From Knowledge Distillation to Self-Knowledge Distillation: A Unified Approach with Normalized Loss and Customized Soft Labels

Zhendong Yang[1,2*] Ailing Zeng[2] Zhe Li[3] Tianke Zhang[1] Chun Yuan[1†] Yu Li[2†]
[1]Tsinghua Shenzhen International Graduate School [2]International Digital Economy Academy (IDEA)
[3]Institute of Automation, Chinese Academy of Sciences
{yangzd21,ztk21}@mails.tsinghua.edu.cn axel.li@outlook.com
yuanc@sz.tsinghua.edu.cn {zengailing, liyu}@idea.edu.cn

Abstract

Knowledge Distillation (KD) uses the teacher's logits as soft labels to guide the student, while self-KD does not need a real teacher to require the soft labels. This work unifies the formulations of the two tasks by decomposing and reorganizing the generic KD loss into a Normalized KD (NKD) loss and customized soft labels for both target class (image's category) and non-target classes named Universal Self-KD (USKD). We decompose the KD loss and find the non-target loss from it forces the student's non-target logits to match the teacher's, but the sum of the two non-target logits is different, preventing them from being identical. NKD normalizes the non-target logits to equalize their sum. It can be generally used for KD and self-KD to better use the soft labels for distillation. USKD generates customized soft labels for both target and non-target classes without a teacher. It smooths the target logit of the student as the soft target label and uses the rank of the intermediate feature to generate the soft non-target labels with Zipf's law. For KD with teachers, NKD achieves state-of-the-art performance on CIFAR-100 and ImageNet, boosting the ImageNet Top-1 accuracy of Res-18 from 69.90% to 71.96% with a Res-34 teacher. For self-KD without teachers, USKD is the first method that can be effectively applied to both CNN and ViT models with negligible additional time and memory cost, resulting in new state-of-the-art results, such as 1.17% and 0.55% accuracy gains on ImageNet for MobileNet and DeiT-Tiny, respectively. Code is available at https://github.com/yzd-v/cls_KD.

1. Introduction

Deep convolutional neural networks (CNNs) have significantly advanced the performance in many tasks [8, 9, 27, 29]. In general, a larger model performing better needs more computing resources. On the other hand, smaller models have lower computation complexity but are less competitive than larger models. To bridge this gap and improve the performance of smaller models, knowledge distillation (KD) has been proposed [11]. The core idea of KD is to employ the teacher's prediction logits as soft labels to guide the student. Self-knowledge distillation (self-KD) [33, 47] is inspired by the knowledge distillation method, but it does not require an actual teacher. Instead, it designs soft labels through auxiliary branches or special distribution. The similarity between KD and self-KD is that they utilize soft labels for distillation loss, while the key difference is in how they obtain the soft labels. This paper aims to 1) improve the utilization of soft labels for distillation loss and 2) propose a general and effective method to obtain customized soft labels for self-KD. The targets make us obtain the soft labels with a teacher and use our modified distillation loss for better performance. Alternatively, when we lack a teacher, we can use the proposed self-KD method

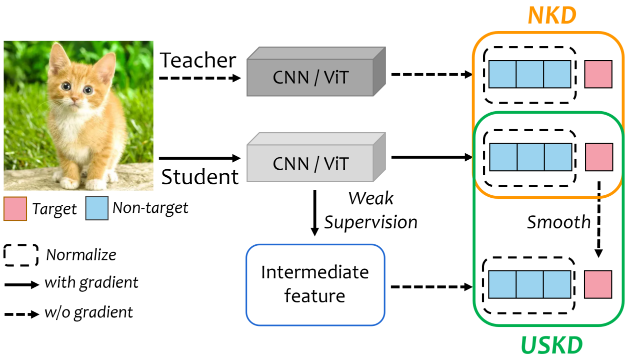

Figure 1. Illustration of the proposed NKD and USKD for distillation loss calculations. NKD normalizes the non-target logits, using the soft labels more effectively, and achieves better performance. Meanwhile, USKD sets customized soft labels for both target and non-target classes, and can be applied to both CNNs and ViTs.

[*] This work was done when Zhendong was an intern at IDEA.
[†] Corresponding authors.

to obtain the soft labels and then calculate the loss.

The original cross-entropy (CE) loss for classification calculates the loss on the target class (the image's category). While the soft labels from the teacher include target and non-target class, thus the KD loss also includes both target and non-target loss. The decoupled method has been proven effective for KD in DKD [49]. Unlike DKD's method of adjusting hyper-parameters on target and non-target loss, we present a simple yet effective way to decompose the KD loss. We decompose the KD loss into a combination of the target loss (like the original CE loss) and the non-target loss in CE form. The non-target loss transforms the internal distribution of the student's non-target logits to match the teacher's distribution. However, we find that the sum of the student's and teacher's non-target logits is changing and different, which hinders the alignment of their distributions. To address this issue, we normalize the non-target logits to equalize their sum, transferring teacher's non-target knowledge. With this slight modification, we introduce our Normalized Knowledge Distillation (NKD) loss, as depicted in Fig.1, significantly enhancing KD's performance.

Our proposed NKD utilizes the teacher's target logit and normalized non-target logits to guide the student, resulting in state-of-the-art performance. This demonstrates the effectiveness of NKD loss formulation. Also, it can be generally used for self-KD to calculate the distillation loss, but how to generate the soft labels without a real teacher generally and efficiently is also important.

Various self-KD methods have explored using manually designed soft labels to enhance students with less time than KD. These methods [14, 38, 44, 47] typically obtain the labels from auxiliary branches or contrastive learning, as depicted in Fig. 2 (a), (b), and (c). However, despite requiring less time than KD, they still involve significant overhead compared to training the model directly. Recently, state-of-the-art Zipf's LS [17], as shown in Fig. 2 (d), introduced soft non-target labels based on a special distribution that can significantly reduce resource and time requirements. It classifies the student's feature in the spatial dimension and determines the rank of the non-target class using Zipf's law [25]. However, it requires the pixel-level features before average pooling, making it unsuitable for ViT-like models [6] with patch-wise tokenization.

To address the limitations of existing methods, we propose a general and effective way to obtain soft labels. We design customized soft labels available for both CNN and ViT models. Following the NKD loss formulation, our customized soft labels comprise soft target label and soft non-target labels for corresponding loss. For the soft target label, we replace the teacher's target logit with the smoother label value obtained from the student's prediction. Since the student's predictions vary drastically during training, especially in the beginning, we smooth the student's target output within each training batch to stabilize the label values. For the soft non-target labels, we need their rank and distribution. First, for the rank, we get it from the intermediate feature, making our method available for both CNN and ViT models. We take weak supervision on the intermediate feature to get weak logit. Then, we normalize and combine it with the final logit and sort for the rank, as shown in Fig. 2 (f). The soft non-target labels' distribution follows Zipf's Law [25]. With the soft target and non-target labels, we set our customized soft labels and propose Universal Self-Knowledge Distillation (USKD) as shown in Fig. 1. Besides, USKD only needs an extra linear layer for weak supervision. So it just takes a few more computing resources and time than training the model directly. USKD is a simple and effective method that achieves state-of-the-art performance on both CNN and ViT models.

As described above, we normalize KD's non-target logits and propose NKD, using the soft labels better and improving KD's performance significantly. For the generation of soft labels without a real teacher, we set soft target and non-target labels, proposing USKD for self-KD. In a nutshell, the contributions of this paper are:

- We normalize the non-target logits in the classical KD, making it better to optimize the cross-entropy loss. With this minor change, we propose Normalized KD (NKD) loss, using teacher's soft labels better and improving KD's performance significantly.
- We propose a novel way to set customized soft labels without a real teacher, including target and non-target classes for self-KD. We utilize the weak logit to obtain soft non-target labels. Besides, we enlarge the difference between student's target logit for different images and soften them for soft target labels.
- We propose a simple and effective self-KD method USKD with our customized soft labels, which applies to both CNN and ViT models. Importantly, USKD requires only almost negligible additional time and resources compared to training the model directly.
- We conduct extensive experiments on CIFAR-100 and ImageNet to verify the effectiveness of NKD and USKD, achieving state-of-the-art performance. Additionally, we demonstrate the efficacy of models trained with our self-KD method on COCO for detection.

2. Related work

2.1. Knowledge Distillation

Knowledge distillation is a method to improve the model while keeping the network unchanged. It was first proposed by Hinton *et al.* [11], where the student is supervised by the hard and soft labels from the teacher's output. Many

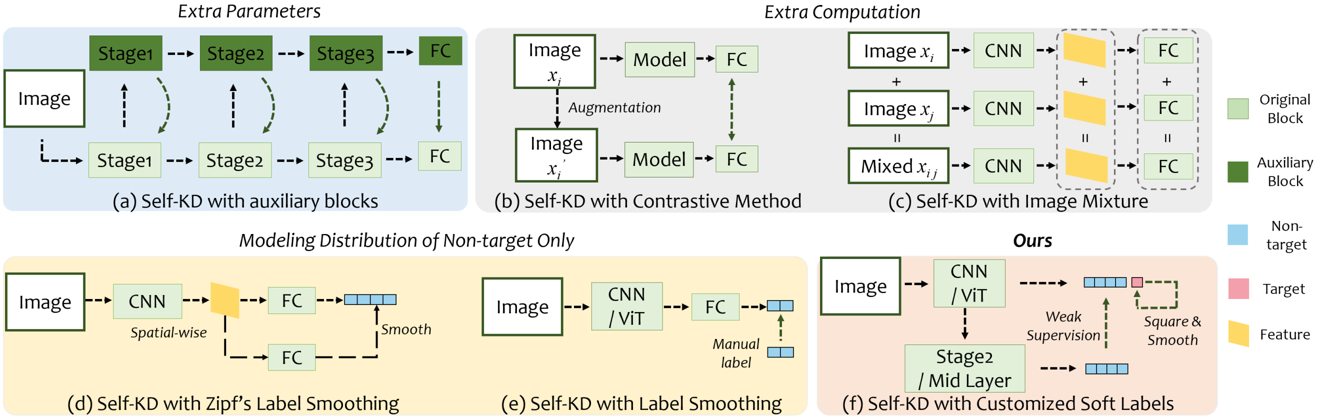

Figure 2. Comparison of different self-KD methods with **(a)** *extra blocks and parameters* [14], **(b)** *contrastive method* [37], **(c)** *image mixture* [38], **(d)** *Zipf's label smoothing* [17], and **(e)** *label smoothing* [22, 34]. **(f)** Self-KD with our *customized soft labels* including both target and non-target class for distillation. Note some methods can only be applied to CNN, while ours works for both CNN and ViT.

following works focus on better using soft labels to transfer more knowledge. WSLD [50] analyzes soft labels and distributes different weights for them from a perspective of bias-variance trade-off. SRRL [39] forces the output logits of the teacher's and student's features after the teacher's linear layer to be the same. DKD [49] decouples the logit and distributes different weights for the target and non-target classes. DIST [13] uses the Pearson correlation coefficient to replace the KL divergence and transfers the inter-relation and intra-relation together.

Besides distillation on logits, some works [1, 31, 40, 42] aim at transferring knowledge from intermediate features. FitNet [28] distill the semantic information from the intermediate feature directly. AT [45] transfers the attention of feature maps to the students. OFD [10] designs the margin ReLU and modifies the measurement for the distance between students and teachers. RKD [23] extracts the relation from the feature map. CRD [35] applies contrastive learning to distillation successfully. KR [3] transfers knowledge from multi-level features for distillation. TaT [18] helps the student to learn the teacher's every spatial component. MGD [41] masks the student's feature and forces it to generate the teacher's feature.

2.2. Self-Knowledge Distillation

Self-Knowledge Distillation has been proposed as an alternative approach to Knowledge Distillation that does not rely on an external teacher model. Self-KD aims to utilize the information within the student model to guide its learning process. Several self-KD methods have been proposed in recent years. DKS [33] introduces auxiliary supervision branches and pairwise knowledge alignments, while BYOT [47] adds blocks and layers to every stage and uses shallow and deep features as student and teacher, respectively. KDCL [7] trains two models for online knowledge distillation, while FRSKD [14] adds a new branch supervised by the original feature and uses the logit of the new branch for self-KD. DDGSD [37] transfers knowledge between different distorted versions of the same training data. OLS [46] sets a matrix that is made up of the soft label for every class, while MixSKD [38] proposes incorporating self-knowledge distillation with image mixture and aggregates multi-stage feature to produce soft labels. Tf-FD [16] includes intra-layer and inter-layer distillation, reusing the channel-wise and layer-wise features to provide knowledge without an additional model. However, these methods require auxiliary architecture, adapt layers for alignment, or data augmentation, consuming much more time and computing resources than training the model directly.

In contrast, some self-KD methods require little extra or even no more time than training the model directly. For example, Label Smoothing [34] sets the labels manually by distributing the same values to all non-target classes. Tf-KD [43] revisits KD via label smoothing, using a high temperature to generate the manual logit for distillation. Zipf's LS [17] utilizes the student's linear layer to obtain several logits for each pixel of the student's last feature map. The method uses these logits to vote for the non-target class's rank with Zipf distribution [25]. These methods set soft labels to achieve self-knowledge distillation without contrastive learning, data augmentation, or auxiliary branches, saving much training time and computing resources.

3. Method

3.1. Normalized KD for Better Using Soft Labels

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

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

Using T_i denote the teacher's soft labels. The classical KD utilizes them for distillation loss as:

$$L_{kd} = -\sum_{i}^{C} T_i log(S_i) \quad (2)$$

$$= -T_t log(S_t) - \sum_{i \neq t}^{C} T_i log(S_i). \quad (3)$$

As L_{kd} shows, the first loss $-T_t log(S_t)$ is about the target class like the original L_{ori}. While for the second non-target loss $-\sum_{i \neq t}^{C} T_i log(S_i)$, it has the same form as CE loss $-\sum p(x) log(q(x))$. The CE loss aims at making $q(x)$ be the same as $p(x)$. However, $\sum_{i \neq t}^{C} T_i = 1 - T_t$ and $\sum_{i \neq t}^{C} S_i = 1 - S_t$. Since the trainable S_i is unlikely to exactly match the fixed T_i during training, the sum of the two non-target logits is always different, preventing the two distributions from being the same. Thus we normalize T_i and S_i to force the sum of the two distributions to be the same. With $\mathcal{N}(\cdot)$ denoting the normalized operation, we modify the KD loss and propose our **N**ormalized **K**nowledge **D**istillation (NKD) loss as follows:

$$L_{nkd} = -T_t log(S_t) - \gamma \cdot \lambda^2 \cdot \sum_{i \neq t}^{C} \mathcal{N}(T_i^{\lambda}) log(\mathcal{N}(S_i^{\lambda})), \quad (4)$$

where γ is a hyper-parameter to balance the loss and λ is the temperature for KD [11]. Finally, combining the original loss L_{ori}, and NKD loss L_{nkd}, we train the students with:

$$L_{all} = L_{ori} + L_{nkd}. \quad (5)$$

3.2. Customized Soft Labels for Self-KD

We utilize NKD for better use of the soft labels. Following our NKD loss in Eq. 4, we propose a general and effective self-KD method, which sets soft labels on target and non-target classes without teachers.

3.2.1 Soft Target Label

First, for the soft target label, the weight T_t in Eq. 4 is obtained from the teacher's 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 differences between T_t and S_t mainly focus on two parts. The first is that T_t is fixed, and S_t varies gradually during training. The second part is that the difference between different samples' S_t is much smaller than T_t at the beginning of training. To overcome the problem, we first square S_t to enlarge the difference between different samples' S_t. Then we propose a way to adjust S_t, making it smoother to fit the training set without teachers. This strategy can be applied directly to different models, including CNN-liked and ViT-liked models. In this way, we get the soft target label P_t, which is detached and zero-grad in training for self-KD:

$$P_t = S_t^2 + V_t - mean(S_t^2). \quad (6)$$

With the P_t, we follow our NKD for the target loss:

$$L_{target} = -P_t log(S_t), \quad (7)$$

where V_t denote the original target label value, *e.g.* [0.8,0.2] for a mixed image. And $mean(\cdot)$ is calculated across different samples in a training batch. We discuss the effects of the smoothing ways for S_t in Sec. 5.6.

3.2.2 Soft Non-target Labels

The knowledge from the teacher's soft non-target labels includes its rank and distribution. We also need them to set the soft non-target labels for self-KD. To get the rank, we first obtain a new weak logit by setting weak supervision on the intermediate feature. Using $\mathcal{F}$ denote the feature of stage 2 of the CNN-liked model or the mid layer's classification token of the ViT-liked model, GAP denote the global average pooling, FC denote a new linear layer, the weak logit of the CNN-liked model can be formulated as:

$$W_i = softmax(FC(GAP(\mathcal{F}))), \quad (8)$$

As for the ViT-liked model, the weak logit is as follows:

$$W_i = softmax(FC(\mathcal{F})).$$

We aim to obtain another smoother non-target logit different from the original final logit. So we utilize a smooth label and take weak supervision to get the weak logit. Using V_i denote the label, which is the original label processed with label smoothing, the loss for obtaining the weak logit is as follows:

$$L_{weak} = -\mu \cdot \sum_{i}^{C} V_i log(W_i), \quad (9)$$

where $\mu < 1$ is a hyper-parameter to achieve weak supervision. With the weak logit, we obtain the soft non-target labels' rank by normalizing and combining it with the final logit. This operation balances the two logits' effects for the rank, and the analysis is shown in Sec. 5.7.

$$R_i = \frac{W_i}{1 - W_t} + \frac{S_i}{1 - S_t}. \quad (10)$$

The soft non-target labels' distribution follows Zipf's law [25], which has been applied in Zipf's LS [17]. The formulation can be formulated as follows:

$$Z_i = \frac{i^{-1}}{\sum_{i=1}^{C} i^{-1}}. \tag{11}$$

With the distribution, we sort it with the rank of R_i and obtain the soft non-target labels Z_i for self-KD. Following our NKD loss in Eq. 4, the non-target loss is as follows:

$$L_{non} = -\sum_{i \neq t}^{C} \mathcal{N}(Z_i) log(\mathcal{N}(S_i)). \tag{12}$$

3.2.3 Overall for Self-KD

With the proposed soft target label and soft non-target labels, we calculate the corresponding loss and propose USKD as shown in Fig. 1. We train all the models with the total loss for self-KD as follows:

$$L_{all} = L_{ori} + \alpha \cdot L_{target} + \beta \cdot L_{non} + L_{weak}, \tag{13}$$

where L_{ori} is the original loss for the models among all the tasks, α and β are two hyper-parameters to balance the loss scale. L_{weak} is used to generate the weak logit in Eq. 8.

4. Experiments

4.1. Datasets and Details

We conduct the experiments on CIFAR-100 [15] and ImageNet [5], which contain 100 and 1000 categories, respectively. For CIFAR-100, we use 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 KD with teachers, NKD has two hyper-parameters γ and λ in Eq. 4. For all the experiments, we adopt $\{\gamma = 1.5, \lambda = 1\}$ on ImageNet. While for CIFAR-100, we follow the training setting from DKD [49] for a fair comparison. And USKD has two hyper-parameters α and β to balance the loss scale in Eq. 13. Another hyper-parameter μ is used to achieve the weak supervision in Eq. 9. For all the experiments, we adopt $\{\alpha = 1, \beta = 0.1, \mu = 0.005\}$ on ImageNet and $\{\alpha = 0.1, \beta = 0.1, \mu = 0.1\}$ on CIFAR-100. The other training setting for KD and self-KD is the same as training the students without distillation. We use 8 GPUs to conduct the experiments with MMClassition [4] based on Pytorch [24]. More details and experimental results about the hyper-parameters are shown in the supplement.

4.2. Normalized KD with Teachers

When we get the soft labels from a real teacher, we can use NKD loss for better performance. To prove this, we first conduct experiments with various teacher-student distillation pairs on CIFAR-100, shown in Tab. 1. In this setting, we evaluate our method on several models with different architectures, including VGGNet [32], ResNet [9], ShuffleNet [48] and MobileNetV2 [30]. We compare our method with KD [11] and several other state-of-the-art distillation methods. As the results show, our method brings the students remarkable accuracy gains over other methods. Our method achieves the best performance among logit-based distillation methods and even surpasses the feature-based distillation methods in some settings.

To further demonstrate the effectiveness and robustness of our NKD, we test it on a more challenging dataset, ImageNet. We set two popular teacher-student pairs, which include homogeneous and heterogeneous teacher-student structures for distillation. The homogeneous distillation is ResNet34-ResNet18, and the heterogeneous distillation is ResNet50-MobileNet. The results of different KD methods on ImageNet are shown in Tab. 1. As the results show, our method outperforms all the previous methods. It 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 3.37% accuracy gains with the knowledge transferred from the teacher's logits, respectively.

As described above, our NKD enhances KD's performance significantly with a slight modification. And in various settings, it also surpasses DKD, a method that improves KD according to a different decoupled way.

4.3. Universal Self-KD without Teachers

When we lack a teacher, we use the proposed self-KD method USKD to obtain the soft labels and then calculate the loss. To evaluate its effectiveness, we first conduct experiments with ResNets [9] and MobileNet [12] on CIFAR100 and ImageNet datasets. We compare with the other state-of-the-art methods, which also set manual labels and only bring little extra time consumption, including label smoothing [34], Tf-KD [43] and Zipf's LS [17]. As shown in Tab. 2, our method surpasses the previous related self-knowledge distillation methods on various settings and brings the model remarkable gains. For example, it brings MobileNet and ResNet-18 1.17% and 0.89% Top-1 accuracy gains on ImageNet.

Furthermore, we also test our USKD on more models, which include lighter models MobileNetV2 [30], ShuffleNetV2 [21] and a deeper ResNet. The results are shown in Tab. 3. For both the lightweight models, including MobileNetV2 and ShuffleNetV2, and the stronger model ResNet-101, our method also achieves considerable improvements. Besides, we also compare the time consumption to train the model for an epoch in Tab. 3. Compared with the baseline, the extra time we need for self-knowledge

Model	Teacher	VGGNet13	ResNet32x4	VGGNet13	ResNet50	ResNet34	ResNet50
	Student	VGGNet8	ResNet8x4	MobileNetV2	MobileNetV2	ResNet18	MobileNet
Accuracy	Teacher	74.64	79.42	74.64	79.34	73.62	76.55
	Student	*70.36*	*72.50*	*64.60*	*64.60*	*69.90*	*69.21*
Feature	RKD [23]	71.48	71.90	64.52	64.43	71.34	71.32
	CRD [35]	73.94	75.51	69.73	69.11	71.17	71.40
	OFD [10]	73.95	74.95	69.48	69.04	71.08	71.25
	KR [3]	74.84	75.63	**70.37**	69.89	71.61	72.56
Logit	KD [11]	72.98	73.33	67.37	67.35	71.03	70.68
	WSLD [50]	74.36	76.05	69.02	70.15	71.73	72.02
	DKD [49]	74.68	76.32	69.71	70.35	71.70	72.05
	NKD (ours)	**74.86**	**76.35**	70.22	**70.67**	**71.96**	**72.58**

Table 1. Results of different knowledge distillation methods on CIFAR-100 (the left four columns) and ImageNet (the right two columns) dataset. The data that is underlined denotes the second-best result among all the results. The metric is the Top-1 accuracy (%).

Dataset	CIFAR100	ImageNet		
Model	ResNet18	ResNet18	ResNet50	MobileNet
Baseline	*78.58*	*69.90*	*76.55*	*69.21*
LS [34]	79.42	69.92	76.64	68.98
Tf-KD [43]	79.53	70.14	76.59	69.20
Zipf's LS [17]	79.63	70.30	76.96	69.59
USKD (ours)	**79.90**	**70.79**	**77.07**	**70.38**

Table 2. The comparative results of different self-knowledge distillation methods on CIFAR100 and ImageNet dataset. We report the models' performance with Top-1 accuracy (%).

	Model	*Baseline*	**USKD (ours)**
Accuracy (%)	ResNet-101	*77.97*	78.54 (+0.57)
	MobileNet-V2	*71.86*	72.41 (+0.55)
	ShuffleNet-V2	*69.55*	70.30 (+0.75)
Time (min)	ResNet-101	*13.78*	13.95
	MobileNet-V2	*10.17*	10.18
	ShuffleNet-V2	*8.63*	8.68

Table 3. The results of the models' performance are Top-1 accuracy (%) on ImageNet dataset. The time data are reported with minutes (min) for a training epoch.

Model	*Baseline*	**USKD (ours)**
RegNetX-1.6GF	*76.84*	77.30 (+0.46)
DeiT-Tiny	*74.42*	74.97 (+0.55)
DeiT-Small	*80.55*	80.77 (+0.22)
Swin-Tiny	*81.18*	81.49 (+0.31)

Table 4. Results of training more models, including ViT-liked models with our proposed method on ImageNet dataset. All the results are reported with Top-1 accuracy (%).

distillation is very limited. Specifically, the time of training MobileNet-V2 with our proposed USKD for an epoch is 10.18 minutes, which is just 0.01 minutes higher than training the model directly. Our method brings considerable improvements to various models with negligible extra time consumption.

4.4. Universal Self-KD for More Models

The previous self-KD methods are specially designed for CNN-liked models. However, some models like ViT [6] translate the image into different tokens and have a completely different architecture. Those self-KD methods fail to benefit ViT-liked models. For ViT-liked models, USKD can also set customized soft labels and bring remarkable improvements. As described in Eq. 8, we use an extra linear layer connected to ViT's middle layer for classification and obtain the weak logit. The rest operations are the same as CNN-liked models. To show the generalization of USKD, we apply it to more models, including RegNet [26], DeiT [36], and Swin-Transformer [20], which can be seen in Tab. 4. All the models can achieve remarkable Top-1 accuracy gains. Besides, our method even can bring 0.55% gains for DeiT-Tiny. And it also brings 0.31% Top-1 accuracy gains for the latest state-of-the-art model Swin-Transformer. The results of more models show our method is both effective and general.

5. Analysis

5.1. Effects of Normalizing the Non-target Logits

In this paper, we proposed normalizing the non-target logits in the original KD to help the student perform better. In this subsection, we conduct experiments to demonstrate the effectiveness of our modification. As shown in Tab. 5, using the target loss alone leads to a 1.16% increase in accuracy. Combining the knowledge from both target and non-target classes allows us to better use the teacher's knowledge, resulting in a significant improvement of 2.06% Top-1 accuracy for the student. Furthermore, we normalize the non-target logits in the KD loss for distillation and compare

Loss	ResNet34 - ResNet18			
Target	✗	✓	✓	✓
KD's Non-target	✗	✗	✓	✗
NKD's Non-target	✗	✗	✗	✓
Top-1 Acc. (%)	*69.90*	71.06	71.33	**71.96**
Top-5 Acc. (%)	*89.43*	89.51	90.25	**90.48**

Table 5. Effects of our normalized non-target loss on ImageNet dataset. The teacher is ResNet34 and the student is ResNet18.

the non-target loss from both KD and NKD. Our non-target loss brings much greater gains than KD's non-target loss, as shown in the comparison. These results demonstrate the effectiveness of our proposed modification in improving KD's performance.

5.2. Difference between Our NKD and DKD

To better use the soft labels, we decompose KD loss and normalize the non-target logits for a better performance. The decoupled method is inspired by DKD [49]. However, this paper presented a more straightforward and efficient decomposition method. DKD decouples KD loss as:

$$L_{kd} = -T_t log(S_t) - (1 - T_t) log(1 - S_t) \quad (14)$$

$$- (1 - T_t) \sum_{i \neq t}^{C} \hat{T}_i log(\hat{S}_i) \quad (15)$$

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

DKD analyzed the effects of KD's components and set hyper-parameters for a new formulation:

$$L_{dkd} = \alpha \cdot \big(-T_t log(S_t) - (1 - T_t) log(1 - S_t) \big) \quad (16)$$

$$- \beta \cdot \big(\sum_{i \neq t}^{C} \hat{T}_i log(\hat{S}_i) \big) \quad (17)$$

While we decompose KD loss into the target loss (like the original CE loss) and non-target loss in CE form:

$$L_{kd} = -T_t log(S_t) - \sum_{i \neq t}^{C} T_i log(S_i) \quad (18)$$

Then we find the sum of student's and teacher's non-target logits is different, making them hard to be the same. So we normalize them to equalize the sum as follows:

$$L_{nkd} = -T_t log(S_t) - \gamma \cdot \sum_{i \neq t}^{C} \mathcal{N}(T_i) log(\mathcal{N}(S_i)) \quad (19)$$

With this slight modification, we present our NKD loss and achieve better performance than DKD, as shown in Tab. 1.

Loss	Top-1 Accuracy (%)			
Target	✗	✓	✗	✓
Non-target	✗	✗	✓	✓
MobileNet	*69.21*	70.18	69.43	**70.38**
RegNetX-1.6GF	*76.84*	76.87	77.25	**77.30**

Table 6. Ablation study of USKD's target and non-target loss. The experiments are conducted on the ImageNet dataset. All the results are the Top-1 accuracy (%).

Method	ImageNet	COCO	
	Top-1 Acc. (%)	AP^{box}	AR^{box}
Baseline (Res50)	*76.55*	*38.0*	*52.4*
USKD (ours)	77.07	38.3	52.9

Table 7. The detection results on the COCO dataset. We pre-train the backbone with USKD and use Mask-RCNN as the detector.

5.3. Effects of USKD's Target and Non-target Loss

We propose a novel self-KD method called USKD, which utilizes customized soft labels that incorporate information from both target and non-target classes. To evaluate the impact of each type of information, we conduct experiments on MobileNet and RegNetX-1.6GF in Tab. 6. The experimental findings demonstrate that both types of information are beneficial and important for the two models, and their combination leads to further improvements in performance. For instance, by combining the target and non-target class distillation together, MobileNet achieves 70.38%, which surpasses the accuracy achieved by using distillation on either the target or non-target class alone.

5.4. Models with USKD for Downstream Task

Our self-KD method yields remarkable accuracy gains for the classification task on CIFAR-100 and ImageNet datasets. To further evaluate its effectiveness and generalization, we also apply the pre-trained model to object detection using Mask R-CNN [8] as the detector and evaluate the model's performance with AP^{box} and AR^{box} on the COCO val2017 dataset [19]. We conduct the detection experiments for 12 epochs using MMDetection [2]. As shown in Tab. 7, the ResNet-50 backbone trained with our method improves the detector's performance by 0.3 mAP and 0.5 mAR. The results demonstrate that our self-KD method not only improves the model's classification performance but also generalizes well to downstream tasks like object detection.

5.5. Customized Soft Labels for USKD

Our self-KD method, USKD, leverages customized soft labels for every image during training. Fig.3 shows several samples' soft labels, including the value for the target class and the top-3 non-target classes. For the target class, the value may be larger than 1 due to the smoothing method

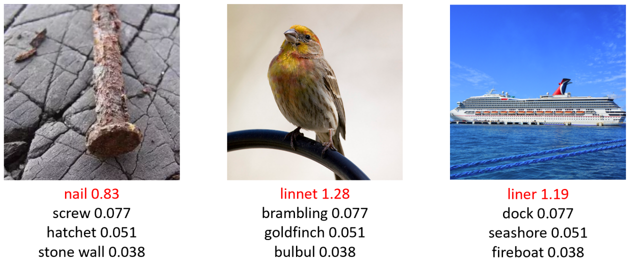

Figure 3. The visualization of the target and top-3 non-target class values of our customized soft labels. The target class value is obtained by squaring and smoothing. The non-target class value is obtained by Zipf's law.

described in Eq. 6. For the non-target classes, USKD distributes larger values to the categories that are similar to the target class. For instance, the labels for the second image not only include the target 'linnet,' but also assign higher values to similar non-target classes like 'brambling', 'goldfinch', and 'bulbul'. This approach ensures that each image receives an appropriate customized soft label, which enables successful self-KD. The visualization also demonstrates how our USKD helps the model to perform better.

5.6. Different Smooth Ways for Soft Target Label

Our USKD utilizes customized soft labels that include target and non-target classes for all the images during training. To create the soft labels, we replace the weights of NKD's target loss with a smoothed version of the student's target output S_t, as shown in Eq. 6. In this subsection, we investigate the impact of different methods for smoothing the student's target output S_t on the performance of the model. We conduct experiments on ResNet18 trained on ImageNet to compare these methods, as shown in Tab. 8. The experiments show that all the methods significantly improve the student model. Notably, using $S_t+V_t-mean(S_t)$ as the weights for smoothing the student's target output achieves 0.86% improvement, even outperforming using the teacher ResNet18's output as the soft target label. Based on this observation, we choose this method for smoothing the student's target output and use it as our soft target label.

5.7. Different Ranks for Soft Non-target Labels

In Eq. 10, we combine the normalized weak and final logit to obtain the soft non-target labels' rank. In this subsection, we explore the effects of different ways for the rank, as shown in Tab. 9. Specifically, we compare the performance when using the weak logit's rank alone, the final logit's rank alone, or a combination of both but without normalization. Interestingly, our results show that all these methods help improve the model's accuracy, with the combination of the two normalized logits achieving the best performance. It is also noteworthy that even directly using the final logit to obtain the rank can satisfy accuracy gains. These findings provide insights into the effectiveness of various approaches for the rank of the non-target labels.

Smooth Ways	Teacher	Top-1 Acc. (%)
Baseline	-	*69.90*
$S_t + V_t - mean(S_t)$	✗	**70.76**
$softmax(S_t) * sum(V_t)$	✗	70.57
$\sqrt{S_t - min(S_t)}$	✗	70.57
$S_t/max(S_t)$	✗	70.53
$S_t/mean(S_t)$	✗	70.50
Trained ResNet-18	✓	**70.75**

Table 8. Results of training ResNet18 with target loss on ImageNet dataset. All the operations are calculated with different samples in a training batch. Because the trained Res-18's output is S_t, we drop the square operation for different smoothing ways.

Model	ResNet-18	RegNetX-1.6GF
Baseline	*69.90*	*76.84*
Weak Logit	70.65	77.28
Final Logit	70.72	77.15
Combination	70.71	77.25
Normalization	70.79	77.30

Table 9. Results of different ranks for soft non-target labels. **Normalization** and **Combination** mean combining weak and final logit with normalization and without normalization, respectively.

6. Conclusion

In this paper, we decompose KD loss into the target loss like original CE loss and non-target loss in a CE form. We then normalize the non-target logits to enhance the student's learning from the teacher. With this normalization, we introduce Normalized KD (NKD), which helps students to achieve state-of-the-art performance. Building on our NKD loss formulation, we further propose a new self-KD method, USKD, that works for both CNN-like and ViT-like models. USKD uses customized soft labels that include target and non-target classes for self-KD. We first square and smooth the student's target output logit as the soft target label. For the soft non-target labels, we use weak supervision to obtain the rank and utilize Zipf's law to generate the labels. In this way, USKD needs negligible extra time and resources than training the model directly. Extensive experiments on various models with different datasets demonstrate that both our NKD and USKD are simple and efficient.

Acknowledgement. This work was supported by the National Key R&D Program of China (2022YFB4701400/4701402), the SZSTC project Grant (JCYJ20190809172201639, WDZC20200820200655001), Shenzhen Key Laboratory (ZDSYS20210623092001004).

References

[1] Weihan Cao, Yifan Zhang, Jianfei Gao, Anda Cheng, Ke Cheng, and Jian Cheng. Pkd: General distillation framework for object detectors via pearson correlation coefficient. *Advances in Neural Information Processing Systems*, 2022. 3

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 7

[3] 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*, pages 5008–5017, 2021. 3, 6

[4] MMClassification Contributors. Openmmlab's image classification toolbox and benchmark. https://github.com/open-mmlab/mmclassification, 2020. 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2, 6

[7] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020. 3

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 1, 7

[9] 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*, pages 770–778, 2016. 1, 5

[10] 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*, pages 1921–1930, 2019. 3, 6

[11] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1, 2, 4, 5, 6

[12] 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. 5

[13] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. *Advances in Neural Information Processing Systems*, 2022. 3

[14] Mingi Ji, Seungjae Shin, Seunghyun Hwang, Gibeom Park, and Il-Chul Moon. Refine myself by teaching myself: Feature refinement via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10664–10673, 2021. 2, 3

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[16] Lujun Li. Self-regulated feature learning via teacher-free feature distillation. In *European Conference on Computer Vision*, pages 347–363, 2022. 3

[17] Jiajun Liang, Linze Li, Zhaodong Bing, Borui Zhao, Yao Tang, Bo Lin, and Haoqiang Fan. Efficient one pass self-distillation with zipf's label smoothing. In *European Conference on Computer Vision*, pages 104–119, 2022. 2, 3, 5, 6

[18] Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge distillation via the target-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10915–10924, 2022. 3

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. 7

[20] 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*, pages 10012–10022, 2021. 6

[21] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *European conference on computer vision*, pages 116–131, 2018. 5

[22] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in Neural Information Processing Systems*, 32, 2019. 3

[23] 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*, pages 3967–3976, 2019. 3, 6

[24] 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. 5

[25] David M. W. Powers. Applications and explanations of zipf's law. In *Conference on Computational Natural Language Learning*, 1998. 2, 3, 5

[26] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design

spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10428–10436, 2020. 6

[27] 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. 1

[28] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015. 3

[29] 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*, pages 234–241, 2015. 1

[30] 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*, pages 4510–4520, 2018. 5

[31] 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*, pages 5311–5320, 2021. 3

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[33] Dawei Sun, Anbang Yao, Aojun Zhou, and Hao Zhao. Deeply-supervised knowledge synergy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6997–7006, 2019. 1, 3

[34] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016. 3, 5, 6

[35] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019. 3, 6

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021. 6

[37] Ting-Bing Xu and Cheng-Lin Liu. Data-distortion guided self-distillation for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5565–5572, 2019. 3

[38] Chuanguang Yang, Zhulin An, Helong Zhou, Linhang Cai, Xiang Zhi, Jiwen Wu, Yongjun Xu, and Qian Zhang. Mixskd: Self-knowledge distillation from mixup for image recognition. In *European Conference on Computer Vision*, pages 534–551, 2022. 2, 3

[39] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020. 3

[40] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4643–4652, 2022. 3

[41] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. In *European Conference on Computer Vision*, pages 53–69, 2022. 3

[42] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Practical guidelines for vit feature knowledge distillation. *arXiv preprint arXiv:2209.02432*, 2022. 3

[43] 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*, pages 3903–3911, 2020. 3, 5, 6

[44] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13876–13885, 2020. 2

[45] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3

[46] Chang-Bin Zhang, Peng-Tao Jiang, Qibin Hou, Yunchao Wei, Qi Han, Zhen Li, and Ming-Ming Cheng. Delving deep into label smoothing. *IEEE Transactions on Image Processing*, 30:5984–5996, 2021. 3

[47] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722, 2019. 1, 2, 3

[48] 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*, pages 6848–6856, 2018. 5

[49] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11953–11962, 2022. 2, 3, 5, 6, 7

[50] 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. 3, 6