RETHINKING KNOWLEDGE DISTILLATION VIA CROSS-ENTROPY

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge Distillation (KD) has developed extensively and boosted various tasks. The classical KD method adds the KD loss to the original cross-entropy (CE) loss. We try to decompose the KD loss to explore its relation with the CE loss. Surprisingly, we find it can be regarded as a combination of the CE loss and an extra loss which has the identical form as the CE loss. However, we notice the extra loss forces the student's relative probability to learn the teacher's absolute probability. Moreover, the sum of the two probabilities is different, making it hard to optimize. To address this issue, we revise the formulation and propose a distributed loss. In addition, we utilize teachers' target output as the soft target, proposing the soft loss. Combining the soft loss and the distributed loss, we propose a new KD loss (NKD). Furthermore, we smooth students' target output to treat it as the soft target for training without teachers and propose a teacher-free new KD loss (tf-NKD). Our method achieves state-of-the-art performance on CIFAR-100 and ImageNet. For example, with ResNet-34 as the teacher, we boost the ImageNet Top-1 accuracy of ResNet18 from 69.90% to 71.96%. In training without teachers, MobileNet, ResNet-18 and SwinTransformer-Tiny achieve 70.04%, 70.76%, and 81.48%, which are 0.83%, 0.86%, and 0.30% higher than the baseline, respectively.

1 INTRODUCTION

Over the last decade, deep convolutional neural networks (CNNs) have significantly advanced the performance in many computer vision tasks (He et al., 2016; Ren et al., 2015; Ronneberger et al., 2015; He et al., 2017). Generally, a larger model scores higher but needs more computing resources. In contrast, a smaller model has less computation complexity and runs faster, but it performs less competitively than the larger one. Knowledge distillation (KD) is proposed (Hinton et al., 2015) to bridge the gap to boost the small model in the training stage, causing no extra cost in the test time. Its core idea is when training the small student model; besides the supervision from the label, it also inherits the knowledge from the large teacher model as additional guidance. The distillation methods have been successfully applied to various domains, such as image classification (Yang et al., 2020; Zhou et al., 2020; Chen et al., 2021; Zhao et al., 2022; Yang et al., 2022), object detection (Chen et al., 2017; Li et al., 2017; Wang et al., 2019; Guo et al., 2021; Yang et al., 2021), and semantic segmentation (Liu et al., 2019; He et al., 2019; Shu et al., 2021; Yang et al., 2022).

The classical distillation method (Hinton et al., 2015) utilizes the teacher's prediction as the soft label to guide the student. In addition to the predicted labels given by the teacher, there are also artificially given soft labels. Label smooth can be regarded as a particular case of the soft label and (Müller et al., 2019) show it helps the models to represent the samples from the same class to the group in tight clusters. Tf-KD (Yuan et al., 2020) explore the knowledge distillation from label smoothing regularization and propose a novel teacher-free knowledge distillation method. DKD (Zhao et al., 2022) regards the image's category as the target class, reformulating the knowledge distillation loss into target and non-target class loss.

However, previous works lack the consideration of the relation between the original CE loss and KD loss. From this perspective, we investigate the classical KD loss and find that the KD loss can be reformulated as a combination of the original CE loss and an extra loss. The extra loss mainly introduces the knowledge of all classes except the target class, which we call non-target distribution. Besides, the extra loss has the same form as CE loss. However, the extra loss aims to force the

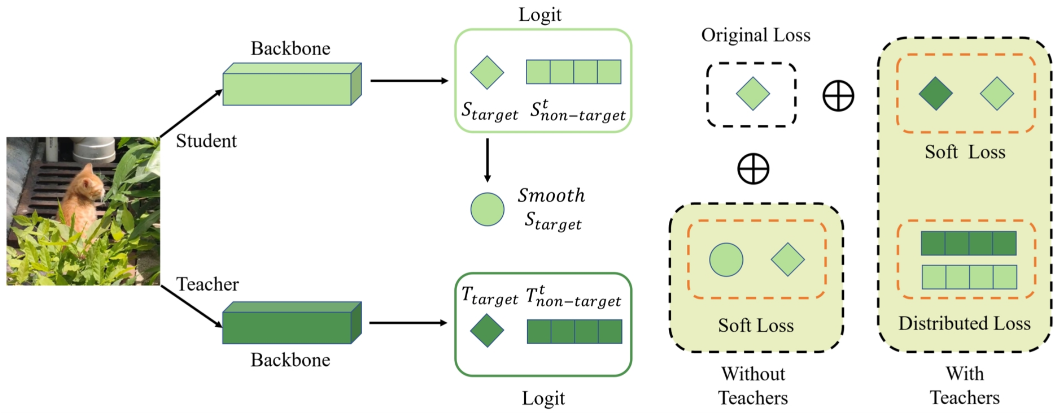


Figure 1: Illustration of the proposed NKD and tf-NKD.

student's relative probability to be the same as the teacher's absolute probability. To this end, although the decomposition provides a term that looks like CE loss, the student's output should not be similar to the teacher's output when converging. To solve this problem, we modify the formula and proposed distributed loss to transfer the knowledge of non-target distribution.

Besides the non-target distribution, we believe the target information should also be reasonably introduced into the knowledge distillation. Inspired by the phenomenon that the soft label is easier than the ground-truth (GT) label for a compact model to fit, we argue that the teacher's target output can also be viewed as the soft target. This soft target gives a much smoother label value than the GT label value. Based on the method, we apply the soft targets to the samples and propose the soft loss. Since the distributed loss provides non-target distribution knowledge and the soft loss provides the soft target knowledge, we can use these two losses in combination. In this way, we present our **N**ew **K**nowledge **D**istillation (NKD) loss. The students achieve significant improvements and state-of-the-art performance with our new loss.

Furthermore, as the student's prediction can also give the sample a much smoother label value, we try to use soft loss without teachers. Since student's predictions vary gradually during training, we smooth student's target output to make it more stable during training, propose our **t**eacher **f**ree **N**ew **K**nowledge **D**istillation (tf-NKD) loss. We conduct various experiments to validate the effectiveness and robustness of our tf-NKD loss. Besides, the weights are all obtained by adjusting the student's target output. Therefore compared with the baseline, it does not take extra time to train a model.

As we analyzed above, we propose a new paradigm of knowledge distillation loss, including distributed loss and soft loss. Combining these two losses with the original CE loss, we achieve state-of-the-art performance in various models. Besides, in the absence of teachers, we also get a considerable boost in many models using the soft loss. The two methods we propose are shown in Figure 1. In a nutshell, the contributions are as follows:

- We demonstrate that the classical KD loss can be regarded as the combination of the original CE loss and an extra loss. The extra loss mainly introduces the knowledge of non-target distribution and is hard to optimize. To address this issue, we propose distributed loss.
- Inspired by the soft label method, we propose soft loss, which uses the teacher's target output as the soft target for distillation. Combining distributed and soft loss, we propose NKD loss, achieving state-of-the-art performance.
- We smooth the student's target output as the soft target to train the students without teachers. In this way, we propose tf-NKD loss, which can also bring considerable improvements for the students without extra time costs.

2 Related work

Knowledge distillation (KD) is a method to improve the model while keeping the network structure unchanged. It was first proposed by Hinton et al. (Hinton et al., 2015), where the student is supervised by the hard labels and the soft labels from the teacher's output. Many following works focus on making better use of soft labels to transfer more knowledge. WSLD (Zhou et al., 2020) analyzes soft labels and distributes different weights for them from a perspective of bias-variance trade-off. DKD (Zhao et al., 2022) divides the classical KD according to the teacher's prediction and modifies the formulation of KD, achieving state-of-the-art performances. SRRL (Yang et al., 2020) forces the output logits of teacher's and student's features after the teacher's linear layer to be the same.

Besides distillation on logits, some works aim at transferring knowledge from intermediate features. FitNet (Romero et al., 2014) distills the semantic information from intermediate feature directly. OFD (Heo et al., 2019) designs the margin ReLU and modifies the measurement for the distance between students and teachers. RKD (Park et al., 2019) extracts the relation from the feature map. CRD (Tian et al., 2019) applies contrastive learning to distillation successfully. KR (Chen et al., 2021) transfers knowledge from multi-level features for distillation. SRRL (Yang et al., 2020) utilizes the teacher's classifier to train the student's feature. MGD (Yang et al., 2022) proposes a new distillation method that makes the student generate the teacher's feature instead of mimicking.

3 METHOD

3.1 DISTILLATION WITH TEACHERS

Using t denotes the target class, C denotes the number of classes, V_i denotes the label value, and S_i denotes the student's output probability. The original loss for image classification can be formulated:

$$L_{ori} = -\sum_{i}^{C} V_i log(S_i) = -V_t log(S_t) = -log(S_t). \tag{1}$$

Using λ denotes the temperature for knowledge distillation, T_i denotes the teacher's output probability. The classical KD loss can be formulated as:

$$L_{kd} = -\sum_{i}^{C} T_i^\lambda log(S_i^\lambda) = -\sum_{i}^{C} T_i^\lambda log(S_t^\lambda * \frac{S_i^\lambda}{S_t^\lambda}) \tag{2}$$

$$= -\sum_{i}^{C} T_i^\lambda log(S_t^\lambda) - \sum_{i}^{C} T_i^\lambda log(\frac{S_i^\lambda}{S_t^\lambda}). \tag{3}$$

Because $\sum_i^C T_i^\lambda = \sum_i^C S_i^\lambda = 1$ and $T_t^\lambda log(S_t^\lambda / S_t^\lambda) = 0$, L_{kd} can be simplified as:

$$L_{kd} = -log(S_t^\lambda) - \sum_{i \neq t}^{C} T_i^\lambda log(\frac{S_i^\lambda}{S_t^\lambda}). \tag{4}$$

As L_{kd} shows, $-log(S_t^\lambda)$ has the same form as L_{ori} and does not bring new knowledge for the student. While for the extra loss $-\sum_{i \neq t}^{C} T_i^\lambda log(S_i^\lambda / S_t^\lambda)$, it has the same form as CE loss $-\sum p(x) log(q(x))$ and mainly introduces the non-target knowledge to the student. The CE loss aims at making $q(x)$ to be the same as $p(x)$. Therefore the sum of the two distributions needs to be equal. However, T_i^λ is the absolute probability and $\sum_{i \neq t}^{C} T_i^\lambda = 1 - T_t^\lambda$. While $S_i^\lambda / S_t^\lambda$ is the relative probability and $\sum_{i \neq t}^{C} S_i^\lambda / S_t^\lambda = (1 - S_t^\lambda)/S_t^\lambda$. So S_i/S_t is hard to be similar to T_i. In short, when combining L_{ori} and L_{kd} together for distillation, we need to view the two losses together. So we try to decompose L_{kd} via L_{ori} and find the optimization problem we describe above.

To transfer the non-target knowledge and overcome the problem, we propose distributed loss:

$$L_{distributed} = -\sum_{i \neq t}^{C} \hat{T}_i^\lambda log(\hat{S}_i^\lambda). \tag{5}$$

$$\hat{T}_i^\lambda = \frac{T_i^\lambda}{1 - T_t^\lambda}, \quad \hat{S}_i^\lambda = \frac{S_i^\lambda}{1 - S_t^\lambda}.$$

In this case, we can see $\sum_{i\neq t}^{C}\hat{T}_i^{\lambda} = \sum_{i\neq t}^{C}\hat{S}_i^{\lambda} = 1$, making the student learn teacher's non-target knowledge easier.

However, $L_{distributed}$ lacks the teacher's target knowledge. Some previous KD methods (Hinton et al., 2015; Zhao et al., 2022) have proved that the teacher's predictions can be utilized as soft labels to accelerate the convergence and improve student performance. Inspired by this soft label method, we regard the teacher's target output probability T_t as the soft target directly. Based on the soft target the teacher introduces, we propose the soft loss for distillation with teachers:

$$L_{soft} = -T_t log(S_t). \tag{6}$$

Finally, combining the original loss L_{ori}, distributed loss $L_{distributed}$ and soft loss L_{soft}, we propose our **N**ew **K**nowledge **D**istillation (NKD) loss as follows:

$$L_{NKD} = -log(S_t) - T_t log(S_t) - \alpha * \lambda^2 * \sum_{i\neq t}^{C}\hat{T}_i^{\lambda} log(\hat{S}_i^{\lambda}), \tag{7}$$

where α is a hyper-parameter to balance the loss.

3.2 Training without teachers

The weight T_t and $\hat{T}_i^{\lambda}$ in Equation 7 are both got from teacher's output. Therefore L_{NKD} is just suitable for distillation with teachers. The $\hat{T}_i^{\lambda}$ is about the non-target distribution knowledge, which needs to be calculated from a trained model. However, the soft target T_t is the target output probability for the input image. We wonder if the soft target can be provided by adjusting the student's target output S_t. The difference is that T_t is fixed and S_t varies gradually during training. From this perspective, we adjust S_t, making it smoother to fit the training setting without teachers. This method can be applied to different models directly and does not take any extra time compared with the baseline. For the training without teachers, we propose our **t**eacher **f**ree **N**ew **K**nowledge **D**istillation (tf-NKD) loss:

$$L_{tf-NKD} = -log(S_t) - (S_t + V_t - mean(S_t)) log(S_t), \tag{8}$$

where V_t denotes the target label value for the sample and $mean(\cdot)$ is calculated across different samples in a batch. Comparing with L_{soft} in Equation 6, we replace the T_t with adjustable S_t. The experiments of L_{tf-NKD} are shown in Subsection 4.3. We also discuss the effects of different ways to smooth the S_t in Subsection 5.3.

In short, we propose L_{NKD} for distillation with teachers and L_{tf-NKD} for training without teachers.

4 Experiments

4.1 Datasets and Details

We conduct the experiments on CIFAR-100 (Krizhevsky et al., 2009) and ImageNet (Deng et al., 2009), which contains 100 and 1000 categories, respectively. For CIFAR-100, we use the 50k images for training and 10k for validation. For ImageNet, we use 1.2 million images for training and 50k images for validation. In this paper, we use accuracy to evaluate all the models.

For distillation with teachers, NKD has two hyper-parameters α and λ in Equation7. For all the experiments, we adopt $\{\alpha = 1.5, \lambda = 1\}$ on ImageNet. The training setting for distillation is the same as training the students without distillation. We use 8 Tesla-V100 GPUs to conduct the experiments with MMClassition (Contributors, 2020) based on Pytorch (Paszke et al., 2019). While for CIFAR-100, we follow the training setting from DKD (Zhao et al., 2022).

4.2 Distillation with teachers

For the distillation with teachers, we first conduct experiments with various teacher-student distillation pairs on CIFAR-100, shown in Table 1. In this setting, we evaluate our method on various models with different architectures including VGG (Simonyan & Zisserman, 2014), ResNet (He et al.,

Table 1: Results of different distillation methods on the CIFAR-100 validation. The models in the first row and the second row are the teacher and student, respectively.

Method	Type	VGG13 VGG8	ResNet32x4 ResNet8x4	VGG13 MobileNetV2	ResNet50 MobileNetV2	ResNet32x4 ShuffleNetV1
baseline	-	70.36	72.50	64.60	64.60	70.50
RKD	Feature	71.48	71.90	64.52	64.43	72.28
CRD	Feature	73.94	75.51	69.73	69.11	75.11
OFD	Feature	73.95	74.95	69.48	69.04	75.98
KR	Feature	74.84	75.63	**70.37**	69.89	**77.45**
KD	Logit	72.98	73.33	67.37	67.35	74.07
WSLD	Logit	74.36	76.05	69.02	70.15	75.46
DKD	Logit	74.68	76.32	69.71	70.35	76.45
Ours	Logit	**74.86**	**76.35**	70.22	**70.67**	76.54

Table 2: Results of different distillation methods on ImageNet dataset. **T** and **S** indicate the teacher and student, respectively.

Type	Method	Top-1	Top-5	Method	Top-1	Top-5
	ResNet-34 (**T**)	73.62	91.59	ResNet-50 (**T**)	76.55	93.06
	ResNet-18 (**S**)	69.90	89.43	MobileNet (**S**)	69.21	89.02
Feature	AT	70.59	89.73	AT	70.72	90.03
	OFD	71.08	90.07	OFD	71.25	90.34
	RKD	71.34	90.37	RKD	71.32	90.62
	CRD	71.17	90.13	CRD	71.40	90.42
	KR	71.61	90.51	KR	72.56	91.00
	MGD	71.69	90.49	MGD	72.49	90.94
Logit	KD	71.03	90.05	KD	70.68	90.30
	WSLD	71.73	**90.53**	WSLD	72.02	90.70
	DKD	71.70	90.41	DKD	72.05	**91.05**
	Ours	**71.96**	90.48	**Ours**	**72.58**	90.96
Feature + Logit	SRRL	71.73	90.60	SRRL	72.49	90.92
	Ours+MGD	**72.01**	**90.84**	**Ours**+MGD	**73.10**	**91.32**

2016), MobileNetV2 (Sandler et al., 2018) and ShuffleNet (Zhang et al., 2018). We compare our method with the classical KD (Hinton et al., 2015) and several other state-of-the-art distillation methods for both heterogeneous and homogeneous distillation. As the results show, our method brings the students remarkable accuracy gains over other methods. For both heterogeneous and homogeneous distillation, our method achieves the best performance among logit-based distillation and even surpasses feature-based distillation in some distillation settings.

To further demonstrate the effectiveness and robustness of our method, we also test it on a more challenging dataset, ImageNet. We set two popular teacher-student pairs, which include homogeneous and heterogeneous distillation. The homogeneous distillation is ResNet34-ResNet18, and the heterogeneous distillation is ResNet50-MobileNet.

The results of different methods on ImageNet are shown in Table 2. As the results show, our method outperforms all the previous methods by just distilling on the logit. Our method brings consistent and significant improvements to the students for both distillation settings. The student ResNet18 and MobileNet achieve 71.96% and 72.58% Top-1 accuracy, getting 2.06% and 2.37% accuracy gains with the knowledge transferred from the teacher's logit, respectively. Furthermore, we try to combine our method with the SOTA feature-based distillation method MGD (Yang et al., 2022) to explore the upper bound for the distillation pairs. In this way, the student ResNet18 and MobileNet can achieve 72.01% and 73.10% Top-1 accuracy, getting another 0.05% and 0.52% accuracy gains, respectively.

Table 3: Results of different smooth methods. **Extra time cost** means whether needs a teacher, which needs the time for training a teacher first and inference the teacher during training.

Methods	Extra time cost	ResNet18	ResNet50
baseline	-	69.90	76.55
Label smooth	-	69.92 (+0.02)	76.64 (+0.09)
Tf-KD	-	70.14 (+0.24)	76.59 (+0.04)
Our tf-NKD	-	**70.76 (+0.86)**	**76.93 (+0.38)**

Table 4: Top-1 accuracy of different models with our tf-NKD on ImageNet dataset.

Model	Params (M)	Flops (G)	Baseline	+ tf-NKD
MobileNet	4.2	0.575	69.21	70.04 (+0.83)
MobileNetV2	3.5	0.319	71.86	72.08 (+0.22)
ShuffleNetV2	2.3	0.149	69.55	69.93 (+0.38)
ResNet-18	11.69	1.82	69.90	70.76 (+0.86)
ResNet-50	25.56	4.12	76.55	76.93 (+0.38)
ResNet-101	44.55	7.85	77.97	78.30 (+0.33)
Swin-Tiny	28.29	4.36	81.18	81.48 (+0.30)
Swin-Small	49.61	8.52	83.02	83.08 (+0.06)
Swin-Base	87.77	15.14	83.36	83.36

4.3 Training without teachers

Our tf-NKD is designed for training the students without teachers. The soft targets in tf-NKD are all obtained by smoothing the student's target output, so there is not any extra time cost compared with training the student directly. To evaluate the effectiveness of tf-NKD, we first compare it with two other methods that smooth the labels, including label smooth and Tf-KD (Yuan et al., 2020). As the results shown in Table3, tf-NKD brings a much larger improvement than the two methods. The label smooth and Tf-KD mainly introduce the knowledge about non-target distribution, while our tf-NKD is inspired by the KD method and brings the soft target knowledge.

To further evaluate the effectiveness and generalization of tf-NKD, we apply tf-NKD to various models with different architectures and sizes, including MobileNet (Howard et al., 2017), MobileNetV2 (Sandler et al., 2018), ShuffleNetV2 (Ma et al., 2018), ResNet (He et al., 2016) and SwinTransformer (Liu et al., 2021). As the results are shown in Table 4, the tf-NKD is beneficial to all different architectures, including lightweight models, CNN-based models and hybrid models. All the architectures can achieve Top-1 accuracy gains. Even for SwinTransformer-Tiny, it also brings 0.3% accuracy gains. Besides, tf-NKD improves even more for lightweight models. For example, it brings MobileNet and ResNet18 0.83% and 0.86% Top-1 accuracy gains, while the performance of Swin-Base keeps the same. The results show our tf-NKD is general and effective.

5 Analysis

5.1 Effects of soft loss and distributed loss for NKD

In this paper, we propose soft loss and distributed loss, which transfer the knowledge of the teacher's soft target and non-target distribution, respectively. In this subsection, we conduct experiments on soft loss and distributed loss to investigate their influences. As shown in Table 5, the soft loss and distributed loss lead to 1.16% and 1.48% accuracy improvements respectively, which shows both the knowledge of the soft target and non-target distribution is helpful to the student. Besides, the knowledge that soft loss and distributed loss transfer are independent. Combining them together allows us to make better use of the teacher's knowledge. In this way, the student achieves 71.96% Top-1 accuracy, which is significantly greater than using just using soft loss or distributed loss.

Table 5: Ablation study of distributed loss and soft loss.

Method	ResNet34 - ResNet18				
soft loss	-	-	✓	-	✓
distributed loss	-	-	-	✓	✓
classical KD	-	✓	-	-	-
Top-1 Acc	69.90	71.03	71.06	71.38	**71.96**
Top-5 Acc	89.43	90.05	89.51	90.47	**90.48**

Moreover, we modify classical KD and propose distributed loss for distillation. Here we also compare the distributed loss with classical KD. As it shows, distributed loss brings 1.48% gains while classical KD brings 1.03%. The comparison also validates the effectiveness of distributed loss.

5.2 Soft loss with different soft target

For the soft loss, we utilize the teacher's target output T_t as the soft target in Equation 6. In this subsection, we explore the effects of different soft targets T_t for the soft loss.

We conduct experiments by distilling ResNet50 on ImageNet, which is shown in Table 6. The performance of the teachers varies from MobileNetV2's 71.86% to ResNet50's 76.64%. We also design a perfect teacher, which has 100% accuracy and the T_t for each sample is 1. For the teacher trained without label smooth, a better teacher's label output probability is closer to gt-label. For example, the same sample's T_t of ResNet-50 is closer to 1 than that of ResNet-34. That is to say: strong teachers have high target output values, making the 'soft target' harder. Such harder 'soft target' bring the student fewer improvements. As the results show, the worst teacher MobileNetV2 brings 0.43% gains while ResNet50 just brings 0.33%. When we use the perfect teacher, which means $T_t = 1$, the soft loss even harms the student and causes a 0.1% Top-1 accuracy drop.

However, when the teacher is trained with label smooth, the teacher with high performance can also benefit the student a lot. Specifically, the accuracy of ResNet-50 with label smooth is 76.64%, which is 0.09% higher than ResNet-50's 76.55%. But it brings the student 0.54% Top-1 accuracy gains, while ResNet-50 just brings 0.33% Top-1 accuracy gains. We try to decompose the cross-entropy loss with label smooth to explore this phenomenon:

$$L_{ls} = -\sum_{i}^{C} V_i log(S_i) \tag{9}$$

$$= -\sum_{i}^{C} V_i log(\frac{S_i}{S_t}) - \sum_{i}^{C} V_i log(S_t). \tag{10}$$

Because $\sum_i^C V_i = 1$, $V_i = \alpha/C(i \neq t)$ and $V_t log(S_t/S_t) = 0$, L_{ls} can be simplified as:

$$L_{ls} = -\frac{\alpha}{C}\sum_{i\neq t}^{C} log(\frac{S_i}{S_t}) - log(S_t), \tag{11}$$

where C is the number of classes and α is a hyper-parameter. As the equation shows, the loss with label smooth can be regarded as the combination of the original cross-entropy loss and extra loss on the non-target output. The extra loss can prevent the model from over-fitting, which will decrease T_t and get a smoother output. Such models can get softer 'soft targets' and thus benefit the student more through the soft loss.

5.3 Different ways to smooth student's target output for tf-NKD

For tf-NKD, we replace the weights of NKD's soft loss by smoothing the student's target output S_t according to Equation 8. In this subsection, we explore the effects of different ways to smooth the student's target output S_t.

Table 6: Results of distilling ResNet-50 only by soft loss with different teachers on ImageNet dataset. The perfect teacher means the model has 100% accuracy and the target prediction T_t for each image is 1. **ls** means training the model with label smooth. **T** and **S** means the teacher and student.

Type	Extra time cost	Top-1 Acc (**T**)	Top-1 Acc (**S**)	Top-5 Acc (**S**)
baseline	-	-	76.55	93.06
$S_t + V_t - mean(S_t)$	-	-	76.93 (+0.38)	93.21
MobileNetV2	✓	71.86	76.98 (+0.43)	93.22
ResNet-34	✓	73.62	76.89 (+0.34)	93.22
ResNet-50	✓	76.55	76.88 (+0.33)	93.16
ResNet-50 (ls)	✓	76.64	77.09 (+0.54)	93.26
Perfect T	-	100.0	76.45 (-0.10)	93.06

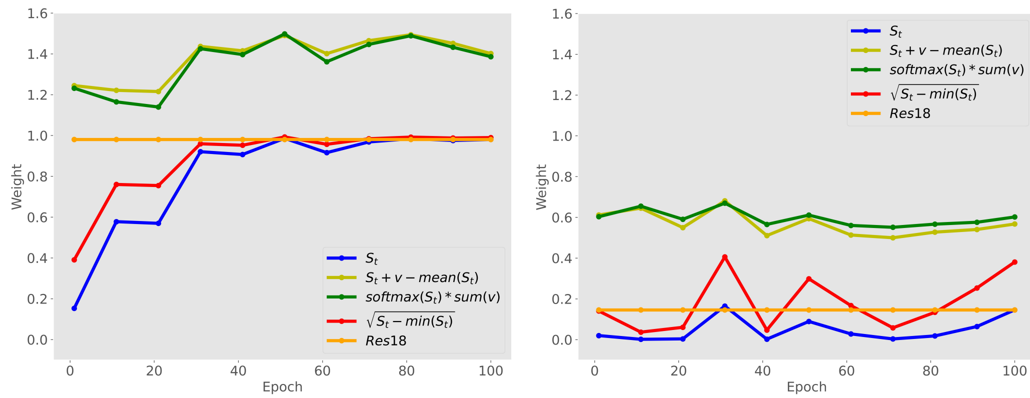
Table 7: Results of training ResNet18 directly with tf-NKD on ImageNet dataset. The weights for tf-NKD are obtained by smoothing the student's target prediction S_t in different ways. V_t means the gt-label value for the input sample. **mean**, **sum**, **max** and **softmax** are all calculated with different samples in a training batch.

smooth ways	Extra time cost	Top-1 Acc	Top-5 Acc
baseline	-	69.90	89.43
S_t	-	70.50 (+0.60)	89.38
$S_t + V_t - mean(S_t)$	-	**70.76 (+0.86)**	89.30
$softmax(S_t) * sum(v)$	-	70.57 (+0.67)	89.34
$\sqrt{S_t - min(S_t)}$	-	70.57 (+0.67)	89.13
$S_t/max(S_t)$	-	70.53 (+0.63)	89.39
$S_t/mean(S_t)$	-	70.50 (+0.60)	88.99
ResNet18	✓	**70.75 (+0.85)**	89.53

We need a fixed label for training, such as 1 for an image or (0.8,0.2) for a mixed-up image. However, the student's target output S_t gradually increases during training. Especially, as shown in Fig 2, for easy samples, the S_t is less than 0.2 at the beginning but larger than 0.9 at the last epochs. When the S_t is small, the student's output can not describe the sample's true category actually. In this case, the purpose of optimizing the model is to learn a very small label. We propose two key points for smoothing the student's target output S_t: **1) reflect the sample's true category in every epoch** and **2) prevent the soft target of every sample varying greatly in different epochs**.

Based on the two guidelines, we devise several different smooth methods. We conduct experiments by training ResNet18 on ImageNet to explore these smooth methods, which are shown in Table 7. As the results show, tf-NKD with all the methods can bring the student considerable improvements. Especially, the model gets 0.86% gains when using $S_t + V_t - mean(S_t)$ as the weights, which is even higher than using the teacher ResNet18's output as the soft target. We finally apply this way to smooth the student's target output for tf-NKD, which is shown in Equation 8. The results of applying tf-NKD to more models can be seen in Table 4.

Furthermore, we select two representative samples and visualize the soft targets calculated in different ways in different epochs. As Figure 2 shows, the student's target output S_t is very small even for the easy samples at the beginning and varies greatly from different epochs. This makes it hard to represent different samples' true categories accurately. However, as Figure 2 shows, the weights obtained by smoothing S_t according to Equation 8 **1) reflect the true category of different samples in every epoch** and **2) vary smoothly in different epochs**. In this way, we can utilize the weights to get a better model with the same training time as the baseline.



(a) easy sample

(b) hard sample

Figure 2: The curve of the weights of two representative samples during training. Different formulations mean adjusting the S_t in different ways.

Table 8: Results on the CIFAR-100 validation with different temperature. We use ResNet-34 as the teacher to distill the student ResNet-18.

baseline	λ	0.5	1.0	2.0	3.0	4.0	5.0
78.58	Top-1	80.42	80.55	**80.76**	80.72	80.54	80.50
94.10	Top-5	94.89	95.14	**95.14**	95.11	95.05	95.11

5.4 THE EFFECT OF THE TEMPERATURE

The temperature λ in Equation 7 is a hyper-parameter used to adjust the distribution of the teacher's logit. KD always applies $\lambda > 1$ to make the logit become more smooth, which causes the logit contains more non-target distribution knowledge. The target output probability of the same model will get a higher value on an easy dataset, such as CIFAR-100. This causes $\hat{T}_i^{\lambda}$ in Equation 7 contains less knowledge, which may bring adverse effect to the distillation. In this subsection, we explore the effects by using different temperatures to distill the student ResNet18 on CIFAR-100, which is shown in Table 8. The results show that temperature is an important hyper-parameter.

6 CONCLUSION

In this paper, we first analyze the relation between classical KD loss and original CE loss. From this point of view, we modify the formulation of KD and propose distributed loss to transfer the knowledge of non-target distribution. Besides, we propose soft loss, which regards the teacher's target output as the soft target for the student to learn. We propose New Knowledge Loss (NKD) which includes distributed and soft loss, helping students achieve state-of-the-art performance. Furthermore, we smooth student's target output as the soft target to train the model directly, which also brings the students considerable improvements without teachers or extra time costs.

LIMITATIONS

We try several attempts to smooth the student's target output for tf-NKD. However, the way to adjust the student's target output to the sample's soft target is still naive and left as future work. Moreover, we transfer the target knowledge from the student to the student successfully. We can consider whether it is available to transfer the non-target knowledge from the student to the student too. We believe it's meaningful work and worth exploring.

REFERENCES

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. *Advances in Neural Information Processing Systems*, 30, 2017.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5008–5017, 2021.

MMClassification Contributors. Openmmlab's image classification toolbox and benchmark. `https://github.com/open-mmlab/mmclassification`, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Jianyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2154–2164, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge adaptation for efficient semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 578–587, 2019.

Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930, 2019.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *Proceedings of the ieee Conference on Computer Vision and Pattern Recognition*, pp. 6356–6364, 2017.

Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2604–2613, 2019.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32, 2019.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5311–5320, 2021.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019.

Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4933–4942, 2019.

Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020.

Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. *arXiv preprint arXiv:2111.11837*, 2021.

Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. *arXiv preprint arXiv:2205.01529*, 2022.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911, 2020.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, 2018.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. *arXiv preprint arXiv:2203.08679*, 2022.

Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias–variance tradeoff perspective. In *International Conference on Learning Representations*, 2020.