Supplementary Material - Logit Standardization in Knowledge Distillation

Shangquan Sun[1,2], Wenqi Ren[3†], Jingzhi Li[1], Rui Wang[1,2], Xiaochun Cao[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]School of Cyber Science and Technology, Shenzhen Campus of Sun Yat-sen University
{sunshangquan,lijingzhi,wangrui}@iie.ac.cn, {renwq3,caoxiaochun}@mail.sysu.edu.cn

Overview

In Sec. 1, we first provide detailed proves regarding the properties of the standardized logit by the proposed $\mathcal{Z}$-score function. We then elaborate the implementation of our proposed method on CTKD, DKD and MLKD in detail in Sec. 2. Finally, we show more experimental results and analyses in Sec. 3.

1. Proof

In this section, we prove three properties of $\mathcal{Z}(\mathbf{z}_n;\tau)$ mentioned in Section 4.3 of the manuscript, i.e., zero mean, finite standard deviation, and boundedness. Its forth property of monotonicity is trivial by considering the fact that $\mathcal{Z}$-score function is a kind of linear transformation function.

1.1. Proof of Zero Mean Porperty

We want to prove the "Zero mean" property of $\mathcal{Z}(\mathbf{z}_n;\tau)$ in Section 4.3 of the manuscript, i.e.,

$$\frac{1}{K}\sum_{k=1}^{K}\mathcal{Z}(\mathbf{z}_n;\tau)^{(k)}=0$$

Proof. We can easily obtain

$$\begin{aligned}\frac{1}{K}\sum_{k=1}^{K}\mathcal{Z}(\mathbf{z}_n;\tau)^{(k)}&=\frac{1}{K}\sum_{k=1}^{K}\frac{\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n}{\sigma(\mathbf{z}_n)\tau}\\&=\frac{1}{K\sigma(\mathbf{z}_n)\tau}\sum_{k=1}^{K}\left(\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n\right)\\&=0,\end{aligned}$$

since the vector mean $\overline{\mathbf{z}}_n=\frac{1}{K}\sum_{k=1}^{K}\mathbf{z}_n^{(k)}$. The same proof holds for $\mathbf{v}_n$ and thus we omit it. □

1.2. Proof of Finite Standard Deviation

We mention the "finite standard deviation" property of standardized logit in Section 4.3 of the manuscript. Here, we want to prove $\mathcal{Z}(\mathbf{z}_n;\tau)$ has the standard deviation $\frac{1}{\tau}$, i.e.,

$$\sigma(\mathcal{Z}(\mathbf{z}_n;\tau))=\frac{1}{\tau}$$

Proof. We have proved the "zero mean" of $\mathcal{Z}(\mathbf{z}_n;\tau)$, so we have

$$\begin{aligned}&\sigma(\mathcal{Z}(\mathbf{z}_n;\tau))\\=&\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\mathcal{Z}(\mathbf{z}_n;\tau)^{(k)}-\frac{1}{K}\sum_{m=1}^{K}\mathcal{Z}(\mathbf{z}_n;\tau)^{(m)}\right)^2}\\=&\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\mathcal{Z}(\mathbf{z}_n;\tau)^{(k)}\right)^2}\\=&\sqrt{\frac{1}{K\sigma(\mathbf{z}_n)^2\tau^2}\sum_{k=1}^{K}\left(\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n\right)^2}\\=&\frac{1}{\tau},\end{aligned}$$

since $\sigma(\mathbf{z}_n)=\sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n\right)^2}$. The same proof holds for $\mathbf{v}_n$ and thus we omit it. □

1.3. Proof of Boundedness

We want to prove that the logit after the weighted $\mathcal{Z}$-score $\mathcal{Z}(\mathbf{z}_n;\tau)$ in Section 4.3 of the manuscript is bounded within $[-\sqrt{K-1}/\tau,\sqrt{K-1}/\tau]$.

Proof. Let $\overline{\mathbf{z}}_n^*=\frac{1}{K-1}\sum_{m\neq t_0}^{K}\mathbf{z}_n^{(m)}$ for an arbitrary index t_0, then we have the equation:

$$\begin{aligned}\sum_{k=1}^{K}\left(\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n\right)^2&=\sum_{k=1}^{K}\left(\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n^*\right)^2-\frac{1}{K}\left(\mathbf{z}_n^{(t_0)}-\overline{\mathbf{z}}_n^*\right)^2\\&=\sum_{k\neq t_0}^{K}\left(\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n^*\right)^2+\frac{K}{K-1}\left(\mathbf{z}_n^{(t_0)}-\overline{\mathbf{z}}_n\right)^2.\end{aligned}$$

Table 1. The Top-1 Accuracy (%) of different knowledge distillation methods on the validation set of CIFAR-100 [10]. The teacher and student have distinct architectures. The KD methods are sorted by the types, i.e., feature-based and logit-based. We apply our logit standardization to the existing logit-based methods and use Δ to show its performance gain. The values in blue denote slight enhancement and those in red non-trivial enhancement no less than 0.15. The best and second best results are in **bold** and underlined respectively.

Type		ResNet32×4	ResNet32×4	WRN-40-2
	Teacher	79.42	79.42	75.61
	Student	SHN-V1	VGG8	SHN-V1
		70.50	70.36	70.50
Feature	FitNet [16]	73.59	72.91	73.73
	AT [22]	71.73	72.74	73.32
	RKD [15]	72.28	72.84	74.21
	CRD [20]	75.11	73.54	76.05
	OFD [6]	75.98	73.85	75.85
	ReviewKD [2]	77.45	74.35	77.14
	SimKD [1]	77.18	75.76	75.65
	CAT-KD [4]	**78.26**	75.92	77.35
Logit	KD [7]	74.07	72.73	74.83
	KD+Ours	74.44	73.23	75.64
	Δ	0.37	0.50	0.81
	KD+CTKD [13]	74.71	72.47	75.64
	KD+CTKD+Ours	74.85	73.51	76.32
	Δ	0.14	1.04	0.68
	DKD [25]	76.45	74.48	76.70
	DKD+Ours	76.77	74.61	76.95
	Δ	0.32	0.13	0.25
	MLKD [9]	77.18	74.58	77.44
	MLKD+Ours	78.15	**75.98**	**78.28**
	Δ	0.97	1.40	0.84

This yields the inequality:

$$\sigma(\mathbf{z}_n)^2 \geq \frac{1}{K-1}\left(\mathbf{z}_n^{(t_0)} - \overline{\mathbf{z}}_n\right)^2.$$

Eventually, we obtain

$$\Rightarrow \left|\mathcal{Z}(\mathbf{z}_n;\tau)^{(t_0)}\right| = \left|(\mathbf{z}_n^{(t_0)} - \overline{\mathbf{z}}_n)/\sigma(\mathbf{z}_n)/\tau\right| \leq \left|\sqrt{K-1}/\tau\right|.$$

The same proof holds for $\mathbf{v}_n$ and thus we omit it. □

2. Implementation Details

We follow the same experimental settings as previous works [2, 9, 25].

For the experiments on **CIFAR-100**, the optimizer is SGD [19] and the epoch number is 240, except for MLKD [9] of 480. The batch size is 64. The learning rate is set initially 0.01 for MobileNets[8, 17] and ShuffleNets [24] and 0.05 for other architectures consisting of ResNets [5], WRNs [21] and VGGs [18]. The learning rate shrinks by a decay rate of 0.1 at 150-th, 180th and 210th epochs. The momentum and weight decay are respectively 0.9 and $5e-4$. The weight of CE loss is unchanged from its original value (0.1 for KD [7] and CTKD [13], 1.0 for DKD [25] and MLKD [9]). We choose the base temperature $\tau = 2$ and the weight of KD loss $\lambda_{KD} = 9$ by default according to the ablation studies. For the experiments of DKD [25], the weight of KD loss is set to 12.

For the settings on **ImageNet**, SGD solver is used and the total epoch number is 100. The batch size is 512. The learning rate is set to 0.2 and divided by 10 at 30th, 60th, 90th epochs. The momentum and weight decay are respectively 0.9 and $1e-4$. The weight of CE loss is unchanged from its original value (0.1 for KD [7] and CTKD [13], 0.5 for DKD [25] and MLKD [9]). We choose the base temperature $\tau = 2$ and the weight of KD loss $\lambda_{KD} = 9$ by default. For the experiments of DKD [25], the weight of KD loss is set to 12.

2.1. Implement ours on CTKD, DKD and MLKD

CTKD [13] is a method of choosing adaptive sample-wise temperature by adversarial learning. Therefore, we combine ours with CTKD [13] by using CTKD to choose the base temperature τ. The weight of KL loss is set to 9 for all experiments of CTKD [13]+Ours.

DKD [25] is a KD method of decoupling KL divergence into two terms, i.e., target class knowledge distillation (TCKD) and non-target class knowledge distillation (NCKD). Considering the fact that it is a variant of KL divergence, we just apply our pre-process directly. The weight of DKD loss is set to 12 for all experiments of DKD [13]+Ours.

MLKD [9] is a logit-based KD method that regulates the alignment not only at the instance level but also at batch and class levels. Therefore, the approach leverages the `softmax` function involving temperatures at three levels. We thus apply our pre-process in the `softmax` of the levels. The weight of MLKD loss is set to 9 for all experiments of MLKD [9]+Ours.

3. More Analysis

3.1. More experiments on CIFAR-100

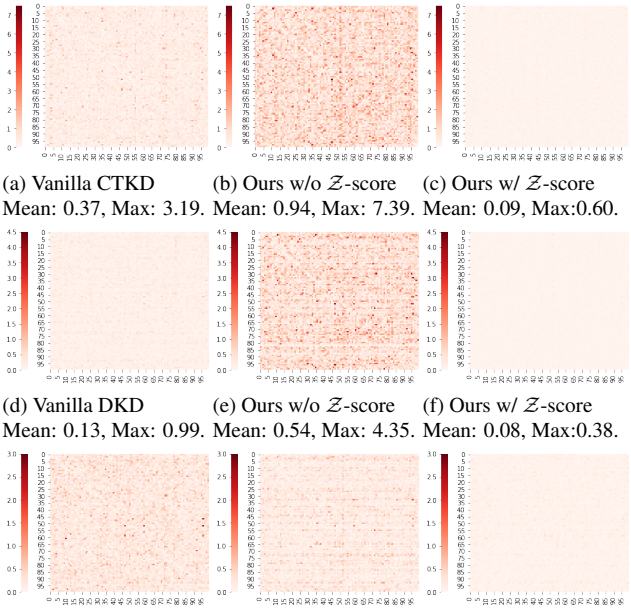
We conduct several more experiments on CIFAR-100 where the teacher and student have different architectures. Due to the page limit, we put the results in the supplementary materials, which are shown in Tab. 1. As implied, our preprocess benefits all the logit-based KD methods. For the experiments of other baselines like ReviewKD [2], CAT-KD [4], etc., we use their original codes and default configurations to run experiments for any missing setting. The additional experiments further validate the advantage of our method for boosting the existing logit-based KD methods.

Table 2. The ablation studies under different settings in $\mathcal{Z}$-score. The base temperature τ is set to be 1. By default $\lambda_{CE} = 0.1$. The logit vector of teacher $\mathbf{v}_n$ and student $\mathbf{z}_n$ are abbreviated as $\mathbf{z}$ for succinctness. The teacher and student are ResNet32×4 and ResNet8×4.

λ_{KD}	$\mathbf{z}$ (KD)	$\mathbf{z}-\overline{\mathbf{z}}$	$\frac{\mathbf{z}}{\sigma(\mathbf{z})}$	$\frac{(\mathbf{z}-\overline{\mathbf{z}})}{\sigma(\mathbf{z})}$ (Ours)
0.9	73.09	72.60	74.81	74.96
3.0	73.79	73.65	75.89	75.85
6.0	73.91	73.87	75.97	76.31
9.0	73.67	73.97	75.61	76.23
12.0	73.60	74.00	75.89	76.11
15.0	73.19	73.20	75.72	75.91
18.0	72.48	72.95	75.65	75.72

Table 3. The ablation studies under different settings in $\mathcal{Z}$-score. The base temperature τ is set to be 4. By default $\lambda_{CE} = 0.1$. The logit vector of teacher $\mathbf{v}_n$ and student $\mathbf{z}_n$ are abbreviated as $\mathbf{z}$ for succinctness. The teacher and student are ResNet32×4 and ResNet8×4.

λ_{KD}	$\mathbf{z}$ (KD)	$\mathbf{z}-\overline{\mathbf{z}}$	$\frac{\mathbf{z}}{\sigma(\mathbf{z})}$	$\frac{(\mathbf{z}-\overline{\mathbf{z}})}{\sigma(\mathbf{z})}$ (Ours)
0.9	73.76	73.61	73.03	73.83
3.0	74.07	74.09	74.15	74.33
6.0	74.14	73.93	76.11	76.40
9.0	74.19	74.24	76.17	76.54
12.0	73.96	73.97	76.12	76.39
15.0	73.08	73.85	76.14	76.38
18.0	69.98	70.84	76.18	76.43



(a) Vanilla CTKD Mean: 0.37, Max: 3.19. (b) Ours w/o $\mathcal{Z}$-score Mean: 0.94, Max: 7.39. (c) Ours w/ $\mathcal{Z}$-score Mean: 0.09, Max:0.60.

(d) Vanilla DKD Mean: 0.13, Max: 0.99. (e) Ours w/o $\mathcal{Z}$-score Mean: 0.54, Max: 4.35. (f) Ours w/ $\mathcal{Z}$-score Mean: 0.08, Max:0.38.

(g) Vanilla MLKD Mean: 0.23, Max: 2.75. (h) Ours w/o $\mathcal{Z}$-score Mean: 0.18, Max: 1.91. (i) Ours w/ $\mathcal{Z}$-score Mean: 0.06, Max:0.44.

Figure 1. The heatmaps of the average logit difference between the teacher and student. The **1st Row** is for CTKD, **2nd Row** is for DKD and **3rd Row** is for MLKD. Our pre-process indeed enables the student to generate the logits of divergent range from the teacher as shown in 1b, 1e&1h, while its standardized logits (1c, 1f&1i) are more closer to the teacher's than vanilla CTKD (1a), DKD (1d) and MLKD (1g).

3.2. More Ablation Studies

We conduct more ablation studies for the extra cases of base temperatures $\tau = 1$ and $\tau = 4$. The results are shown in Tab. 2 and Tab. 3 respectively. We can find that for any base temperature increasing the weight of KD loss does not enhance the performance of vanilla KD. However, a relatively large weight of KD loss for our pre-process yields a significant performance gain (such as $\lambda_{KD} = 6, 9, 12, 15$). Besides, the proposed pre-process is not very sensitive to the base temperature and the weight of KD loss. A relatively large range of their values constantly leads to a satisfactory result. For example, when $\tau = 1, 2, 4$ and $\lambda_{KD} = 6, 9, 12, 15$, almost all the accuracy values exceed 76%. Therefore, we choose $\tau = 2$ and $\lambda_{KD} = 9$ by default.

ATKD [3] has a similar idea regarding standard deviation with our $\mathcal{Z}$-score. As shown in the tables, our results are consistently better than $\frac{\mathbf{z}}{\sigma(\mathbf{z})}$. Besides, their derivation involves an approximation of Taylor expansion, while ours is more mathematically accurate. They also assume the mean of logits equal to zero. In contrast, as shown in the bivariate histograms, the logits are always numerically divergent from zero. Such an assumption may reduce its accuracy.

Note that different from existing works, our pre-process is not an individual KD method. Instead, our work is like CTKD [13] and can serve as an assistant for all logit-based KD involving temperature. So we apply our pre-process to three existing KD methods and help them get better and many of their results exceed SOTA performance.

Regarding the reason on the necessity of the relatively large weight λ_{KD} for our pre-process, we give the derivation of the gradient of loss with respect to $\mathbf{z}_n^{(k)}$ and try to answer it from a perspective of gradient compensation.

We know the expression of KD loss is given by

$$\mathcal{L}_{KD} = \mathcal{L}_{\mathrm{KL}}\left(q(\mathbf{v})||q(\mathbf{z})\right) = \sum_{k=1}^{K} q(\mathbf{v})^{(k)} \log\left(\frac{q(\mathbf{v})^{(k)}}{q(\mathbf{z})^{(k)}}\right).$$

Then its gradient with respect to $\mathbf{z}_n^{(k)}$ is

$$\begin{aligned}\frac{\partial \mathcal{L}_{KD}}{\partial \mathbf{z}_n^{(k)}} &= \frac{\partial \mathcal{L}_{KD}}{\partial q(\mathbf{z}_n)^{(k)}}\frac{\partial q(\mathbf{z}_n)^{(k)}}{\partial \mathbf{z}_n^{(k)}} + \sum_{m\neq k}\frac{\partial \mathcal{L}_{KD}}{\partial q(\mathbf{z}_n)^{(m)}}\frac{\partial q(\mathbf{z}_n)^{(m)}}{\partial \mathbf{z}_n^{(k)}} \\ &= -\frac{q(\mathbf{v}_n)^{(k)}}{q(\mathbf{z}_n)^{(k)}}\frac{\partial q(\mathbf{z}_n)^{(k)}}{\partial \mathbf{z}_n^{(k)}} - \sum_{m\neq k}\frac{q(\mathbf{v}_n)^{(m)}}{q(\mathbf{z}_n)^{(m)}}\frac{\partial q(\mathbf{z}_n)^{(m)}}{\partial \mathbf{z}_n^{(k)}}.\end{aligned}$$

Before computing $\frac{\partial q(\mathbf{z}_n)^{(k)}}{\partial \mathbf{z}_n^{(k)}}$ and $\frac{\partial q(\mathbf{z}_n)^{(m)}}{\partial \mathbf{z}_n^{(k)}}$, we need the fol-

lowing intermediate step, i.e.,

$$\frac{\partial}{\partial \mathbf{z}_n^{(k)}}\left(e^{\frac{\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n}{\sigma(\mathbf{z}_n)\tau}}\right)=\frac{1}{\sigma(\mathbf{z}_n)\tau}e^{\frac{\mathbf{z}_n^{(k)}-\overline{\mathbf{z}}_n}{\sigma(\mathbf{z}_n)}}.$$

We then compute the following terms,

$$\begin{cases}\dfrac{\partial q(\mathbf{z}_n)^{(k)}}{\partial \mathbf{z}_n^{(k)}}=\dfrac{1}{\sigma(\mathbf{z}_n)\tau}\left(q(\mathbf{z}_n)^{(k)}-q^2(\mathbf{z}_n)^{(k)}\right)\\ \dfrac{\partial q(\mathbf{z}_n)^{(m)}}{\partial \mathbf{z}_n^{(k)}}=-\dfrac{1}{\sigma(\mathbf{z}_n)\tau}q(\mathbf{z}_n)^{(k)}q(\mathbf{z}_n)^{(m)}.\end{cases}$$

Finally, we plug the above terms back and obtain

$$\begin{aligned}\frac{\partial \mathcal{L}_{KD}}{\partial \mathbf{z}_n^{(k)}}=&-\frac{q(\mathbf{v}_n)^{(k)}}{q(\mathbf{z}_n)^{(k)}}\frac{1}{\sigma(\mathbf{z}_n)\tau}\left(q(\mathbf{z}_n)^{(k)}-q^2(\mathbf{z}_n)^{(k)}\right)\\&-\sum_{m\neq k}\frac{q(\mathbf{v}_n)^{(m)}}{q(\mathbf{z}_n)^{(m)}}\frac{-1}{\sigma(\mathbf{z}_n)\tau}q(\mathbf{z}_n)^{(k)}q(\mathbf{z}_n)^{(m)}\\=&\frac{1}{\sigma(\mathbf{z}_n)\tau}\left(q(\mathbf{z}_n)^{(k)}-q(\mathbf{v}_n)^{(k)}\right).\end{aligned}$$

By applying Taylor formula, we can get

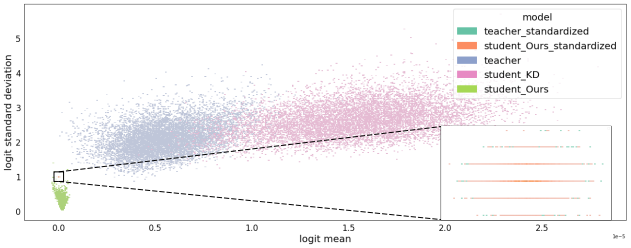
$$\frac{\partial \mathcal{L}_{KD}}{\partial \mathbf{z}_n^{(k)}}\approx\frac{1}{K\sigma(\mathbf{z}_n)^2\tau^2}\left(\mathbf{z}_n^{(k)}-\mathbf{v}_n^{(k)}\right)$$

Besides the factor of the base temperature τ, considering the decline of gradient due to the $\frac{1}{\sigma(\mathbf{z}_n)^2}$ where $\sigma(\mathbf{z}_n)$ is usually greater than 1, we consider increasing the weight of KD loss manually to compensate the gradient decline. We also tried to make the weight of KD loss exactly $\sigma(\mathbf{z}_n)^2$. However, gradients turn to be unstable.

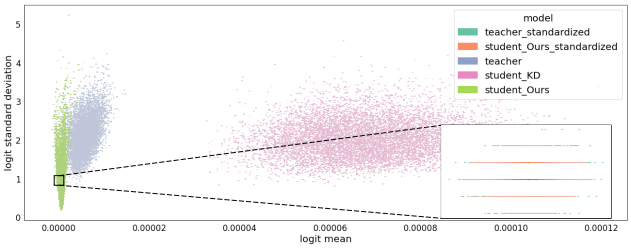
3.3. More visualizations

We show more visualizations of CTKD [13], DKD [25] and MLKD [9] regarding logit difference in Fig. 1 and logit bivariate histogram in Fig. 2. In the visualization section, the teacher and student for all experiments are ResNet32×4 and ResNet8×4.

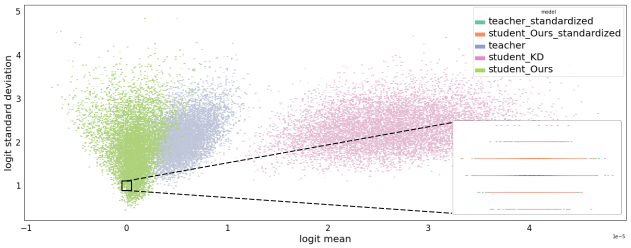
The results are similar to the cases of KD [7]. One noteworthy observation is that in Fig. 2b and 2c the students distilled by vanilla DKD [25] and MLKD [9] have already generated the logits diverging from those of the teacher. Considering DKD [25] and MLKD [9] perform much better than KD [7] and CTKD [13], a strict constraint of logit mimicking may indeed impedes the performance of students. Namely, a distribution of the student logits differing from the teacher can yield better accuracy. Compared to vanilla DKD [25] and MLKD [9], our pre-process achieves better results and, at the same time, enables a perfect match between standardized teacher and student logits.



(a) Teacher and student distilled by CTKD only and CTKD with ours



(b) Teacher and student distilled by DKD only and DKD with ours



(c) Teacher and student distilled by MLKD only and MLKD with ours

Figure 2. The bivariate histogram of logit mean and logit standard deviation for multiple models on CIFAR-100.

3.4. More experiments on distilling ViT

Distilling ViT using our method is feasible as our work is model-agnostic, and experiments are conducted on CIFAR-100 following the same setting as [11, 12]. The results are presented in Tab. 4. Notably, our method consistently and significantly improves KD for all ViTs. We only attempt KD+Ours with default λ_{KD} and τ but still achieve comparable results against AutoKD [12]. It outperforms AutoKD [12] considerably, especially when the student is hierarchical and relatively large (78.43 vs 77.48). These findings show its efficiency in mitigating ViTs' data-hungry issue. Fig. 3 plots two validation logs during training, showing that ours consistently transfers rich latent knowledge, helping KD surpass LG [11] after the 120-th epoch.

3.5. Experiments with More baselines

More comparisons against TAKD [14], ATKD [3], and PT-Loss [23] are presented in Tab. 5. Since the code is unavailable, we implement PT-Loss ourselves. Though TAKD could alleviate teacher-student gap, TA-student or teacher-TA gap may exist and hinder its final performance. ATKD and PT-Loss yield comparable performances against

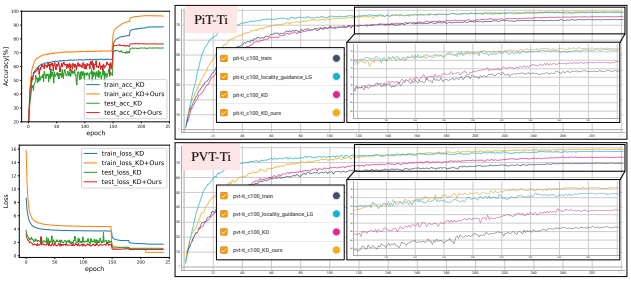


Figure 3. **Left**: Accuracy & loss of RN32×4-RN8×4. **Right**: Top-1 validation accuracy on CIFAR-100 for PiT-Ti & PVT-Ti.

Table 4. The Top-1 Acc. (%) of KD methods on CIFAR-100. Teacher is ResNet56. Hie. indicates if model is hierarchy structure. No logs and codes for Auto-KD are available yet. The values from column names without † are reported by [11, 12].

Student	Hie.	Size	Train	AT	LG[1]	AutoKD[5]	KD	KD+Ours†
DeiT-Ti	✗	5M	65.08	73.51	78.15	**78.58**	73.25	78.55 +5.30
T2T-ViT$_7$	✗	4M	69.37	74.01	78.35	**78.62**	74.15	78.43 +4.28
PiT-Ti	✔	5M	73.58	76.03	78.48	78.51	75.47	**78.76** +3.29
PVT-Ti	✔	13M	69.22	74.66	77.07	77.48	73.60	**78.43** +4.83

Table 5. Results of various teachers on CIFAR-100. Student is W-16-2. We set TA in TAKD as the student in the left previous column. The values of row names without † are from our paper.

Teacher	VGG13 74.64	W-28-2 75.45	W-40-2 75.61	W-16-4 77.51	W-28-4 78.60	RN50 79.34
TAKD$^\dagger_{TA}$	74.97$_{74.8}$	75.54$_{75.9}$	75.02$_{77.0}$	75.92$_{76.5}$	75.27$_{78.0}$	75.35$_{80.1}$
ATKD†	75.01	76.14	75.89	76.32	75.61	76.10
PT-Loss†	75.03	76.31	76.12	76.47	75.58	76.12
KD+Ours	75.03 +0.1	76.32 +1.0	76.11 +1.2	**76.72** +0.9	75.77 +0.7	76.24 +0.9
DKD+Ours	**75.56** +0.1	**76.39** +0.5	**76.39** +0.2	76.68 +0.7	**76.67** +0.2	**76.82** +0.2

KD+Ours but could not outperform DKD+Ours.

3.6. Experiments on Additional Datasets

We also conduct experiments on two additional datasets e.g., COCO and CUB.

For COCO, we adhere to the exact settings of ReviewKD and DKD and present results in Tab. 6. Our method is able to improve the performance of the object detector consistently.

For CUB, we choose ResNet34 as teacher and train it in two ways. The first is to train from scratch, and the second is to fine-tune the model pre-trained on ImageNet. We select two distillation methods, KD and DKD, to distill students. The patch size is 448. The learning rate is 0.2, the same as distilling RN18 on ImageNet. Other settings follow the settings as Tab. 5 on CIFAR-100. The results are in Tab. 7.

Table 6. AP Results on MS-COCO (val2017) based on Faster RCNN-FPN. Teacher-student pair is ResNet50 & MobileNet-V2. The values of columns without † are from ReviewKD & DKD.

	Tea.	Stu.	ReviewKD	KD	KD+Ours†	DKD	DKD+Ours†
AP	40.22	29.47	33.71	30.13	31.74 +1.61	32.34	**33.98** +1.64
AP$_{50}$	61.02	48.87	53.15	50.28	52.77 +2.49	53.77	**54.93** +1.16
AP$_{75}$	43.81	30.90	36.13	31.35	33.40 +2.05	34.01	**36.34** +2.33

Table 7. Accuracy on CUB (RN34*: pretrained on ImageNet)

Model setting	Acc.	Tea.	Stu.	SimKD	CAT-KD	KD	KD+Ours
RN34/RN18	top-1	65.03	61.13	65.96	66.36	65.84	**67.35** +1.51
	top-5	85.36	83.05	86.07	86.55	85.81	**86.69** +0.88
RN34*/RN18	top-1	83.40	61.13	78.95	78.86	78.81	**79.53** +0.72
	top-5	95.63	83.05	94.08	93.64	93.80	**94.13** +0.33

References

[1] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *CVPR*, 2022. 2

[2] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, 2021. 2

[3] Jia Guo. Reducing the teacher-student gap via adaptive temperatures, 2022. 3, 4

[4] Ziyao Guo, Haonan Yan, Hui Li, and Xiaodong Lin. Class attention transfer based knowledge distillation. In *CVPR*, 2023. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[6] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, 2019. 2

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 4

[8] 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. 2

[9] Ying Jin, Jiaqi Wang, and Dahua Lin. Multi-level logit distillation. In *CVPR*, 2023. 2, 4

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2

[11] Kehan Li, Runyi Yu, Zhennan Wang, Li Yuan, Guoli Song, and Jie Chen. Locality guidance for improving vision transformers on tiny datasets. In *ECCV*, 2022. 4, 5

[12] Lujun Li, Peijie Dong, Zimian Wei, and Ya Yang. Automated knowledge distillation via monte carlo tree search. In *ICCV*, 2023. 4, 5

[13] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation. In *AAAI*, 2023. 2, 3, 4

[14] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI*, 2020. 4

[15] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, 2019. 2

[16] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou,

Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *ICLR*, 2015. 2

[17] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2

[18] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[19] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 2

[20] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *ICLR*, 2020. 2

[21] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 2

[22] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ICLR*, 2017. 2

[23] Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. Do not blindly imitate the teacher: Using perturbed loss for knowledge distillation. *arXiv preprint arXiv:2305.05010*, 2023. 4

[24] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 2

[25] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *CVPR*, 2022. 2, 4