EXPLORING THE CAPACITY MISMATCH PROBLEM IN KNOWLEDGE DISTILLATION FROM THE VIEW OF SOFT LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation (KD) has been extensively employed to transfer the knowledge using the soft label from a large teacher model to the smaller students, 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 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 show that PESF-KD can significantly reduce the training cost while obtaining competitive results compared to advanced online distillation methods.

1 INTRODUCTION

Knowledge distillation (KD) (Hinton et al., 2015), as an important method for model compression, has been widely used in various fields (Jin et al., 2019; Tian et al., 2020; Zhang et al., 2022) of deep learning. This traditional paradigm (Tian et al., 2020; Passalis & Tefas, 2018; Park et al., 2019) 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) (Szegedy et al., 2016) 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 (Müller et al., 2019). 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 (Chen et al., 2021; Zhu & Wang, 2021; Mirzadeh et al., 2020; Cho & Hariharan, 2019), which is called the capacity mismatch problem (Park et al., 2021; Zhu & Wang, 2021; Jin et al., 2019; Mirzadeh et al., 2020) and makes the knowledge transfer difficulty of the such soft label (Gou et al., 2021) in KD.

One of the solutions to reducing this transfer difficulty is to manually smooth the teacher's output. Chandrasegaran et al. (2022) point 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. Müller et al. (2019) indicate 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 (Zhang et al., 2018), KDCL (Guo et al., 2020) and SFTN (Park et al., 2021)) 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 Figure 2), 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 settings as the vanilla KD, 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 paper, 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 (Houlsby et al., 2019) into KD and propose a novel framework PESF-KD, as shown in Figure 1(b), 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:

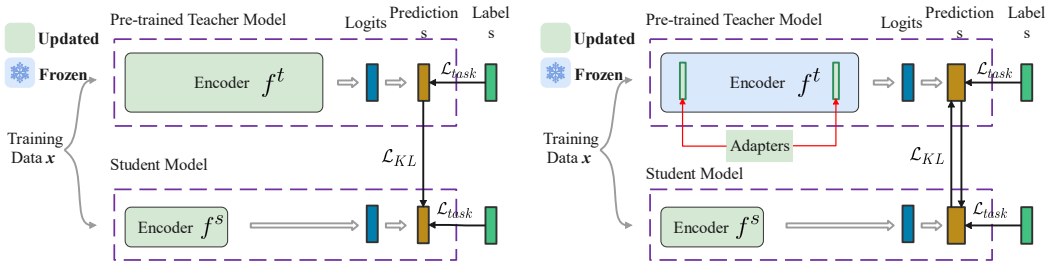
- We provide evidences to show the smoothness of soft labels affect the KD. (§4)
- We propose a parameter-efficient and student-friendly distillation (PESF-KD) framework, which can better facilitate the knowledge transfer by automatically updating the soft labels provided by the teacher. (§3 and §4.2)
- We empirically validate the effectiveness and efficiency of our PESF-KD upon several vision and language models compared to existing knowledge distillation methods. (§5)

2 BACKGROUND

2.1 LABEL SMOOTHING AND KNOWLEDGE DISTILLATION

Label Smoothing (Szegedy et al., 2016) 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 natural language processing (NLP) and computer vision (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 (Hinton et al., 2015) 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



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

Figure 1: Comparison between FT-KD and PESF-KD. **Green** means the parameter needs to be updated, while **blue** means not. (a) Our proposed FT-KD. Unlike vanilla KD (teacher networks in the distillation process are fixed, resulting a gap of knowledge transfer), our FT-KD is quite like DML (Zhang et al., 2018) and other online KD methods, which get better knowledge transfer and need training teachers and students together. (b) 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 better.

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 $k-\text{th}$ class. So the vanilla KD 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.

2.2 Online Distillation

The teacher and the student are jointly trained to make the teacher's knowledge more friendly to the student called online distillation (Park et al., 2021; Jin et al., 2019; Xu et al., 2020; Bhat et al., 2021). These methods, such as DML (Zhang et al., 2018) and KDCL (Guo et al., 2020), usually update most or even all the parameters of the teacher by using the real labels (hard labels) and the feedback information (soft labels) of the students, as shown in Figure 1(a). However, in the setting of online distillation, a large teacher network and a student network need to be trained simultaneously for each new downstream task from scratch, which is too time-consuming and inefficient. In contrast to these above methods, we propose an adaptive knowledge transfer learning method (also can be regarded as online KD) that can dynamically generate a soft target distribution at each time step for different contexts under the constraint of the student's logits distribution with less training time costs.

2.3 Gap Between Teacher and Student

Label smoothness also can be called output sharpness. The sharpness of two networks of their labels significantly exacerbates the knowledge transfer difficulty in KD Chandrasegaran et al. (2022); Müller et al. (2019). 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 directly calculate network output logits unlike Guo (2022) measuring the logits after temperature scaling that is actually applied to the KL-Loss in Equation 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 (Müller et al., 2019).

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* (Stanton et al., 2021) and *loyalty* (Xu et al., 2021), 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:

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

3 ADAPTIVE KNOWLEDGE TRANSFER LEARNING

3.1 KNOWLEDGE DISTILLATION WITH FINE-TUNED TEACHER (FT-KD)

The effectiveness of KD can be considered in two ways, i.e., Consistency, which can be called Fidelity (Stanton et al., 2021) and loyalty (Xu et al., 2021), and Generalization. In general, most studies (Zhang et al., 2018; Guo et al., 2020) consider the generalization of the student model, which is the final classification accuracy. However, Stanton et al. (2021) show that good student accuracy does not imply good distillation consistency. In this work, we further explore the link between generalization and consistency and find that both the consistency and generalization among models increase significantly through the co-training of the teacher-student networks even without the additional designed losses compared to offline KD. To produce a more straightforward and robust baseline FT-KD (see §4.2), we modify the DML (Zhang et al., 2018) settings by removing an explicit KL loss term in the student-to-teacher direction and using the label adjustment technique.

Training Objectives. As shown in Figure 1(a), different from vanilla KD (Hinton et al., 2015), a new baseline method 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: $\mathcal{L}_t = \mathcal{L}_{task}(\theta^t) = -\sum_{i\in|X|}\sum_{c\in C}\left[\mathbf{1}\left[\mathbf{y}_i = c\right]\cdot\log p\left(\mathbf{y}_i = c \mid \mathbf{x}_i;\theta^t\right)\right]$, where c is a class label and C denotes the set of class labels.

In vanilla KD (Hinton et al., 2015), students receive soft label supervision (Equation 1) from the teacher as well as hard label (the ground truth label) supervision. The teacher network provides soft labels that help students learn, but there may be some generalization errors (maximum probability that the labels are not the true labeled labels) that lead to a decrease in student performance. Therefore, to reduce the impact of this part of the error, we modified the teacher's soft labels called "label adjustment", i.e., the maximum probability labels are guaranteed to be the labeled true labels. So the final formulation can be written as follows:

$$\mathcal{L}_s = \alpha\mathcal{L}_{KL}\left(\boldsymbol{p}(\theta^s), \boldsymbol{p}(\theta^t)\right) + (1-\alpha)\mathcal{L}_{task}(\theta^s), \tag{4}$$

where the task loss $\mathcal{L}_{task}$ follows the same format as the teacher network. 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 teacher-to-student direction $\mathcal{L}_{KL}$ term.

3.2 KNOWLEDGE DISTILLATION WITH ADAPTER (PESF-KD)

Adapter Module. Many online KD methods require training the whole teacher network, 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 (Houlsby et al., 2019) for KD with updating parameters of this adapter module while the original parameters of the teacher network are fixed. The adapters 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 a up-projection function with weight matrix $\boldsymbol{W}_{\text{up}} \in \mathbb{R}^{r\times d}$ to increase the dimension to the original dimension. Usually, these modules use a residual connection, and the final form is as follows (He et al., 2022):

$$\boldsymbol{h} \leftarrow \boldsymbol{h} + f\left(\boldsymbol{h}\boldsymbol{W}_{\text{down}}\right)\boldsymbol{W}_{\text{up}} \tag{5}$$

Training Objectives. PESF-KD achieves better performance and less training time compared to DML (Zhang et al., 2018) and FT-KD 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. During training, we find that maintaining the KL loss from teacher-to-student and student-to-teacher enhances the training stability of the adapter, so here we do not remove the KL loss in the student-to-teacher direction as in FT-KD. Formally, the training loss of the student and teacher network can be formulated as: $\mathcal{L}_s = \alpha\mathcal{L}_{KL}\left(\boldsymbol{p}(\theta^s), \boldsymbol{p}(\theta^{ta})\right) + (1-\alpha)\mathcal{L}_{task}(\theta^s)$, and $\mathcal{L}_t = \alpha\mathcal{L}_{KL}\left(\boldsymbol{p}(\theta^{ta}), \boldsymbol{p}(\theta^s)\right) + (1-\alpha)\mathcal{L}_{task}(\theta^{ta})$, where θ^{ta} is the parameter of the adapter of the teacher network needed to update.

4 A Closer Look at Teacher-Student Relationship in Distillation

4.1 How to Narrow the Gap?

In this section, we will explore what factors affect this gap (Equation 3). We first approximate this expression using a Taylor second expansion:

$$\begin{aligned} G_{gap} =& log\sum_{j}^{K}\exp\left(z_j(\theta^t)\right) - log\sum_{j}^{K}\exp\left(z_j(\theta^s)\right) \\ \approx& \log\left(K + \sum_{j}^{K} z_j(\theta^t) + \frac{1}{2}\sum_{j}^{K} z_j(\theta^t)^2\right) - \log\left(K + \sum_{j}^{K} z_j(\theta^s) + \frac{1}{2}\sum_{j}^{K} z_j(\theta^s)^2\right) \end{aligned} \tag{6}$$

Following Hinton's assumption (Hinton et al., 2015) and also through experimental phenomena (Guo, 2022), 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:

$$\begin{aligned} G_{gap} &= \log\left(K + \frac{1}{2}\sum_{j}^{K}\left(z_j(\theta^t)\right)^2\right) - \log\left(K + \frac{1}{2}\sum_{j}^{K}\left(z_j(\theta^s)\right)^2\right) \\ &= \log\left(1 + \frac{1}{2K}\sum_{j}^{K} z_j(\theta^t)^2\right) - \log\left(1 + \frac{1}{2K}\sum_{j}^{K} z_j(\theta^s)^2\right) \\ &= \log\left(1 + \frac{1}{2} * \sigma_t^2\right) - \log\left(1 + \frac{1}{2} * \sigma_s^2\right), \end{aligned} \tag{7}$$

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 KD with different temperatures, and our proposed methods (FT-KD and PESF-KD) to check out how temperature and respective methods affect the gap.

In the real situation, the variance of students' logits will be affected by the variance of teachers' logits, which makes our directly mathematical analysis of Equation 7 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 Figure 2, the gap comparison in Figure 3 and the Top-1 accuracy of the respective methods in Figure 4.

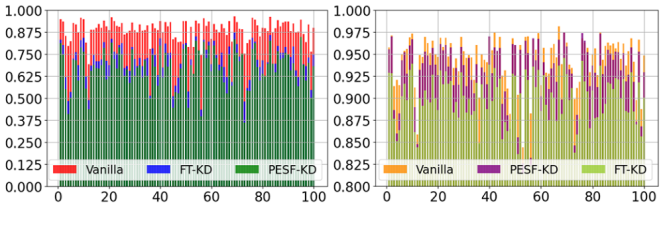


Figure 2: Normalized major logits distribution (Left: student; Right: teacher).

4.2 Gap Analysis

Appropriate soft labels for teacher networks can reduce the sharpness gap. In Figure 2, we show the output variance of the student and teacher can be greatly affected by different KD methods. The logit has the largest value that represents the model's category prediction. Obviously, vanilla KD 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 teacher network, the output of the teacher network becomes smoother compared to vanilla KD.

The smoothing of the teacher's network by gradient automatic adjustment not only reduces the trouble of manually adjusting the temperature, but also can obtain lower gap values in different temperature ranges, as shown in Figure 3. Consistent with the constrain of Equation 1 and 7, since the teacher's output is unchanged, that is, the σ_t calculated in Equation 7 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 Equation 1), which reduces the distinction of category information and causes students to fail to learn useful information, and the variance of its output is instead smaller, so gap becomes larger.

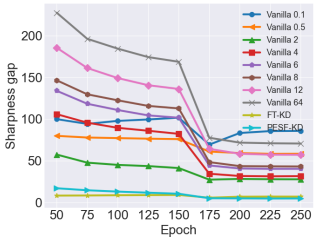


Figure 3: Sharpness gap comparison.

In Figure 4, 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 (Guo, 2022) scales the probability for all categories equally, erasing inter-class information (Müller et al., 2019). 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 KD with the best temperature setting, our methods have improved a lot about the accuracy, and a large reduction in the sharpness gap).

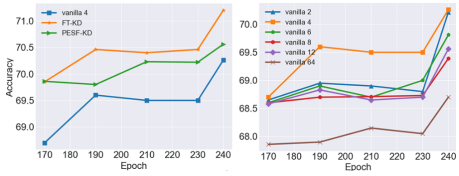


Figure 4: Comparison of Top-1 accuracy.

Appropriate soft labels for teacher networks can make the knowledge transfer better. 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 Figure 3, KL-Divergence in Figure 5 & 6 and CKA (Kornblith et al., 2019) in Figure 7 & 8) and accuracy in Figure 4. 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 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 Equation 7, it is clear that the gap also decreases when the student network is trained with the vanilla KD. 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).

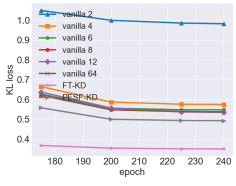


Figure 5: KL loss comparison.

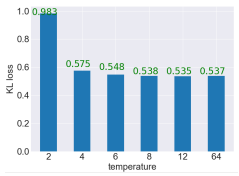


Figure 6: KL comparison of vanilla KD with different temperature.

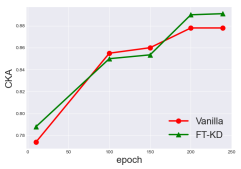


Figure 7: CKA of feature logits (last layer before classifier).

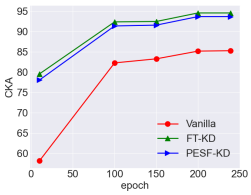
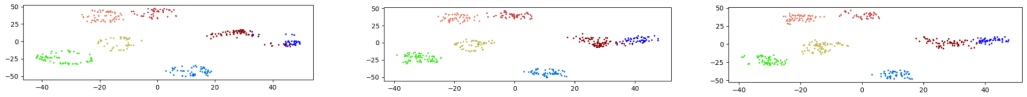


Figure 8: CKA of predictions vanilla KD and proposed method

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 KD with different temperatures).



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

Figure 9: Visualize of penultimate layer representation of with 7 semantically different classes.

Table 1: Results on CIFAR-100 test set. We compare the top-1 accuracy on various teacher-student combinations. The best results are **bold**. The second best results are underlined.We report the averaged results over 3 random seeds.

	method	resnet56-20	resnet110-32	vgg13-8	resnet56-vgg8
offline KD	Vanilla KD(Hinton et al., 2015)	$70.95_{0.51}$	$73.08_{0.42}$	$73.36_{0.24}$	$73.98_{0.33}$
	PKT(Passalis & Tefas, 2018)	71.27 (+0.32)	73.67 (+0.59)	73.40 (+0.04)	74.10 (+0.12)
Representation KD	CRD(Tian et al., 2020)	71.44 (+0.49)	73.62 (+0.54)	73.31 (-0.05)	74.06 (+0.08)
	RKD (Park et al., 2019)	71.47 (+0.52)	73.53 (+0.45)	74.15 (+0.79)	73.35 (-0.63)
online KD	KDCL (Guo et al., 2020)	70.11 (-0.84)	72.87 (-0.21)	73.99 (+0.63)	73.16 (-0.82)
	DML(Zhang et al., 2018)	71.40 (+0.45)	72.21 (-0.87)	74.18 (+0.82)	73.86 (-0.12)
ours (online KD)	FT-KD	$\underline{71.65_{0.11}(+0.70)}$	$\underline{73.90_{0.22}(+0.82)}$	$\underline{73.52_{0.14}(+0.16)}$	$\underline{74.40_{0.2}(+0.42)}$
	PESF-KD	$\underline{71.84_{0.27}(+0.89)}$	$\underline{74.23_{0.26}(+1.15)}$	$\underline{74.74_{0.39}(+1.38)}$	$\underline{74.67_{0.28}(+0.69)}$

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 Figure 9 using T-SNE (Van der Maaten & Hinton, 2008). 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 KD.

5 Experiments

5.1 Datasets and Baselines.

Datasets Two types of tasks including image classification (CIFAR-100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009)) and natural language understanding (GLUE (Wang et al., 2019)) 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 (Jiao et al., 2020): SST-2 (Socher et al., 2013) for Sentiment Classification; RTE (Wang et al., 2019) for the Natural Language Inference; QQP [1] for Paraphrase Similarity Matching.

Baselines. We report several knowledge distillation methods for comparison, including vanilla KD (Hinton et al., 2015), knowledge distillation via collaborative learning (KDCL) (Guo et al., 2020), deep mutual learning (DML) (Zhang et al., 2018), contrastive representation distillation (CRD) (Tian et al., 2020), relational knowledge distillation (RKD) (Park et al., 2019) and probabilistic knowledge transfer(PKT) (Passalis & Tefas, 2018). According to Gou et al. (2021) 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 KD, PKT) and representation KD (CRD, RKD). On NLP datasets, we compare one offline distillation method (Vanilla KD) 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.

Experimental Setup. For CV tasks, we follow previous works (Tian et al., 2020) using various combinations of student & teacher networks. Each pair of student & teacher networks are from 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

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

Table 2: Results on ImageNet test set. ResNet18/ResNet50 is the student/teacher model. We compare the training time consumption (batch time) and the amount of parameters needed to update (params) and the corresponding accuracy. The best results are **bold**. The second best results are underlined. We report the averaged results over 3 random seeds.

	method	batch time	params	@1
offline KD	Vanilla KD (Hinton et al., 2015)	0.46	47M	$69.20_{0.25}$
	PKT (Passalis & Tefas, 2018)	0.47	47M	69.69 (+0.49)
Representation KD	CRD (Tian et al., 2020)	0.57	63M	69.33 (+0.13)
	RKD (Park et al., 2019)	1.53	47M	69.52 (+0.32)
online KD	KDCL (Guo et al., 2020)	1.56	149M	70.17 (+0.97)
	DML (Zhang et al., 2018)	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	$70.55_{0.14}(+1.35)$

Table 3: Results on the three NLP datasets. The best results are **bold**. The second best results are underlined. We report the averaged results over 3 random seeds.

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 KD (Hinton et al., 2015)	66M	1.0x	67.7	91.2	87.3
RCO (Jin et al., 2019)	>176M	>2.66x	67.6 (-0.1)	91.4 (+0.2)	87.4 (+0.1)
TAKD (Mirzadeh et al., 2020)	>132M	>2.00x	68.5 (+0.8)	91.4 (+0.2)	87.5 (+0.2)
DML (Zhang et al., 2018)	176M	2.66x	68.4 (+0.7)	91.5 (+0.3)	87.4 (+0.1)
SFTN (Park et al., 2021)	>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)**

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 Top 1 Accuracy (@1) for image classification experiments as a network performance metric. For QQP we report F1. For other NLP tasks, we report accuracy.

5.2 RESULTS

1) Our method gets better (in most cases) or comparable performance compared to the competitors, which demonstrates the power of our simple method 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 1) and a larger dataset ImageNet (Table 2), our PESF-KD achieves the best top-1 accuracy among all distillation methods. On most NLP datasets (Table 3), PESF-KD also achieves the best results across various online KD methods.

2) Even if the accuracy is slightly lower, our method significantly reduces training resource consumption compared to the competitors (online KD), see Table 2 and Table 3. As shown in Table 2 and Table 3, 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 for training. The more detailed training consumption results can be seen in the appendix. On the CV datasets (Table 1 and Table 2), PESF-KD improves student's accuracy even though our PESF-KD uses fewer training resources and less batch time. On NLP datasets (Table 3), 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 increase in distillation accuracy of our method is greater, confirming the role of soft labels 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 §4.2), and the improvement is relatively insignificant in datasets with fewer categories (SST-2 and QQP, see Table 3) 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 continues to improve results compared to traditional online distillation even when inter-class information is more absent.

Orthogonality to other KD methods *PESF-KD on the bottom of other KD methods can further get improvements.* Table 4 shows the results of four knowledge distillation approaches combined with PESF-KD on the CV dataset (CIFAR-100) and Table 5 shows our results of PESF-KD combined with intermediate layer distillation (PKD and TinyBERT[2]) on the NLP dataset (RTE and SST-2).

Our method can improve other KD methods on the CV dataset, but the middle-layer distillation on NLP can be a little 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 Xu et al. (2021), PESF-KD improves consistency by changing the teachers' output logits, which is helpful in mitigating the loss of consistency from middle-layer distillation. We give a more detailed explanation in the appendix.

Table 4: Comparison of orthogonality with existing methods on the CIFAR-100 dataset.

Teacher/Student	ResNet56/ ResNet20	
Method	Stadard	PESF-KD
KD (Hinton et al., 2015)	70.95	71.84 (+0.89)
PKT (Passalis & Tefas, 2018)	71.27	71.90 (+0.63)
CRD (Tian et al., 2020)	71.44	71.97 (+0.53)
RKD (Park et al., 2019)	71.47	71.72 (+0.25)

Table 5: TinyBERT and PKD combined with PESF-KD on RTE and SST-2. We present the results with 3 different random seeds in the form of mean (standard deviation).

Method	Accuracy(↑)	Standard / PESF-KD PL(↑) (Xu et al., 2021)	Agree(↑) (Stanton et al., 2021)	Gap*10^-2(↓)
		RTE		
TinyBERT (Jiao et al., 2020)	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 (Sun et al., 2019)	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 (Jiao et al., 2020)	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 (Sun et al., 2019)	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)

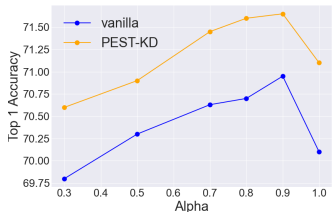


Figure 10: Loss weight results on CIFAR-100.

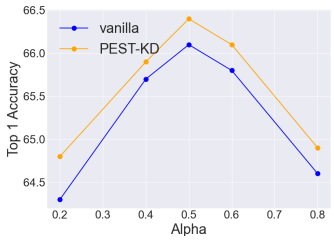


Figure 11: Loss weight on results on RTE.

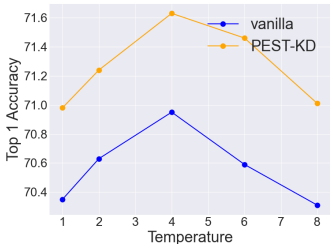


Figure 12: Temperature results on CIFAR-100.

Loss Weight and Temperature *PESF-KD shows better performance and robustness compared to vanilla KD in all experiments.* In vanilla, the hyper-parameter α is a loss weight that needs non-trivial tuning like hyperparameter search. Bigger 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 Figure 10 and 11. It shows that 1) properly 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. Figure 12 shows the performance of student models in different temperatures. Vanilla KD and our proposed approach have a similar trends in different temperatures.

6 Conclusion

In this paper, we show the smoothness of labels affects the teacher-student mismatch. To reduce this mismatch and to balance the difficulty and cost of training, we present PESF-KD, a novel knowledge distillation framework by applying 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 demonstrate the robustness and effectiveness of our method.

[2]TinyBERT is a two-stage distillation method, and for time and fairness reasons we only performed the second stage of distillation (without distillation on pretrain 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-uncased for the distillation of the intermediate and prediction layers.

REFERENCES

Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *CVPR*, 2019.

Prashant Bhat, Elahe Arani, and Bahram Zonooz. Distill on the go: Online knowledge distillation in self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2678–2687, 2021.

Keshigeyan Chandrasegaran, Ngoc-Trung Tran, Yunqing Zhao, and Ngai-Man Cheung. Revisiting label smoothing and knowledge distillation compatibility: What was missing? In *ICML*, 2022.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, 2021.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *IJCV*, 2021.

Jia Guo. Reducing the teacher-student gap via adaptive temperatures. *OpenReview preprint*, 2022.

Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *CVPR*, 2020.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *ICLR*, 2022.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI*, 2019.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NeurIPS*, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *ICML*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *EMNLP Findings*, 2020.

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *ICCV*, 2019.

Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *NeurIPS*, 2018.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *ICML*, 2019.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, 2020.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *NeurIPS*, 2019.

Dae Young Park, Moon-Hyun Cha, Daesin Kim, Bohyung Han, et al. Learning student-friendly teacher networks for knowledge distillation. *NeurIPS*, 2021.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019.

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, 2018.

Baoyun Peng, Xiao Jin, Dongsheng Li, Shunfeng Zhou, Yichao Wu, Jiaheng Liu, Zhaoning Zhang, and Yu Liu. Correlation congruence for knowledge distillation. In *ICCV*, 2019.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew Gordon Wilson. Does knowledge distillation really work? In *NeurIPS*, 2021.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In *EMNLP*, 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *ICCV*, 2019.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. BERT-of-theseus: Compressing BERT by progressive module replacing. In *EMNLP*, 2020.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian J. McAuley, and Furu Wei. Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression. In *EMNLP*, 2021.

Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *CVPR*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

Lin Zhang, Li Shen, Liang Ding, Dacheng Tao, and Ling-Yu Duan. Fine-tuning global model via data-free knowledge distillation for non-iid federated learning. In *CVPR*, 2022.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *CVPR*, 2018.

Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. Bert learns to teach: Knowledge distillation with meta learning. In *ACL*, 2022.

Yichen Zhu and Yi Wang. Student customized knowledge distillation: Bridging the gap between student and teacher. In *ICCV*, 2021.

Table 6: Statistics of datasets

dataset	Training set	Testing set	Development set	Classes
CIFAR-100(Krizhevsky et al., 2009)	50000	10000	-	100
ImageNet(Deng et al., 2009)	≈1200k	≈100k	≈50k	1000
SST-2	67350	1821	873	2
QQP	363870	390965	40431	2
RTE	2491	3000	277	2

A Appendix

A.1 Training details

Datasets Table 6 shows the statistics used in our experiments. The training set, testing set, and development set refer to the scale of data in each part of the dataset. Table 6 shows that ImageNet is much bigger than CIFAR-100 in scale. And that's the reason we choose the two datasets, to verify our proposed approach in datasets of different scales. We are also concerned about people's consent and privacy in datasets. From the information search from the website, faces in ImageNet have been blurred in order to protect privacy after being blamed to collect information from people without consent. But we do not find a similar announcement on the website of CIFAR-100 and GLUE. Maybe it's because they have pre-processed the data or the data itself contains little privacy information. It's better for the websites to make the announcement of not containing personally identifiable information or offensive content.

Infrastructure We implement our models with Pytorch, and our experiments are as follows:

1. CPU: 256 AMD EPYC 7742 64-Core Processor
2. RAM: 386840MB
3. GPU: 8x GeForce RTX 3090
4. Operating System: Ubuntu 18.04 LTS
5. Tools: Python3.7, tensorflow2,2,0, sklearn 0.23.2

Hyper-parameter search In CV experiment, we follow previous works (Tian et al., 2020), the settings on CIFAR-100 and ImageNet dataset are the same as these works. In CIFAR-100 we train the student model by SGD optimizer with a momentum of 0.9, a batch size of 64 and weight decay of 5×10^{-4}. The learning rate starts from 0.05 and decays by 10 every 30 epochs after 150 epochs. And on ImageNet we train the student model by SGD optimizer with a momentum of 0.9, a batch size of 256 and weight decay of 1×10^{-4}. The learning rate starts from 0.1 and decays by 10 every 30 epochs after 30 epochs. In the experiment of FT-KD, we train the teacher model along with the student model during the training process with a learning rate of 1×10^{-3}. And in the experiment of a student trained with a teacher with the adapter module, also called adaptive teacher, the learning rate of the trainable part in the teacher model is set to 1×10^{-4}. Notably, classification loss from the teacher model is appended to the loss of students by multiplying a hyper-parameter of 0.5.

For NLP, we inherit parameters like maximum sequence length, temperature and batch size according to setting from previous works (Zhou et al., 2022). 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}, the weight of KD loss from {0.3, 0.5, 0.8, 0.9} for a better performance. We evaluate our methods on the dev set of the GLUE benchmark.

A.2 More detailed results

We show more detailed experimental results (number of trainable parameters, batch time and accuracy), as shown in the Table 7 and Table 9. To further illustrate the superiority of our methods, we further compare the current typical distillation methods like previous work (Tian et al., 2020), as shown in Table 8.

Table 7: Results on CIFAR-100 test set. We compare the training time consumption (batch time) and the amount of parameters needed to update (params) and the corresponding accuracy (@1) on various teacher-student combinations. The best results are **bold**. The second best results are underlined.

		resnet56-20			resnet110-32			vgg13-8			resnet56-vgg8		
	method	batch time	params	@1	batch time	params	@1	batch time	params	@1	batch time	params	@1
offline KD	Vanilla KD(Hinton et al., 2015)	0.219	1.11M	70.95	0.319	1.89M	73.08	0.097	15.86M	73.36	0.134	15.86M	73.98
	PKT(Passalis & Tefas, 2018)	0.225	1.11M	71.27	0.336	1.89M	73.67	0.120	15.86M	73.40	0.139	15.86M	74.10
Representation KD	CRD(Tian et al., 2020)	0.234	1.18M	71.44	0.353	1.96M	73.62	0.139	16.39M	73.31	0.159	15.93M	74.06
	RKD (Park et al., 2019)	0.234	1.11M	71.47	0.846	1.89M	73.53	0.207	15.86M	74.15	0.444	15.86M	73.35
online KD	KDCL (Guo et al., 2020)	0.520	4.56M	70.11	1.035	8.84M	72.87	0.190	53.71M	73.99	0.433	19.31M	73.16
	DML(Zhang et al., 2018)	0.524	4.56M	71.40	0.912	8.84M	72.21	0.190	53.71M	<u>74.18</u>	0.437	19.31M	73.86
ours (online KD)	FT-KD	0.481	4.56M	<u>71.65</u>	0.819	8.84M	<u>73.90</u>	0.156	53.71M	73.52	0.443	19.31M	<u>74.40</u>
	PESF-KD	0.228	1.16M	**71.84**	0.325	1.94M	**74.23**	0.117	15.91M	**74.74**	0.136	15.91M	**74.67**

Table 8: More Results on CIFAR-100 test set. The best results are **bold**. The second best results are underlined.

teacher student	resnet56 resnet20	resnet110 resnet32	vgg13 vgg8	resnet56 vgg8
teacher	72.34	74.31	74.64	79.34
student	69.06	71.14	70.36	70.36
KD (Hinton et al., 2015)	70.66	73.08	72.98	73.81
FitNet (Romero et al., 2015)	69.21	71.06	71.02	70.69
AT (Zagoruyko & Komodakis, 2017)	70.55	72.31	71.43	71.84
SP (Tung & Mori, 2019)	69.67	72.69	72.68	73.34
CC (Peng et al., 2019)	69.63	71.48	70.71	70.25
VID (Ahn et al., 2019)	70.38	72.61	71.23	70.30
AB (Heo et al., 2019)	69.47	70.98	70.94	70.65
FT (Kim et al., 2018)	69.84	72.37	70.58	70.29
FSP (Yim et al., 2017)	69.95	71.89	70.23	73.90
FT-KD	<u>71.65</u>	<u>73.90</u>	<u>73.52</u>	<u>74.40</u>
PESF-KD	**71.84**	**74.23**	**74.74**	**74.67**

Table 9: Results on ImageNet test set. ResNet18 is the student model and ResNet50 is the teacher model. We report the averaged results over 3 random seeds. The best results are **bold**.

	method	batch time	params	@1	KL loss	GAP
offline KD	Vanilla KD (Hinton et al., 2015)	0.46	47M	69.20	10.3	19.3
	PKT (Passalis & Tefas, 2018)	0.47	47M	69.69	10.3	19.2
Representation KD	CRD (Tian et al., 2020)	0.57	63M	69.33	10.5	20.0
	RKD (Park et al., 2019)	1.53	47M	69.52	9.6	20.5
online KD	KDCL (Guo et al., 2020)	1.56	149M	<u>70.17</u>	**6.8**	14.8
	DML (Zhang et al., 2018)	1.45	149M	70.00	**6.7**	**14.1**
ours (online KD)	FT-KD	1.44	149M	70.00	**6.7**	<u>14.9</u>
	PESF-KD	0.47	54M	**70.55**	7.2	15.2

A.3 Further Analysis on our methods

A.3.1 Discrepancy between teacher and student predictions

In this section, we compare the associated discrepancy metrics for teachers and students. These metrics are: Probability Loyalty (PL↑) (Xu et al., 2021), Kullback-Leibler divergence (KL↓), Average Top-1 Agreement (Agree↑) (Stanton et al., 2021) and Sharpness Gap (Gap↓), where ↑ means the greater the better and ↓ means smaller the better.

Discrepancy in CV dataset First, we further measure the relevant consistency metrics in the CV dataset(CIFAR-100), as shown in Table 10. We use resnet56/20 as the teacher-student combination. Compared to vanilla KD, 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 (second best) and FT-KD (best) outperform vanilla KD 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 4.2.***

Table 10: Results of PESF-KD on CIFAR-100 with standard deviation. PESF-KD gets higher accuracy and lower deviation compared with vanilla KD. We present the results with 3 different random seeds in the form of mean/standard deviation. † denotes the results without label adjustment strategy.

Method	Accuracy(↑)	KL(↓)	PL(↑)	Agree(↑)
Vanilla KD	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

Discrepancy in NLP dataset We experiment further on two datasets, SST-2 (67K, high resource) and RTE (2.5K, low resource). And the results are shown in Table 11. We discover as Xu et al. (2021) 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((Xu et al., 2021) tests consistency in MNLI (393K)). ***We found that PESF-KD performed better in both consistency metrics in the high resource case (SST-2), although in the low resource case (RTE), only KL/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 in the main part achieve distillation goals based on prediction layers, while PKD (Sun et al., 2019) and TinyBERT (Jiao et al., 2020) utilize distillation of intermediate layers for further combinations, and we show the total training objects for PKD:

$$\mathcal{L}_{PKD} = (1-\alpha)\mathcal{L}^s_{CE} + \alpha\mathcal{L}_{KL} + \beta\mathcal{L}_{PT} \tag{8}$$

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 (Sun et al., 2019), 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.

Table 11: The results of PESF-KD on SST-2 and RTE dataset compared with vanilla KD and PKD and TinyBert. PESF-KD shows better performance on consistency metric (PL, KL, Agree, Gap) compared with vanilla KD and PKD. **Gap is computed by soft-max prediction differed from the main part using original prediction. We present the results with 3 different random seeds in the form of mean/standard deviation. † denotes the results without label adjustment strategy.**

	SST-2(67K)				RTE(2.5K)			
Method	PL(↑)	KL*10^{-2}(↓)	Agree(↑)	**Gap***10^{-2}(↓)	PL(↑)	KL*10^{-2}(↓)	Agree(↑)	**Gap***10^{-2}(↓)
Vanilla KD	92.1/0.7	6.8/0.4	91.0/0.7	4.0/0.2	85.7/2.1	10/0.3	71.8/3.9	1.9/0.2
FT-KD	96.3/0.2	2.4/0.2	98.1/0.7	1.8/0.3	87.8/1.3	8.6/0.5	83.2/3.0	1.8/0.6
PESF-KD†	97.5/0.9	1.8/0.5	98.3/0.7	1.7/0.4	88/1.76	8.3/0.5	83.2/5.0	1.8/0.3
PKD	92.6/0.5	6.7/0.4	91.8/0.4	4.0/0.2	88.2/0.7	9.7/0.5	76.2/1.5	1.7/0.3
TinyBERT	97.3/0.3	2.1/0.2	99.2/0.2	1.1/0.5	96.4/0.5	8.6/0.4	84.2/3.0	1.2/0.3

A.3.2 ORTHOGONALITY TO OTHER KD METHODS

Table 12: Comparison of orthogonality with existing methods on the CIFAR-100 dataset. † denotes the results without label adjustment strategy.

Teacher/Student	ResNet56/ ResNet20	
Method	Stadard	PESF-KD†
KD (Hinton et al., 2015)	70.95	71.63
PKT (Passalis & Tefas, 2018)	71.27	71.74
CRD (Tian et al., 2020)	71.44	71.76
RKD (Park et al., 2019)	71.47	71.52

Table 13: TinyBERT and PKD combined with PESF-KD on RTE and SST-2. **Gap is computed by soft-max prediction differed from the main part using original prediction. We present the results with 3 different random seeds in the form of mean/standard deviation. † denotes the results without label adjustment strategy.**

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

We also test PESF-KD in other methods to verify the potential improvement of other knowledge distillation methods even without the label adjustment strategy.

Table 12 shows results of four knowledge distillation approaches (vanilla KD (Hinton et al., 2015), PKT (Passalis & Tefas, 2018), CRD (Tian et al., 2020) and RKD(Park et al., 2019)) combined with PESF-KD on the CV dataset(CIFAR-100). Every traditional KD method combined with PESF-KD can get better performance.

Table 13 shows our results of PESF-KD combined with intermediate layer distillation (PKD and TinyBERT) on the NLP dataset(RTE and SST-2). 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 Xu et al. (2021), PESF-KD improves consistency by changing the teachers' output logits, which is helpful in mitigating the loss of consistency from middle-layer distillation.

Compared with different knowledge distillation approaches varying from NLP and CV, results with PESF-KD get better performance on accuracy and indicate that PESF-KD has the potential to become a plug-in method on top of different KD methods.

A.3.3 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, 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 has the great potential to apply to a broad range of tasks.*

Adapter dimension As shown in Figure 13, we test the impact of different scaling dimensions of adapter on classification tasks on CIFAR-100. 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 He et al. (2022) to make the dimension of the model lower than the input dimension and finally chose 32.

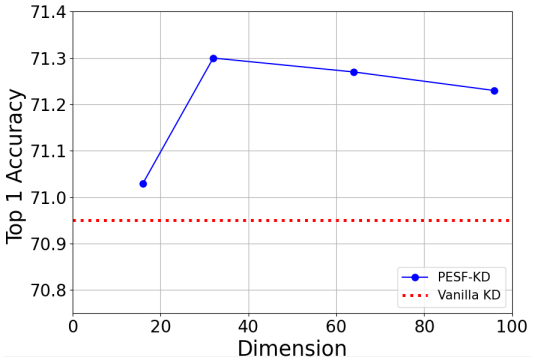


Figure 13: Influence of adapter dimension on results of PESF-KD. We report the averaged results over 3 random seeds. We set dimension from the set of $\{16, 32, 64, 96\}$. **Dimension = 32 is an appropriate choice.**

Table 14: Ablation on different structures of adapter in VGG. Experiments are performed from CIFAR-100. **The simple adapter shows decent performance compared with other parameter efficient methods in top 1 accuracy, CKA and time consuming.**

Method	Batch Time(↓)	Parameter(↓)	Top-1(↑)	CKA(↑)
vanilla KD (Hinton et al., 2015)	0.097	15.86M	73.36	0.8537
Adapter (Houlsby et al., 2019)	0.137	15.95M	73.43	0.8758
LoRA (Hu et al., 2022)	0.166	15.95M	73.49	0.8579
Scaled PA (He et al., 2022)	0.165	15.95M	73.13	0.8594

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 KD and three classic adapter structures, and report their top-1 accuracy and CKA consistency. We use the teacher-student combination of vgg13-8 and conduct comparative experiments on CIFAR-100. As can be seen from Table 14, the simple and efficient adapter achieves second performance with top-1 accuracy, best CKA scores, and is less time-consuming. Therefore we use the simple adapter module in all experiments.