Student Helping Teacher: Teacher Evolution via Self-Knowledge Distillation

Zheng Li,1,4* Xiang Li,2 Lingfeng Yang,2,4 Jian Yang,2 Zhigeng Pan3†

1 Hangzhou Normal University
2 Nanjing University of Science and Technology
3 Nanjing University of Information Science and Technology
4 MEGVII Technology
lizheng1@stu.hznu.edu.cn

Abstract

Knowledge distillation usually transfers the knowledge from a pre-trained cumbersome teacher network to a compact student network, which follows the classical teacher-teaching-student paradigm. Based on this paradigm, previous methods mostly focus on how to efficiently train a better student network for deployment. Different from the existing practices, in this paper, we propose a novel student-helping-teacher formula, Teacher Evolution via Self-Knowledge Distillation (TESKD), where the target teacher (for deployment) is learned with the help of multiple hierarchical students by sharing the structural backbone. The diverse feedback from multiple students allows the teacher to improve itself through the shared feature representations. The effectiveness of our proposed framework is demonstrated by extensive experiments with various network settings on two standard benchmarks including CIFAR-100 and ImageNet. Notably, when trained together with our proposed method, ResNet-18 achieves 79.15% and 71.14% accuracy on CIFAR-100 and ImageNet, outperforming the baseline results by 4.74% and 1.43%, respectively. The code is available at: https://github.com/zhengli427/TESKD.

Introduction

Deep convolutional networks (CNNs) have achieved remarkable success in various computer vision applications, including image classification (He et al. 2016; Huang et al. 2017; Xie et al. 2017; Li et al. 2019), object detection (Girshick 2015; Ren et al. 2015; Tian et al. 2019; Li et al. 2020a; 2021a), semantic segmentation (Long, Shelhamer, and Darrell 2015; Ronneberger, Fischer, and Brox 2015) and pose estimation (Newell, Yang, and Deng 2016; Chu et al. 2017). With the growing number of model parameters, a large amount of computational resource is required to achieve state-of-the-art accuracy. However, networks with millions of parameters are hard to be deployed to platforms with limited computing resources. To address this issue, a variety of network compression approaches such as quantization (Chen et al. 2015; Wu et al. 2016), network pruning (Molchanov et al. 2016; Li et al. 2016) and knowledge distillation (Hinton, Vinyals, and Dean 2015), have been exploited to obtain a small network that can work as well as the large network while effectively reducing the computational costs and memory consumption.

Knowledge distillation, as one of the main network compression techniques, becomes increasingly popular recently. It usually transfers the knowledge of a cumbersome pre-trained teacher network in the form of soft predictions (Hinton, Vinyals, and Dean 2015) or intermediate representations (Romero et al. 2014; Zagoruyko and Komodakis 2016; Yim et al. 2017), aiming at improving the generalization ability of a compact student network. Such a learning process can be typically viewed as a teacher-teaching-student paradigm, where the fixed strong teacher teaches weak student and the well-trained student is used for final deployment, as shown in Fig. 1(a). Instead of knowledge transfer between a static teacher and a compact student, another distillation paradigm student-teaching-student has also been proposed in DML (Zhang et al. 2018). In this paradigm, the pre-trained teacher network no longer exists, and all the individual networks are treated as students. Two or more students are trained simultaneously in a cooperative peer-teaching manner and gain extra knowledge from each other. Such a one-stage learning process significantly improve the training efficiency. In test, the best student is selected for final deployment. The whole mutual learning procedure is shown in Fig. 1(b).

Different from the existing paradigms, we propose a novel student-helping-teacher formula, Teacher Evolution via Self-Knowledge Distillation (TESKD), which introduces multiple hierarchical classifiers (students) to facilitate the learning of the backbone network (teacher) for deployment, as illustrated in Fig. 1(c). Specifically, we design a backbone teacher network with target complexity for deployment, and construct multiple student sub-networks in a FPN-like (Lin et al. 2017) way by sharing various stages of teacher backbone features. During training, once the teacher provides high-quality soft labels to guide the hierarchical students, it also offers the opportunity for the teacher to make meaningful improvements based on students’ diverse feedback via the shared intermediate representations. In order to obtain the effective feedback, we propose the Mixed Fusion Module (MFM) to build hierarchical student sub-networks with a top-down architecture. Specifically, MFM consists of both addition and concatenation operators, which diversely and sufficiently bridge the information flow be-

---

*This work is done when Zheng Li is a research intern at Megvii
†Corresponding author
Figure 1: Comparison of three distillation paradigms. The blue line is the forward path and the dashed green line is the soft label distillation. The arrow points to the distilled target. As opposed to previous approaches that follow the classical teacher-teaching-student or student-teaching-student paradigm, we propose a novel student-helping-teacher distillation paradigm. When the target teacher distills knowledge to the students, it can also be significantly improved based on the students’ diverse feedback via shared intermediate representations. We omit the supervised learning loss and feature distillation loss for simplicity.

The overall contributions of this paper are summarized as follows:

- To our best knowledge, we are the first to propose the student-helping-teacher paradigm in knowledge distillation area. The target deployed teacher can make improvements by learning from the feedback of multiple hierarchical students, through their shared backbone features in our proposed one-stage self-distillation framework.
- To further improve the effectiveness of the feedback, we propose the Mixed Fusion Module (MFM) to build multiple hierarchical student sub-networks with a top-down architecture.
- Extensive experiments on CIFAR-100 and ImageNet-2012 datasets based on four popular network architectures demonstrate the effectiveness of our proposed self-distillation framework.

Related Work

Knowledge Distillation Knowledge distillation aims at effectively learning a compact and comparable student model by transferring the knowledge of a pre-trained cumbersome teacher model, which follows the classical teacher-teaching-student distillation paradigm. Hinton et al. (Hinton, Vinyals, and Dean 2015) proposes to match the output probability distributions of two models by minimizing the Kullback-Leibler (KL) divergence loss. To improve distillation performance, existing methods have designed various forms of knowledge transfer. The method in FitNet (Romero et al. 2014) proposes to let the student mimic the intermediate representations of the teacher network. AT (Zagoruyko and Komodakis 2016) tries to transfer the attention map of teacher to student. FSP (Yim et al. 2017) proposes to generate the FSP matrix from the layer feature and use this matrix to guide the learning of student. The paraphraser and translator network is introduced in FT (Kim, Park, and Kwak 2018) to aid the knowledge transfer process. Many works (Xu and Liu 2019; Tung and Mori 2019; Yun et al. 2020) have also explored the relationship between data samples in distillation. RKD (Park et al. 2019) proposes a distillation framework that penalizes structural differences in relationships based on distance and angle. CCKD (Peng et al. 2019) introduces a kernel-based method for capturing complex inter-instance correlations. In addition to the traditional classification task, knowledge distillation was also used to effectively obtain a light-weight network when training and designing networks for object detection (Li, Jin, and Yan 2017; Chen et al. 2017), semantic segmentation (Liu et al. 2019), and human pose estimation (Zhang, Zhu, and Ye 2019; Li et al. 2021b).

Unlike the two-stage offline knowledge distillation methods we described above, online counterparts reduces the training process to one-stage by eliminating the need for a large pre-trained teacher network. DML (Zhang et al. 2018) proposes a novel student-teaching-student paradigm which conducts distillation collaboratively for peer student networks by learning from each other. DCM (Yao and Sun 2020) adds multiple auxiliary classifiers to the certain hidden layers of student networks in the two-way distillation method. ONE (Zhu, Gong et al. 2018) trains a multi-branch student network while the teacher is established on-the-fly. OKDDip (Chen et al. 2020) introduces the concept of two-level distillation and uses the self-attention mechanism (Vaswani et al. 2017) to build diverse peer networks. KDCL (Guo et al. 2020) removes the shared low-level structures and enables models of various capacities to learn collaboratively in the ensemble method. Li et al. (Li et al. 2020b) further improves the ensemble-based distillation method by enhancing the branch diversity. Walawalkar et al. (Walawalkar, Shen, and Savvides 2020) proposes to simultaneously train multiple student networks with different compression rates in one training procedure.

Self-Knowledge Distillation Self-knowledge distillation can be approximately divided into two parts: data augmentation based approaches (Xu and Liu 2019; Yun et al. 2020) and auxiliary branch based approaches (Zhang et al. 2019; Ji et al. 2021). The data augmentation based method trans-
fers the knowledge between the different augmented versions of the same training data without the need for additional teacher network. Xu et al. (Xu and Liu 2019) proposes to learn the consistent feature or posterior distributions between the different augmented versions of the same training samples. CS-KD (Yun et al. 2020) proposes the class-wise regularization term that penalizes the predictions between different samples of the same label.

The auxiliary branch based self-distillation method introduces multiple convolutional branches on the backbone network to exploit its own knowledge. By simplifying the structure of the auxiliary branches, the self-distillation method achieves higher distillation efficiency than the online counterparts. DKS (Sun et al. 2019) first explores the possibility of utilizing the knowledge learned by the auxiliary branch to regularize the training of the backbone network. The backbone network is treated as the teacher and student at the same time. In BYOT (Zhang et al. 2019), the backbone network is divided into several blocks according to its structures and depth. Additional bottleneck and fully connected layers are set after each block, which constitutes the auxiliary classifier. Each auxiliary classifier in the network is treated as a student. During training, the backbone network plays the role of a teacher, providing the high-accuracy soft labels to train the multiple student sub-networks. BYOT typically follows the classical teacher-teaching-student paradigm and all well-trained students can be utilized independently to meet the needs of different compression rates. But it neglects that the teacher network is also being optimized during training. The trained backbone teacher network shows superior performance to the baseline method. This gives us the inspiration that the teacher can actually learn from students and get meaningful improvements. Inspired by FPN (Lin et al. 2017), FRSKD (Ji et al. 2021) constructs an auxiliary self-teacher network that distills the refined feature maps to the original classifier. The soft label generated by the self-teacher is also utilized for the distillation. A major drawback of the above work is that they neglect the advantages of the diverse supervision signals provided by multiple auxiliary classifiers.

**Teacher Evolution via Self-Knowledge Distillation**

In this section, we first briefly review the basic concept of the classical knowledge distillation that follows the teacher-teaching-student paradigm. Then we propose our novel student-helping-teacher framework and introduce the mixed fusion module.

**Background and Notations**

Knowledge distillation (Hinton, Vinyals, and Dean 2015), as one of the main network compression techniques (Wu et al. 2016; Molchanov et al. 2016), has been widely used in many tasks (Li, Jin, and Yan 2017; Liu et al. 2019; Zhang, Zhu, and Ye 2019). The traditional two-stage distillation process usually starts with a pre-trained cumbersome teacher network. Then a compact student network will be trained under the supervision of the teacher network in the form of soft predictions or intermediate representations (Romero et al. 2014; Yim et al. 2017). After the distillation, the student can master the expertise of the teacher and thus is used for efficient deployment. Such a learning process can be typically viewed as a teacher-teaching-student paradigm. Given the labeled classification dataset \(D = \{(x_i, y_i)\}_{i=1}^{n}\), the Kullback-Leibler (KL) divergence loss is used to minimize the discrepancy between the soften output probabilities of the student network and teacher network:

\[
L_{KL} = \sum_{i=1}^{n} T^2 KL(q(s)_i, q(t)_i),
\]

where \(T\) is the temperature parameter to scale the smoothness of distribution, \(q(s)_i\) and \(q(t)_i\) denotes the soften probability produced by the student and the teacher, respectively. The predicted probabilities are calculated with a softmax layer built on logits \(t_i\), i.e., \(q_i = softmax(t_i/T)\). A larger temperature \(T\) will make the probability distribution softer.

To train a multi-class classification network, we also minimize the traditional Cross-Entropy (CE) loss between the predicted probabilities \(q(s)_i\) and the ground-truth one-hot label \(y_i\) of each training sample:

\[
L_{CE} = \sum_{i=1}^{n} CE(q(s)_i, y_i).
\]

With both hard labels and soft labels, the final loss function of the conventional knowledge distillation is written with the balancing parameter \(\lambda\) as follows:

\[
L_{total} = L_{CE} + \lambda L_{KL}.
\]

**Our Method**

As opposed to traditional distillation approaches that follow the teacher-teaching-student paradigm, in this work, we explore a novel and reverse aspect that studies the student-helping-teacher scheme, aiming at improving the teacher network through the effective feedback from multiple students. An overview of our proposed distillation framework is illustrated in Fig. 2. The whole framework mainly consists of two components:

1. The backbone teacher network \(T\) with \(B\) stages for deployment, which meets the requirement of the target complexity. Note that it is different from the most existing practices that usually train student network for deployment.
2. \(B−1\) auxiliary hierarchical student sub-networks \(S\) constructed with a top-down architecture, each of which shares the feature maps from the corresponding teacher backbone stages.

Specifically, given an input RGB image, we can obtain the backbone teacher feature set \(T = \{T_1, T_2, ..., T_B\}\) and the output probability distribution \(q(t)\) after the feed forward computation. The student sub-network takes the last teacher feature \(T_B\) as initial input and obtain \(S_{B}\). For every \(1 \leq b \leq B−1\), it generates higher resolution features \(S_b\) iteratively by upsampling spatially coarser, but semantically stronger feature maps \(S_B\). Then these features \(S_b\) are fused with semantically weaker, but spatially finer, features \(T_{b−1}\) via the proposed Mixed Fusion Module (MFM).
Figure 2: An overview of the our proposed student-helping-teacher method, Teacher Evolution via Self-Knowledge Distillation (TESKD). We divide the target backbone teacher into four blocks and construct three hierarchical student sub-networks #1, #2 and #3 in a FPN-like way by sharing various stages of the teacher backbone features. When the teacher provides high-quality soft labels to guide the hierarchical student, it also make improvements based on students’ diverse feedback via the shared intermediate representations.

The details of MFM are elaborated in the following section. In this way, we can obtain the hierarchical student feature set $S = \{S_1, S_2, ..., S_B\}$. Additional convolutional blocks and fully connected layers are set after each student block, each of which serves as an independent student classification model to generate the soft probability $q(s)_b$ for distillation.

The backbone teacher network has two learning tasks during distillation. It not only learns to generate high-accuracy labels for prediction and distillation, but also tries to provide robust intermediate features to guide the learning of multiple hierarchical students. Through such optimization, the backbone network can learn more generalized features and significantly outperforms the baseline network. In test, multiple auxiliary student sub-networks can be simply removed while keeping the well-trained teacher network for deployment.

The Mixed Fusion Module

Inspired by MLN (Wang et al., 2018), we propose the Mixed Fusion Module (MFM) which consists of both addition and concatenation operators, leading to the multiple hierarchical student sub-networks with a top-down architecture. The architecture of our proposed MFM is depicted in Fig. 2(b).

Previous works (Lin et al., 2017; Ronneberger, Fischer, and Brox, 2015) use addition and concatenation operation independently to fuse the features from the encoder (backbone) and decoder network. But simply adding or concatenating two features may impede the information flow or bring redundancy to the network, resulting in performance degradation. In this work, we mix these two operations to effectively bridge the information flow between the backbone teacher and multiple hierarchical students, in order to combine both the advantages of them and avoid the possible limitations.

Specifically, with a spatially coarser feature $S_b$, we up-sample the spatial resolution by a factor of 2. The following $1 \times 1$ convolution operation is used to align the channel dimension between two input features. Then the upsampled features are fused with the corresponding student features $S_{b-1}$ through the addition and concatenation operations, which can be formulated as Eqn. (4):

$$S_b = \text{Conv}(f_t(T_b) + f_s(S_{b+1})) \quad || \quad f_s(S_{b+1}), b \in [1, B - 1],$$

where the symbol “+” and “||” denotes addition and concatenation operation, respectively. $f_t(\cdot)$ is the function of upsampling and the $1 \times 1$ convolution operation. $f_s(\cdot)$ is the function of the $1 \times 1$ convolution block, which is used to align the number of channels between two features. \(\text{Conv}\) is a $1 \times 1$ convolution block, which fuse the features after the concatenation operation and halves the number of feature channels to $C_b$, as shown in Fig. 2(b). This fusion process is iterated until the latest feature map $S_1$ is generated. We set the channel dimension of all student features to $C_B$, which is the same channel dimension as the last teacher feature $T_B$.

Through our proposed MFM, even the shallowest student sub-network can still obtain sufficient spatial and semantic information at the same time, resulting in better representation ability. Multiple stronger student networks can provide more diverse and sufficient feedback signals, from which the backbone teacher can learn and be significantly improved.
Feature Distillation

After the feed forward computation through the FPN-like multi-branch distillation framework, we can obtain the proportionally sized student’s feature set $S = \{S_1, S_2, ..., S_B\}$, teacher’s feature set $T = \{T_1, T_2, ..., T_B\}$ and branches’ intermediate feature set $F = \{F_1, F_2, ..., F_{B-1}\}$. Specifically, we aim to make the backbone teacher classifier guide the learning process of all auxiliary student classifiers. The guided layer is selected as the last layer of the convolutional block in the auxiliary branch. We minimize the L2 loss between the intermediate feature maps in the main teacher classifier and other auxiliary student classifiers, which can be written as:

$$L_{FEA} = \sum_{b=1}^{B-1} \|F_b - T_B\|_2^2.$$  \hspace{1cm} (5)

Note that the convolution blocks in student sub-network are made of regular convolution layers, batch normalization (Ioffe and Szegedy 2015) and ReLU activation function. For each student, instead of the bottleneck design in BYOT (Zhang et al. 2019), we construct a simple single conv-bn-relu block to first reduce the feature map size (e.g., to $4 \times 4$ for CIFAR and $7 \times 7$ for ImageNet). Then a global average pooling layer is applied for feature distillation.

Overall To get a better understanding of our proposed TESKD, the full training procedure is summarized in Algorithm 1. The overall loss of our proposed self-distillation framework is given as:

$$L_{total} = \alpha_1 \sum_{b=1}^B L_{CE}(q_b, y) + \alpha_2 \sum_{b=1}^{B-1} L_{KL}(q(t)_b, q(s)) + \beta L_{FEA}. \hspace{1cm} (6)$$

where $\alpha_1, \alpha_2$ and $\beta$ are the hyperparameters to control the impact of each loss term which also satisfy $\alpha_1 + \alpha_2 = 1$. $q$ denotes all the predicted results generated by the whole network, i.e. $\{q(t), q(s)_1, q(s)_2, q(s)_3\}$. The first loss term is the total cross-entropy loss to the ground truth labels of both all auxiliary students and the backbone teacher network.

**Experiments**

We evaluate our proposed TESKD framework on four popular neural networks (VGG [Simonyan and Zisserman 2014], ResNet, SENet [Hu, Shen, and Sun 2018], ResNeXt (Xie et al. 2017)) and two benchmark datasets (CIFAR-100 (Krizhevsky and Hinton 2009), ImageNet-2012 (Deng et al. 2009)). We compare our method with closely related self-distillation and online distillation works (ONE (Zhu, Gong et al. 2018), CS-KD (Yun et al. 2020), FRSKD (Ji et al. 2021) and BYOT). Detailed ablation studies on the network components are also conducted to demonstrate its effectiveness. All evaluations are made in comparison to state-of-the-art approaches based on standard experimental settings and reported in means and standard deviations over 3 runs.

**Algorithm 1: Student Helping Teacher: Teacher Evolution via Self-Knowledge Distillation.**

**Input:** Labelled Training dataset $D = \{(x_i, y_i)\}_{i=1}^n$; Training Epoch Number $\epsilon$; A target teacher model $\theta^t$; Three hierarchical student models $\theta^{s}_1$, $\theta^{s}_2$, $\theta^{s}_3$.

**Output:** A well-trained target teacher model;

**Initialize:** Epoch $\epsilon=1$; Randomly initialize $\theta^s_1$ and $\theta^s_2$; \(\begin{aligned}
&1:\text{while } \epsilon \leq \epsilon \text{ do } \text{ end while}\n&2:\text{Forward propagation and obtain the intermediate features in teacher and multiple students. (Eqn. (4));}
&3:\text{Compute the predictions of all auxiliary students } q^{(s)}(s) \text{ and teacher } q(t); \text{ (Eqn. (4));}
&4:\text{Compute the Cross-Entropy loss } L_{CE}. \text{ (Eqn. (2));}
&5:\text{Distill the knowledge from the teacher model to multiple hierarchical student models. (Eqn. (1));}
&6:\text{Align the feature representations between teacher and multiple students. (Eqn. (5));}
&7:\text{Obtain the final loss function. (Eqn. (6));}
&8:\text{Update the model parameters } \theta^s_1 \text{ and } \theta^s_2; \text{ (6).}
&\end{aligned}\)

**Dataset.** The CIFAR-100 dataset consists of colored natural images with $32 \times 32$ pixels. The training and testing sets contain 50K and 10K images, respectively. Same as previous works (Zhang et al. 2019; Ji et al. 2021), the network structures are modified to fit the tiny images in CIFAR-100.

The ImageNet-2012 classification dataset is more challenging than CIFAR. It contains 1.2M images for training, 50K for validation, from 1K classes. The resolution of input images after pre-processing is $224 \times 224$.

**Implementation details.** All the methods are implemented by PyTorch (Paszke et al. 2019). For CIFAR-100, we follow the standard data augmentation scheme for all training images as in [Zhang et al. 2019; Ji et al. 2021], i.e. random cropping and horizontal flipping. We use the stochastic gradient descents (SGD) as the optimizer with momentum 0.9 and weight decay 5e-4 during training. The learning rate starts from 0.1 and is divided by 10 at 100 and 150 epochs, for a total of 200 epochs. For ImageNet, we set the initial learning rate to 0.1 and divide the learning rate by 10 at 30 and 60 epochs, for a total of 90 epochs. Weight decay is set to 1e-4. Same data augmentation scheme is adopted as in Ji et al. 2021. The mini-batch size is 128 and 256 for CIFAR and ImageNet, respectively. We set $\alpha_1$ to 0.2 and 0.8 in Eqn. (6) for CIFAR-100 and ImageNet, respectively. $\beta$ is usually set to 1e-7. We set $B$ to 4 for all methods.

**Experiments on CIFAR-100**

Table 3 shows the top-1 classification accuracy on CIFAR-100 based on seven varying capacity state-of-the-art neural networks. Note that we set the branch number in ONE to 3 and the reported results are the averaged accuracy of all branches. For BYOT, we report the results of the main network (i.e. backbone network). From this table, we can observe that all different networks benefit significantly from our proposed TESKD, particularly for small models achiev-
Table 1: Accuracy (%) comparison of various distillation approaches on CIFAR-100 dataset. “w/o F” denotes without feature distillation. The best performing model is indicated as boldface.

| Models          | VGG-16      | VGG-19      | ResNet-18    | ResNet-34    | SENet-34    | ResNeXt-18   | ResNeXt-34   |
|-----------------|-------------|-------------|--------------|--------------|-------------|--------------|--------------|
| Baseline        | 72.70±0.24  | 72.81±0.25  | 74.40±0.08   | 74.51±0.16   | 74.54±0.14  | 75.38±0.17   | 76.08±0.06   |
| ONE             | 73.24±0.10  | 72.13±0.11  | 77.01±0.28   | 77.24±0.23   | 76.19±0.15  | 77.42±0.09   | 78.25±0.14   |
| CS-KD           | 73.43±0.14  | 73.13±0.10  | 77.26±0.14   | 76.82±0.29   | 76.85±0.18  | 78.32±0.26   | 77.35±0.11   |
| BYOT            | 73.70±0.25  | 74.18±0.19  | 77.22±0.20   | 77.93±0.06   | 77.58±0.14  | 74.28±0.21   | 75.23±0.27   |
| FRSKD           | 69.24±0.05  | 70.38±0.11  | 77.51±0.15   | 77.58±0.09   | 77.46±0.18  | 78.13±0.03   | 77.11±0.11   |
| Ours w/o F      | 74.73±0.20  | 74.44±0.14  | 78.61±0.09   | 78.90±0.15   | 78.69±0.06  | 79.57±0.14   | 79.49±0.23   |
| Ours            | 74.90±0.17  | 75.01±0.22  | 79.14±0.11   | 79.60±0.16   | 78.97±0.06  | 79.65±0.12   | 79.77±0.08   |

Figure 3: Top-1 accuracy (%) comparison of student sub-networks on CIFAR-100 dataset. “Stu” is the abbreviation of “student”.

Table 2: Top-1 classification accuracy (%) comparison with other SOTA self-distillation methods for ResNet-18 on ImageNet-2012 dataset.

| Models      | Baseline | BYOT       | FR SKD | Ours      |
|-------------|----------|------------|--------|-----------|
| ResNet-18   | 69.71±0.12 | 69.84±0.12 | 70.17±0.12 | 71.14±0.12 |

Experiment on ImageNet

Table 3 shows the results on a large-scale image classification dataset ImageNet-2012 (Deng et al. 2009) based on

ResNet-18. We compare our method with two closely related state-of-the-art self-knowledge distillation approaches on ImageNet. The ResNet-18 network trained with our proposed method shows 71.14% accuracy, outperforming the baseline by 1.43% margin. This demonstrates that our method can still be applied to large-scale dataset effectively.

Compared with Traditional Distillation

Traditional knowledge distillation methods usually follow the teacher-teaching-student paradigm. Table 3 compares the classification accuracy of our proposed method with five traditional distillation methods on CIFAR-100, including vanilla KD, FitNet (Romero et al. 2014), AT (Zagoruyko and Komodakis 2016), SP (Tung and Mori 2019) and VID (Ahn et al. 2019). The results of the vanilla KD are also included for comparison. Each column includes the results of corresponding student models which are generated by the super-
Table 3: Top-1 accuracy comparison with traditional distillation method on CIFAR-100 dataset.

| Teacher   | Student     | Baseline | KD  | FitNet | AT  | SP  | VID | Ours        |
|-----------|-------------|----------|-----|--------|-----|-----|-----|-------------|
| ResNet-101| ResNet-18    | 74.40±0.08| 77.12±0.24| 77.38±0.18| 77.41±0.25| 77.84±0.14| 77.34±0.18| 79.14±0.11|
| ResNet-101| ResNet-34    | 74.51±0.16| 77.70±0.21| 77.97±0.06| 77.43±0.16| 77.50±0.05| 77.73±0.14| 79.60±0.16|
| ResNeXt-152| ResNeXt-18  | 75.38±0.17| 78.81±0.18| 79.08±0.08| 78.85±0.13| 78.84±0.16| 78.72±0.13| 79.65±0.12|
| ResNeXt-152| ResNeXt-34  | 76.08±0.06| 78.65±0.06| 78.77±0.16| 78.72±0.19| 78.68±0.13| 78.58±0.08| 79.77±0.08|

Table 4: Ablation Study: Impact of different connection operations in TESKD. (ResNet-18 on CIFAR-100 dataset)

| Method                          | Top-1 Acc |
|---------------------------------|-----------|
| w/o MFM-No Connection           | 78.40±0.13|
| w/o MFM-Concat (i.e., FPN Style)| 78.63±0.09|
| w/o MFM-Add (i.e., UNet Style)  | 78.77±0.10|
| w/o Knowledge Distillation       | 77.25±0.22|
| TESKD                           | 79.14±0.11|

Impact of the student sub-networks

We evaluate the impact of the student sub-networks on the performance of our branch-based self-distillation approach. As shown in Fig. 4, we have three sub-networks in our proposed framework. When we reduce the number of student sub-networks, the diversity of feedback that students provide to the teacher will also decrease. This will affect the teacher’s performance. The results are summarized in Table 5. Multiple student sub-networks are removed one-by-one to measure their effect. Noted that if we remove all students, our method will not be able to perform the distillation operation, so we keep the last student #3. From Table 5, we can observe that when we remove the auxiliary students one-by-one, the performance of the teacher model used for deployment gradually decreases. This verifies our idea that feedback from multiple students can indeed affect teacher’s learning.

Conclusion

Different from the existing teacher-teaching-student and student-teaching-student paradigm, in this paper, we propose a novel student-helping-teacher formula, Teacher Evolution via Self-Knowledge Distillation (TESKD) where the target teacher is learned with the help of multiple hierarchical students by sharing structural backbone. The well-trained teacher is used for final deployment. Extensive experiments have validated the effectiveness of our proposed TESKD on two popular benchmark datasets.
References

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9163–9171.

Chen, D.; Mei, J.-P.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online Knowledge Distillation with Diverse Peers. In Proceedings of the AAAI Conference on Artificial Intelligence, 3430–3437.

Chen, G.; Choi, W.; Yu, X.; Han, T.; and Chandraker, M. 2017. Learning efficient object detection models with knowledge distillation. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 742–751.

Chen, W.; Wilson, J.; Tyree, S.; Weinberger, K.; and Chen, Y. 2015. Compressing neural networks with the hashing trick. In International conference on machine learning, 2285–2294. PMLR.

Chu, X.; Yang, W.; Ouyang, W.; Ma, C.; Yuille, A. L.; and Wang, X. 2017. Multi-context attention for human pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1831–1840.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, 248–255. Ieee.

Girshick, R. 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, 1440–1448.

Guo, Q.; Wang, X.; Wu, Y.; Yu, Z.; Liang, D.; Hu, X.; and Luo, P. 2020. Online knowledge distillation via collaborative learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11020–11029.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531.

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, 7132–7141.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4700–4708.

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

Ji, M.; Shin, S.; Hwang, S.; Park, G.; and Moon, I.-C. 2021. Refine Myself by Teaching Myself: Feature Refinement via Self-Knowledge Distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10664–10673.

Kim, J.; Park, S.; and Kwak, N. 2018. Paraphrasing complex network: Network compression via factor transfer. arXiv preprint arXiv:1802.04977.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report.

Li, H.; Kadav, A.; Durandanovic, I.; Samet, H.; and Graf, H. P. 2016. Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710.

Li, Q.; Jin, S.; and Yan, J. 2017. Mimicking very efficient network for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6356–6364.

Li, X.; Wang, W.; Hu, X.; Li, J.; Tang, J.; and Yang, J. 2021a. Generalized focal loss v2: Learning reliable localization quality estimation for dense object detection. In Advances in Neural Information Processing Systems.

Li, Z.; Huang, Y.; Chen, D.; Luo, T.; Cai, N.; and Pan, Z. 2020b. Online Knowledge Distillation via Multi-branch Diversity Enhancement. In Proceedings of the Asian Conference on Computer Vision.

Li, Z.; Ye, J.; Song, M.; Huang, Y.; and Pan, Z. 2021b. Online Knowledge Distillation for Efficient Pose Estimation. arXiv preprint arXiv:2108.02092.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2117–2125.

Liu, Y.; Chen, K.; Liu, C.; Qin, Z.; Luo, Z.; and Wang, J. 2019. Structured knowledge distillation for semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2604–2613.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3431–3440.

Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; and Kautz, J. 2016. Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In European Conference on Computer Vision, 483–499. Springer.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3967–3976.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.
et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, 8026–8037.

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation congruence for knowledge distillation. In Proceedings of the IEEE International Conference on Computer Vision, 5007–5016.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28: 91–99.

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

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, 234–241. Springer.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sun, D.; Yao, A.; Zhou, A.; and Zhao, H. 2019. Deeply-supervised knowledge synergy. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6997–7006.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF international conference on computer vision, 9627–9636.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 1365–1374.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, 5998–6008.

Walawalkar, D.; Shen, Z.; and Savvides, M. 2020. Online Ensemble Model Compression using Knowledge Distillation. In European Conference on Computer Vision, 18–35. Springer.

Wang, W.; Li, X.; Yang, J.; and Lu, T. 2018. Mixed Link Networks. ArXiv, abs/1802.01808.

Wu, J.; Leng, C.; Wang, Y.; Hu, Q.; and Cheng, J. 2016. Quantized convolutional neural networks for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4820–4828.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1492–1500.

Xu, T.-B.; and Liu, C.-L. 2019. Data-distortion guided self-distillation for deep neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 5565–5572.