Parameter-Efficient and Student-Friendly Knowledge Distillation

Jun Rao, Xv Meng, Liang Ding, *Member, IEEE*, Shuhan Qi, Xuebo Liu, Min Zhang, and Dacheng Tao, *Fellow, IEEE*

***Abstract*—Pre-trained models are frequently employed in multimodal learning. However, these models have too many parameters and need too much effort to fine-tune the downstream tasks. Knowledge distillation (KD) is a method to transfer knowledge using the soft label from this pre-trained teacher model to a smaller student, where the parameters of the teacher are fixed (or partially) during training. Recent studies show that this mode may cause difficulties in knowledge transfer due to the mismatched model capacities. To alleviate the mismatch problem, adjustment of temperature parameters, label smoothing and teacher-student joint training methods (online distillation) to smooth the soft label of a teacher network, have been proposed. But those methods rarely explain the effect of smoothed soft labels to enhance the KD performance. The main contributions of our work are the discovery, analysis, and validation of the effect of the smoothed soft label and a less time-consuming and adaptive transfer of the pre-trained teacher's knowledge method, namely PESF-KD by adaptive tuning soft labels of the teacher network. Technically, we first mathematically formulate the mismatch as the sharpness gap between teacher's and student's predictive distributions, where we show such a gap can be narrowed with the appropriate smoothness of the soft label. Then, we introduce an adapter module for the teacher and only update the adapter to obtain soft labels with appropriate smoothness. Experiments on various benchmarks including CV and NLP show that PESF-KD can significantly reduce the training cost while obtaining competitive results compared to advanced online distillation methods.**

***Index Terms*—Knowledge distillation, parameter-efficient, image classification.**

Manuscript received 13 January 2023; revised 2 August 2023; accepted 18 September 2023. Date of publication 5 October 2023; date of current version 8 March 2024. This Work was done when Jun was interning at JD Explore Academy. This work was supported in part by the National Natural Science Foundation of China under Grant 62372139, in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies under Grant 2022B1212010005, in part by Shenzhen Foundational Research Funding under Grant JCYJ20220818102414030, and in part by PINGAN-HITsz Intelligence Finance Research Center. The Associate Editor coordinating the review of this manuscript and approving it for publication was Prof. Jianguo Zhang. *(Jun Rao and Xv Meng contributed equally to this work.) (Corresponding author: Shuhan Qi.)*

Jun Rao was with the JD Explore Academy, China. He is now with the Harbin Institute of Technology, Shenzhen 518055, China (e-mail: rao7jun@gmail.com).

Xv Meng, Xuebo Liu, and Min Zhang are with the Harbin Institute of Technology, Shenzhen 518055, China (e-mail: mxx0822@foxmail.com; liuxuebo@hit.edu.cn; minzhang@suda.edu.cn).

Liang Ding and Dacheng Tao are with the the School of Computer Science, University of Sydney, NSW 2006, Australia (e-mail: liangding.liam@gmail.com; dacheng.tao@sydney.edu.au).

Shuhan Qi is with the the Harbin Institute of Technology, Shenzhen 518055, China, and with the Peng Cheng Laboratory, Shenzhen 518000, China, and also with the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies, Shenzhen 518000, China (e-mail: shuhanqi@cs.hitsz.edu.cn).

Digital Object Identifier 10.1109/TMM.2023.3321480

I. Introduction

KNOWLEDGE distillation (KD) [1], as an important method for model compression, has been widely used in various fields [2], [3], [4] of deep learning. This traditional paradigm [3], [5], [6] utilizes a pre-trained teacher network to obtain a student network that is close to the teacher network but with fewer parameters and the prediction output (soft label) is produced by the fixed teacher.

Label smoothing (LS) [7] is another method to produce soft labels to train a model. Compared with KD, it can be harmful to the training of the network because teachers in KD can understand the nuances of different classes, and such inter-class information brings more information than label smoothing and helps students generalize some unseen data [8]. However, when independently training the model from scratch, the larger model is more likely to output sharper values and obtain better accuracy, while the smaller model is more likely to output smoother values and obtain poorer accuracy [9], [10], [11], [12], which is called the capacity mismatch problem [2], [10], [11], [13] and makes the knowledge transfer [14] difficulty of such soft label [15] in KD.

One of the solutions to reducing this transfer difficulty is to smooth the teacher's output manually. [16] points out that KD can be compatible with LS when the temperature is low, i.e., the teacher network trained by label smoothing can produce smoother soft labels to train a better student. [8] indicates that the label smoothness of the target provided by the teacher exerts a great influence on the student network, and the difference in information between classes determines whether the student's performance can be improved. But manual conditioning is quite difficult and inefficient. Manually selecting the hyperparameters of LS to get the right teacher to provide a soft label is too resource-intensive. And the label smoothness controlled by the temperature may cause the loss of inter-class information when the temperature is not right.

The other type of method that can reduce this mismatch problem is online distillation (e.g., DML [17], KDCL [18] and SFTN [13]) not requiring manual conditioning. The idea behind online KD methods is the same: using joint training so that the teacher network can be optimized, which makes it easier for students to learn from the teacher. In our preliminary experiments (see Figs. 2 and 3), we found an interesting phenomenon of online distillation: if the teacher network continues to fine-tune through the ground truth labels with the rest of the

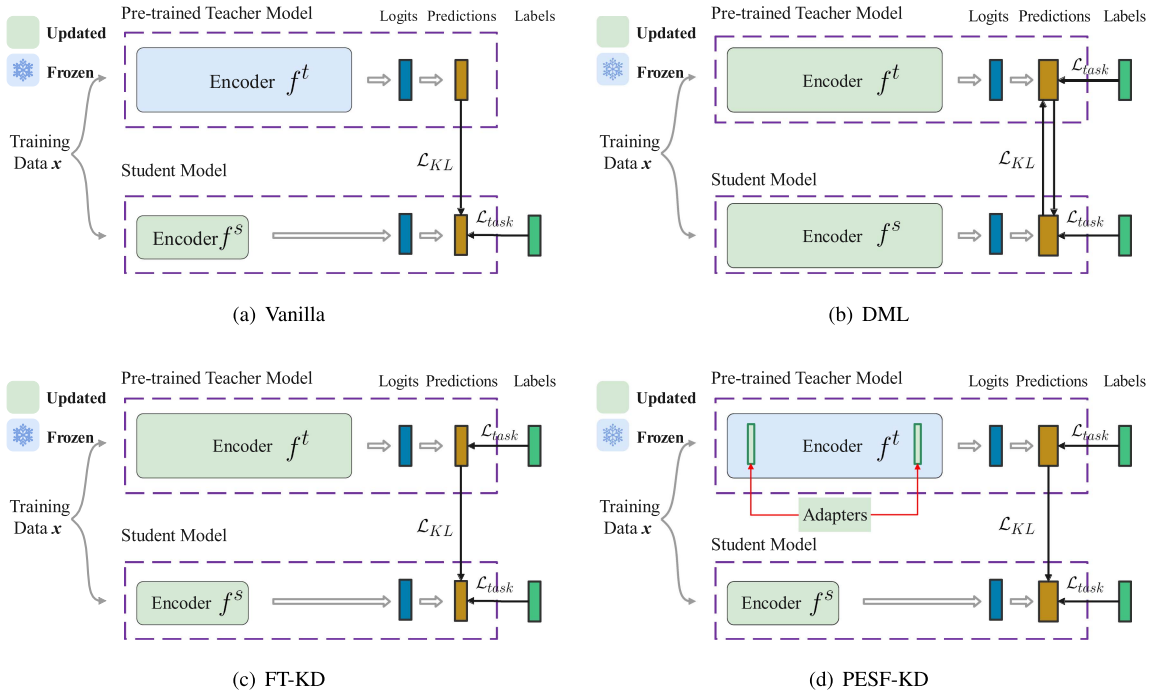


Fig. 1. Comparison between Vanilla, DML, FT-KD and PESF-KD. Green means the parameter needs to be updated, while blue means not. (a) Vanilla, teachers and students are trained independently, resulting in a gap in knowledge transfer. (b) DML [17], an online KD method, which gets better knowledge transfer and needs training teachers and students together. However, the teacher network and student network of DML are not like the traditional distillation i.e., the teacher network is much larger than the student network. When the teacher-student capacity gap is too large, student-to-teacher supervision may limit the further optimization of both networks. (c) FT-KD, a variant of DML, does not have an explicit KL loss term in the student-to-teacher direction unlike DML, which enables the teacher and student networks to further enhance the distillation results of the student network by adjusting the parameters of two networks. (d) Our proposed PESF-KD, updates the parameters of adapter modules of the teacher with ground-truth labels and the feedback from student outputs, while the rest of the parameters of the teacher are all fixed, which makes knowledge transfer more efficient.

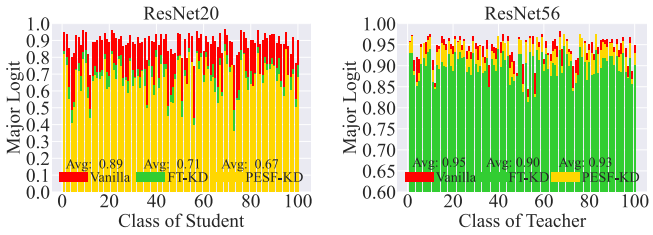


Fig. 2. Normalized major logits distribution of isomorphism student-teacher (Left: Student. Right: Teacher).

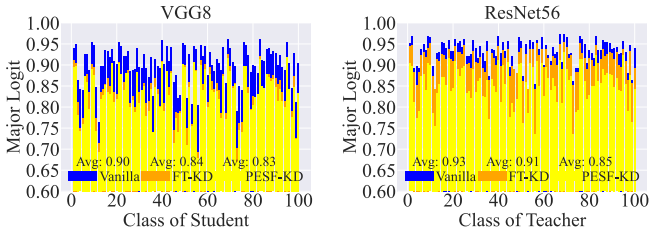


Fig. 3. Normalized major logits distribution of isomerism student-teacher (Left: Student. Right: Teacher).

settings as the Vanilla, where teacher network can also get the gradient information from the KD loss term, then the accuracy of the distilled student network is better and the teacher's labels become smoother. But the drawbacks of the online KD mainly lie in the need for iterative updates which greatly increases the training time and the reason why this co-training can enhance the transferability of knowledge is unclear. These questions motivate us to explore the relationship between the label smoothness of KD and how to further reduce the co-training cost of online KD.

In this article, for the first time, we give a unified explanation of the teacher-student mismatch problem on KD: The smoothness of the labels is a vital factor that affects the teacher-student mismatch problem. Our work provides the discovery, analysis, and validation of the effect of the smoothed soft label. To get better suitable smoothing labels and save training costs, we introduce the idea of efficient fine-tuning [19] into KD and propose a novel framework PESF-KD, as shown in Fig. 1(d), which achieves both parameter-efficient (fewer parameters to be updated) and student-friendly (better teacher-student consistency) KD. This framework also looks at online distillation from a new perspective: teachers learn to soften their own category distribution more appropriately under the supervision of a network of students. This supervision can be seen as a kind of transfer learning (i.e., adapting the student distribution). Based on extensive experiments and analyses, we show that our framework can utilize the information from ground-truth labels and student supervision to train the adapter modules, and further narrow the gap between the teacher and student models, which makes knowledge transfer easier.

In summary, our contributions are:

- By analyzing and exploring the smoothness in knowledge distillation methods, we empirically show that online distillation provides smoother prediction than offline distillation, which fixes the discrepancy between teacher-student capability. (Section VI)
- We propose a parameter-efficient and student-friendly distillation (PESF-KD) framework, which can better facilitate knowledge transfer by automatically updating the soft labels provided by the teacher. (Section IV and Section VI-A)
- Unlike previous distillation methods that only consider unimodal data (only test on the CV or NLP task), we use multimodal data (both CV and NLP tasks) to test the effectiveness and efficiency. (Section V)

II. Related Works

A. Knowledge Distillation.

Knowledge distillation is a popular technique for training compacting pre-trained models from a big teacher network to a small student network. It can be applied to many domains like natural language processing [20], image-text retrieval [21] and image classification [11], [22], [23]. According to whether to update the parameters of the teacher network, there are two types of knowledge distillation, offline and online. In offline distillation, the teacher network does not update its parameters during student training. Existing works of offline distillation mainly focus on promoting the design of knowledge transfer. Most researchers pay attention to including more information from the training data [3], [6], [24], [25]. [26] and [27] use features from layers except for the classification layers for supervision. [28] dynamically changes the training hyper-parameters for better distillation. The knowledge transfer in offline distillation is one-way, from the teacher to the student network. That means the teacher cannot dynamically adjust to downstream tasks. Besides, the capacity gap between the teacher and student networks exists and hinders knowledge transfer in offline knowledge distillation [29].

B. Online Distillation

The teacher and the student are jointly trained to make the teacher's knowledge more friendly to the student, called online distillation [2], [13], [30], [31]. Some works [17], [18] focus on updating all parameters in both networks using labels (hard labels) and feedback information (soft labels) in the training process, as shown in Fig. 1(b). Others [32], [33] balance multiple teachers and use ensemble prediction to guild the student. [11] and [34] bring additional training parts to extract information from teacher networks. [29] argues that online distillation performs better than offline ones because it narrows the capacity gap between teacher and student. However, online distillation also suffers from the extra training consumed because online approaches must train more parameters, mainly from teacher networks, than offline ones.

In contrast to these above methods, we first apply parameter-efficient learning to online knowledge distillation to accelerate the training process while not hindering prediction accuracy.

C. Parameter-Efficient Learning.

Parameter-efficient learning is a popular framework that applies the large neural network to downstream tasks with only a small number of extra parameters to be updated while keeping most pre-trained parameters frozen [19], [35], [36], [37]. [35] introduces a unified framework for parameter-efficient learning. Adapter [19] inserts small trainable modules in the network. Prefix tuning [38] introduces a trainable prefix vector in the input embedding. LoRA [39] injects trainable low-rank matrices into the neural network. In PESF-KD, we use the adapter module almost as the same in the original work [19] to reduce training costs for online distillation.

III. Background

A. Label Smoothing and Knowledge Distillation

Label Smoothing [7] is a method to soften and weigh traditional hard labels with a uniform distribution. This approach has successfully improved the effectiveness of several deep learning models and has been widely validated in NLP and CV And to date, this approach has also been used as a training trick to improve the training of models. We provide a mathematical description of the label smoothing process. First, we show the original cross-entropy: $H(\boldsymbol{y}, \boldsymbol{p}) = \sum_{k=1}^{K} -y_k \log(p_k)$, where y_k is "1" for the correct class and "0" for the rest. Then the label smoothing is achieved by increasing the smoothing parameter α to change y_k to y_k^{LS}: $y_k^{LS} = y_k(1-\alpha) + \alpha/K$. When a network is trained with label smoothing, the differences between the logits of the correct and incorrect classes become a constant that is dependent on α, while KD provides dynamic soft labels to let the network learn the distribution of teachers.

KD [1] often employs a pre-trained teacher network with the goal of transferring the teacher's knowledge to a small group of students. In the classification task, one of the simplest forms is to provide the soft label information by forwarding the teacher's output. The initial teacher and student model can be defined as: teacher $\boldsymbol{p}(\theta^t)$ and student $\boldsymbol{p}(\theta^s)$, respectively, where θ is the model parameters and $\boldsymbol{p_k}(\cdot) = \frac{\exp(\boldsymbol{z}_k(\theta)/\tau)}{\sum_{j=1}^{K}\exp(\boldsymbol{z}_j(\theta)/\tau)}$ is the probability predict of the matching label and K is the number of classes and z_k is the logical output of the $k-\mathrm{th}$ class. So the loss measuring the KL-Divergence of teachers and students can be formulated as:

$$\mathcal{L}_{KL}\left(\boldsymbol{p}(\tau|\theta^s), \boldsymbol{p}(\tau|\theta^t)\right) = \tau^2 \sum_j \boldsymbol{p}_j(\tau|\theta^t) \cdot \log \frac{\boldsymbol{p}_j(\tau|\theta^t)}{\boldsymbol{p}_j(\tau|\theta^s)}, \tag{1}$$

where τ is the temperature, which controls how much to rely on the teacher's soft predictions.

B. Sharpness Gap Between Teacher and Student

Label smoothness also can be called output sharpness. The sharpness of the two networks of their labels significantly exacerbates the knowledge transfer difficulty in KD [8], [16]. We use a simple and intuitive sharpness metric to get a smooth approximation to the maximum function considering the overall information of each class without the smoothing parameter τ

to calculate network output logits directly. [28] measuring the logits after temperature scaling that is actually applied to the KL-Loss in (1). They take an offline distillation approach and control the smoothness of the network by a uniform scaling factor (temperature), which, similar to LS, causes the problem of inter-class information elimination [8].

If we use K to denote K classes, the sharpness is defined as the logarithm of the exponential sum of logits:

$$S_{sharpness} = log \sum_{j}^{K} \exp z_j(\theta) \tag{2}$$

Similar to *fidelity* [40] and *loyalty* [41], measuring the resemblance between the soft labels of student and teacher from different aspects, the *sharpness gap* can measure the difference of *the label sharpness* between teacher and student networks:

$$G_{gap} = log \sum_{j}^{K} \exp\left(z_j(\theta^t)\right) - log \sum_{j}^{K} \exp\left(z_j(\theta^s)\right) \tag{3}$$

In the setting of sharpness gap of [28], they found that the $G_{gap}(\tau)$ with temperature decrease faster of ATKD [28] ($\frac{1}{\tau^3}$) compared with Vanilla ($\frac{1}{\tau^2}$), which results of the $G_{gap}(\tau)$ of ATKD would be smaller than KD in a considerable margin with the same temperature. However, the temperature adjustment leads to variance changes in the student's logit, which makes the $G_{gap}(\tau)$ may not decrease in some temperature situations. Different from [28], we focus on the network's actual output and do not consider the temperature smoothing of the network using (3), thus transforming the teacher-student capacity mismatch problem to study the smoothing of the soft label of the two networks. We also show in Section VI-A the effect of different temperatures on the gap and verify on various experiments that our methods significantly reduce the gap compared to Vanilla.

IV. Adaptive Knowledge Transfer Learning

A. Knowledge Distillation With Fine-Tuned Teacher (FT-KD)

Training Objectives: As shown in Fig. 1(c), different from Vanilla [1], our distillation method named "FT-KD" requires fine-tuning the parameters of the teacher network. The teacher network needs to output soft labels to supervise the student network, it also requires ground-truth labels to train itself. Take the classification task as an instance, the corresponding teacher's loss is:

$$\begin{aligned}\mathcal{L}_t &= \mathcal{L}_{task}(\theta^t) \\ &= -\sum_{i\in|X|}\sum_{c\in C}\left[\mathbf{1}\left[\mathbf{y}_i = c\right]\cdot \log p\left(\mathbf{y}_i = c \mid \mathbf{x}_i; \boldsymbol{\theta}^t\right)\right],\end{aligned} \tag{4}$$

where c is a class label and C denotes the set of class labels.

In Vanilla [1], students receive soft label supervision (1) from the teacher as well as hard label (the ground truth label) supervision. The final formulation of the student can be written as follows:

$$\mathcal{L}_s = \alpha\mathcal{L}_{KL}\left(\boldsymbol{p}(\theta^s), \boldsymbol{p}(\theta^t)\right) + (1-\alpha)\mathcal{L}_{task}(\theta^s), \tag{5}$$

where the task loss $\mathcal{L}_{task}$ follows the same format as the teacher network (4). In this mode, the teacher network can adjust its own smoothness of labels by implicitly acquiring the optimization signals related to the student network through the $\mathcal{L}_{KL}$ term. Unlike DML [17], this approach does not have an explicit KL loss term in the student-to-teacher direction, yet this baseline approach also enables the teacher to provide smoother soft labels, see Section VI-A.

B. Knowledge Distillation With Adapter (PESF-KD)

Adapter Module: Many online KD methods require retraining teacher networks, so it is desirable to participate in training a teacher network with only a small number of parameters. To reduce the over-consumption of the fully trained teacher network, we propose to adopt a standard adapter module [19] for knowledge distillation with updating parameters of this adapter module while the original parameters of the teacher network are fixed. The adapter consists of an up-sampling layer, a nonlinear activation function, and a down-sampling layer. It can be written as $proj_{down} \rightarrow$ non-linear $\rightarrow proj_{up}$ architecture. Specifically, the adapter firstly projects the input h to a lower-dimensional space with dimension r, utilizing a down-projection weight matrix $\boldsymbol{W}_{\text{down}} \in \mathbb{R}^{d\times r}$. Then through a nonlinear activation function and then through an up-projection function with weight matrix $\boldsymbol{W}_{\text{up}} \in \mathbb{R}^{r\times d}$ to increase the dimension to the original dimension. With such down-sampling and up-sampling operations, the feature dimensions of output h by the adapter module are consistent with the dimensions of input h. Usually, the adapter module uses a residual connection for preserving the information in input, and the output feature h of the adapter module is as follows [35]:

$$\boldsymbol{h} \leftarrow \boldsymbol{h} + f\left(\boldsymbol{h}\boldsymbol{W}_{\text{down}}\right)\boldsymbol{W}_{\text{up}} \tag{6}$$

Training Objectives: Our approach achieves better performance and less training time compared to DML [17] by introducing the adapter module and adjusting soft labels to provide the smoothed soft labels obtained by network training that are more reasonable compared to LS and temperature adjustment, respectively. Formally, the training loss of the student and teacher network can be formulated as follows:

$$\mathcal{L}_s = \alpha\mathcal{L}_{KL}\left(\boldsymbol{p}(\theta^s), \boldsymbol{p}(\theta^{ta})\right) + (1-\alpha)\mathcal{L}_{task}(\theta^s), \tag{7}$$

$$\mathcal{L}_t = \alpha\mathcal{L}_{KL}\left(\boldsymbol{p}(\theta^{ta})\boldsymbol{p}(\theta^s)\right) + (1-\alpha)\mathcal{L}_{task}(\theta^{ta}), \tag{8}$$

where θ^{ta} is the parameter of the adapter of the teacher network needed to update.

V. Experiments

A. Experimental Setup

Datasets: Two types of tasks including image classification (CIFAR-100 [42] and ImageNet [43]) and natural language understanding (GLUE [44]) are adopted for a series of experiments. For natural language understanding tasks, we test three commonly used and with a large amount of data datasets [45]:

SST-2 [46] for Sentiment Classification; RTE [44] for the Natural Language Inference; QQP [1] for Paraphrase Similarity Matching.

Baselines. We report several knowledge distillation methods for comparison, including Vanilla [1], knowledge distillation via collaborative learning (KDCL) [18], deep mutual learning (DML) [17], contrastive representation distillation (CRD) [3], relational knowledge distillation (RKD) [6] and probabilistic knowledge transfer (PKT) [5]. According to [15] on CV datasets, KD methods can be divided into two groups, online distillation and offline distillation. For a more fine-grained comparison, we further split them into three different kinds, online KD (DML, KDCL), offline KD (Vanilla, PKT) and representation KD (CRD, RKD). On NLP datasets, we compare one offline distillation method (Vanilla) and four online distillation (RCO, TAKD, DML and SFTN). Besides these methods, we report the result of our methods "FT-KD" and "PESF-KD" to support our argument about less sharpness gap helps the student to perform better to absorb the knowledge of the teacher.

For CV tasks, we follow previous works [3] using various combinations of student & teacher networks. Each pair of student & teacher networks are from a different capacity and architecture. We run isomorphic distillation and isomeric distillation. For isomorphic distillation, we run three different combinations (ResNet56-ResNet20, ResNet 110-ResNet32 and VGG13-VGG8). For isomerism distillation, the results of ResNet-56 to VGG-8 are reported. For NLU tasks, we first fine-tune the pre-trained teacher (12-layers of BERT-Base) and then train the student model (6-layers of BERT-Base) on each downstream task. We report accuracy for image classification experiments as a network performance metric. For QQP we report F1. For other NLP tasks, we report accuracy.

Training Details: [2] In CV experiment, we follow previous works [3]. We train the student model by SGD optimizer with a momentum of 0.9, a batch size of 64/256, and weight decay of 5×10^{-4}. The learning rate starts from 0.05/0.1 and decays by 10 every 30 epochs after 150/30 epochs for CIFAR-100/ImageNet, respectively. In the experiment of FT-KD, we train the teacher and student models during the training process with a learning rate of 1×10^{-3}. For NLP, we inherit parameters like maximum sequence length, temperature, and batch size according to [20]. We also perform grid-search over the sets of the student learning rate λ from {1e-5, 2e-5, 3e-5, 4e-5, 5e-5}, teacher learning rate μ from {2e-6, 1e-5, 2e-5}, batch size from {32, 64}, and the weight of KD loss from {0.3, 0.5, 0.8, 0.9} for better performance.

B. Results

1) Our method gets better (in most cases) or comparable performance compared to the competitors, demonstrating our simple method's power in both CV and NLP tasks: Generally, the distillation results of the online distillation methods are significantly higher than those of the offline distillation methods, especially in the case of excessive differences in network capacity between teachers and students (VGG13-8). On CIFAR-100 (Table I) and a larger dataset ImageNet (Table II), our PESF-KD achieves the best accuracy among all distillation methods. On most NLP datasets (Table III), PESF-KD also achieves the best results across various online KD methods. PESF-KD achieves higher accuracy, including CV and NLP, compared to Vanilla while increasing time consumption by only a tiny amount (1.05x). In many cases, PESF-KD performs better than FT-KD. We find possible explanations from several studies [39], [48], [49] demonstrating that larger models require fewer parameters to be updated during fine-tuning and the additional trainable parameters of PESF-KD may alleviate catastrophic forgetting by keeping pre-trained model parameters frozen.

2) Even if the accuracy is slightly lower, our method significantly reduces training resource consumption compared to the competitors (online KD), see Tables II and III: As shown in Tables II and III, online KD will greatly increase the training cost due to the need to update the parameters of the teacher network and the student network synchronously (see the batch time change), while PESF-KD significantly reduces the number of parameters that need to be updated for training and reduces the time required.

On the CV datasets (Tables I and II), PESF-KD improves student's accuracy even though our PESF-KD uses fewer training resources and less batch time. On NLP datasets (Table III), our PESF-KD can also obtain similar results to other baselines and require minimal training cost, and even surpass other baselines on SST-2 and QQP, showing the generalizability.

3) As the number of categories increases, the increased distillation accuracy of our method is greater, confirming the role of soft label smoothing: Most of the GLUE datasets are binary classification tasks, with CIFAR-100/ImageNet being a 100/1000 classification task. This leads to most of the online KD, which we attribute the result improvement to co-training leading to smoothing the teacher's soft labels (discussed in Section VI-B). The improvement is relatively insignificant in datasets with fewer categories (SST-2 and QQP, see Table III) and get more promotion on more classes of tasks (CIFAR-100 and ImageNet) because of richer inter-class information. However, by getting more reasonable soft labels (adapting the student distribution), our PESF-KD improves results compared to traditional online distillation even when inter-class information is more absent.

4) Orthogonality to other KD methods: PESF-KD on the bottom of other KD methods can improve further, which is a significant advantage over Vanilla. Table IV shows the results of four knowledge distillation approaches combined with PESF-KD on the CV dataset (CIFAR-100) and Table V shows our results of PESF-KD combined with intermediate layer distillation (PKD and TinyBERT[3]) on the NLP dataset (RTE and SST-2).

[1][Online]. Available: https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs

[2]The code is available at: https://github.com/1170300319/PESF-KD.git

[3]TinyBERT is a two-stage distillation method, and for time and fairness reasons we only performed the second stage of distillation (without distillation on pre-train stage). We used the 6-layer model provided by TinyBERT as the student after the first stage distillation, and the teacher used the PESF-KD fine-tuned BERT-Base for the distillation of the intermediate and prediction layers.

TABLE I
RESULTS ON CIFAR-100 TEST SET

	Method	ResNet56-20	ResNet110-32	VGG13-8	ResNet56-VGG8
Offline KD	Vanilla [1]	$70.95_{0.51}$	$73.08_{0.42}$	$73.36_{0.24}$	$73.98_{0.33}$
	PKT [5]	71.27 (+0.32)	73.67 (+0.59)	73.40 (+0.04)	74.10 (+0.12)
Representation KD	CRD [3]	71.44 (+0.49)	73.62 (+0.54)	73.31 (-0.05)	74.06 (+0.08)
	RKD [6]	71.47 (+0.52)	73.53 (+0.45)	74.15 (+0.79)	73.35 (-0.63)
Online KD	KDCL [18]	70.11 (-0.84)	72.87 (-0.21)	73.99 (+0.63)	73.16 (-0.82)
	DML [17]	71.40 (+0.45)	72.21 (-0.87)	74.18 (+0.82)	73.86 (-0.12)
Ours (Online KD)	FT-KD	$71.65_{0.11}$ (+0.70)	$73.90_{0.22}$ (+0.82)	$73.52_{0.14}$ (+0.16)	$74.40_{0.20}$ (+0.42)
	PESF-KD	**$71.84_{0.27}$ (+0.89)**	**$74.23_{0.26}$ (+1.15)**	**$74.74_{0.39}$ (+1.38)**	**$74.67_{0.28}$(+0.69)**

We report the averaged results over 3 random seeds. We compare the accuracy of various teacher-student combinations. The best results are **bold**. The second best results are underlined. We present the results with 3 different random seeds [47] in the form of the mean (standard deviation).

TABLE II
RESULTS ON IMAGENET TEST SET

	Method	Batch Time	Params	Accuracy
Offline KD	Vanilla [1]	0.46	47M	$69.20_{0.25}$
	PKT [5]	0.47	47M	69.69 (+0.49)
Representation KD	CRD [3]	0.57	63M	69.33 (+0.13)
	RKD [6]	1.53	47M	69.52 (+0.32)
Online KD	KDCL [18]	1.56	149M	70.17 (+0.97)
	DML [17]	1.45	149M	70.00 (+0.80)
Ours (Online KD)	FT-KD	1.44	149M	$70.00_{0.13}$ (+0.80)
	PESF-KD	0.47	54M	**$70.55_{0.14}$ (+1.35)**

ResNet18/ResNet50 is the student/teacher model. We compare the training time consumption (batch time) and the number of parameters needed to update (params) and the corresponding accuracy.

TABLE III
COMPARISON OF INFERENCE EFFICIENCY (#PARAMS AND TIME) AND MODEL PERFORMANCE RESULTS ON THE THREE NLP DATASETS

Method	#Params	Time	RTE $(2.5K)$	SST-2 $(67K)$	QQP $(364K)$
BERT-Base$_{teacher}$	-	-	71.4	93.0	88.5
BERT-Base$_{student}$	-	-	67.9	91.1	86.9
Vanilla [1]	66M	1.0x	67.7	91.2	87.3
RCO [2]	>176M	>2.66x	67.6 (-0.1)	91.4 (+0.2)	87.4 (+0.1)
TAKD [11]	>132M	>2.00x	68.5 (+0.8)	91.4 (+0.2)	87.5 (+0.2)
DML [17]	176M	2.66x	68.4 (+0.7)	91.5 (+0.3)	87.4 (+0.1)
SFTN [13]	>176M	>2.66x	**69.4 (+1.7)**	91.5 (+0.3)	87.5 (+0.2)
FT-KD	176M	2.66x	68.2 (+0.5)	91.7 (+0.5)	87.2 (-0.1)
PESF-KD	66M	1.05x	69.0 (+1.3)	**91.9 (+0.7)**	**87.7 (+0.4)**

TABLE IV
COMPARISON OF ORTHOGONALITY WITH EXISTING METHODS ON THE CIFAR-100 DATASET

Teacher/Student	ResNet56/ ResNet20	
Method	Stadard	PESF-KD
Vanilla [1]	70.95	71.84 (+0.89)
PKT [5]	71.27	71.90 (+0.63)
CRD [3]	71.44	71.97 (+0.53)
RKD [6]	71.47	71.72 (+0.25)

Our method can improve other KD methods on the CV dataset, but the middle-layer distillation on NLP can be slightly different. Compared with the results of TinyBERT, the results using PESF-KD performed better in terms of consistency metrics, and the standard deviation was reduced. After combining PESF-KD, both intermediate layer distillation methods (PKD and TinyBERT) improved in consistency metrics. Combined with the findings of [41], PESF-KD improves consistency by changing the teachers' output logits, which helps mitigate the loss of consistency from middle-layer distillation.

VI. CLOSER LOOK AT TEACHER-STUDENT RELATIONSHIP IN DISTILLATION

A. Revisit Gap in Knowledge Distillation

This section will explore factors affecting this gap (3). We first approximate this expression using a Taylor second expansion:

$$G_{gap} = log \sum_{j}^{K} \exp\left(z_j(\theta^t)\right) - log \sum_{j}^{K} \exp\left(z_j(\theta^s)\right)$$

TABLE V
TINYBERT AND PKD COMBINED WITH PESF-KD ON RTE AND SST-2

Method	Accuracy(↑)	Standard / PESF-KD PL(↑) [47]	Agree(↑) [46]	Gap*10^-2(↓)
		RTE		
TinyBERT [51]	67.9(0.8)/68.4(0.4)	96.4(0.5)/96.7(0.1)	84.2(3.0)/84.6(0.9)	0.2(0.06)/0.2(0.04)
PKD [29]	67.6(0.4)/67.8(0.3)	88.2(0.7)/88.9(0.8)	76.2(1.5)/80.3(1.5)	1.7(0.30)/1.4(0.20)
		SST-2		
TinyBERT [51]	92.0(0.4)/92.2(0.3)	97.3(0.3)/97.4(0.3)	99.2(0.1)/99.3(0.1)	0.1(0.06)/0.1/(0.05)
PKD [29]	91.1(0.3)/91.4(0.2)	97.3(0.1)/97.3(0.1)	98.1(0.1)/98.1(0.1)	0.7(0.05)/0.6(0.05)

We present the results with three different random seeds in the form of the mean (standard deviation).

$$\approx \log\left(K+\sum_j^K z_j(\theta^t)+\frac{1}{2}\sum_j^K z_j(\theta^t)^2\right) - \log\left(K+\sum_j^K z_j(\theta^s)+\frac{1}{2}\sum_j^K z_j(\theta^s)^2\right) \quad (9)$$

Following Hinton's assumption [1] and also through experimental phenomena [28], it can be known that the logits of each training sample are approximately zero-meaned so that $\sum_j^K z_j(\theta^s)=\sum_j^K z_j(\theta^t)=0$. So the gap can be rewritten as:

$$\begin{aligned} G_{gap} &= \log\left(K+\frac{1}{2}\sum_j^K \left(z_j(\theta^t)\right)^2\right) - \log\left(K+\frac{1}{2}\sum_j^K \left(z_j(\theta^s)\right)^2\right) \\ &= \log\left(1+\frac{1}{2\,K}\sum_j^K z_j(\theta^t)^2\right) - \log\left(1+\frac{1}{2\,K}\sum_j^K z_j(\theta^s)^2\right) \\ &= \log\left(1+\frac{1}{2}*\sigma_t^2\right)-\log\left(1+\frac{1}{2}*\sigma_s^2\right), \end{aligned} \quad (10)$$

where the $\sigma^2=\frac{1}{K}\sum_j^K z_j(\theta)^2$ is the variance of logits. So the change in the gap only comes from the change in the variance of logits, making our discussion easier. Once these logits become smoother, then the corresponding variance becomes smaller, if the logits become sharper then the variance becomes larger. The smoothness of the final logits results in a change in the gap. In the following sections, we compared three methods, namely, Vanilla with different temperatures, and our proposed methods (FT-KD and PESF-KD), to determine how temperature and respective methods affect the gap.

In the actual situation, the variance of students' logits will be affected by the variance of teachers' logits, which makes our direct mathematical analysis of (10) hard.

To more intuitively show the effect of logits output smoothness on the gap, we show the average major logit distribution of the student network in Figs. 2 and 3, the gap comparison in Fig. 4 and the accuracy of the respective methods in Fig. 6.

B. Analysing of Sharpness Gap in Knowledge Distillation

Appropriate soft labels for teacher networks can reduce the sharpness gap: In Figs. 2 and 3, we show that different KD methods can greatly affect the output variance of the student and teacher. The logit has the largest value that represents the model's category prediction. Obviously, Vanilla brings more sharp logit output of the student networks, that is, greater variance, while the student's variance of FT-KD and PESF-KD decreases sequentially with a large margin. Due to the tuning of the teacher network, the output of the teacher network becomes smoother compared to Vanilla.

The smoothing of the teacher's network by gradient automatic adjustment reduces the trouble of manually adjusting the temperature and can obtain lower gap values in different temperature ranges, as shown in Fig. 4. Consistent with the constrain of (1) and (10), since the teacher's output is unchanged, that is, the σ_t calculated in (10) is unchanged, by increasing the temperature, the student's output learning from a smoother teacher (the output scaled by temperature, $\boldsymbol{p_j}(\tau|\theta^t)$) is indeed smoother, resulting in a larger gap. However, with much lower temperatures (0.1 and 0.5), the soft label is closer to the hard label (see (1)), which reduces the distinction of category information and causes students to fail to learn helpful information, and the variance of its output is instead smaller, so gap becomes larger.

In Fig. 6, generally, lower gaps are associated with higher accuracy (lower temperature, lower gap value), but very close gaps may introduce anomalies (see the case of temperatures 2 and 4), which illustrates that even with different output smoothness of the teacher's soft labels, students' final performance can be close in some cases. This phenomenon shows that the appropriate soft labels of the teacher network by appropriate temperature adjustment within a certain range can improve accuracy and reduce the gap. However, manual adjustment τ or adaptive adjustment τ with the logits variance [28] scales the probability for all categories equally, erasing inter-class information [8]. While our methods, the soft labels can be *dynamically adjusted* according to the gradient of each sample to achieve better knowledge transfer (compared to the Vanilla with the best temperature setting, our methods have improved a lot about the accuracy, and a large reduction in the sharpness gap).

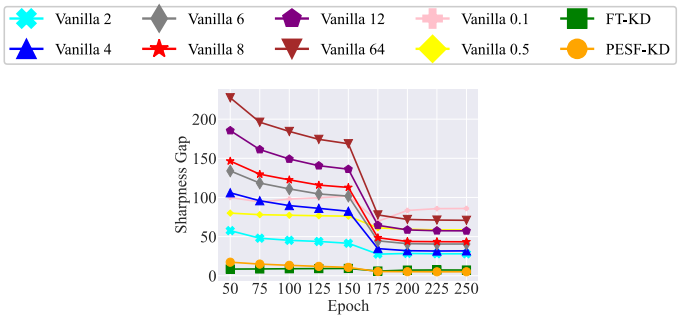


Fig. 4. Sharpness gap comparison with training epochs, including extreme setting.

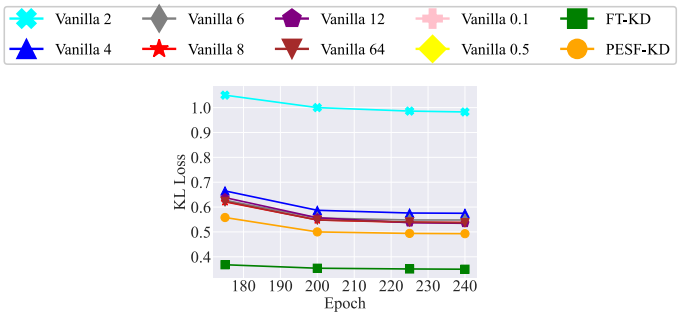


Fig. 5. KL loss comparison of student model with the lastest training epochs.

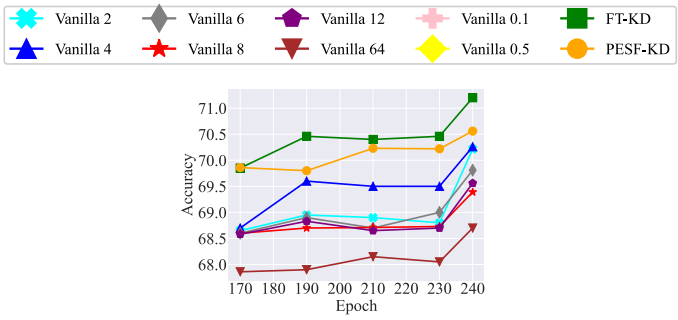


Fig. 6. Accuracy of student model with the lastest training epochs.

Appropriate soft labels for teacher networks can improve knowledge transfer: In this part, we show that trainable part in teacher models can narrow the gap between student and teacher and make the knowledge transfer process more student-friendly, thus achieving better accuracy.

We further explore the relationship between **NETWORK CONSISTENCY** (sharpness gap in Fig. 4, KL loss in Figs. 5 and 7 and CKA [50] in Fig. 8) and accuracy in Fig. 6. The three metrics mentioned above measure the final degree of consistency of the teacher-student network from different perspectives. A lower sharpness gap represents a closer knowledge representation of the teacher-student, and a lower KL loss represents the final convergence degree of the lower bound through distillation learning, while a larger CKA represents the larger similarity between the students and teachers.

We get the following interesting findings:

1) From (10), it is clear that the gap also decreases when the student network is trained with the Vanilla. This reduction comes from the fact that the output of the student network becomes sharper (σ_s become bigger) i.e., more similar to the output of the larger teacher network (both KL loss and Gap decrease).

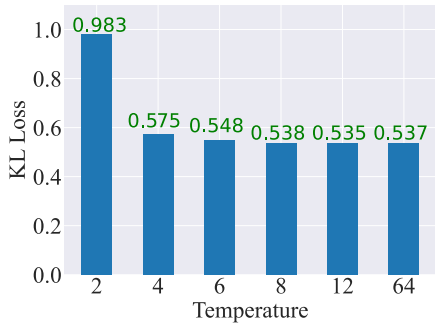


Fig. 7. KL comparison of Vanilla with different temperatures.

2) Both our proposed methods can reduce the gap, and the model trained with FT-KD can bring a great reduction (from 36.3 to 16.8). Our methods also make the output of teachers and students more consistent (the KL loss, and gap of the two methods are significantly lower and the CKA is higher than those of Vanilla with different temperatures).

3) The accuracy of the final student network trained by our methods and the gap between the two networks, the KL loss (0.42 vs. 0.23), CKA of logits, and predictions (almost the same) of our proposed methods are closer, which shows that our methods can guarantee the consistency of teacher and student characteristics. It also illustrates that gap and KL-loss interpret the similarity of output distributions from different perspectives.

4) We observe that clusters in our proposed approach are tighter because the student model is encouraged to learn more information from all other class templates in the training data set by narrowing the sharpness gap between teacher and student networks, as shown in Fig. 9 using T-SNE [51]. Besides, when looking at the projections, some clusters, i.e., crimson and dark blue ones, are more discernible in our proposed methods than in Vanilla.

VII. Further Analysis on Our Methods

A. Discrepancy Between Teacher and Student Predictions

This section compares the associated discrepancy metrics for teachers and students. These metrics are: Probability Loyalty (PL↑) [41], Kullback-Leibler divergence (KL↓), Average Top-1 Agreement (Agree↑) [40] and Sharpness Gap (Gap↓), where ↑ means the greater the better and ↓ means smaller the better.

1) Discrepancy in CV Dataset: First, we measure the relevant consistency metrics in the CV dataset (CIFAR-100), as shown in Table VI. We use resnet56/20 as the teacher-student combination. Compared to Vanilla, PESF-KD has a lower offset in accuracy. This phenomenon is also reflected in the combination of PESF-KD with PKD and TinyBERT. Also, the bias value of PESF-KD is greater over both PL and Agree on consistency metrics. We speculate that this is because the online distillation approach has greater logit variation for both models compared to the student-only training KD approach, resulting in a greater degree of bias. Overall, the results show that both our PESF-KD

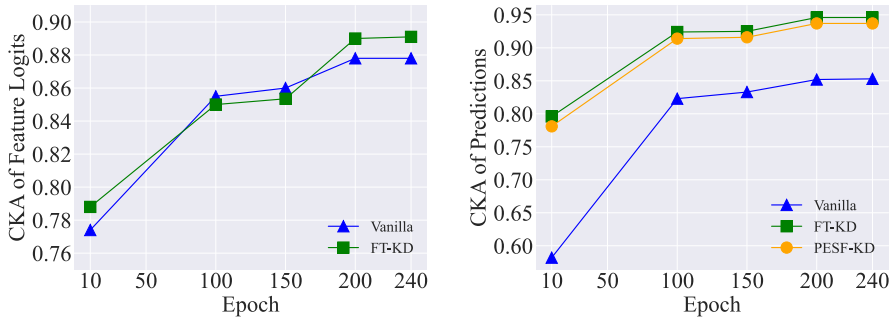
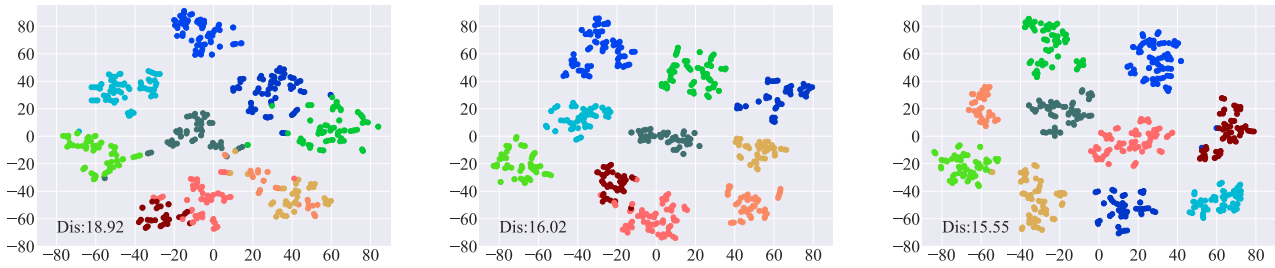


Fig. 8. CKA between teacher and student of feature logits (last layer before classifier) and predictions, respectively. PESF-KD does not affect feature logits, so we do not plot it on the left figure.



(a) Vanilla (b) FT-KD (c) PESF-KD

Fig. 9. Visualize of penultimate layer representation with seven semantically different classes. We compare three KD methods with average cluster distance (Dis). Clusters are tighter and more distinguishable in our proposed PESF-KD. Better view in color.

TABLE VI
RESULTS OF PESF-KD ON CIFAR-100 WITH STANDARD DEVIATION

Method	Accuracy(↑)	KL(↓)	PL(↑)	Agree(↑)
Vanilla [1]	70.95/0.51	1.84/0.07	78.6/0.7	80.0/0.2
PESF-KD	71.63/0.27	0.52/0.09	78.5/0.8	80.4/0.4
FT-KD	71.65/0.11	0.43/0.11	79.0/1.1	80.1/0.3

PESF-KD gets higher accuracy and lower deviation compared with vanilla. We present the results with three different random seeds in the form of mean/standard deviation.

(second best) and FT-KD (best) outperform Vanilla in terms of the relevant consistency metrics as well as accuracy, *which is consistent with the findings of our main part of the analysis in Section VI-B.*

2) Discrepancy in NLP Dataset: We further test on two datasets, SST-2 (67 K, high resource) and RTE (2.5 K, low resource). And the results are shown in Table V. We discover, as [41] pointed out, that this type of distillation method (PKD, TinyBERT) combined with an intermediate layer distillation term reduces the consistency of the label between teachers and students in high resource settings ([41] tests consistency in MNLI (393 K)). We find that PESF-KD performs better in both consistency metrics in the high resource case (SST-2), although in the low resource case (RTE), only Agree outperformed PKD.

However, we conjecture that an approach based on intermediate layer distillation (e.g., PKD) may lead to a failure of the relevant consistency metric, i.e., higher consistency in the output of the teacher-student network does not lead to better distillation results. Most of the distillation methods achieve distillation goals based on prediction layers, while PKD [26] and TinyBERT [45] utilize the distillation of intermediate layers for further combinations. We show the total training objects for PKD:

$$\mathcal{L}_{PKD} = (1-\alpha)\mathcal{L}^s_{CE} + \alpha\mathcal{L}_{KL} + \beta\mathcal{L}_{PT}, \qquad (11)$$

where $\mathcal{L}^s_{CE}$ denotes the standard cross-entropy loss as task loss, $\mathcal{L}_{KL}$ denotes distillation loss as described in the main part and $\mathcal{L}_{PT}$ denotes the middle layer distillation loss. The α and β are hyper-parameters that weigh the importance of each term.

According to the original PKD settings [26], we set α to 0.5 and β to 100 for the above training objectives. In the actual experiment, we found that the loss of the $\mathcal{L}^s_{CE}$ term was largest and much larger than the values of the other two distillation terms when the student model was nearly converged, which we guess is why the value of β needs to be so large to highlight its constraining effect. And this causes the $\mathcal{L}_{KL}$ term to lose its usefulness (the loss is small enough compared with the other two terms to be optimized more rarely) and also causes a relatively low consistency of the labeled predicted values compared to prediction layer-based distillation methods such as FT-KD and PESF-KD.

B. Robustness

In this section, we test the effects of a series of hyperparameters on the distillation results. Specifically, the effects of adapter structure and dimensionality, the ratio of different losses,

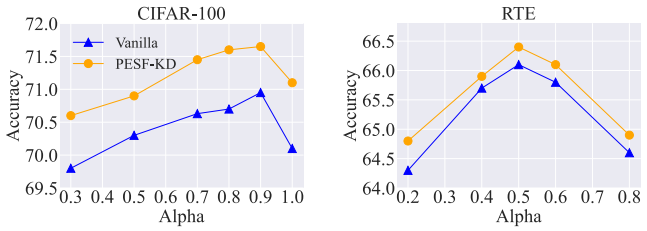


Fig. 10. Comparison of loss weight sensitivity of different methods.

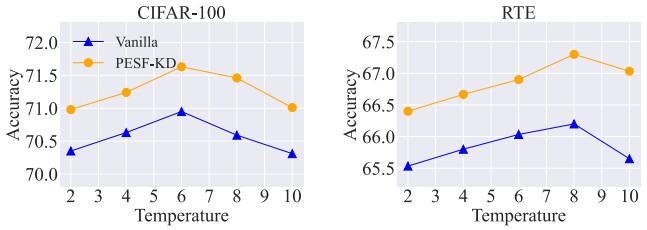


Fig. 11. Comparison of temperature sensitivity of different methods.

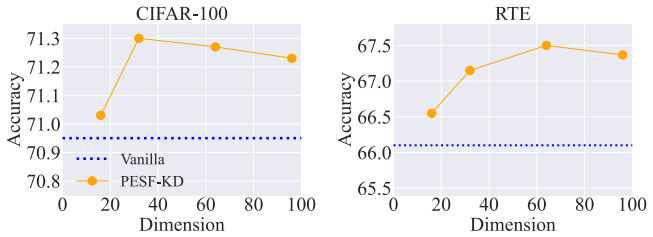


Fig. 12. Influence of adapter dimension on CIFAR-100 (left) and RTE (right) of PESF-KD. Dimension $=$ 32 and 64 are appropriate choices for CIFAR-100 and RTE, respectively.

and temperature are included. All the findings confirm that our PESF-KD is easy-to-use and robust to promote knowledge distillation, making the strategy great potential to apply to a broad range of tasks.

1) Adapter Dimension: As shown in Fig. 12, we test the impact of different scaling dimensions of the adapter on classification tasks on CIFAR-100 and RTE. As mentioned in the main part, the adapter structure is w_{down}, non-linear, and w_{up}. We modify the output dimension of w_{down} and the input dimension of w_{up}. As seen, dimension spanning 32, 64, and 96 seems not to affect the performance, further reducing (e.g. 16) slightly worsens the performance (drop $<$ 0.3), and all of them still outperform the baseline, showing the robustness to different adapter dimensions. We follow the setting of [35] to lower the model's dimension than the input dimension and finally chose 32 for CIFAR-100 and 64 for RTE, respectively.

2) Adapter Architecture: The adapter module is our recipe for success in the above performance comparisons. To explore the influence of adapter structure on classification results, we compare four methods, including Vanilla and three classic adapter structures, and report their accuracy and CKA consistency. We conduct comparative experiments and use the teacher-student model of VGG on CIFAR-100 and BERT-Base on RTE, respectively. As can be seen from Table VII,

TABLE VII
ABLATION ON DIFFERENT STRUCTURES OF ADAPTER

Method	Batch Time(↓)	Top-1(↑)	CKA(↑)
CIFAR-100			
Vanilla [1]	0.097	73.36	0.854
Adapter [19]	0.137	73.43	0.876
LoRA [42]	0.166	73.49	0.858
Scaled PA [38]	0.165	73.13	0.859
RTE			
Vanilla [1]	0.393	66.10	0.629
Adapter [19]	0.403	67.50	0.639
LoRA [42]	0.412	67.15	0.644
Scaled PA [38]	0.406	67.29	0.638

Experiments are performed in VGG on CIFAR-100 and BERT-base on RTE. The simple adapter shows decent performance compared-with other parameter-efficient methods in accuracy, CKA, and time consumption.

the simple and efficient adapter achieves competitive performance with accuracy and CKA scores and is less time-consuming. Therefore we use the simple adapter module in all experiments.

3) Loss Weight and Temperature: PESF-KD shows better performance and robustness than Vanilla in all experiments: In vanilla, the hyper-parameter α is a loss weight that needs non-trivial tuning like hyperparameter search. Larger α means a higher percentage of KL loss. To test the robustness of our PESF-KD, we use the teacher-student combination of Resnet56/20 and BERT, and run experiments on CIFAR-100 and RTE with different α as shown in Fig. 10. It shows that 1) proper grid searching indeed obtains better performance, and 2) our PEST-KD consistently outperforms Vanilla ranging from 0.3 to 1.0 (for CIFAR-100) and 0.2 to 0.8 (for RTE), confirming the robustness of our method in terms of different loss weights. Fig. 11 shows the performance of student models in different temperatures on CIFAR-100 and RTE. PESF-KD shows better performance and robustness compared to Vanilla.

VIII. CONCLUSION

This article shows how the smoothness of labels affects the teacher-student mismatch. To reduce this mismatch and balance the difficulty and cost of training, we present PESF-KD, a novel KD framework that applies adapters to optimize the teacher network for better knowledge transfer to the student network. Through detailed analysis, we point out that the decline in sharpness and a better ability to distinguish within classes lead to better knowledge transfer, which leads to better results. Extensive experiments on CV and NLP demonstrate the robustness and effectiveness of our method. Future works of embedding PESF-KD into existing distillation frameworks are expected to generalize our methods. Our future work continues to explore more elaborate structural designs of KD.

ACKNOWLEDGMENT

The computing resources of Pengcheng Cloud Brain are used in this research.

REFERENCES

[1] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. Neurips*, 2015, pp. 1–9.

[2] X. Jin et al., "Knowledge distillation via route constrained optimization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1345–1354.

[3] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–19.

[4] L. Zhang, L. Shen, L. Ding, D. Tao, and L.-Y. Duan, "Fine-tuning global model via data-free knowledge distillation for non-IID federated learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10174–10183.

[5] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 268–284.

[6] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3967–3976.

[7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2818–2826.

[8] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–10.

[9] P. Chen, S. Liu, H. Zhao, and J. Jia, "Distilling knowledge via knowledge review," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 5006–5015.

[10] Y. Zhu and Y. Wang, "Student customized knowledge distillation: Bridging the gap between student and teacher," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 5057–5066.

[11] S. Mirzadeh et al., "Improved knowledge distillation via teacher assistant," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5191–5198.

[12] J. H. Cho and B. Hariharan, "On the efficacy of knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4794–4802.

[13] D. Y. Park et al., "Learning student-friendly teacher networks for knowledge distillation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 13292–13303.

[14] J. Rao et al., "Student can also be a good teacher: Extracting knowledge from vision-and-language model for cross-modal retrieval," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 3383–3387.

[15] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, pp. 1789–1819, 2021.

[16] K. Chandrasegaran, N. Tran, Y. Zhao, and N. Cheung, "Revisiting label smoothing and knowledge distillation compatibility: What was missing?," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 2890–2916.

[17] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, "Deep mutual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4320–4328.

[18] Q. Guo et al., "Online knowledge distillation via collaborative learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11020–11029.

[19] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," in *Proc. Int. Conf. Mach. Learn.*2019, pp. 2790–2799.

[20] W. Zhou, C. Xu, and J. J. McAuley, "BERT learns to teach: Knowledge distillation with meta learning," in *Proc. 60th Annu. Meeting Assoc. Computat. Linguistics*, 2022. pp. 7037–7049.

[21] J. Rao et al., "Dynamic contrastive distillation for image-text retrieval," *IEEE Trans. Multimedia*, early access, Apr. 14, 2023, doi: 10.1109/TMM.2023.3236837.

[22] Z. Hao, Y. Luo, Z. Wang, H. Hu, and J. An, "CDFKD-MFS: Collaborative data-free knowledge distillation via multi-level feature sharing," *IEEE Trans. Multimedia*, vol. 24, pp. 4262–4274, 2022.

[23] M. Yuan and Y. Peng, "CKD: Cross-task knowledge distillation for text-to-image synthesis," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1955–1698, Aug. 2020.

[24] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–10.

[25] B. Peng et al., "Correlation congruence for knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 5006–5015.

[26] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for BERT model compression," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4323–4332.

[27] B. Heo, M. Lee, S. Yun, and J. Y. Choi, "Knowledge transfer via distillation of activation boundaries formed by hidden neurons," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3779–3787.

[28] J. Guo, "Reducing the teacher-student gap via adaptive temperatures," OpenReview preprint, 2022.

[29] L. Li and J. Zhe, "Shadow knowledge distillation: Bridging offline and online knowledge transfer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 635–649.

[30] C. Xu, W. Zhou, T. Ge, F. Wei, and M. Zhou, "BERT-of-Theseus: Compressing BERT by progressive module replacing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*2020, pp. 7859–7869.

[31] P. Bhat, E. Arani, and B. Zonooz, "Distill on the go: Online knowledge distillation in self-supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2678–2687.

[32] D. Chen, J.-P. Mei, C. Wang, Y. Feng, and C. Chen, "Online knowledge distillation with diverse peers," in in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 3430–3437.

[33] X. Lan, X. Zhu, and S. Gong, "Knowledge distillation by on-the-fly native ensemble," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7528–7538.

[34] Z. Liu, Y. Liu, and C. Huang, "Semi-online knowledge distillation," in *Proc. Brit. Mach. Vis. Conf.*, 2021, pp. 1–13.

[35] J. He et al., "Towards a unified view of parameter-efficient transfer learning," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–15.

[36] Q. Zhong, L. Ding, J. Liu, B. Du, and D. Tao, "Panda: Prompt transfer meets knowledge distillation for efficient model adaptation," 2022, *arXiv:2208.10160*.

[37] S. He, L. Ding, D. Dong, J. Zhang, and D. Tao, "SparseAdapter: An easy approach for improving the parameter-efficiency of adapters," in *Proc. Findings Assoc. Comput. Linguistics*, 2022, pp. 2184–2190.

[38] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4582–4597.

[39] E. J. Hu et al., "LORA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*2022, pp. 1–13.

[40] S. Stanton, P. Izmailov, P. Kirichenko, A. A. Alemi, and A. G. Wilson, "Does knowledge distillation really work?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 6906–6919.

[41] C. Xu et al., "Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression," in *Proc. Conf. Empirical Methods Natural Lang. Process.*2021, pp. 10653–10659.

[42] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.

[43] J. Deng et al., "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[44] A. Wang et al., "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. Int. Conf. Learn. Representations*2019, pp. 1–20.

[45] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," in *Proc. Findings Assoc. Conf. Empirical Methods Natural Lang. Process. Findings*, 2020, pp. 4163–4174.

[46] R. Socher et al., "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*2013, pp. 1631–1642.

[47] J. Rao et al., "Where does the performance improvement come from-A reproducibility concern about image-text retrieval," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 2727–2737.

[48] A. Aghajanyan, S. Gupta, and L. Zettlemoyer, "Intrinsic dimensionality explains the effectiveness of language model fine-tuning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 7319–7328.

[49] V. Lialin, V. Deshpande, and A. Rumshisky, "Scaling down to scale up: A guide to parameter-efficient fine-tuning," 2023, *arXiv:2303.15647*.

[50] S. Kornblith, M. Norouzi, H. Lee, and G. E. Hinton, "Similarity of neural network representations revisited," in *Proc. Int. Conf. Mach. Learn.*2019, pp. 3519–3529.

[51] L. V. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9 pp. 2579–2605, 2008.

Jun Rao is currently working toward the Ph.D. degree with the Harbin Institute of Technology, Shenzhen, Shenzhen, China. He authored or coauthored several papers at top conferences and international journals, including MM, TMM, CIKM, and SIGIR. His long-term research goal is to build socially intelligent embodied agents with the ability to perceive and engage in multimodal human communication. As steps towards this goal, his research interests include the fundamentals of multimodal learning, specifically the representation, translation, fusion, and alignment of heterogeneous data sources, human-centered language, vision, and their applications, the real-world deployment of efficiency involves both the amount of computation and data required for (pre-)training and using NLP models.

Xv Meng is currently working toward the graduation degree with the Harbin Institute of Technology, Shenzhen, China. His long-term research goal is to build interactive, inexpensive commercial AI with computer vision and natural language processing technology. To achieve his goal, he is researching on visual and language common sense and question answering, image classification, natural language understanding, and named entity recognition.

Liang Ding (Member, IEEE) received the Ph.D. degree from the University of Sydney, Camperdown, NSW, Australia. He was a Research Scientist and led the NLP Research Group, JD Explore Academy, JD.com, China. He works on deep learning for NLP, including language model pretraining, language understanding, generation, and translation. He has authored or coauthored more than 40 papers in NLP/ML venues, including ACL, EMNLP, COLING, NAACL, ICLR, ICML, AAAI, IJCAI, SIGIR, CVPR, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, and IEEE TRANSACTIONS ON MULTIMEDIA. He was the Area Chair (or Session Chair) for ACL, AAAI, and SDM. He was the recipient of many AI challenges, including SuperGLUE/ GLUE, WMT2022, IWSLT 2021, WMT 2020, and WMT 2019.

Shuhan Qi received the M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China. He was a Visiting Scholar with the National University of Singapore, Singapore. He is currently an Associate Researcher with the School of Computer Science and Technology, Harbin Institute of Technology. His main part-time positions include double-appointed assistant Researcher with the Network Intelligence Department of Pengcheng Laboratory and Deputy Director of Internet Application Technology Engineering Laboratory of Shenzhen Development and Reform Commission. He is with the Computer Application Research Center, Harbin Institute of Technology. He has authored or coauthored more than 30 papers in IEEE TRANSACTION ON MULTIMEDIA, SIGIR, ACM MM, and other important international journals and conferences.

Xuebo Liu received the Ph.D. degree in computer science from the University of Macau, Macau, China, in 2021. He is currently an Assistant Professor with the Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, Shenzhen, China. His research interests include natural language processing and machine translation.

Min Zhang received the bachelor's and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1991 and 1997, respectively. He is currently a Professor with the Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, Shenzhen, China. He has authored 150 papers in leading journals and conferences and has co-edited ten books that were published by Springer and IEEE. His research interests include machine translation, natural language processing, information extraction, large-scale text processing, intelligent computing, and machine learning. He has been actively contributing to the research community by organizing many conferences as the Chair, Program Chair, and Organizing Chair and by giving talks at many conferences and lectures.

Dacheng Tao (Fellow, IEEE) was the President of JD Explore Academy and a Senior Vice President of JD.com. He is currently an Advisor and Chief Scientist of the Digital Science Institute, University of Sydney, Camperdown, NSW, Australia. He mainly applies statistics and mathematics to artificial intelligence and data science. His research interests include across computer vision, data science, image processing, machine learning, and video surveillance. His research results have been presented in one monograph and more than 500 publications in prestigious journals and at prominent conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, T-IP, JMLR, IJCV, NIPS, ICML, CVPR, ICCV, ECCV, ICDM, and ACM SIGKDD, with several Best Paper awards, such as the Best Theory/Algorithm Paper Runner Up Award at IEEE ICDM07, the Best Student Paper Award at IEEE ICDM13, the Distinguished Student Paper Award at the 2017 IJCAI, the 2014 ICDM ten Year Highest-Impact Paper Award, and the 2017 IEEE Signal Processing Society Best Paper Award. He was the recipient of the 2015 Australian Scopus-Eureka Prize, the 2015 ACS Gold Disruptor Award, and the 2015 UTS Vice-Chancellor's Medal for Exceptional Research. He is a Fellow of AAAS, OSA, IAPR, and SPIE.