with the teacher.

2. Related work

Since Ba and Caruana [23] first introduced the view of teacher supervising student for model compression, many variant works have emerged based on their teacher-student learning framework. To achieve promising model compression results, these works focus on how to better capture the knowledge of the large teacher network to supervise the training process of the small student network. The most representative work about knowledge distillation (KD) comes from Hinton et al. [12]. They defined knowledge as the teacher's softened logits and encouraged the student to mimic

it instead of the raw activations before softmax [23], because they argued that the intra-class and inter-class relationships learned by the pre-trained teacher network can be described more accurately by the softened logits than by the raw activations. However, simply using logits as the learning goal for the student may limit the information that the teacher distills to the student. To introduce more knowledge of the teacher network, FitNets [24] added element-wise supervision on the feature maps at intermediate layers to assisting the training of the student. However, this approach only works well when one supervision loss is applied to one intermediate layer. It can not achieve satisfying results when more losses are added to supervising more intermediate layers [18]. This is because supervision on the feature maps at many different layers makes the constraints become too restrictive. There are many methods trying to soften the constraints while preserving meaningful information of feature maps. Zagoruyko and Komodakis [25] condensed feature maps to attention maps based on channel statistics of each spatial location. Instead of defining knowledge based on feature maps of a single layer, Yim et al. [18] employed Gramian matrix to measure the correlations between feature maps from different layers. Besides, Kim et al. [26] utilized convolutional auto-encoder, and Lee et al. [27] used Singular Value Decomposition (SVD) to perform dimension reduction on feature maps. Heo et al. [19,15] demonstrated that simply using activation status of neurons could also assist the distillation process. Ahn et al. [28] proposed a method based on information theory which maximizes the mutual information between the teacher network and the student network. To enable the student to acquire more meaning information, a lot of studies defined knowledge based on relations rather than individual feature maps [14,29–31,16,32].

On the other hand, our proposed IAKD is essentially different from aforementioned knowledge distillation methods. IAKD does not require distillation losses to drive the distillation process (see Eqs. (1) and (5)). In the learning process of the student, we directly involve the teacher's powerful feature transformation ability to improve the relatively weak features extracted by the student, since better features are critical for getting a better prediction.

3. Proposed method

3.1. Conventional knowledge distillation

In general, the conventional, feature-based knowledge distillation methods [19,28,13,18,15] decompose the student network and the teacher one into M modules. The student network is trained to minimize the following loss function:

$$\mathscr{L}_{CKD} = \gamma(y, \hat{y}) + \sum_{m \in M} \lambda_m \varphi_m(F_m^S, F_m^T), \qquad (1)$$

where $\gamma(\cdot, \cdot)$ is the task-specific loss. y represents the ground truth. $\hat{y}$ is the predicted result of the student network. F_m^S and F_m^T denote the transformed output of the student network and the one of the teacher network produced by the m-th module. λ_m is a tunable balance factor for balancing the different losses. $\varphi_m(\cdot, \cdot)$ is the m-th distillation loss for narrowing down the distance between F_m^S and F_m^T. $\varphi_m(\cdot, \cdot)$ varies from method to method because of different definitions of knowledge. As can be seen in Eq. (1), the conventional knowledge distillation methods essentially aim at forcing the student to mimic the teacher representation space.

3.2. Interactive knowledge distillation

To achieve the interaction between the student network and the teacher network, our IAKD randomly swaps in the teacher blocks to replace the student blocks at each step. The swapped-in

teacher blocks can respond to the output of the previous student block and then provide better features to motivate the next student block. To make all student blocks fully participate in the distillation process, after each training step, we put back the replaced student blocks to their original positions in the student network. Due to the return of replaced student blocks, the architecture of the student is kept consistent for the next swapping-in operation.

Hybrid block for swapping-in operation. As shown in Fig. 2, for effective implementation of such interactive strategy, we devise a two-path hybrid block, which is composed of one student block (the student path) and several teacher blocks to be swapped in (the teacher path). We use the hybrid block to replace the original student block in the student network to form a hybrid network. However, the blocks and layers that are shared by the teacher and the student are not replaced by the hybrid blocks,[1] since the shared parts have the same learning capacity. The number of hybrid blocks depends on the number of non-shared student blocks. Every non-shared student block will pair with one or several non-shared teacher blocks to form the hybrid block. All non-shared teacher blocks will be taken to form the hybrid blocks.

For the i-th hybrid block in the hybrid network, let $a_i \in \{0, 1\}$ denotes a Bernoulli random variable, which indicates whether the student path is selected ($a_i = 1$) or the teacher path is chosen ($a_i = 0$). Therefore, the output of the i-th hybrid block can be formulated as

$$H_{i+1} = a_i f_i^S(H_i) + (1 - a_i) f_i^T(H_i), \qquad (2)$$

where H_i and H_{i+1} denote the input and the output of the i-th hybrid block. $f_i^S(\cdot)$ is the function of the student block. $f_i^T(\cdot)$ is the nested functions of the teacher blocks.

If $a_i = 1$, the student path is chosen and Eq. (2) reduces to:

$$H_{i+1} = f_i^S(H_i). \qquad (3)$$

This means the hybrid block degrades to the original student block. If $a_i = 0$, the teacher path is chosen and Eq. (2) reduces to:

$$H_{i+1} = f_i^T(H_i). \qquad (4)$$

This implies the hybrid block degrades to the teacher blocks. Next, we denote the probability of selecting the student path in the i-th hybrid block as $p_i = P(a_i = 1)$. Therefore, each student block is randomly replaced by the teacher blocks with probability $(1 - p_i)$. The swapped-in teacher blocks can interact with the student blocks to inspire the potential of the student.

The student blocks are replaced by the teacher blocks in a random manner, which means the teacher can guide the student in many different situations through interaction. In contrast, swapping in the teacher blocks by manually-defined manners allows the teacher to guide the student only in limited situations. Therefore, to traverse as many different situations as possible, the student blocks are replaced by the teacher blocks in a random manner. Besides, we only assume the swapped-in teacher blocks can output semantically similar but better feature maps than the original student block. We do not need the teacher and the student networks in the same pattern or same depth.

Distillation phase. When the image features go through each hybrid block, the path is randomly chosen during the distillation phase. Let us consider two extreme situations. The first is that $p_i = 1$ for all hybrid blocks. The student path is chosen in every hybrid block and the hybrid network becomes the student network. The second is that $p_i = 0$ for all hybrid blocks. Under such circumstances, all teacher paths are chosen, and the hybrid network becomes the teacher network. In other cases ($0 < p_i < 1$),

---

[1] The fully-connected layer and the transition blocks in ResNet [33] are considered as the shared parts.
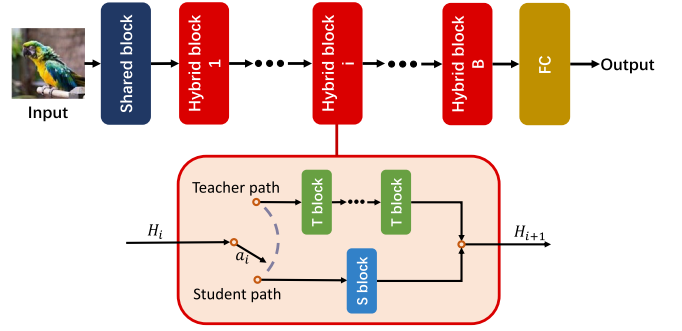

Fig. 2. The diagram of the hybrid network which can effectively implement the proposed interactive knowledge distillation (IAKD). Specifically, the hybrid network enables the teacher blocks to randomly replace the student block of the original student network. FC denotes the fully-connected layer.

the input randomly goes through the student path or the teacher path in each hybrid block. The swapped-in teacher blocks are trained with the student blocks, thus they can respond to the output of the previous student block and further provide better features to motivate the following student block. Before distillation phase, the parameters of the teacher blocks in each hybrid block are loaded from the corresponding teacher blocks in the pretrained teacher network. During distillation process, those parameters are frozen, which means the teacher parameters will not be updated, and the batch normalization layers still use the mini-batch statistics. The student block in each hybrid block, and other shared parts are randomly initialized. The optimized loss function of the hybrid network $\mathscr{L}_h$ can be formulated as

$$\mathscr{L}_h = \gamma(y, \hat{y}_h), \qquad (5)$$

where $\gamma(\cdot, \cdot)$ is the task-specific loss. y denotes the ground truth. $\hat{y}_h$ is the predicted result of the hybrid network. At each step, only the student parts which the feature maps go through are updated (the shared parts are considered as the student parts since they are trained from scratch).

Comparing Eq. (5) with Eq. (1), we can see our proposed method does not need the additional distillation losses to drive the knowledge distillation process. Therefore, laboriously searching the optimal hyper-parameters to balance the task-specific loss and different distillation losses can be avoided. Besides, the complicated knowledge transformation and extra forward computation of the teacher network are unnecessarily required by our proposed method, leading to an efficient distillation process (see Section 4.5 for training time comparison).

Test phase. Since knowledge distillation aims at improving the performance of the student network, we test the whole student network to check whether the student can achieve performance improvement or not. For convenient validation, we fix the probability p_i to 1 in the test phase. Under this circumstances, the hybrid network becomes the student network during the test phase. Meanwhile, for practical deployment, it is not reasonable to directly use the cumbersome hybrid network. We can reduce the burden of resource consumption by separating the student network from the hybrid network for further inference.

3.3. Probability schedule

The interaction level between the teacher network and the student is controlled by p_i. The smaller value of p_i indicates the student network has more interaction with the corresponding teacher blocks. For simplicity, we assume $p_1 = p_2 = \ldots = p_i = \ldots = p_B$, where B is the number of hybrid blocks. This suggests that every hybrid block shares the same interaction rule. To properly utilize

the interaction mechanism, we propose three kinds of probability schedules for the change of p_i: 1) uniform schedule, 2) linear growth schedule and 3) review schedule. These schedules are discussed in detail as follows.

Uniform schedule. The simplest probability schedule is to fix $p_i = p_{start} \in (0, 1)$ for all epochs during the distillation phase. This means the interaction level does not change. See Fig. 3a for a schematic illustration.

Linear growth schedule. With the epoch increasing, the teacher network could participate less in the distillation process as [24,18,19] do, because the student network gradually becomes more powerful to handle the task by itself. Thus, we set the value of p_i according to a function of the number of epoch, and propose a simple linear growth schedule. The probability of choosing the student path in each hybrid block increases linearly from $p_i = p_{start} \in (0, 1)$ for the first epoch to $p_i = 1$ for the last epoch (see Fig. 3b). The linear growth schedule indicates that as the knowledge distillation process proceeds, the interaction level between the student and the teacher gradually decreases. Vividly speaking, the teacher teaches the student through frequent interaction in early epochs, and the student attempts to solve the problem by himself in late epochs.

Review schedule. The proposed schedules above ignore the change of the learning rate during training. When the learning rate decreases, the optimization algorithm narrows the parameter search radius for more refined adjustments. Thus, we reset p_i to p_{start} when the learning rate decreases so that the teacher can *reinteract* with the student for better guidance. Within the interval where the learning rate remains unchanged, the probability increases linearly from $p_i = p_{start} \in (0, 1)$ to $p_i = 1$. Fig. 3c shows a schematic illustration. We call this kind of probability schedule as the review schedule because it is like the teacher gives review lessons after finishing a unit so that the student can grasp the knowledge more firmly.

The effectiveness of different probability schedules will be validated experimentally in Section 4.1.

4. Experiments

In this section, we first investigate the effectiveness of different probability schedules and then conduct a series of experiments to analyze the impact of the interaction mechanism on the distillation process. At last, we compare our proposed IAKD with conventional, *non-interactive* methods on several public classification datasets to demonstrate the superiority of our proposed method.

4.1. Different probability schedules

To investigate the effectiveness of different probability schedules, we conduct experiments on CIFAR-10 [25] and CIFAR-100 [25] datasets. The CIFAR-10 dataset consists of 60000 images of size 32×32 in 10 classes. The CIFAR-100 dataset [25] has 60000 images of size 32×32 from 100 classes. For all experiments in this subsection, we use SGD optimizer with momentum 0.9 to optimize the hybrid network. We set batchsize to 128 and set weight decay to 1×10^{-4}. The hybrid network is trained for 200 epochs. The learning rate starts at 0.1 and is multiplied by 0.1 at 100, 150 epochs. Images of size 32×32 are randomly cropped from zero-padded 40×40 images. Each image is horizontally flipped with a probability of 0.5 for data augmentation.

4.1.1. Uniform schedule

We first conduct experiments with the uniform schedule mentioned in Section 3.3. We employ ResNet [33] as our base network architecture and denote ResNet-d as the ResNet with a specific

depth d. ResNet-44 and ResNet-26 are used as the teacher/student (T/S) pair. Therefore, in each hybrid block, there is one student block in the student path, and two teacher blocks in the teacher path.

The experimental results are shown in Fig. 4. Note that $p_{start} = 0$ and $p_{start} = 1$ are not reasonable options. When $p_{start} = 0$, we train the teacher network with many frozen pretrained blocks. When $p_{start} = 1$, we train the student network individually without interaction. As we can see from Fig. 4, IAKD with the uniform schedule (denoted as IAKD-U for short) can actually improve the performance of the student network when the value of p_{start} is large (CIFAR-10: $p_{start} \in \{0.7, 0.8, 0.9\}$, CIFAR-100: $p_{start} = 0.9$). However, we notice the performance of the student drops dramatically when the value of p_{start} is small (for example, $p_{start} = 0.1$ for both CIFAR-10 and CIFAR-100). This is due to the small value of p_{start} results in frequently choosing the teacher path in each hybrid block, which leads to the insufficient training of the student.

4.1.2. Linear growth schedule

We still employ ResNet-44/ResNet-26 as the T/S pair when conducting experiments with the linear growth schedule. In this schedule, p_i increases linearly from p_{start} for the first epoch to 1 for the last epoch. During early epochs when the value of p_i is relatively small, the selected student blocks and the shared parts frequently interact with the frozen teacher blocks, which means those blocks are supposed to have better initialization when the value of p_i is relatively large in late epochs (when the hybrid network is more "student-like"). As p_i gradually increases, the hybrid network transitions from "teacher-like" to "student-like". Such distillation process is related to the two-stage knowledge distillation methods [24,18,19]. In these methods, the student network is better initialized based on the distillation losses in the first stage, and is trained for the main task based on the task-specific loss in the second stage. However, the two-stage knowledge distillation methods make the abrupt transition from the initialization stage to the task-specific training stage. As a result, the teacher's knowledge may disappear as the training process proceeds, because the teacher is not involved in guiding the student in the second stage. On the contrary, as p_i increases linearly, IAKD makes a smooth transition from the distillation stage to the task-specific training.

The experimental results of IAKD with linear growth schedule (denoted as IAKD-L) are shown in Fig. 5. Comparing Fig. 5 with Fig. 4, we have two main observations. The first is IAKD-L improves the original student performance on almost every value of p_{start} while IAKD-U only works for large p_{start}. The second is the best distillation results of IAKD-L on both CIFAR-10 and CIFAR-100 datasets are better than the ones of IAKD-U. These two observations indicate that, as the student network becomes powerful, the interaction between the teacher network and student network is more effective when at a gradually reduced rate other than a fixed level.

4.1.3. Review schedule

For fair comparison, we employ ResNet-44/ResNet-26 as the T/S pair to conduct experiments on IAKD with the review schedule (denoted as IAKD-R for short). The experimental results of IAKD-R are shown in Fig. 6. IAKD-R, which considers the change of learning rate, is actually an advanced version of IAKD-L. Comparing Fig. 6 with Fig. 5, we can see that IAKD-R could further improve the performance of the original student in contrast to IAKD-L. Even for very small value of p_{start}, IAKD-R still provides improvement for the original student network.

For more clearly comparison, we summarize the best results of different probability schedules in Table 1. It can be seen that IAKD-R obtains the highest classification accuracy in comparison with IAKD-U and IAKD-L on both CIFAR-10 and CIFAR-100 datasets. From above discussions and comparisons, we can conclude that

(a) Uniform schedule
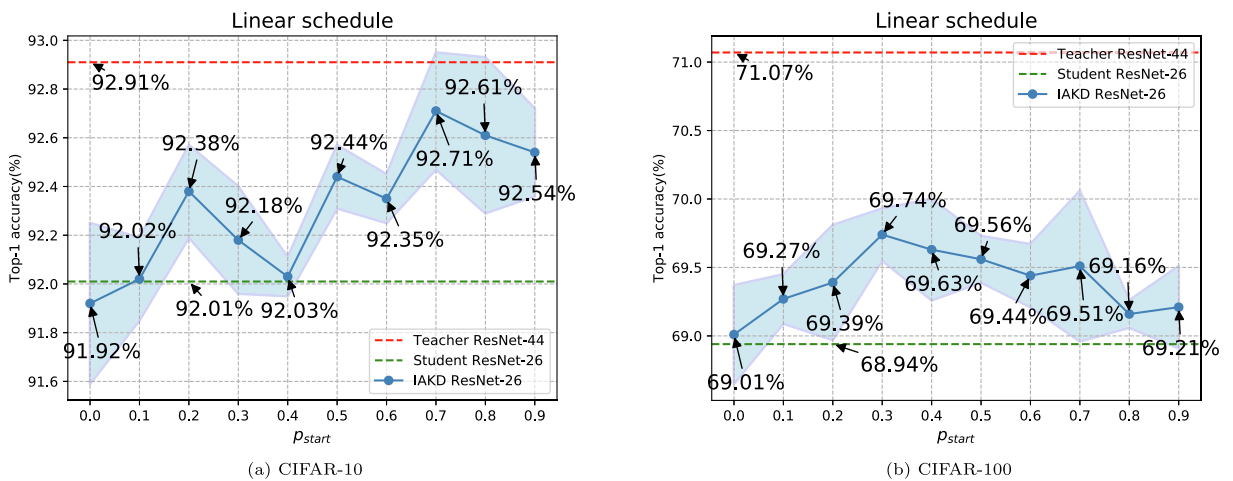(b) Linear growth schedule
(c) Review schedule
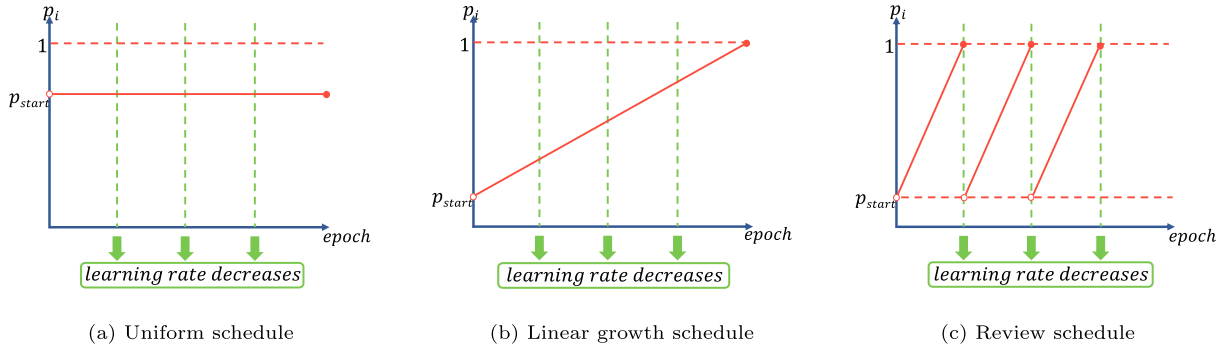
Fig. 3. Schematic illustrations for p_i with different probability schedules. The vertical green dashed lines indicate when the learning rate decreases.
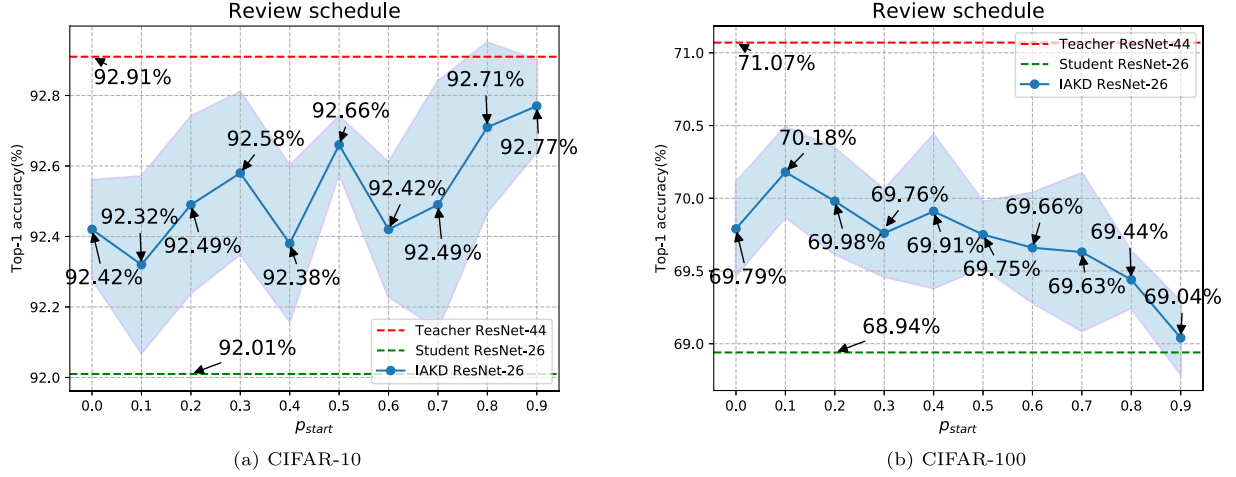

(a) CIFAR-10
(b) CIFAR-100
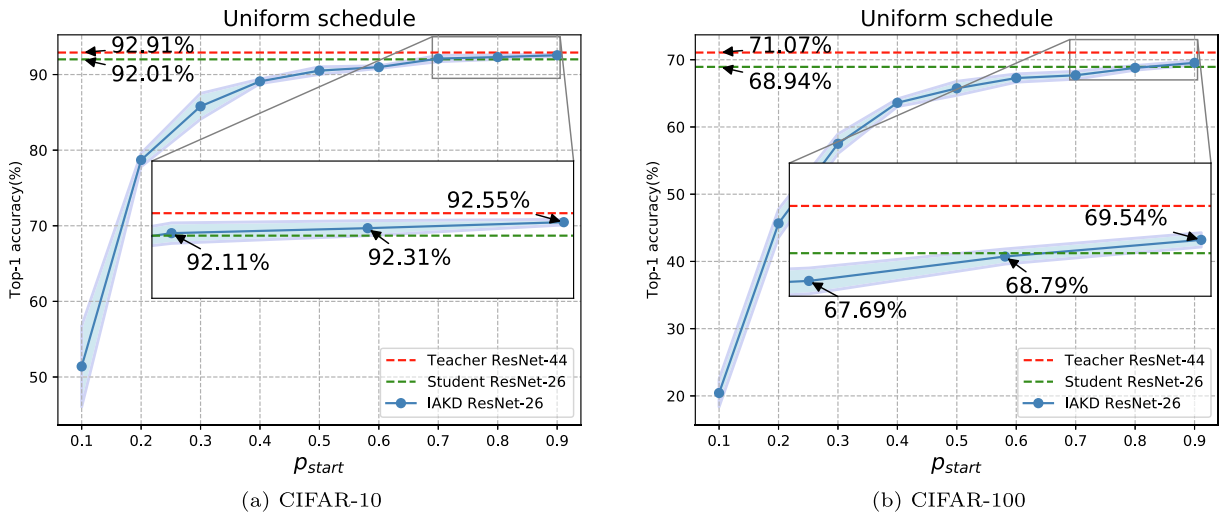
Fig. 4. The results of IAKD-U with different p_{start}. The figure shows Top-1 accuracy on test datasets. Each experiment is repeated 5 times, and we take the mean value as the result.

(a) CIFAR-10
(b) CIFAR-100

Fig. 5. The results of IAKD-L with different p_{start}. The figure shows Top-1 accuracy on test datasets. Each experiment is repeated 5 times, and we take the mean value as the result.

compared with uniform schedule and linear growth schedule, review schedule is the most effective interaction mechanism which makes the teacher network re-interact with the student network once the learning rate decreases. Thus, we take the review schedule as our final probability schedule setting when comparing

with other state-of-the-art knowledge distillation methods. Besides, it is worth noticing that for IAKD-L and IAKD-R, we need to set p_{start} to a small value when the number of classes is large, while we need to set p_{start} to a large value when the number of classes is small. This reflects that the distillation process can be

(a) CIFAR-10

(b) CIFAR-100

Fig. 6. The results of IAKD-R with different p_{start} . The figure shows Top-1 accuracy on test datasets. Each experiment is repeated 5 times, and we take the mean value as the result.

Table 1
The best results of each probability schedule on CIFAR-10 and CIFAR-100. The numbers in brackets represent the value of p_{start} when achieving the corresponding result. The best performance among different probability schedules for each dataset is shown in **bold**.

Method/Model	CIFAR-10	CIFAR-100
ResNet-44 (Teacher)	92.91%	71.07%
ResNet-26 (Student)	92.01%	68.94%
IAKD-U	92.55% ± 0.11% (0.9)	69.54% ± 0.32% (0.9)
IAKD-L	92.71% ± 0.24% (0.7)	69.74% ± 0.19% (0.3)
IAKD-R	**92.77% ± 0.13% (0.9)**	**70.18% ± 0.31% (0.1)**

assisted well by adequate interaction based on the difficulty of a certain task.

Moreover, to ensure the generalization of the proposed review schedule, we conduct more experiments with different schedules, using different architecture combinations[2]. The value of p_{start} in each probability schedule on CIFAR-100 is the same as the one in Table 1. Experimental results are shown in Table 2. As we can see, review schedule (*i.e.*, IAKD-R) consistently outperforms other two schedules (IAKD-U and IAKD-L) and can achieve promising performance improvement over the student networks in different teacher/student settings, proving the generalization of the review schedule.

4.2. Study of interaction mechanism

In this part, the training settings are the same with those in Section 4.1. Each experiment is repeated 5 times, and we take the mean value as the final result.

4.2.1. Without interaction

By randomly swapping in the teacher blocks at each step, we achieve the interaction between the teacher and the student. To investigate the effectiveness of the interaction mechanism, we turn off the interaction between the teacher network and the student one during the distillation process, *i.e.*, we do not replace the student block with the corresponding teacher blocks. Correspondingly, taking the whole pretrained teacher network and the student network, we force the output of each student block to be similar with the output of the corresponding teacher blocks, which

pair with the student block in hybrid blocks. Given that IAKD randomly swaps in the teacher blocks to replace the student block, we can randomly measure the difference between the output of each student block and that of the corresponding teacher blocks. Therefore, the student is trained by the following loss function:

$$\mathcal{L}_{NIA} = \gamma(y, \hat{y}_s) + \sum_{n \in B} a_n \varphi(F_n^S, F_n^T), \qquad (6)$$

where y is the ground truth. $\hat{y}_s$ is the output of the student network. B is the number of student blocks except the blocks shared by the teacher and the student. F_n^S and F_n^T are the outputs of the student block and the corresponding teacher blocks, respectively. NIA denotes No Interaction between the student and the teacher. $\gamma(\cdot, \cdot)$ is the cross-entropy loss. $\varphi(\cdot, \cdot)$ is the L_2 loss, penalizing the difference between F_n^S and F_n^T. $a_n \in \{0, 1\}$ denotes a Bernoulli random variable, and indicates whether the n-th loss is added ($a_n = 1$) or ignored ($a_n = 0$). We denote $p_n = P(a_n = 0)$, which represents the probability of ignoring the n-th loss. For simplicity and fair comparison, we set the probability schedule of p_n to be the same as the probability schedule of p_i in IAKD-U.

The experimental results are shown in Tables 3 and 4. Compared with IAKD-U when showing better performance improvement on CIFAR-10 ($p_i \in \{0.7, 0.8, 0.9\}$) and CIFAR-100 ($p_i = 0.9$), the students trained without interaction (w/o IA) achieve less performance improvement than the ones trained with interaction (IAKD-U). This proves interaction mechanism can actually assist the student to achieve much better performance.

4.2.2. Group-wise Interaction

Until now, we achieve IAKD by block-wise interaction (one student block pairs with several teacher blocks). To verify the effectiveness of block-wise interaction, we reduce the interaction level to group-layer-wise [13] interaction (one student group pairs with one teacher group). A group layer is defined as blocks that have outputs of the same spatial size, as suggested in [13]. For simplicity, we only choose IAKD-U to achieve group-layer-wise interaction (denoted as IAKD-U-G). ResNet-44/ResNet-26 are employed as the T/S pair. As shown in Tables 3 and 4, IAKD-U outperforms IAKD-U-G on both datasets. Besides, IAKD-U-G fails to achieve performance improvement on CIFAR-100, because of the significant decrease in the interaction level caused by the group-layer-wise interaction. The student blocks, except the first or the last one in each group, have no chance to interact with the teacher blocks.

---

[2] Replacing the "Conv3×3-BN-ReLU" with "group Conv3×3-BN-ReLU-Conv1×1-BN-ReLU" to achieve the cheap ResNet, as in [34].

Table 2

Evaluations on different architecture combinations with different probability schedules. Average over 5 runs on CIFAR-100. The table gives the Top-1 accuracy. The best results are shown in **bold**.

Setup	(a)	(b)	(c)	(d)	(e)
Teacher	ResNet-44 [33]	ResNet-18 [33]	VGG-19 [35]	ResNet-50x4 [33]	ResNet-56-bottleneck [33]
Student	cheap ResNet-26 [34]	VGG-8 [35]	VGG-11 [35]	ShuffleNet-v1* [36]	cheap ResNet-56-bottleneck [34]
Teacher	71.07%	72.32%	72.74%	72.79%	73.13%
Student	65.92%	68.91%	68.84%	67.11%	71.60%
IAKD-U (p_{start}=0.9)	66.55%(±0.43%)	69.60%(±0.12%)	70.21%(±0.27%)	66.13%(±0.41%)	72.21%(±0.20%)
IAKD-L (p_{start}=0.3)	66.53%(±0.43%)	68.98%(±0.20%)	69.96%(±0.30%)	66.47%(±0.27%)	72.35%(±0.14%)
IAKD-R (p_{start}=0.1)	**67.15%(±0.33%)**	**69.80%(±0.23%)**	**70.94%(±0.15%)**	**68.25%(±0.10%)**	**73.00%(±0.17%)**

* The total number of layers in the ShuffleNet-v1 is 50 layers.

Table 3

Experimental results on CIFAR-10 when training the student without interaction (denoted as "w/o IA") or with group-wise interaction (denoted as "IAKD-U-G"). Better performance is shown in **bold**.

Model	w/o IA	IAKD-U-G	IAKD-U	p_n
ResNet-44 (Teacher)		92.91%		–
ResNet-26 (Student)		92.01%		–
ResNet-26	92.10%(±0.02%)	92.05%(±0.21%)	**92.11%(±0.41%)**	0.7
ResNet-26	92.18%(±0.05%)	92.28%(±0.05%)	**92.31%(±0.30%)**	0.8
ResNet-26	92.18%(±0.06%)	92.37%(±0.05%)	**92.55%(±0.11%)**	0.9

Table 4

Experimental results on CIFAR-100 when training the student without interaction (denoted as "w/o IA") or with group-wise interaction (denoted as "IAKD-U-G"). Better performance is shown in **bold**.

Model	w/o IA	IAKD-U-G	IAKD-U	p_n
ResNet-44 (Teacher)		71.07%		–
ResNet-26 (Student)		68.94%		–
ResNet-26	69.20%(±0.15%)	68.68%(±0.16%)	**69.54%(±0.32%)**	0.9

4.2.3. Visualizations of class activation mapping

To show how the interaction between the teacher and the student network helps the learning of the student, we visualize the class activation mapping (CAM) from ResNet-44 (teacher), ResNet-26 (student) and ResNet-26 (trained with IAKD-R) using Grad-CAM [37], which is commonly used to localize the discriminative regions for image classification. All the networks are trained on CIFAR-100 dataset. The experimental results are shown in Fig. 7. As we can see, the ResNet-26 network trained with IAKD-R (Fig. 7b) have larger CAM areas than the student (Fig. 7d), which are more similar to the teacher CAM (Fig. 7a).Due to the interaction learning strategy, the IAKD-R has activation maps that tend to cover the whole object, while activation maps of student only cover parts of the object in the image. Such ability of precisely highlighting CAM regions makes the small network trained with IAKD-R have better generalization capacity.
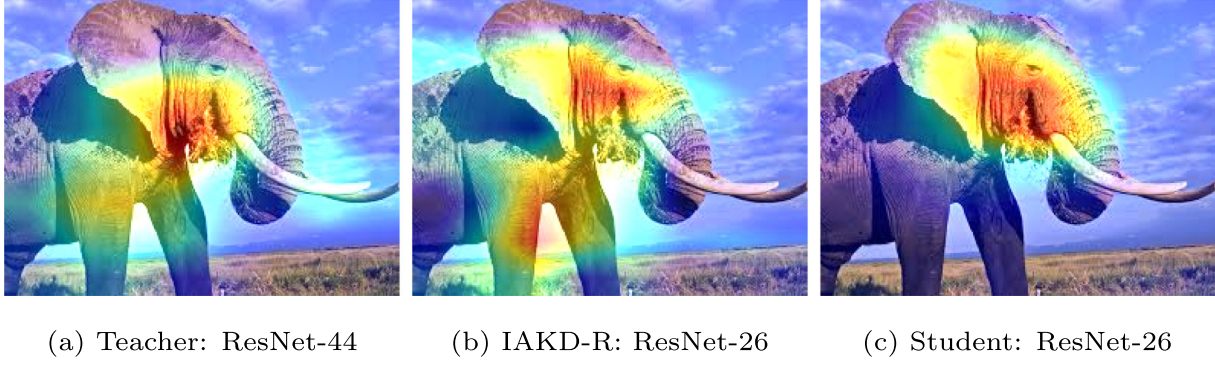
4.3. IAKD-R with deeper teachers

As the depth of a network increases, we can usually get better performance. One might expect that as the teacher becomes more accurate, the features that the teacher extracted may be more powerful and thus serves as better supervision to the student. Therefore, intuitively, we might expect the student benefits from the deeper teacher.

We evaluate if this is true on CIFAR-10 and CIFAR-100 datasets with a deeper teacher (e.g., ResNet-80). The experimental results are shown in Table 5. From these experiments, we find out that deeper teachers with better performance does not necessarily bring more performance improvements on the student. On CIFAR-10, ResNet-26 trained with ResNet-80 gives better distillation results than the one trained with ResNet-44. However, on

CIFAR-100, ResNet-26 trained with ResNet-44 shows better distillation results. One possible reason is that as the teacher becomes more powerful, it can extract more discriminative features for more difficult classification tasks (e.g., classification on CIFAR-100), which makes it more difficult for the outputs of the student blocks to be similar to the corresponding teacher outputs. In other words, due to larger capacity gap between the deeper teacher and the student, the student with much lower capacity is unable to find a solution in its own space that matches well to the deeper teacher, which is also demonstrated in [38].

4.4. Discussions about more knowledge distillation points

From the view of the conventional, *non-interactive* knowledge distillation methods, the end of each set of swapped-in teacher blocks can be regarded as the knowledge distillation point where the knowledge of the teacher network is transferred to the student network. Randomly swapping in the teacher blocks to replace the student blocks implicitly increases the number of knowledge distillation points. To verify more knowledge distillation points do not necessarily lead to more performance improvement and may even harm the student performance, we introduce more connections between the teacher and the student to multi-connection knowledge based methods (*i.e.*, AT [13] and FSP [18]). The experiments are conducted by ResNet-44/ResNet-26 pair on CIFAR-10 and CIFAR-100 datasets. Accordingly, we modify AT to transfer attention maps at the end of each non-shared student block rather than the end of each group, and name it as 9-point AT because there are totally 9 knowledge distillation points. Similarly, we modify FSP to calculate the Gramian matrix by the input and the output of each non-shared student block instead of each group, and name it as 9-point FSP.

(a) Teacher: ResNet-44   (b) IAKD-R: ResNet-26   (c) Student: ResNet-26

Fig. 7. Visualizations of class activation mapping. All the networks are trained on CIFAR-100 dataset.

Table 5

Experimental results with deeper teacher networks on CIFAR-10 and CIFAR-100 datasets. Average over 5 runs. The table gives the Top-1 accuracy, and the best results are shown in **bold**.

Dataset	CIFAR-10				CIFAR-100			
Teacher	ResNet-80		ResNet-44		ResNet-80		ResNet-44	
Student		ResNet-26				ResNet-26		
Teacher	93.49%		92.91%		72.04%		71.07%	
Student		92.01%				68.94%		
IAKD-R	**92.81**%±0.24%(p_{start}=0.9)		92.77%±0.13% (p_{start}=0.9)		69.85%±0.18% (p_{start}= 0.1)		**70.18**%±0.31% (p_{start}=0.1)	

Experimental results are shown in Table 6. It turns out, for conventional, *non-interactive* knowledge distillation methods, more knowledge distillation points do not necessarily lead to more performance improvement. For AT, more distillation points even make the distillation performance worse than the performance of the original AT. This is because excessive supervision causes a hard constraint for the student network. However, as shown in Tables 1 and 2, the proposed IAKD, which also introduced 9 distillation points with ResNet-44/ResNet-26, achieved better performance improvement over 9-point AT and 9-point FSP. This suggests the effectiveness of the interaction mechanism.

4.5. Comparison with state-of-the-arts

We compare our proposed IAKD-R with other representative state-of-the-art knowledge distillation methods: HKD [12], AT [13], FSP [18], Feature distillation [15], AB [19], VID-I [28] and RKD-DA [14]. As the discussion above, all these comparison methods are *non-interactive*. We conduct experiments on three datasets: CIFAR-10 [25], CIFAR-100 [25], and TinyImageNet [39]. TinyImageNet contains images of size 64×64 with 200 classes. Each experiment is repeated 5 times, and we take the mean value as the final report result. For other compared methods, we directly use the author-provided codes if publicly available, or implement those methods based on the original paper. The hyper-parameters of contrastive methods are listed as follows.

1. **HKD** [12]: temperature $\mathscr{T}$ = 4, balance factor: α = 0.9 (Eq. (20) in [16]).
2. **AT** [13]: balance factor: β = 1000 (Eq. (2) in [13]).
3. **FSP** [18]: No balance factor for the task-specific and the distillation loss because the distillation is in the initialization stage where only the distillation loss applies. In the initialization stage, we set λ_i, the hyper-parameter to balance each FSP loss term in Eq. (2) in [18], to 1.
4. **AB** [19]: No balance factor for the task-specific loss and the distillation loss because the distillation is in the initialization stage where only the distillation loss applies. In the initialization

Table 6

The experimental results when non-interactive methods take more knowledge distillation points. The table shows the Top-1 accuracy. Average over 5 runs. The best results are shown in **bold**.

Model/Method	CIFAR-10	CIFAR-100
ResNet-44 (Teacher)	92.91%	71.07%
ResNet-26 (Student)	92.01%	68.94%
9-point AT	92.01%(±0.19%)	69.56%(±0.22%)
AT	92.55%(±0.12%)	69.72%(±0.17%)
9-point FSP	92.45%(±0.27%)	69.07%(±0.43%)
FSP	92.47%(±0.06%)	69.10%(±0.34%)
IAKD-R (Ours)	**92.77%(±0.13%)**	**70.18%(±0.31%)**

stage, there are three hyper-parameters: λ_1, λ_2 and λ_3 to balance the different terms of the distillation loss. We set λ_1, λ_2 and λ_3 to $\frac{1}{4} \times 10^{-3}$, $\frac{1}{2} \times 10^{-3}$, and 1×10^{-3}, respectively (See [19] for explanations of these hyper-parameters).

5. **VID-I** [28]: balance factor: λ_k = 1 (Eq. (2) in [28]).
6. **RKD-DA** [14]: balance factors: λ_{RKD-D}=25 and λ_{RKD-A}=50 (See [14] for explanations of these hyper-parameters).
7. **Feature distillation**[15]: balance factor: α=10^{-3} (Eq. (6) in [15]). In the distillation loss function in Eq. (5) in [15], the loss is multiplied by 2 when the spatial size of the feature maps is decreased by twice as the input moves from shallow to deep layers.

For fair comparison, the optimization configurations are identical for all methods on one dataset.

For CIFAR-10 and CIFAR-100, we use the same optimization configurations as described in Section 4.1. Specially, for two-stage methods like FSP and AB, 50 epochs are used for initialization and 150 epochs are used for classification training. The learning rate at the initialization stage is 0.1 and 0.001 for AB and FSP, respectively. For AB and FSP, the initial learning rate at the classification training stage is 0.1, and is multiplied by 0.1 at 75, 110 epochs. For TinyImageNet, we employ ResNet-80/ResNet-26 as the T/S pair. Horizontal flipping is applied for data augmentation. We optimize the network using SGD with batchsize 128, momentum 0.9 and weight decay 1×10^{-4}. The number of total training

Table 7

Comparison of test accuracy by different knowledge distillation methods on CIFAR-10, CIFAR-100, and TinyImageNet datasets. The best and second best results on Top-1 accuracy are shown in **bold** and <u>underlined</u>, respectively.

Method	Model	CIFAR-10		CIFAR-100		TinyImageNet	
		Size/FLOPs	Top-1	Size/FLOPs	Top-1	Size/FLOPs	Top-1
Teacher	ResNet-44/80*	0.66M/98.51M	92.91%	0.67M/98.52M	71.07%	1.26M/737.24M	55.73%
Student	ResNet-26	0.37M/55.62M	92.01%	0.38M/55.62M	68.94%	0.38M/222.47M	50.04%
HKD [12]			92.90%(±0.19%)		69.59% (±0.26%)		50.25%(±0.46%)
AT [13]			92.55%(±0.12%)		69.72%(±0.17%)		52.07%(±0.21%)
FSP [18]			92.47%(±0.06%)		69.10%(±0.34%)		50.97%(±0.50%)
AB [19]			92.59%(±0.16%)		69.40%(±0.42%)		48.48%(±2.96%)
VID-I [28]			92.48%(±0.10%)		69.90%(±0.23%)		51.98%(±0.15%)
RKD-DA [14]			92.70%(±0.16%)		69.46%(±0.39%)		51.69%(±0.43%)
Feature distillation [15]			92.29%(±0.10%)		69.11%(±0.17%)		52.56%(±0.22%)
IAKD-R (Ours)			92.77%(±0.13%)		<u>70.18%(±0.31%)</u>		<u>53.28%(±0.13%)</u>
IAKD-R+HKD (Ours)			**93.31%(±0.08%)**		**71.02%(±0.16%)**		**53.50%(±0.31%)**

* ResNet-44 is used as the teacher on CIFAR-10 and CIFAR-100, while ResNet-80 is used as the teacher on TinyImageNet.

epochs is 300. The initial learning rate is 0.1 and is multiplied by 0.2 at 60, 120, 160, 200, 250 epochs. Specially, for FSP and AB, 50 epochs are used for initialization and 250 epochs are used for classification training. The learning rate at the initialization stage is 0.1 and 0.001 for AB and FSP, respectively. For AB and FSP, the initial learning rate at the classification training is 0.1, and is multiplied by 0.2 at 60, 120, 160, 200 epochs.

For the proposed IAKD, we use the review schedule as the final probability schedule setting. p_{start} is set to 0.9 for comparison on CIFAR-10, and is set to 0.1 for comparison on CIFAR-100 and TinyImageNet. The configurations are based on the experimental results and arguments in Section 4.1.3.

In Table 7, we list the comparison results on CIFAR-10, CIFAR-100, and TinyImageNet by different methods. As we can see, the proposed IAKD improves the performance of the student network more effectively than most contrastive methods. Meanwhile, the distillation result of IAKD is competitive with HKD on CIFAR-10. We then combine HKD with our proposed IAKD through adding distillation loss to Eq. (5), and denote it as IAKD-R+HKD. Surprisingly, IAKD-R+HKD even outperforms the performance of the original teacher on CIFAR-10. On CIFAR-100 and TinyImageNet, IAKD-R +HKD outperforms all other contrastive methods. IAKD-R also shows highly competitive results on CIFAR-100 and TinyImageNet. In addition, our proposed method, robust as we can see, consistently achieves significant performance improvement on three datasets while most other contrastive methods do not.

We also provide Fig. 8 to show a more intuitive comparison of convergence rate among different knowledge distillation methods. We conduct the experiment on CIFAR-100 dataset, using ResNet-44/ResNet-26 as the teacher/student pair. Compared with non-interactive knowledge distillation methods, IAKD-R gives a special convergence curve. As p_i (the probability of choosing the student path) increases from p_{start} to 1, the performance of IAKD-R shows the tendency to rising continuously instead of converging after a few epochs. The performance of IAKD-R decreases when p_i is reset to p_{start}, but we can also observe the fact that IAKD-R shows the tendency to sustained increase within the range where the learning rate remains unchanged. Such learning process of IAKD-R shows the advantage of interaction between the teacher and the student.

Advantages on learning efficiency. Our proposed IAKD-R is also benefited from better learning efficiency. Conventional knowledge
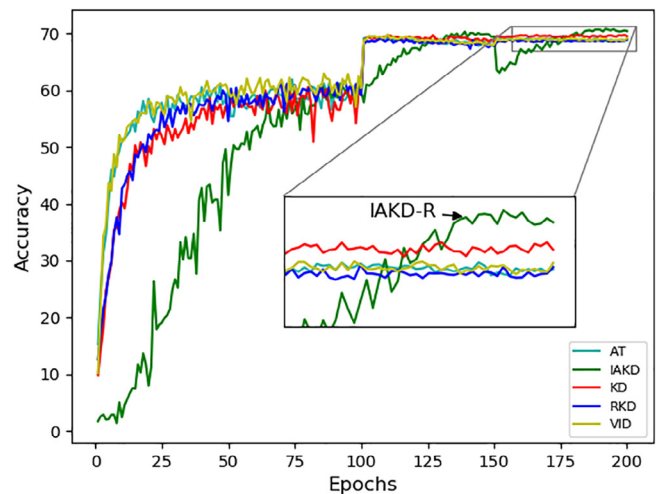
Fig. 8. The comparison of convergence rate among different methods.

distillation methods update all student blocks along the whole training steps. Our work reveals that this laborious distillation is not necessary for promising performance. In our IAKD-R, the student blocks replaced by the teacher ones will not be updated at that step. Thus, in each training step, a *portion* of the student network is updated. For the possible replaced student block, our training epochs number, the mathematical expectation, is 190, 110 and 165 on CIFAR-10, CIFAR-100 and TinyImageNet, respectively, which accounts for 95%, 55% and 55% of other contrastive methods on corresponding datasets. This demonstrates the IAKD-R enjoys efficient learning capacity, and also validates that the student network indeed benefits well from the interaction with the teacher network.

As mentioned in Section 1, our proposed IAKD discards the cumbersome knowledge transformation process and gets rid of the feature extraction of the entire teacher network, we compare the training time of different knowledge distillation methods, using ResNet-80/ResNet-26 teacher-student pair on TinyImageNet dataset. We conduct training time comparison on a single NVIDIA RTX 2080Ti GPU with Intel Core i7-8700 CPU and PyTorch framework. As shown in Table 8, our proposed method gives a highly

Table 8

Training time comparison on TinyImageNet.

Method	Teacher	Student	HKD [12]	AT [13]	FSP [18]	AB [19]	VID-I [28]	RKD-DA [14]	Feature distillation [15]	IAKD-R (Ours)
Time Cost	11 h 47 m	4 h 8 m	6 h 47 m	11 h 29 m	7 h 13 m	4 h 21 m	13 h 44 m	6 h 48 m	7 h 5 m	4 h 40 m

Table 9

Evaluations on different architecture combinations. Average over 5 runs on CIFAR-100. The table gives the Top-1 accuracy. The best and second best results on Top-1 accuracy are shown in **bold** and <u>underlined</u>, respectively. ↓ denotes underperforming the corresponding students trained without knowledge distillation.

Setup	(a)	(b)	(c)	(d)
Teacher	ResNet-44 [33]	ResNet-18 [33]	VGG-19 [35]	ResNet-50x8 [33]
Student	cheap ResNet-26	VGG-8 [35]	VGG-11 [35]	ShuffleNet-v1 [36]
Teacher	71.07%	72.32%	72.74%	75.15%
Student	65.92%	68.91%	68.84%	67.11%
HKD [12]	65.55%(±0.40%) ↓	**70.63%(±0.27%)**	71.09%(±0.13%)	<u>68.58%(±0.43%)</u>
AT [13]	65.90%(±0.37%) ↓	68.54%(±0.40%) ↓	70.93%(±0.14%)	68.13%(±0.30%)
RKD-DA [14]	65.66%(±0.21%) ↓	68.69%(±0.31%) ↓	70.84%(±0.14%)	66.69%(±0.38%) ↓
AB [19]	65.64%(±0.76%) ↓	69.86%(±0.24%)	<u>71.21%(±0.27%)</u>	66.89%(±1.46%) ↓
IAKD-R	**67.15%(±0.33%)**	69.80%(±0.23%)	70.94%(±0.15%)	68.25%(±0.10%)
IAKD-R+HKD	<u>66.11%(±0.72%)</u>	<u>70.51%(±0.22%)</u>	**71.76%(±0.21%)**	**68.65%(±0.29%)**

competitive result in contrast to other methods. It only needs 4 h and 40 min to complete the training process of the student network, which reflects that the high training efficiency of our proposed IAKD.

4.6. Different architecture combinations

To ensure the applicability of IAKD, we explore more architecture combinations on CIFAR-100, as shown in Table 9. We replace the "Conv3 × 3-BN-ReLU" with "group Conv3 × 3-BN-ReLU-Conv1 × 1-BN-ReLU" to achieve the cheap ResNet, as suggested by [34]. The experimental settings are the same as those described in Section 4.5. As we can see from Table 9, in all the architecture combinations, IAKD-R and IAKD-R+HKD can consistently achieve performance improvement over the student network. On the other hand, contrastive methods give different knowledge distillation performance in different settings. In the case of (a), only IAKD-R and IAKD-R+HKD can successfully achieve performance improvement over the student cheap ResNet-26. AT and RKD-DA fail to improve the student performance in the case of (b). RKD-DA and AB fail to achieve performance improvement in the case of (d). Therefore, the proposed IAKD, a novel knowledge distillation method without additional distillation losses, can also achieve promising performance improvement in different teacher/student settings.

5. Conclusion

In this paper, we proposed a simple yet effective knowledge distillation method named Interactive Knowledge Distillation (IAKD). By randomly swapping in the teacher blocks to replace the student blocks in each step, we accomplish the interaction between the teacher network and the student one. To properly utilize the interaction mechanism during distillation process, we proposed three kinds of probability schedules. Our IAKD does not need additional distillation losses to drive the distillation process, and is complementary to conventional knowledge distillation methods. Extensive experiments demonstrated that the proposed IAKD could boost the performance of the student network on diverse image classification tasks. Instead of mimicking the teacher's feature representation space, our proposed IAKD aims to directly leverage the teacher's powerful feature transformation ability to motivate the student, providing a new perspective for knowledge distillation.

CRediT authorship contribution statement

Shipeng Fu: Conceptualization, Methodology, Software, Writing - original draft, Visualization. **Zhen Li:** Data curation, Visualization, Investigation. **Zitao Liu:** Validation, Formal analysis, Resources. **Xiaomin Yang:** Validation, Supervision, Resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

References

[1] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: CVPR, 2017, pp. 4700–4708.

[2] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: CVPR, 2019, pp. 4401–4410.

[3] K. Sun, B. Xiao, D. Liu, J. Wang, Deep high-resolution representation learning for human pose estimation, in: CVPR, 2019, pp. 5693–5703.

[4] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, W. Wu, Feedback network for image super-resolution, in: CVPR, 2019, pp. 3867–3876.

[5] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, P. Torr, Res2net: A new multi-scale backbone architecture, IEEE TPAMI doi:10.1109/TPAMI.2019.2938758..

[6] P. Luo, Z. Zhu, Z. Liu, X. Wang, X. Tang, Face model compression by distilling knowledge from neurons, in: AAAI, vol. 30, 2016..

[7] G. Chen, W. Choi, X. Yu, T. Han, M. Chandraker, Learning efficient object detection models with knowledge distillation, in: NeurIPS, 2017, pp. 742–751.

[8] Y. Chen, N. Wang, Z. Zhang, Darkrank: Accelerating deep metric learning via cross sample similarities transfer, in: AAAI, vol. 32, 2018..

[9] T. Wang, L. Yuan, X. Zhang, J. Feng, Distilling object detectors with fine-grained feature imitation, in: CVPR, 2019, pp. 4933–4942.

[10] Y. Liu, K. Chen, C. Liu, Z. Qin, Z. Luo, J. Wang, Structured knowledge distillation for semantic segmentation, in: CVPR, 2019, pp. 2604–2613.

[11] Y. Hou, Z. Ma, C. Liu, C.C. Loy, Learning lightweight lane detection cnns by self attention distillation, in: ICCV, 2019, pp. 1013–1021..

[12] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: NeurIPS Deep Learning and Representation Learning Workshop, 2015..

[13] S. Zagoruyko, N. Komodakis, Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer, in: ICLR, 2017..

[14] W. Park, D. Kim, Y. Lu, M. Cho, Relational knowledge distillation, in: CVPR, 2019, pp. 3967–3976.

[15] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, J.Y. Choi, A comprehensive overhaul of feature distillation, in: ICCV, 2019, pp. 1921–1930..

[16] Y. Tian, D. Krishnan, P. Isola, Contrastive representation distillation, in: ICLR, 2020.

[17] S.-I. Mirzadeh, M. Farajtabar, A. Li, H. Ghasemzadeh, Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher, arXiv preprint arXiv:1902.03393..

[18] J. Yim, D. Joo, J. Bae, J. Kim, A gift from knowledge distillation: Fast optimization, network minimization and transfer learning, in: CVPR, 2017, pp. 4133–4141.

[19] S.Y. Byeongho Heo, Minsik Lee, J.Y. Choi, Knowledge transfer via distillation of activation boundaries formed by hidden neurons, in: AAAI, vol. 33, 2019, pp. 3779–3787..

[20] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: ECCV, Springer, 2014, pp. 818–833..

[21] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: ICML, 2014, pp. 647–655..

[22] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: CVPRW, 2014, pp. 806–813.
[23] J. Ba, R. Caruana, Do deep nets really need to be deep?, in: NeurIPS, vol. 27, 2014, pp. 2654–2662..
[24] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: Hints for thin deep nets, in: ICLR, 2015..
[25] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Tech. rep., 2009..
[26] J. Kim, S. Park, N. Kwak, Paraphrasing complex network: Network compression via factor transfer, in: NeurIPS, 2018, pp. 2760–2769.
[27] S. Hyun Lee, D. Ha Kim, B. Cheol Song, Self-supervised knowledge distillation using singular value decomposition, in: ECCV, 2018, pp. 339–354..
[28] S. Ahn, S.X. Hu, A. Damianou, N.D. Lawrence, Z. Dai, Variational information distillation for knowledge transfer, in: CVPR, 2019, pp. 9163–9171.
[29] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, Y. Duan, Knowledge distillation via instance relationship graph, in: CVPR, 2019, pp. 7096–7104.
[30] S. Lee, B.C. Song, Graph-based knowledge distillation by multi-head self-attention network, in: BMVC, 2019.
[31] F. Tung, G. Mori, Similarity-preserving knowledge distillation, in: ICCV, 2019, pp. 1365–1374..
[32] G. Xu, Z. Liu, X. Li, C.C. Loy, Knowledge distillation meets self-supervision, in: ECCV, Springer, 2020, pp. 588–604..
[33] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: CVPR, 2016, pp. 770–778.
[34] E.J. Crowley, G. Gray, A.J. Storkey, Moonshine: Distilling with cheap convolutions, in: NeurIPS, 2018, pp. 2888–2898..
[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, ICLR (2015).
[36] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: CVPR, 2018, pp. 6848–6856..
[37] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: ICCV, 2017, pp. 618–626..
[38] J.H. Cho, B. Hariharan, On the efficacy of knowledge distillation, in: ICCV, 2019, pp. 4794–4802.
[39] L. Yao, J. Miller, Tiny imagenet classification with convolutional neural networks, CS 231N..
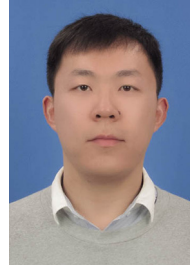
Shipeng Fu received the B.S. degree in information security from Sichuan University, Chengdu, China, where he is currently working toward the Ph.D. degree in signal and information processing. His research interests include deep learning, knowledge distillation, and image restoration.
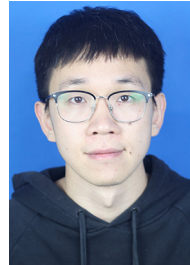
Zhen Li received the M.S. degree in electronics and communication engineering from Sichuan University, Chengdu, China, in 2019. He is currently working toward the Ph.D. degree in Nankai University, Tianjin, China. His research interests include efficient learning and image restoration.

Zitao Liu is the Director of AI Solution at TAL Education Group (NYSE:TAL). His research is in the area of machine learning, and includes contributions in the areas of artificial intelligence in education, multimodal knowledge representation and user modeling. He has published his research in highly ranked conference proceedings and serves as the executive committee of the International AI in Education Society and top tier AI conference/workshop organizers/program committees. Before joining TAL, Zitao was a senior research scientist at Pinterest and received his Ph.D degree in Computer Science from University of Pittsburgh.

Xiaomin Yang received the B.S. and Ph.D. degrees in communication and information system from Sichuan University, Chengdu, China, in 2002 and 2007, respectively. She held a postdoctoral position with the University of Adelaide. She is currently a Professor with the College of Electronics and Information Engineering, Sichuan University. Her research interests are image processing and pattern recognition.