Distilling a Powerful Student Model via Online Knowledge Distillation

Shaojie Li[, Mingbao Lin[, Yan Wang[, Yongjian Wu[, Yonghong Tian[, Fellow, IEEE[, Ling Shao[, Fellow, IEEE, and Rongrong Ji[, Senior Member, IEEE

Abstract—Existing online knowledge distillation approaches either adopt the student with the best performance or construct an ensemble model for better holistic performance. However, the former strategy ignores other students’ information, while the latter increases the computational complexity during deployment. In this article, we propose a novel method for online knowledge distillation, termed feature fusion and self-distillation (FFSD), which comprises two key components: FFSD, toward solving the above problems in a unified framework. Different from previous works, where all students are treated equally, the proposed FFSD splits them into a leader student set and a common student set. Then, the feature fusion module converts the concatenation of feature maps from all common students into a fused feature map. The fused representation is used to assist the learning of the leader student. To enable the student to absorb more diverse information, we design an enhancement strategy to increase the diversity among students. Besides, a self-distillation module is adopted to convert the feature map of deeper layers into a shallower one. Then, the shallower layers are encouraged to mimic the transformed feature maps of the deeper layers, which helps the students to generalize better. After training, we simply adopt the leader student, which achieves superior performance, over the common students, without increasing the storage or inference cost. Extensive experiments on CIFAR-100 and Imagenet demonstrate the superiority of our FFSD over existing works. The code is available at https://github.com/SJLeo/FFSD.

Index Terms—Feature fusion, knowledge distillation, online distillation, self-distillation.

I. INTRODUCTION

D

EEP neural networks (DNNs) have achieved unprecedented success in various visual tasks. Nevertheless, their extensive memory and computational requirements hinder their deployment in resource-limited devices. Several methods have been developed to derive a lightweight model with negligible performance compromise. Examples include network pruning [1], [2], parameter quantization [3], [4], low-rank decomposition [5], [6], and knowledge distillation [7], [8].

Among them, knowledge distillation has received particular attention, transferring knowledge from a high-capacity teacher [7]–[9], or an online ensemble [10]–[12], to a student model. As illustrated in Fig. 1(a), Traditional knowledge distillation methods use a two-stage optimization where a cumbersome teacher network has to be trained in advance in order to yield a high-capacity model, which then serves as supervision information to guide the training of a lightweight student network. Though progress has been made, these methods heavily rely on an appropriate teacher model. As stressed in [13] and [14], it is difficult to choose a suitable teacher model for the student model.

This has motivated the community to simplify the training procedure by exploring online knowledge distillation [10], [12], where a collection of student models are trained simultaneously in a collaborative manner without the involvement of a teacher model. As shown in Fig. 1, existing online knowledge distillation can be implemented by either mutual learning [10], [15], [16] or ensemble learning [11], [12], [17]–[19]. The former aligns the soft outputs of all students so as to allow message passing among them. Then, the student model with the optimal performance is adopted as the final model. However, the message passing does not guarantee that one single student will carry all the information of the ensemble. This limits the distillation performance. In contrast, the latter constructs a virtual teacher by ensembling the outputs of all the students, which are then distilled back to foster each student. Nevertheless, this group has to retain all the student models so as to ensemble their output logits together to pursue better performance, which significantly increases the memory consumption and computational complexity since all models have to be stored in disks and evaluated during inference. Thus, online knowledge distillation remains an open problem.
In this article, we present a novel online knowledge distillation method, introducing two key modules, that is, feature fusion and self-distillation (FFSD), in order to solve the above problems in a unified framework. Specifically, we construct a common student set and a leader student, which share the same network architecture. The common student set is trained through mutual learning. In order to take full advantage of the rich information of all common students in the set, we design a feature fusion module similar to an autoencoder to fuse the output feature maps from all common students and then distill it to the leader student. First, it encodes the common students’ output feature maps into a meaningful compact feature map with the same size as the leader student, and then the leader student is encouraged to mimic this fused feature map. Meanwhile, the output feature map of the leader student is decoded back to match the concatenated feature maps of all students. The diverse feature map is further distilled to other students. In addition, we design a self-distillation module that converts high-order supervision information (the fused feature map and the diverse feature map) to low-order supervision information for shallower network layers. Self-distillation facilitates the flow of supervision information from deeper to shallower network layers. It thus enriches the supervision information in the training process. After training, only the leader student is retained for deployment, as it achieves superior performance over all other students.

Our contributions are summarized as follows.

1) A novel online knowledge distillation method, FFSD, is proposed. We design a feature fusion module equipped with a diversity enhancement strategy to integrate the knowledge of students and distill it to the leader student, which improves the generality of the final model.

2) A self-distillation module is proposed to convert high-order supervision information to low-order cues for shallower network layers, which provides richer information for model training.

3) Extensive experiments on two benchmarks demonstrate the effectiveness of our FFSD.

II. RELATED WORK

A. Traditional Knowledge Distillation

Traditional distillation works transfer knowledge from a cumbersome teacher model to a lightweight student model. As such, a large-scale model has to be trained in advance, based on which various knowledge definitions and transfer strategies are proposed to boost the performance of the student model. The pioneering work [7] performs knowledge representation of a teacher model using the softmax output layer, which converts the logit into a soft probability with a temperature parameter. Following this, a large number of works proposed new forms of knowledge, such as output logits [20], [21], intermediate feature maps [8], [22], [23], attention maps [9], second-order statistics [24], contrastive features [25], [26], or structured knowledge [27]–[29]. Another group of methods focus on transfer strategies so as to enable the student model to inherit knowledge from the teacher model. An intuitive solution is to use the Kullback–Leibler divergence or $\ell_p$-loss when the knowledge falls on the soft logit [7], [30] or intermediate representation [8], [9]. Beyond that, Wang et al. [31] utilized the adversarial training scheme in generative adversarial networks (GANs) [32] to transfer knowledge. Jang et al. [33] considered meta-learning to selectively transfer knowledge. In [34], a reinforcement learning-based architecture-aware distillation was proposed to pass the structural knowledge to the student. Recently, there are some works [35]–[38] that used knowledge distillation for GAN compression to ensure effective compression. The surveys [39], [40] summarized the development of knowledge distillation in recent years.

B. Online Knowledge Distillation

Online knowledge distillation has emerged as an alternative that eliminates the dependency on the teacher model. It builds knowledge distillation based on a collection of student models that collaborate through simultaneous training. To this end, Zhang et al. [10] proposed a deep mutual learning strategy where pair-wise students are encouraged to learn from each other by a mimicry loss based on the Kullback–Leibler
divergence. Chen et al. [17] performed two-level distillation by training multiple auxiliary peers and one group leader, separately. The former aims to boost peer diversity, while the latter transfers knowledge from an ensemble of auxiliary peers to the group leader. In [15], online knowledge distillation was built at a feature-map level using the adversarial training framework. Kim et al. [19] fused the intermediate representations of subnetworks, passing the result to an auxiliary classifier. Then, the knowledge from the auxiliary classifier is delivered back to each subnetwork for mutual teaching. In [11], [12], and [18], all student branches are assembled to construct a stronger teacher model, which is, in turn, distilled back to the students to enhance the model learning. EnD2 [41] distilled the distribution of the predictions of the ensemble into a single model. It enables a single model to retain both the performance of the ensemble as well as the ability of uncertainty estimation. EnD2 also performs diversity enhancement to capture uncertain information. However, we enhance the diversity among common students so that student leaders can obtain higher benefits from feature fusion.

C. Self-Distillation

Self-distillation, originally proposed by Furlanello et al. [42], has received a great deal of attention recently, due to its distillation of knowledge within the network itself without the aid of other models. Augmentation-based works [43], [44] focus on self-distillation via data augmentation of the input images. Hou et al. [45] and Zhang et al. [46] distilled deeper parts of the network as the conceptual teacher model to guide the learning of shallower modules. Li et al. [20] revisited knowledge distillation as a type of learned label smoothing regularization and accordingly proposed a novel teacher-free knowledge distillation framework where the student model learns from itself or a manually designed regularization distribution. In addition, some works have achieved superior performance by applying self-distillation on object detection [47], [48] and super-resolution [49].

III. PROPOSED METHOD

A. Preliminaries

Consider a group of $n + 1$ student models $\mathcal{G} = \{S_i\}_{i=0}^{n}$, all of which share the same network structure and consist of $L$ convolutional layers. For each model $S_i$, we denote the output of the feature maps in the $l$th layer as $F_i^l \in \mathbb{R}^{C_i \times H_i \times W_i}$, where $C_i$, $H_i$, $W_i$ denote the channel number, height, and width of a feature map, respectively. Besides, given a labeled dataset $\mathcal{D} = \{(x, y)\}$ with $K$ classes, the logit produced by student $S_i$ is denoted as $z_i^f \in \mathbb{R}^K$.

Then, the prediction probability of the softmax layer is represented by $p_i$, with the $k$th class computed as

$$p_i^k = \frac{\exp(z_i^k / T)}{\sum_{k=1}^{K} \exp(z_i^k / T)} \tag{1}$$

where $T \geq 1$ is the temperature parameter used to soften the output probability. When $T = 1$, it degenerates to the original softmax output. For ease of representation, we consider $p_i$ as having temperature $T = 1$; otherwise, we rewrite it as $\hat{p}_i$.

Existing literature encourages all students $S_i \in \mathcal{G}$ to learn from each other. Then, the resulting model for deployment falls on the optimal student or the ensemble one. As discussed in Sec.I, the former ignores the efficacy of other students, while the latter increases the resource burden. Differently, in this article, we innovatively propose to regard $S_0$ as the leader student and the remaining $\hat{G} = \mathcal{G} - \{S_0\}$ as a common student set. Then, students in $\hat{G}$ learn in a collaborative manner, while the leader student $S_0$ is responsible for learning the knowledge from the common students. All students mentioned below represent the common students in the set $\hat{G}$.

Following [10], we first define the training objective $\mathcal{L}_{base}$ for the collaborative learning of the students as

$$\mathcal{L}_{base} = \mathcal{L}_{CE}(y, p_i) + T^2 \sum_{j=1, j \neq i}^{n} \mathcal{L}_{KL}(\hat{p}_i || \hat{p}_j) \tag{2}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss between the one-hot ground-truth label $y$ and the prediction $p_i$. The $\mathcal{L}_{KL}$ from $p_i$ to $\hat{p}_j$ is computed as

$$\mathcal{L}_{KL}(\hat{p}_j || \hat{p}_i) = \sum_{k=1}^{K} \hat{p}_j^k \log \frac{\hat{p}_j^k}{\hat{p}_i^k}. \tag{3}$$

We multiply $\mathcal{L}_{KL}$ with $T^2$ because the gradients produced by the soft predictions are scaled by $1/T^2$.

B. Feature Fusion

To maximize the usage of the students’ information, we design a feature fusion module in Fig. 3, to fuse the feature maps of all students. The result is then distilled to strengthen the capacity of the leader student $S_0$.

Specifically, we first concatenate the feature maps in the $l$th convolutional layers of all common students, that is, $\{F_i^l\}_{i=1}^{n}$, the result of which is denoted as $F_f$. The fusion module encodes the concatenated feature maps into a meaningful compact feature map $\hat{F}_f$ with the same size as the leader student. This fused feature map is then passed into a fusion classifier supervised by the ground-truth labels and the ensemble logit of students $z_f = (1/n) \sum_{i=1}^{n} z_i$ [10], [11]. We further denote the output logit of the fusion classifier as $z_f$. By transferring these logits into prediction probabilities using (1), the training objective for the fusion classifier is computed as

$$\mathcal{L}_{fusion} = \mathcal{L}_{CE}(y, p_f) + T^2 \mathcal{L}_{KL} (\hat{p}_f || \hat{p}_f). \tag{4}$$
We aim to transfer the high-quality information from the fused feature map to the leader student. To this end, we encourage the last-layer output of the leader student $F_L^m$ to learn from the fused feature map. Meanwhile, the output feature map of the leader student is decoded back to match the concatenated feature maps of all the students to ensure the effectiveness of the feature map learned by the leader student. Hence, we can derive our optimization objective for the output feature map of the leader student as follows:

$$L_{F_L}^m = \left\| \frac{F_L^m}{\|F_L^m\|_2} - \frac{F_f}{\|F_f\|_2} \right\|_2^2 + \left\| \frac{\sigma(F_L^m)}{\|\sigma(F_L^m)\|_2} - \frac{F_x}{\|F_x\|_2} \right\|_2^2 \quad (5)$$

where $\|\cdot\|_2$ is the $\ell_2$-norm and $\sigma(\cdot)$ aligns the channel dimension of $F_L^m$ to $F_x$.

The layer-wise feature amalgamation is performed from multiple teachers in [50], but we only perform feature fusion in the last layer, which is then distilled to the shallower layers. During the fusion process, we set up an additional classifier to supervise the quality of the fused features, which is not available in [50]. Also, as shown in Fig. 2, the output representations in mutual learning [10] or ensemble learning [12] tend to be unified, preventing the leader student from additional information of the feature fusion. Thus, it is necessary to diversify the student outputs, which is ignored in [50]. An intuitive solution can resort to minimizing the negative reconstruction error on the intermediate outputs [8], [51], [52] as

$$L_{div} = -\frac{1}{L} \sum_{l=1}^n \sum_{j=1, j \neq i}^L \sum_{m=1}^L \left\| \frac{F_L^m}{\|F_L^m\|_2} - \frac{F_f^m}{\|F_f^m\|_2} \right\|_2^2 \quad (6).$$

However, (6) raises some issues: 1) Significant computational complexity. A total of $n(n-1)L \ell_2$-norm distances have to be computed; 2) Task independence. The $\ell_2$-norm loss is to reduce the overall error, which will shift the attention of feature maps to a task-independent position; and 3) Attention inconsistency. The per-layer loss is calculated separately, ignoring the coherence of attentions across different layers.

To reduce the diversity computation, we propose to train the first student $S_1$ using (2), where student $S_i(i > 1)$ only performs diversity enhancement calculation with student $S_{i-1}$. Thus, our diversity enhancement learning transfers each student’s knowledge to the next peer student in a one-way chain manner. However, diversifying feature maps from all layers still incurs high complexity. Fortunately, we observe using only the last layer of each residual block for ResNet [53] can perform well. We denote the selected feature maps of student $S_i$ as $\{F_i^m\}_{m=1}^M \in \{F_i\}_{i=1}^L$ for diversity. Note $F_i^m = F_L^m$. The diversity is essentially enabled to enable the attention of students to concentrate on different image positions. We first extract the attention for each feature map $F_i^m$ [9] as

$$A_i^m = \sum_{c=1}^{C_m} F_i^m(c, \ldots)^2 \quad (7)$$

where $C_m$ denotes the channel number of the $m$th layer. Then, the diversity enhancement attention map $\tilde{A}_i^m$ is

$$\tilde{A}_i^m = \left( \frac{P}{2} - A_i^m \right) \times \text{sign}(A_i^m - t) + \frac{P}{2} \quad (8)$$

where $P = \|A_i^m\|_2$ and $t$ is set as the $(H^m \times W^m / 3)$th smallest number in $A_i^m$. The goal of (8) is to shift attention $A_i^m$ to the slightly weaker areas, while maintaining the attention value of task-independent areas. Specially, $\text{sign}(A_i^m - t)$ is used to determine whether the area is task-dependent. When the area is task-independent $(A_i^m < t, \text{sign}(A_i^m - t) = -1)$, the value of $\tilde{A}_i^m$ is equal to $A_i^m$, and $P = A_i^m$; otherwise, we shift attention to the slightly weaker areas by changing the activation value of the task-dependent area into $P - A_i^m$. Our enhancement for $S_i(i > 1)$ to replace (6) is

$$L_{div} = \sum_{m=1}^M \left\| \frac{A_i^m}{\|A_i^m\|_2} - \frac{\tilde{A}_i^m}{\|\tilde{A}_i^m\|_2} \right\|_2^2. \quad (9)$$

In what follows, we detail our self-distillation to solve the problem of attention inconsistency.

C. Self-Distillation

This section introduces a novel self-distillation module to convert high-order information to low-order information for shallower layers. It consists of $M - 1$ blocks, each of which learns to map $F_i^{m+1}$ back to $F_i^m$. The block is stacked with a transpose convolutional layer, a batch normalization layer, and a ReLU layer. We denote the feature maps of each block as $\{F_i^m\}_{i=1}^{M-1}$ in a top-down order and their attention maps $\{A_i^m\}_{i=1}^{M-1}$ are calculated by (7). Each student is equipped with a self-distillation module, since each student has its own feature mapping. We train the self-distillation module together with the students and, in turn, use it to distill the fused/diversity-enhanced features to the student model.

1) Training of Self-Distillation Module: As shown in Fig. 4(a), the self-distillation module takes the feature maps $\{F_i^m\}_{m=1}^M$ as the training datasets, in which the feature map $F_i^m$ of the last-layer output serves as the input, and the feature maps $\{F_i^m\}_{m=1}^{M-1}$ serve as the target of each block. For student $S_i$, the training objective of its self-distillation module is

$$L_{sdm} = \sum_{m=1}^{M-1} \left\| \frac{A_i^m}{\|A_i^m\|_2} - \frac{A_i^m}{\|\tilde{A}_i^m\|_2} \right\|_2^2 + \alpha \sum_{m=1}^{M-1} \left\| \frac{F_i^m}{\|F_i^m\|_2} - \frac{F_i^m}{\|F_i^m\|_2} \right\|_2^2 \quad (10)$$

where $\alpha$ balances the two loss terms. Due to the limited learning ability of the self-distillation module, we prefer to use it to learn the simple mapping of attention maps. Though an individual self-distillation block cannot completely map $F_i^{m+1}$ back to $F_i^m$, we encourage its output $F_i^m$ to be close to $F_i^m$.

2) Application of Self-Distillation Module: Thanks to the good feature mapping ability of the self-distillation module, we can distill the last layer of the diversity enhancement objective to the shallower layers through the self-distillation module as shown in Fig. 4(b). This achieves the diversity enhancement
of the whole network, while ensuring the attention consistency and task dependence mentioned in Section III-B. Besides, we can use the self-distillation to distill the fused feature map to guide the training of shallower layers of the leader student.

For diversity enhancement, we first transform the diversity attention objective $\bar{A}_{m_1}^n$ back into the diversity feature objective $\bar{F}_{m_1}^n$ as the input of the self-distillation module. Then, the self-distillation module outputs the diversity objective $\{\bar{A}_{m_1}^n\}_{m_1=1}^{M-1}$ of the shallower layers. The diversity enhancement objective of student $S_i (i > 1)$ thus becomes

$$L_{\text{div}} = \sum_{m_1=1}^{M-1} \left\| \bar{A}_{m_1}^n - \bar{A}_{m_1-1}^n \right\|^2_2 + \left\| \frac{\bar{A}_{m}^n}{\bar{A}_{m}^n} - \frac{\bar{A}_{m_1}^n}{\bar{A}_{m_1}^n} \right\|^2_2. \quad (11)$$

The final objective of the common students is as follows:

$$L_{\text{stu}} = L_{\text{base}} + \lambda_{\text{div}} L_{\text{div}} \quad (12)$$

where $\lambda_{\text{div}}$ controls the importance of each term.

Similarly, we use our proposed self-distillation module to distill the fused feature map to shallower layers of the leader student. Specifically, the self-distillation module of the leader student takes the fusion feature map $F_f$ as its input, and outputs the shallower layers’ target feature maps $\{F_{m}^m\}_{m=1}^{M-1}$. The corresponding self-distillation loss becomes

$$L_{\text{self}} = \sum_{m=1}^{M-1} \left\| \frac{F_{0}^m}{F_{0}^m} - \frac{F_{0}^m}{F_{0}^m} \right\|^2_2. \quad (13)$$

The final objective of the leader student is as follows:

$$L_{S_i} = L_{CE} + T^2 L_{KL} + \lambda_{\text{cka}} L_{\text{cka}} + \lambda_{\text{self}} L_{\text{self}} \quad (14)$$

where $\lambda_{\text{cka}}$ and $\lambda_{\text{self}}$ control the importance of each term.

The overall framework of our FFSD is illustrated in Fig. 5. The training process can be referred to Alg. 1.

IV. EXPERIMENTS

A. Experimental Settings

1) Datasets and Architecture: To evaluate the efficacy of our FFSD online knowledge distillation method, we conduct experiments on two widely used datasets, CIFAR-100 [54] and ImageNet [55]. CIFAR-100 contains 50 k images with 100 object classes for training and 10k images for testing. ImageNet is a large-scale dataset containing 1.28 M training images and 50 k validation images of 1000 object classes. The size of each image is $32 \times 32$ for CIFAR-100 and $224 \times 224$ for ImageNet. To verify the generalization of our proposed method on different network architectures, we conduct experiments using ResNet [53], WRN [56], GoogLeNet [57], and DenseNet [58].

2) Implementation Details: We use stochastic gradient descent (SGD) with Nesterov momentum to optimize the training objective. The initial learning rate, momentum, and weight decay are set to 0.1, 0.9, and 1e-4, respectively. For CIFAR-100, the models are trained with a batch size of 128 for 300 epochs and the learning rate is divided by 10 after 150 and 225 epochs. For ImageNet, we train all student models for 90 epochs with a batch size of 256. The learning rate warms up to 0.8 linearly in five epochs and is divided by 10 after 30 and 60 epochs. The feature fusion module uses the same implementation details described above. The self-distillation module is optimized by the ADAM optimizer with an initial learning rate of 0.001, using the same batch size, weight decay, and learning rate decay strategy as the student model. The number of common students and temperature $T$ are set to 2.

Our hyper-parameters include $\lambda_{\text{div}}, \lambda_{\text{cka}}, \lambda_{\text{self}}, \text{ and } \alpha$. For each hyper-parameter, we search its optimal value using grid search with others fixed. For different networks and datasets,
Fig. 5. Framework of our FFSD. First, students 1 and 2 learn from each other in a collaborative way. Then, by shifting the attention of student 1 and distilling it to student 2, the diversity is enhanced among students. Lastly, the feature fusion module fuses all the students’ information into a fused feature map. The fused representation is then used to assist the learning of the leader student. After training, we simply adopt the leader student for deployment.

**TABLE I**

EXPERIMENTAL RESULTS ON CIFAR-100. THE “BASELINE” TRAINS THE MODEL USING GROUND-TRUTH LABELS ONLY, AND “ENS” REPRESENTS THE ACCURACY OF THE CORRECT PREDICTION OF EITHER STUDENT NETWORK. “FUSION” AND “LEADER” REPRESENT THE ACCURACY OF THE FUSION CLASSIFIER AND THE LEADER STUDENT, RESPECTIVELY. ALL RESULTS ARE COMPUTED AS THE MEAN (STANDARD DEVIATIONS) OF THREE RUNS

| Model      | Baseline (%) | Common Student1 (%) | Common Student2 (%) | Ens (%) | Fusion (%) | Leader (%) | Gain(†) |
|------------|--------------|---------------------|---------------------|---------|------------|------------|---------|
| ResNet-20  | 68.58 ± 0.26 | 72.07 ± 0.21        | 71.95 ± 0.12        | 77.44 ± 0.14 | 73.43 ± 0.13 | 72.70 ± 0.13 | 4.12 ± 0.35 |
| ResNet-32  | 69.96 ± 0.30 | 74.30 ± 0.10        | 74.34 ± 0.21        | 79.93 ± 0.11 | 76.04 ± 0.17 | 74.85 ± 0.05 | 4.90 ± 0.25 |
| ResNet-56  | 71.55 ± 0.50 | 75.64 ± 0.23        | 75.78 ± 0.43        | 81.51 ± 0.16 | 77.28 ± 0.26 | 75.80 ± 0.11 | 4.25 ± 0.48 |
| WRN-16-2   | 71.97 ± 0.09 | 75.51 ± 0.12        | 75.44 ± 0.25        | 80.26 ± 0.25 | 76.69 ± 0.11 | 75.81 ± 0.08 | 3.83 ± 0.03 |
| WRN-40-2   | 75.58 ± 0.17 | 78.85 ± 0.26        | 78.83 ± 0.21        | 83.95 ± 0.26 | 80.24 ± 0.29 | 79.14 ± 0.03 | 3.47 ± 0.07 |
| GoogLeNet  | 78.28 ± 0.24 | 81.51 ± 0.07        | 81.58 ± 0.16        | 85.28 ± 0.13 | 82.42 ± 0.08 | 81.60 ± 0.25 | 3.32 ± 0.43 |
| DenseNet-40-12 | 73.70 ± 0.10 | 76.87 ± 0.25        | 77.01 ± 0.06        | 81.60 ± 0.12 | 78.34 ± 0.14 | 77.39 ± 0.23 | 3.68 ± 0.33 |

**TABLE II**

RESULTS OF OUR FFSD COMPARED WITH SEVERAL STATE-OF-THE-ART METHODS ON CIFAR-100. IN KD AND AT, WE USE THE PRE-TRAINED RESNET-56 AND WRN-40-2 AS THE TEACHER MODEL OF RESNET-32 AND WRN-16-2, RESPECTIVELY

| Method      | ResNet-32 (%) | Gain(†) | WRN-16-2 (%) | Gain(†) |
|-------------|---------------|---------|--------------|---------|
| Baseline    | 69.96         | -       | 71.97        | -       |
| KD [7]      | 72.87         | 2.91    | 73.79        | 1.82    |
| AT [9]      | 71.23         | 1.27    | 73.70        | 1.73    |
| DML [10]    | 73.64         | 3.68    | 74.63        | 2.66    |
| AFD [15]    | 74.03         | 4.07    | 75.33        | 3.36    |
| AMLN [16]   | 74.69         | 4.73    | 75.56        | 3.59    |
| ONE [12]    | 73.39         | 3.43    | 74.84        | 2.87    |
| FFL [19]    | 74.44         | 4.48    | 75.26        | 3.29    |
| KDCL [18]   | 74.30         | 4.34    | 75.50        | 3.53    |
| OKD Dip [17] | 74.60       | 4.64    | 75.31        | 3.34    |
| FFSD(Ours)  | 74.85         | 4.90    | 75.81        | 3.84    |

all students is of great help to the final result, and it is inappropriate to only adopt the optimal student. However, such improvements require retention of all students, increasing the storage and inference cost. Thus, we distill the knowledge from the feature fusion module to the leader student, which yields a 3.32%–4.90% improvement. Among the models compared, FFSD achieves a 4.90% improvement on ResNet-32 and 3.83% on WRN-16-2, results of which are superior to all other students.

2) Quantitative Comparison on CIFAR-100: As shown in Table II, we compare FFSD with several state-of-the-art methods on ResNet-32 and WRN-16-2. FFSD surpasses most knowledge distillation methods, including traditional knowledge distillation KD [7] and AT [9], mutual learning-based DML [10], AFD [15] and AMLN [16], and ensemble learning-based ONE [12], FFL [19], KDCL [18], and OKD Dip [17]. For example, with ResNet-32, FFSD achieves

**TABLE III**

RESULTS OF OUR FFSD COMPARED WITH SEVERAL ONLINE KNOWLEDGE DISTILLATION METHODS ON IMAGENET

| Method      | Top1-Acc (%) | Gain(†) |
|-------------|--------------|---------|
| Baseline    | 69.7         | -       |
| DML [10]    | 69.8         | 0.1     |
| ONE [12]    | 70.2         | 0.5     |
| KDCL [18]   | 70.4         | 0.7     |
| FFSD(Ours)  | 70.9         | 1.2     |
| Baseline    | 73.2         | -       |
| DML [10]    | 74.0         | 0.8     |
| ONE [12]    | 74.1         | 0.9     |
| KDCL [18]   | 74.4         | 1.2     |
| FFSD(Ours)  | 74.7         | 1.5     |

B. Experimental Results

1) Results on CIFAR-100: We first evaluate FFSD on CIFAR-100 in Table I. After feature fusion, the fusion classifier improves the accuracy over the baseline and students. This demonstrates that fusing the information of
TABLE IV
COMPARISON BETWEEN DIFFERENT DIVERSITY ENHANCEMENT STRATEGIES. DML DOES NOT ADOPT ANY DIVERSITY ENHANCEMENT STRATEGY. L2 USES (6) FOR DIVERSITY ENHANCEMENT. FFSD IS OUR PROPOSED DIVERSITY ENHANCEMENT STRATEGY. “COSINE” REPRESENTS THE COSINE SIMILARITY BETWEEN THE TWO STUDENT MODELS.

| Model     | DML [10] | L2 | FFSD |
|-----------|----------|----|------|
|           | 2Net Avg | Fusion | Cosine | 2Net Avg | Fusion | Cosine | 2Net Avg | Fusion | Cosine |
| ResNet-32 | 73.64%   | 75.15% | 0.2635 | 73.19%   | 75.03% | 0.2351 | 74.25%   | 75.04% | 0.2550 |
| WRN-16-2  | 74.63%   | 76.18% | 0.2787 | 74.69%   | 75.99% | 0.2570 | 75.41%   | 76.69% | 0.2491 |

an accuracy of 74.85%, which is higher than AMLN's 74.69%. In addition, with WRN-16-2, FFSD can achieve a 3.84% performance improvement, which is superior to AMLN's 3.59% and KDCL's 3.53%. Similarly, End2 [41] also distills the prediction distribution from an ensemble into a single model. Following End2 [41], we compare FFSD with End2 on VGG-16 [59]. FFSD achieves an accuracy of 75.87%, while End2 has only 73.7%. It is worth noting that End2 needs an ensemble of 10 models to capture the diversity of the ensemble, which makes End2 hard to be extended to a larger model, like ResNet-34.

3) Results on ImageNet: We further conduct experiments on the large-scale ImageNet dataset. We choose two most popular models, ResNet-18 and ResNet-34, for verification. As shown in Table III, with ResNet-18, FFSD achieves 70.9% top-1 accuracy, which is superior to that of DML and ONE. Besides, Common student1 and common student2 achieve 70.15% and 70.18% accuracy, respectively, which are also higher than baseline. For ResNet-34, FFSD can obtain 1.5% gains, outperforming the baseline model and other online distillation methods. Even the performance of common students (student1: 74.16% & student2: 74.25%) in FFSD is higher than that of ONE. Hence, FFSD can well generalize to complex datasets.

4) Impact of Student Number: We increase the number of common students in Fig. 6. The accuracy of the students, the leader student, and the fusion classifier increase as the number of students grows. In fact, the fusion classifier achieves an astonishing accuracy of 78.26% after fusing the output feature maps of six students. Keeping only the optimal student after online knowledge distillation wastes the effective knowledge of other students. We observe that the accuracy of the leader student is always higher than the average accuracy of students, even when the number of students is up to six.

C. Detailed Analysis

1) Necessity of the Diversity Enhancement Strategy: We examine the effect of enhancing the diversity among students in the fusion classifier. As shown in Table IV, with ResNet-32, the cosine similarity of the two students in DML reaches 0.2635 without any diversity enhancement, and the fusion classifier can only achieve a 1.51% improvement in accuracy. Although the L2 reduces the cosine similarity between the two students, minimizing (6) greatly affects the performance of the students. The goal of the $\ell_2$-norm loss is to minimize the overall average outputs, which lacks a clear objective. Our FFSD presents a clear diversity enhancement strategy for attention shifting, which decreases the cosine similarity between the two students and improves the performance of the models. Similar results are also observed on WRN-16-2, while the L2 slightly improves the performance of the two students, the performance of the fusion classifier declines.

2) Advancement of the Leader Student and Feature Fusion: We combine the learning strategy of the leader student in FFSD with other online knowledge distillation methods for further experiments. From Table V, we can see that the addition of the leader student injects vitality into other online knowledge distillation methods, and the performance of the leader student is improved. Among them, the combination of DML and the leader student yields a 0.38%–0.52% improvement in accuracy over the original results. The leader student also improves the performance of ONE by 0.59%–0.60%. When removing feature fusion from FFSD, the performance significantly decreases from 74.85% to 74.06%. This well demonstrates the effectiveness of our feature fusion in helping student leader learn more information from common students.

3) Importance of the Self-Distillation: In Fig. 7, we visualize the feature maps with and without self-distillation. There is a significant overlap of attention positions between the activation feature maps of the mid- and top-level of the baseline, which demonstrates the attention consistency. However, it is difficult for us to see this phenomenon with the L2. The activation feature maps of the mid- and top-level only overlap in a very small part. Note that, in the picture

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
In our FFSD and AT as well as the practical training time.

In Table VII, we report the FLOPs of different components for comparison and select WRN-40-2 as its teacher model.

According to Table VI, we can see that the self-distillation module effectively enhances the diversity of the student models and improves their performance. The best accuracy obtained by the self-distillation module is 74.71% (common students) and 74.04% (leader students), which is better than that without self-distillation (74.63% and 74.54%, respectively).

Table VI demonstrates the effect of self-distillation on the performance of the student models. We only retain the last-layer diversity enhancement loss calculation of the L2 to enhance diversity and add self-distillation to it. All indicators are improved. We also investigate the effect of removing the self-distillation from FFSD. Although the performance of students degrades, but it is still better than that with the L2. We also explore the cases that the self-distillation is only applied to common students or leader student. We respectively obtain accuracy of 74.71% (common students) and 74.04% (leader student). In contrast, our FFSD achieves the best performance of 74.85% when the self-distillation module is applied to both common students and student leader.

4) Training Time Consumption: We measure the training time of WRN-16-1 on CIFAR-100. We use AT [9] for comparison and select WRN-40-2 as its teacher model. In Table VII, we report the FLOPs of different components in our FFSD and AT as well as the practical training time.

Though FFSD introduces additional operations from feature fusion and self-distillation, it still achieves lower overhead and better performance (75.81% vs. 73.71% by AT in Table II).

V. CONCLUSION

In this article, a novel online knowledge distillation method, termed FFSD, is proposed using FFSD. Existing online knowledge distillation methods either adopt the student with the best performance or consider the holistic performance using an ensemble model. However, they either ignore other students’ information or increase the computational burden during deployment. To solve these issues, we first design a feature fusion module similar to an autoencoder to fuse the output feature maps from all students into a meaningful and compact fused feature map, which is then distilled to the leader student. At the same time, we design a diversity enhancement strategy to enhance the diversity among students, enabling the leader student to obtain more information during feature fusion. Second, a self-distillation module is proposed to convert the feature maps of deeper layers to shallower ones, which are then distilled to shallower layers. This increases the generalization ability of the model. Extensive experiments on CIFAR-100 and ImageNet demonstrate the superiority of our FFSD.

REFERENCES

[1] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2015, pp. 1135–1143.
[2] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, “Filter pruning via geometric median for deep convolutional neural networks acceleration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 4340–4349.
[3] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, “XNOR-Net: ImageNet classification using binary convolutional neural networks,” in Proc. Eur. Conf. Comput. Vis. (ECCV), New York, NY, USA: Springer, 2016, pp. 525–542.
[4] B. Jacob et al., “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 2704–2713.
[5] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, “Exploiting linear structure within convolutional networks for efficient evaluation,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2014, pp. 1269–1277.
[6] X. Zhang, J. Zou, X. Ning, S. He, and J. Sun, “Efficient and accurate approximations of nonlinear convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1984–1992.
[7] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1503.02531.
[8] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, “FitNets: Hints for thin deep nets,” 2014, arXiv:1412.6550.
[9] N. Komodakis and S. Zagoruyko, “Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017, pp. 1–13.
[10] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4320–4328.
[11] G. Song and W. Chai, “Collaborative learning for deep neural networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 1832–1841.
[12] X. Zhu et al., “Knowledge distillation by on-the-fly native ensemble,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 7517–7527.
[13] X. Jin et al., “Knowledge distillation via route constrained optimization,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 1245–1254.
[14] J. H. Cho and B. Harilaharan, “On the efficacy of knowledge distillation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 4794–4802.
[15] I. Chung, S. Park, J. Kim, and N. Kwak, “Feature-map-level online adversarial knowledge distillation,” in Proc. Int. Conf. Mach. Learn. (ICML), 2020, pp. 2006–2015.
[16] X. Zhu et al., S. Lu, H. Gong, Z. Luo, and M. Liu, “AMLN: Adversarial-based mutual learning network for online knowledge distillation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 158–173.
[17] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, “Online knowledge distillation with diverse peers,” in Proc. AAAI Conf. Artif. Intell. (AAAI), 2020, pp. 3430–3437.
[18] Q. Guo et al., “Online knowledge distillation via collaborative learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11020–11029.
[19] J. Kim, M. Hyun, I. Chung, and N. Kwak, “Feature fusion for online mutual knowledge distillation,” 2019, arXiv:1904.09658.
[20] L. Yuan, F. E. Tay, G. Li, T. Wang, and J. Feng, “Revisiting knowledge distillation via label smoothing regularization,” in Proc. IEEE/CVF Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 3903–3911.
[21] Q. Ding et al., “Knowledge refiner: Learning from decoupled label,” in Proc. AAAI Conf. Artif. Intell. (AAAI), 2021, pp. 7228–7235.
[22] P. Chen, S. Liu, H. Zhao, and J. Jia, “Distilling knowledge via knowledge review,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 5008–5017.
[23] D. Chen et al., “Cross-layer distillation with semantic calibration,” in Proc. AAAI Conf. Artif. Intell. (AAAI), 2021, pp. 7028–7036.
[24] J. Yim, D. Joo, J. Bae, and J. Kim, “A gift from knowledge distillation: Fast optimization, network minimization and transfer learning,” in Proc. IEEE/CVF Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 4133–4141.
[25] Y. Tian, D. Krishnan, and P. Isola, “Contrastive representation distillation,” 2019, arXiv:1910.10699.
[26] G. Xu, Z. Liu, X. Li, and C. C. Loy, “Knowledge distillation meets self-supervision,” 2020, arXiv:2006.07114.
[27] W. Park, D. Kim, Y. Lu, and M. Cho, “Relational knowledge distillation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 3967–3976.
[28] Y. Liu et al., “Knowledge distillation via instance relationship graph,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 7096–7104.
[29] N. Passalis, M. Tzelepis, and A. Tefas, “Heterogeneous knowledge distillation using information flow modeling,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 2330–2348.
[30] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li, “Learning from noisy labels with distillation,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1910–1918.
[31] X. Wang, R. Zhang, Y. Sun, and J. Qi, “KDGAN: Knowledge distillation with generative adversarial networks,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2018, pp. 775–786.
[32] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2014, pp. 2672–2680.
[33] Y. Jiang, H. Lee, S. J. Hwang, and J. Shin, “Learning what and where to transfer,” in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 3030–3039.
[34] Y. Liu et al., “Search to distill: Pearls are everywhere but not the eyes,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 7539–7548.
[35] M. Li, J. Lin, Y. Ding, Z. Liu, J.-Y. Zhu, and S. Han, “GAN compression: Efficient architectures for interactive conditional GANs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 5284–5294.
[36] S. Li et al., “Learning efficient gans for image translation via differentiable masks and co-attention distillation,” 2020, arXiv:2011.08382.
[37] Q. Jin et al., “Teachers do more than teach: Compressing image-to-image models,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 13600–13611.
[38] Y. Li, Z. Shu, Y. Li, Z. Lin, F. Perazzi, and S. Y. Kung, “Content-aware GAN compression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 12156–12166.
Mingbao Lin is currently pursuing the Ph.D. degree with Xiamen University, Xiamen, China.
He has authored or coauthored over ten articles as the first author in international journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI), International Journal of Computer Vision (IJCV), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Neural Information Processing Systems (NeurIPS), the Association for the Advancement of Artificial Intelligence Conference (AAAI), the International Joint Conference on Artificial Intelligence (IJCAI), ACM Multimedia (MM), and so on. His current research interest includes network compression and acceleration, and information retrieval.

Yan Wang received the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 2015.
He works as a Software Engineer in Search with Pinterest, Seattle, WA, USA. He has authored or coauthored over 20 articles on top international conferences and journals. He holds ten U.S. or international patents. He has broad interests on deep learning’s applications on multimedia retrieval.

Yongjian Wu received the master’s degree in computer science from Wuhan University, Wuhan, China, in 2008.
He is currently the Expert Researcher and the Deputy General Manager of Youtu Lab, Tencent Company Ltd., Shanghai, China. His research interests include face recognition, image understanding, and large-scale data processing.

Yonghong Tian (Fellow, IEEE) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.
He is currently a Full Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He has authored or coauthored more than 160 technical articles in refereed journals and conferences and has owned more than 57 Chinese and U.S. patents. His current research interests include machine learning, computer vision, and multimedia big data.

Ling Shao (Fellow, IEEE) is currently the Executive Vice President and a Provost of the Mohamed bin Zayed University of Artificial Intelligence. He is also the CEO and the Chief Scientist of the Inception Institute of Artificial Intelligence (IIAI), Abu Dhabi, United Arab Emirates. His research interests include computer vision, machine learning, and medical imaging.
Mr. Shao is a fellow of the International Association of Pattern Recognition (IAPR), the Institution of Engineering and Technology (IET), and the Chartered Institute for IT (BCS).

Rongrong Ji (Senior Member, IEEE) is currently a Nanqiang Distinguished Professor with Xiamen University, Xiamen, China, the Deputy Director of the Office of Science and Technology, Xiamen University, and the Director of Media Analytics and Computing Lab. He was awarded as the National Science Foundation for Excellent Young Scholars in 2014, the National Ten Thousand Plan for Young Top Talents in 2017, and the National Science Foundation for Distinguished Young Scholars in 2020. His research falls in the field of computer vision, multimedia analysis, and machine learning. He has published 50+ articles in ACM/IEEE TRANSACTIONS, including TPAMI and IJCV, and 100+ full articles on top-tier conferences, such as CVPR and NeurIPS. His publications have got over 10K citations in Google Scholar.
Mr. Ji was the recipient of the Best Paper Award of ACM Multimedia 2011. He has served as Area Chairs in top-tier conferences such as CVPR and ACM Multimedia. He is also an Advisory Member for Artificial Intelligence Construction in the Electronic Information Education Committee of the National Ministry of Education.