Supplementary Materials for *Shadow Knowledge Distillation: Bridging Offline and Online Knowledge Transfer*

Anonymous Author(s)
Affiliation
Address
email

1 Main Experimental Settings

In this section, we provide detailed settings of the classification experiments and extended experiments conducted on vision transformer/object detection.

1.1 Experiments on CIFAR-100

Dataset. CIFAR-100 [14] is the most popular classification dataset for evaluating the performance of distillation methods. It contains 50,000 training images and 10,000 test images with 100 classes.

Implementation. In the comparison experiments with other offline KD methods, we use the same training settings of CRD [24] to implement various KD methods, whose training epochs are 240. We use a mini-batch size of 64 and a standard SGD optimizer with a weight decay of 5×10^{-4}. The multi-step learning rate is initialized to 0.05, decayed by 0.1 at 150, 180 and 210 epochs. For SHAKE, we set λ and τ as 1 and 4, respectively. In the comparison experiments with other online KD methods, we adopt the same training settings with OKDDip [2]. Specifically, all networks are trained for 300 epochs, the batch size is 128, the weight decay is 5×10^{-4}, and the optimizer is SGD. We set the initial learning rate to 0.1, decayed by 0.1 at epochs 150 and 225. To fairly compare these methods with multiple teacher structures, SHAKE chooses three pre-trained teacher models to distill the student models with multiple shadow heads.

1.2 Experiments on ImageNet

Dataset. We also conduct experiments on the ImageNet dataset (ILSVRC12) [23], which is considered the most challenging classification task. It contains about 1.2 million training images and 50 thousand validation images, and each image belongs to one of 1,000 categories.

Implementation. In the ImageNet experiments, the student models (*i.e.*, ResNet-18 [8] and MobileNet [11]) are trained with 100 epochs. The batch size is set to 256 and the multi-step learning rate is initialized to 0.1, decayed by 0.1 at 30, 60, and 90 epochs. Other KD methods are implemented following the hyperparameter settings in the original paper. And SHAKE's detailed settings are the same as those on the CIFAR-100.

1.3 Experiments on vision transformer

Vision transformer. Transformer model [27] has been widely used in natural language processing (NLP). Inspired by its success in NLP, Vision Transformer (ViT) is proposed by Google [5] and DeiT [25] improves its training process by data augmentation and knowledge distillation.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Table 1: Comparison of training time, Top-1 accuracy (%), and teacher-student gap (T-S gap) among the (a) KD, (b) DML, (c) DML† without $KD_{S\to T}$, (d) KD† with $KD_{S\to T}$, and our SHAKE on CIFAR-100. Training time (GPU-seconds) is measured on a single NVIDIA 2080Ti with a batch size of 64. The teacher-student gap [3] is defined as KL divergence between their outputs (lower is better). T (forward) represents the cost of forward propagation of the teacher model. T (head) means only updating head, and T (60) means only updating for 60 epochs. T$'$ (share) denotes proxy teacher shares student's backbone. For SHAKE, we report the accuracy of the proxy model as the performance of the teacher model, and the teacher-student gap in SHAKE refers to the gap between the student and the proxy teacher.

Student & Teacher		[S] ResNet-20 & [T] ResNet-110				[S] WRN-16-2 & [T] WRN-40-2				[S] VGG-8 & [T] VGG-13			
Method	Computation	Time	S-Top-1	T-Top-1	T-S gap	Time	[S] Top-1	[T] Top-1	T-S gap	Time	S-Top-1	T-Top-1	T-S gap
Baseline	S	4,654	69.09			2,888	73.26	-	-	1,601	70.36	-	-
KD	S + T (forward)	6,036	70.66	74.31	1.12	3,980	74.92	75.61	1.56	2,047	72.98	74.64	1.76
DML	S + T	26,076	71.52	75.32	0.28	7,223	75.33	78.11	0.35	3,964	73.64	75.53	0.42
DML†	S + T	26,619	70.55	74.02	0.62	7,016	73.97	76.01	0.69	3,560	71.72	74.79	0.85
KD†	S + T	25,894	71.76	74.36	0.76	6,902	75.58	75.72	0.81	3,560	73.65	74.92	0.77
KD†	S + T (head)	6,821	71.05	74.32	0.92	4,179	75.08	75.55	0.98	2,275	73.22	74.76	0.91
KD†	S + T (60)	12,796	71.13	74.34	0.88	4,577	75.17	75.88	1.12	3,070	73.45	74.88	0.99
SHAKE	S + T$'$	11,167	71.82	71.75	0.33	4,776	76.36	76.22	0.66	2,862	74.35	74.28	0.57
SHAKE	S + T$'$ (share)	7,122	72.02	71.96	0.21	4,278	76.82	76.78	0.26	2,139	74.99	74.92	0.31

Table 2: Top-1 accuracy (%) of SHAKE with proxy teachers under different distillation loss and supervision signals. WRN-16-2 (73.26%) is distilled with pre-trained (Pre.) WRN-40-2 (75.61%), and VGG-8 (70.36%) is distilled by pre-trained VGG-13 (74.64%), respectively.

Distillation signals		Top-1	
Pre. teacher	Ground truth	WRN-16-2	VGG-8
KL	✗	76.82	74.99
CE	✗	76.45	74.52
l_2	✗	76.68	74.86
KL	CE	75.73	74.14

Student architectures. In vision transformer, the input images are firstly divided into a sequence of patches and the transformer network is utilized to extract the image features for visual recognition. First, the patches are flattened and projected into patch embeddings by a linear layer. Next, these patch embeddings are added with a set of learnable position embeddings to maintain positional information. Finally, a class token is concatenated with these enhanced patch embeddings. The inner structure of the vision transformer is composed of position encoding, multi-head self-attention (MSA) blocks, and a feed-forward network, with Layernorm and residual connection add-on. In addition, DeiT introduces a distillation token to learn from the hard labels of the teacher. We extend SHAKE to DeiT-Tiny as the student model with the same convolution teacher RegNetY-16GF [20]. DeiT-Tiny has a hidden dimension of 192, 12 layers (each with three attention heads).

Implementation. For comparison purpose, we use the same data augmentation and regularization methods described in DeiT (*e.g.*, Auto-Augment, Rand-Augment, mixup). The weights of our transformers are randomly initialized by sampling from a truncated normal distribution. We use AdamW as optimizer with learning rate equal to 0.001 and weight decay equal to 0.05. The whole

Table 3: Comparison of training time and Top-1 accuracy (%) of SHAKE with proxy teachers under different weight sharing strategies for ResNet-20 (69.09%) via pre-trained teacher (Pre. T) ResNet-110 (74.31%) on CIFAR-100. Training time is measured on a single 2080Ti GPU. S-Backbone refers to the weight sharing with student's backbone. S-Head means shadow head, and S-BN means separate BatchNorm. $\times$ represents the scaling factor of the channels of the subnetwork.

Weight sharing strategy	Time	Top-1
S-Backbone+S-Head	7,122	72.02
S-Backbone+S-Head+S-BN	7,186	72.16
0.75$\times$S-Backbone+S-Head+S-BN	6,968	71.53
0.5$\times$S-Backbone+S-Head+ S-BN	6,753	71.15

Table 4: Comparison of results with other distillation methods reported in CRD [24] under the same training setting of 240 epochs. Most results of other methods refer to the CRD. We report Top-1 mean accuracies (%) over 3 runs.

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	ResNet-110 ResNet-20	ResNet-110 ResNet-32	ResNet-32x4 ResNet-8×4	VGG-13 VGG-8	ResNet-50 MobileNetV2	ResNet-32×4 ShuffleNetV1	ResNet-32×4 ShuffleNetV2
Teacher	75.61	75.61	74.31	74.31	79.42	74.64	79.34	79.42	79.42
Student	73.26	71.98	69.06	71.14	72.50	70.36	64.60	70.50	71.82
FitNets [22]	73.58	72.24	68.99	71.06	73.50	71.02	63.16	73.59	73.54
AT [30]	74.08	72.77	70.22	72.31	73.44	71.43	58.58	71.73	72.73
SP [26]	73.83	72.43	70.04	72.69	72.94	72.68	68.08	73.48	74.56
CC [19]	73.56	72.21	69.48	71.48	72.97	70.71	65.43	71.14	71.29
VID [1]	74.11	73.30	70.16	72.61	73.09	71.23	67.57	73.38	73.40
RKD [17]	73.35	72.22	69.25	71.82	71.90	71.48	64.43	72.28	73.21
PKT [18]	74.54	73.45	70.25	72.61	73.64	72.88	66.52	74.10	74.69
AB [9]	72.50	72.38	69.53	70.98	73.17	70.94	67.20	73.55	74.31
FT [13]	73.25	71.59	70.22	72.37	72.86	70.58	60.99	71.75	72.50
NST [12]	73.68	72.24	69.53	71.96	73.30	71.53	64.96	74.12	74.68
CRD [24]	75.48	74.14	71.46	73.48	75.51	73.94	69.11	75.11	75.65
CRD+KD	75.64	74.38	71.56	73.75	75.46	74.29	69.54	75.12	76.05
KD [10]	74.92	73.54	70.67	73.08	73.33	72.98	67.35	74.07	74.45
KD†	75.58	74.24	71.76	73.35	74.91	73.65	68.81	75.21	75.95
DML [31]	75.33	73.98	71.52	73.28	74.30	73.64	68.52	75.58	76.44
DML†	74.83	73.26	70.55	72.98	73.15	72.86	67.22	74.02	74.32
SHAKE	76.82	75.62	72.02	74.49	77.95	74.99	70.18	77.46	78.51
SHAKE+FitNets	76.91	75.73	72.15	74.62	78.06	75.15	70.23	77.62	78.69
SHAKE+CRD	77.17	75.89	72.32	74.88	78.13	75.26	70.42	77.86	78.82

Table 5: Top-1 accuracy (%) of SHAKE with one shadow head and multiple shadow heads. VGG-16 is distilled with three pre-trained VGG-16 models as multiple teachers, and other models are distilled with three same-architecture pre-trained models, respectively.

Network	Baseline	One	Multiple
VGG-16	73.81	75.95	76.89
ResNet-110	75.88	78.98	79.61
WRN-20-8	77.50	81.22	81.94

training process includes 300 epochs. The first five epochs are for warm-up, and in the remaining epochs, the learning rate follows a cosine decay function. Following DeiT, SHAKE also employs the distillation token with shadow head as the proxy model. In addition, SHAKE adds mutual distillation between the shadow head and the classification head, resulting in significant gains than DeiT.

1.4 Experiments on object detection.

Dataset. We evaluate SHAKE on MS-COCO dataset [16] , which contains more than 120K images, covering 80 categories. All performance is evaluated on the MS-COCO validation set.

Implementation. We apply SHAKE to two-stage detector (*e.g.*, Faster R-CNN [21]) and one-stage detector (*e.g.*, RetinaNet[15]), which are widely used object detection frameworks. We initialize the backbone with weights pre-trained on ImageNet [23]. Following the common practice [15], all models are trained with 2× learning schedule (24 epochs). Horizontal image flipping is utilized in data augmentation. For SHAKE, we build an extra shadow head with the same architecture as the original classification head, which performs distillation in the detector fine-tuning stage.

2 More Comparisons and Discussions

Detailed comparisons of KD, DML and SHAKE. Table 1 presents more detailed results of KD, DML, and SHAKE on various student models. These results demonstrate that (a) Reversed distillation significantly reduces the teacher-student gap, resulting in accuracy improvements. (b) Thanks to the mutual distillation between proxy teacher and student, SHAKE has the lowest teacher-student gap and the best performance on different student models. (c) SHAKE only needs slight extra training costs than KD and is more efficient than DML.

More analysis with the proxy teacher in SHAKE. The proxy teacher model plays a key role in SHAKE. Table 2 and Table 3 present more results of the proxy teacher with different optimization and weight sharing settings. For different ways to mimic the predictions of the teacher model in Table 2,

Kullback-Leibler (KL) loss achieves the best accuracy, and Cross-Entropy (CE) loss and l_2-loss also obtain considerable gains. Moreover, additive supervision from ground truth labels for proxy teacher results in accuracy reduction since logits diversity collapse. For different weight sharing settings in Table 3, the setting of the separate BatchNorm [29] parameter will improve the accuracy, and some strategies using subnetworks [29] can bring efficiency gains. Note that the proxy model does not appear in the model inference.

More analysis with multiple shadow heads. In the multi-teacher scenario, SHAKE constructs multiple shadow heads to follow each teacher model individually to ensure the diversity of multi-logits. As shown in Table 5, such a multi-shadow head strategy brings a significant improvement compared to single shadow head.

2.1 More comparison with different KD methods.

Apart from logits feature distillation, feature KD and relation distillation can also achieve state-of-the-art performance boosts. Our SHAKE could naturally combine with these methods to obtain additional gains. That is to say, SHAKE is orthogonal to feature and relation distillations. Table 4 introduces more comparisons and combinations of SHAKE with these methods. The results indicate that SHAKE can surpass recent advanced KD methods and achieve considerable gains with other categories of distillation on different student models.

Table 6: Top-1 (%) accuracy of SHAKE with other techniques on CIFAR-100. $\uparrow$ refers to the performance gain than baseline.

Method	ResNet-32	WRN-16-2
Baseline	71.14	73.26
SHAKE	74.49 (3.35$\uparrow$)	76.82 (3.56$\uparrow$)
SHAKE + Cutout	74.68 (3.54$\uparrow$)	76.96 (3.70$\uparrow$)
SHAKE + AutoAug	74.87 (3.73$\uparrow$)	77.12 (3.86$\uparrow$)
SHAKE + Dropblock	74.72 (3.58$\uparrow$)	77.03 (3.77$\uparrow$)
SHAKE + α divergence	74.92 (3.78$\uparrow$)	77.17 (3.91$\uparrow$)

Augmenting SHAKE with other orthogonal techniques. SHAKE serves as the logits KD methods and is orthogonal to other training methods (*e.g.*, data augmentation and feature-based techniques). As shown in Table 6 , Cutout [4] and AutoAugment [6] bring $3.54\% \sim 3.86\%$ accuracy gains, and Dropblock [7] obtains $3.58\% \sim 3.77\%$ gain. Moreover, recent KD design (*e.g.*, α divergence [28]) can also help SHAKE obtain more improvements.

References

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

[2] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. Online knowledge distillation with diverse peers. In *AAAI*, 2020.

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

[4] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint, arXiv:1708.04552*, 2017.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Ekin, Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019.

[7] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Dropblock: A regularization method for convolutional networks. In *NeurIPS*, 2018.

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

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

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

[11] 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.

[12] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

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

[14] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

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

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

[19] Baoyun Peng, Xiao Jin, Jiaheng Liu, Shunfeng Zhou, Yichao Wu, Yu Liu, Dong-sheng Li, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *ICCV*, 2019.

[20] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollar. Designing network design spaces. In *CVPR*, 2020.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint, arXiv:1506.01497*, 2015.

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

[23] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

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

[25] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers amp; distillation through attention. In *ICML*, 2021.

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

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[28] Dilin Wang, Chengyue Gong, Meng Li, Qiang Liu, and Vikas Chandra. Alphanet: Improved training of supernet with alpha-divergence. *arXiv preprint arXiv:2102.07954*, 2021.

[29] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.

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

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