Knowledge Distillation with the Reused Teacher Classifier – Supplementary Material –

Defang Chen[1,2,3] Jian-Ping Mei[4] Hailin Zhang[1,2,3]
Can Wang[1,2,3] Yan Feng[1,2,3] Chun Chen[1,2,3]
[1]Zhejiang University [2]Shanghai Institute for Advanced Study of Zhejiang University
[3]ZJU-Bangsun Joint Research Center [4]Zhejiang University of Technology
defchern@zju.edu.cn, jpmei@zjut.edu.cn, {zzzhl, wcan, fengyan, chenc}@zju.edu.cn

A. Experimental Setting

A.1. Datasets and Training Details

We adopt two datasets including CIFAR-100 [8] and ImageNet [13] to conduct experiments. All images are normalized by channel means and standard deviations. A horizontal flip is used for data augmentation. **CIFAR-100**[1] contains 50,000 training images and 10,000 test images from 100 classes. Each training image is padded by 4 pixels on each size and randomly cropped as a 32×32 sample. **ImageNet**[2] contains about 1.3 million training images and 50,000 validation images from 1,000 classes. Each image is randomly cropped as a 224x224 sample without padding. The top-1 test accuracy of the teacher model (ResNet-50) is 76.26%.

Multi-Teacher Knowledge Distillation. The training hyper-parameters of multi-teacher KD are exactly the same as those of single-teacher KD on CIFAR-100. We first pre-train multiple teacher models with different initialization and then distill their knowledge into a student model. The accuracies of compared AEKD and AEKD-F [4] are obtained by running a public library[3] with default model hyper-parameters on our teacher-student combinations [22]. The top-1 test accuracy of two groups of teacher models used in our main submission are: ① Three ResNet-32x4 models (79.32, 79.43, 79.45), ② Two ResNet-32x4 models (79.43, 79.45) and one ResNet-110x2 model (78.18).

Data-Free Knowledge Distillation. We adopt a public library[4] to reproduce the results of compared approaches: ZSKT [10], DAFL [3] and CMI [5], with the default model hyper-parameters. In our experiment, the top-1 test accuracy of the teacher model (WRN-40-2) is 76.31%. The performance of the student model trained with original dataset is included for comparison.

[1] https://www.cs.toronto.edu/~kriz/cifar.html
[2] http://image-net.org/challenges/LSVRC/2012/index
[3] https://github.com/Rorozhl/CA-MKD
[4] https://github.com/zju-vipa/DataFree

Input dimension	Operator	Output dimension
$H \times W \times C_s$	1x1 Conv	$H \times W \times C_t/r$
$H \times W \times C_t/r$	3x3 Conv	$H \times W \times C_t/r$
$H \times W \times C_t/r$	1x1 Conv	$H \times W \times C_t$

Table S.1. Projector structure. "1x1/3x3Conv" denotes a convolutional layer with 1x1/3x3 kernel size. Standard batch normalization and ReLU activation are used after each convolutional layer. r is the reduction ratio.

Computing Infrastructure. All of the experiments are conducted with PyTorch [11]. CIFAR-100 experiments are conducted on a sever containing eight NVIDIA GeForce RTX 2080Ti GPUs with 11GB RAM. The CUDA version is 11.2. ImageNet experiments are conducted on a sever containing four NVIDIA A40 GPUs with 48GB RAM. The CUDA version is 11.4.

A.2. Network Architectures

We use a large number of teacher-student combinations for performance evaluation, which are composed of several popular neural network architectures: VGG [15], ResNet [6], WRN [20], MobileNetV2 [14], ShuffleNetV1 [23], ShuffleNetV2 [9]. The number behind "VGG-", "ResNet-" denotes the depth of networks. "WRN-d-w" denotes the wide-ResNet with depth d and width factor w. As the previous works do [2, 16], we expand or shrink the number of convolution filters in intermediate layers of some network architectures with a certain ratio and put that ratio behind "x", such as "ResNet-32x4".

A.3. Projector

The detailed structure of our used projector is described in Table S.1. We assume that the spatial dimensions of involved feature maps are the same, and denote them with the notations H and W. Otherwise, an average pooling op-

eration is used in advance for spatial dimension alignment to reduce the computational consumption, as the previous work do [2].

Given the feature maps of teacher and student models, the parameter number of the added projector is a function of the dimension reduction factor r

$$\mathcal{F}(r) = \frac{C_t(C_s + C_t + 4)}{r} + \frac{9C_t^2}{r^2} + 2C_t. \tag{1}$$

Proposition. The extra parameter number $\mathcal{F}(r)$ satisfies the inequality $2\mathcal{F}(2r) < \mathcal{F}(r) < 4\mathcal{F}(2r)$ under some mild conditions.

Proof.
We first prove the left part of the inequality:

$$\begin{aligned} 2\mathcal{F}(2r) &< \mathcal{F}(r) \\ 2 \times \left(\frac{C_t(C_s + C_t + 4)}{2r} + \frac{9C_t^2}{4r^2} + 2C_t \right) &< \\ \frac{C_t(C_s + C_t + 4)}{r} + \frac{9C_t^2}{r^2} + 2C_t & \\ 2 \times \left(\frac{9C_t^2}{4r^2} + 2C_t \right) &< \frac{9C_t^2}{r^2} + 2C_t \\ 2C_t - \frac{9C_t^2}{2r^2} &< 0 \\ 2C_t \left(1 - \frac{9C_t}{4r^2} \right) &< 0. \end{aligned} \tag{2}$$

Generally, the channel dimension C_t in the last feature maps of popular deep neural networks is greater than 128 on CIFAR-100 and is greater than 512 on ImageNet, which means that this equation holds when $r < 16$ and $r < 32$, respectively. This is easy to be satisfied in practice. Since a typical setting for r is 1, 2 and 4 in order to avoid substantial accuracy reduction as shown in Table S.8 and S.9.

We then prove the right part of the inequality:

$$\begin{aligned} 4\mathcal{F}(2r) &> \mathcal{F}(r) \\ 4 \times \left(\frac{C_t(C_s + C_t + 4)}{2r} + \frac{9C_t^2}{4r^2} + 2C_t \right) &> \\ \frac{C_t(C_s + C_t + 4)}{r} + \frac{9C_t^2}{r^2} + 2C_t & \\ 4 \times \left(\frac{C_t(C_s + C_t + 4)}{2r} + 2C_t \right) &> \\ \frac{C_t(C_s + C_t + 4)}{r} + 2C_t & \\ 2 \times \left(\frac{C_t(C_s + C_t + 4)}{2r} + 3C_t \right) &> 0. \end{aligned} \tag{3}$$

Since the channel dimensions C_t and C_s are always greater than zero, this inequality holds automatically. □

Student	VGG-8	WRN-16-2	WRN-16-4
	70.46 ± 0.29	73.51 ± 0.32	77.26 ± 0.24
KD [7]	73.38 ± 0.05	75.40 ± 0.34	79.24 ± 0.23
FitNet [12]	73.63 ± 0.11	75.44 ± 0.22	79.06 ± 0.16
AT [21]	73.51 ± 0.08	75.76 ± 0.29	79.38 ± 0.20
SP [17]	73.53 ± 0.23	75.61 ± 0.34	79.53 ± 0.20
VID [1]	73.63 ± 0.07	75.44 ± 0.24	79.40 ± 0.08
CRD [16]	74.31 ± 0.17	75.86 ± 0.17	79.46 ± 0.19
SRRL [19]	74.25 ± 0.35	75.89 ± 0.12	79.67 ± 0.17
SemCKD [2]	74.43 ± 0.25	75.77 ± 0.11	80.05 ± 0.27
SimKD	**74.93 ± 0.21**	**76.23 ± 0.14**	**80.36 ± 0.04**
Teacher	VGG-13	WRN-40-2	WRN-40-4
	74.64	76.31	79.51

Table S.2. Top-1 test accuracy (%) comparison on CIFAR-100.

Network	Student	SimKD	Teacher
ResNet-34 & ResNet-50	74.01	**74.64**	76.26
ResNet-50 & ResNet-101	76.26	**77.60**	77.80

Table S.3. Top-1 test accuracy (%) comparison on ImageNet.

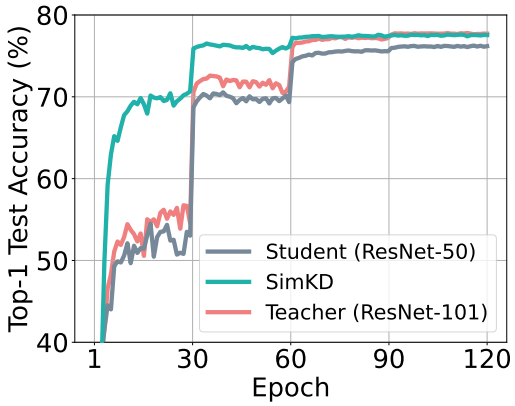


Figure S.1. The test accuracy (%) of ResNet-50 & ResNet-101 on ImageNet. Our SimKD achieves faster model convergence.

B. More Experimental Results

B.1. Comparison of Test Accuracy

Table S.2 and S.3 presents more results on CIFAR-100 and ImageNet datasets with extra *five* teacher-student combinations. Similar observations are obtained as those in the main submission. For ImageNet dataset, we replace the 3x3 convolution as the 3x3 depth-wise separable convolution in the projector (Table S.1) to control the extra parameters.

As shown in Figure S.1, our SimKD achieves *faster convergence* in the whole model training. For example, at 30th epoch, SimKD performance is on the par with the baseline student model performance at 60th epoch. Besides, at 60th epoch, SimKD already outperforms the baseline student model at 120th epoch.

Student	ResNet-8x4 73.09 ± 0.30		ResNet-8x4 73.09 ± 0.30	
	Student Classifier	Teacher Classifier	Student Classifier	Teacher Classifier
$\alpha = 0$ (KD)	**74.42 ± 0.05**	–	**75.28 ± 0.18**	–
$\alpha = 0.2$	74.42 ± 0.10	73.91 ± 0.12	74.83 ± 0.29	73.45 ± 0.31
$\alpha = 0.4$	73.99 ± 0.03	73.83 ± 0.13	74.72 ± 0.17	73.65 ± 0.27
$\alpha = 0.6$	73.93 ± 0.08	73.89 ± 0.26	74.59 ± 0.23	73.81 ± 0.25
$\alpha = 0.8$	73.76 ± 0.26	74.13 ± 0.24	74.29 ± 0.24	74.19 ± 0.22
$\alpha = 0.9$	73.52 ± 0.33	74.42 ± 0.26	73.98 ± 0.06	74.73 ± 0.13
$\alpha = 0.99$	30.77 ± 0.92	77.66 ± 0.22	23.46 ± 0.81	76.68 ± 0.13
$\alpha = 0.999$	5.59 ± 0.60	**77.98 ± 0.19**	4.55 ± 0.68	**76.84 ± 0.28**
$\alpha = 1$ (SimKD)	–	**78.08 ± 0.15**	–	**76.75 ± 0.23**
Teacher	ResNet-32x4 79.42		WRN-40-2 76.31	

Table S.4. Joint training the student feature encoder and classifier with different hyper-parameters.

Student	WRN-40-1 71.92 ± 0.17		MobileNetV2x2 69.06 ± 0.10	
	Student Classifier	Teacher Classifier	Student Classifier	Teacher Classifier
$\alpha = 0$ (KD)	**74.12 ± 0.29**	–	**72.43 ± 0.32**	–
$\alpha = 0.2$	74.33 ± 0.32	73.92 ± 0.31	72.50 ± 0.26	72.04 ± 0.23
$\alpha = 0.4$	74.49 ± 0.30	74.17 ± 0.19	72.48 ± 0.38	72.13 ± 0.25
$\alpha = 0.6$	74.53 ± 0.11	74.39 ± 0.20	72.83 ± 0.34	72.64 ± 0.29
$\alpha = 0.8$	74.93 ± 0.25	74.88 ± 0.22	73.13 ± 0.15	72.95 ± 0.06
$\alpha = 0.9$	75.18 ± 0.20	75.17 ± 0.17	73.44 ± 0.26	73.40 ± 0.27
$\alpha = 0.99$	74.17 ± 0.06	75.35 ± 0.15	74.91 ± 0.30	75.31 ± 0.16
$\alpha = 0.999$	18.39 ± 1.45	75.33 ± 0.08	13.49 ± 1.80	**75.43 ± 0.22**
$\alpha = 1$ (SimKD)	–	**75.56 ± 0.27**	–	**75.43 ± 0.26**
Teacher	WRN-40-2 76.31		ResNet-32x4 79.42	

Table S.5. Joint training the student feature encoder and classifier with different hyper-parameters.

B.2. Joint Training Results

Table S.4 and S.5 present the full joint training results with different hyper-parameters. In the case of $\alpha = 0$ or $\alpha = 1$, only the student classifier or teacher classifier produces meaningful results and the another one degrades into random guess. We denote these random guess as "–".

B.3. Sequential Training Results

The results of sequential training in the main submission are obtained with the regular training procedure. That is to say, we adopt SGD with 0.9 Nesterov momentum and 5×10^{-4} weight decay. The total training epoch is set to 240 and the learning rate is divided by 10 at 150th, 180th and 210th epochs. The initial learning rate is set to 0.01 for MobileNet/ShuffleNet-series architecture and 0.05 for other architectures. The mini-batch size is set to 64.

Learning Rate	Test Accuracy
0.01	52.03 ± 0.15
0.05	**51.97 ± 0.19**
0.1	52.01 ± 0.17
0.5	51.93 ± 0.20

Table S.6. Training a new classifier from scratch with different initial learning rates (Student: ResNet-8x4, Teacher: ResNet-32x4).

	Input (ℓ_2)	Output (ℓ_2)	Input (ℓ_2) + Output (ℓ_2)
Accuracy	**78.08 ± 0.15**	77.09 ± 0.09	77.88 ± 0.30
Loss function	$\|\boldsymbol{f}^t - \boldsymbol{f}^s\|_2^2$	$\|\boldsymbol{W}^t\boldsymbol{f}^t - \boldsymbol{W}^t\boldsymbol{f}^s\|_2^2$	$\|\boldsymbol{f}^t - \boldsymbol{f}^s\|_2^2 + \|\boldsymbol{W}^t\boldsymbol{f}^t - \boldsymbol{W}^t\boldsymbol{f}^s\|_2^2$
Gradient on $\boldsymbol{f}^s$	$-2\left(\boldsymbol{f}^t - \boldsymbol{f}^s\right)$	$-2\mathbf{W}^{t\mathrm{T}}\mathbf{W}^t\left(\boldsymbol{f}^t - \boldsymbol{f}^s\right)$	$-2\{(\boldsymbol{I} + \boldsymbol{W}^{t\mathrm{T}}\boldsymbol{W}^t)(\boldsymbol{f}^t - \boldsymbol{f}^s)\}$

Table S.7. Comparison of different loss functions (Student: ResNet-8x4, Teacher: ResNet-32x4). We omit the projector $\mathcal{P}(\cdot)$ for simplicity.

Student	WRN-40-1 71.92 ± 0.17	ResNet-8x4 73.09 ± 0.30	ResNet-110 74.37 ± 0.17	ResNet-116 74.46 ± 0.09	VGG-8 70.46 ± 0.29	ResNet-8x4 73.09 ± 0.30	ShuffleNetV2 72.60 ± 0.12
$r = 8$	67.20 ± 0.35 (0.55%, 1.05%)	**76.73 ± 0.20** (**0.35%**, 2.11%)	71.71 ± 1.00 (0.18%, 0.35%)	71.96 ± 1.09 (0.18%, 0.33%)	**74.74 ± 0.15** (**0.90%**, 0.86%)	66.26 ± 0.98 (-0.17%, 0.73%)	77.49 ± 0.31 (-0.35%, 3.76%)
$r = 4$	74.29 ± 0.03 (0.99%, 2.81%)	**77.88 ± 0.41** (**0.94%**, 5.67%)	**77.14 ± 0.22** (**0.32%**, 0.92%)	**77.18 ± 0.21** (**0.32%**, 0.87%)	**75.62 ± 0.28** (**1.62%**, 2.19%)	75.57 ± 0.03 (0.41%, 1.78%)	**78.21 ± 0.20** (**0.58%**, 8.85%)
$r = 2$	**75.56 ± 0.27** (**2.5%**, 8.77%)	**78.08 ± 0.15** (**2.88%**, 17.34%)	**77.82 ± 0.15** (**0.82%**, 2.88%)	**77.90 ± 0.11** (**0.82%**, 2.73%)	**75.76 ± 0.12** (**2.98%**, 6.23%)	**76.75 ± 0.23** (**2.18%**, 5.02%)	**78.39 ± 0.27** (**3.16%**, 23.01%)
$r = 1$	75.95 ± 0.30 (7.95%, 30.35%)	78.80 ± 0.13 (9.71%, 58.51%)	**78.00 ± 0.26** (**2.6%**, 9.96%)	**78.15 ± 0.30** (**2.6%**, 9.43%)	75.98 ± 0.21 (11.05%, 19.87%)	76.96 ± 0.07 (8.16%, 15.96%)	78.66 ± 0.08 (11.33%, 67.77%)
Teacher	WRN-40-2 76.31	ResNet-32x4 79.42	ResNet-110x2 78.18	ResNet-110x2 78.18	ResNet-32x4 79.42	WRN-40-2 76.31	ResNet-32x4 79.42

Table S.8. Top-1 test accuracy (%) and pruning ratio (the first element in parenthesis) of SimKD with various dimension reduction factor r on CIFAR-100. We also provide the ratio of the projector parameters to the student parameters (the second element in parenthesis).

Table S.6 gives additional results with different initial learning rates. It is shown that the student accuracy always stays at about 50% when the learning rate ranges from 0.01 to 0.5, which indicates the difficulty of training a satisfactory student classifier from scratch. In contrast, our SimKD achieves **78.08 ± 0.15** test accuracy without any classifier retraining but just reusing the pre-trained teacher classifier.

B.4. Comparison of Loss Function

The default feature alignment loss in our main submission is implemented in the preceding layer of the teacher classifier with a ℓ_2 loss. Its result is reported in the second column of Table S.7. Another implementation is to calculate the loss in the succeeding layer of the teacher classifier with a loss function $\|\boldsymbol{W}^t\boldsymbol{f}^t - \boldsymbol{W}^t\mathcal{P}(\boldsymbol{f}^s)\|_2^2$, and we report its results in the third column of Table S.7. From Table S.7, we find that our default feature alignment loss performs best. Moreover, the gradient comparison of different loss functions indicates that the effect of "Output (ℓ_2)" is to calibrate the gradient of "Input (ℓ_2)" with a symmetric matrix $\mathbf{W}^{t\mathrm{T}}\mathbf{W}^t$.

B.5. Comparison of Pruning Ratio

Table S.8 and S.9 present the top-1 test accuracy and the cost of pruning ratio (the first element in parenthesis) of SimKD versus different dimension reduction factors. We also provide the ratio of the projector parameters to the student parameters (the second element in parenthesis) for comparison. We make those results bold when SimKD achieves state-of-the-art performance and the added projector only requires less than or about 3% pruning ratio cost.

In some cases such as "MobileNetV2x2 & ResNet-32x4" and "ShuffleNetV1 & ResNet-32x4" with $r = 8$, we can see that the pruning ratios of SimKD are even higher than the vanilla KD training, and all competitors accordingly. Moreover, SimKD achieves the second best performance on "ShuffleNetV2 & ResNet-32x4" with $r = 8$ (SimKD: 77.49%, the best performance is achieved by SemCKD: 77.62%), "ShuffleNetV2x1.5 & ResNet-32x4" with $r = 8$ (SimKD: 78.96%, the best performance is achieved by SemCKD: 79.13%), and "ShuffleNetV2 & ResNet-110x2" with $r = 4$, (SimKD: 77.35%, the best performance is achieved by SemCKD: 77.67%). Although the projector needs retaining during the whole training and test stages, a series of trade-off experiments between test accuracy and pruning ratio show that the extra parameters it brought are negligible in most cases.

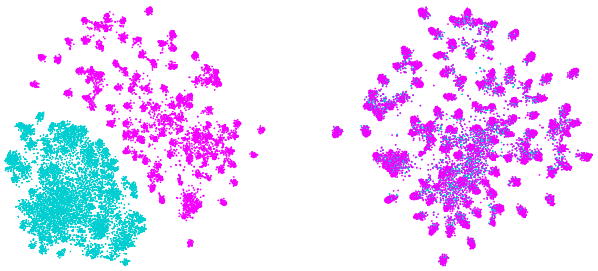
We further extend our technique to the situation where more deep teacher layers are reused for student inference and analyze the accompanying trade-off between accuracy enhancement and complexity increase. As shown in Table S.10, "SimKD+" and "SimKD++" achieve higher performance than "SimKD" but they also bring about a sharp drop of the pruning ratio, which indicates that simply reusing the final teacher classifier strikes a good balance between performance and parameter complexity.

Student	ShuffleNetV1 71.36 ± 0.25	WRN-16-2 73.51 ± 0.32	ShuffleNetV2 72.60 ± 0.12	MobileNetV2 65.43 ± 0.29	MobileNetV2x2 69.06 ± 0.10	WRN-40-2 76.35 ± 0.18	ShuffleNetV2x1.5 74.15 ± 0.22
$r = 8$	**76.68 ± 0.20** (**-0.29%**, 5.16%)	75.41 ± 0.17 (0.47%, 3.13%)	73.50 ± 0.77 (-0.99%, 1.55%)	61.78 ± 1.21 (-4.00%, 3.08%)	**74.97 ± 0.14** (**-0.58%**, 2.51%)	78.55 ± 0.33 (0.13%, 0.98%)	78.96 ± 0.15 (-0.35%, 1.98%)
$r = 4$	**77.22 ± 0.22** (**0.60%**, 12.12%)	**76.69 ± 0.23** (**1.01%**, 8.81%)	77.35 ± 0.18 (-0.63%, 3.39%)	69.43 ± 0.21 (-2.67%, 6.77%)	**75.56 ± 0.34** (**0.45%**, 5.78%)	**79.23 ± 0.06** (**0.27%**, 2.75%)	**79.48 ± 0.12** (**0.58%**, 4.65%)
$r = 2$	**77.18 ± 0.26** (**3.14%**, 32.03%)	**77.17 ± 0.32** (**2.84%**, 28.13%)	**78.25 ± 0.24** (**0.31%**, 8.19%)	**70.71 ± 0.41** (**0.52%**, 15.62%)	**75.43 ± 0.26** (**3.26%**, 14.66%)	**79.29 ± 0.11** (**0.76%**, 8.78%)	**79.54 ± 0.26** (**3.16%**, 12.09%)
$r = 1$	77.58 ± 0.36 (11.20%, 95.15%)	77.65 ± 0.24 (9.45%, 98.01%)	78.58 ± 0.22 (2.99%, 21.83%)	70.90 ± 0.17 (9.43%, 40.34%)	75.71 ± 0.20 (11.87%, 41.86%)	79.26 ± 0.17 (2.55%, 30.59%)	79.72 ± 0.24 (11.33%, 35.61%)
Teacher	ResNet-32x4 79.42	ResNet-32x4 79.42	ResNet-110x2 78.18	WRN-40-2 76.31	ResNet-32x4 79.42	ResNet-110x4 80.20	ResNet-32x4 79.42

Table S.9. Top-1 test accuracy (%) and pruning ratio (the first element in parenthesis) of SimKD with various dimension reduction factor r on CIFAR-100. We also provide the ratio of the projector parameters to the student parameters (the second element in parenthesis).

	Test Accuracy	Pruning Ratio
Student	73.09 ± 0.30	83.40%
SimKD	**78.08 ± 0.15**	**80.52%**
SimKD+	**78.47 ± 0.08**	19.21%
SimKD++	**78.88 ± 0.05**	15.97%
Teacher	79.42	0%

Table S.10. Comparison of reusing different teacher layers.



(a) Vanilla KD [7]. (b) Our SimKD.

Figure S.2. Visualizations of all test images from CIFAR-100 with t-SNE [18]. Features extracted by the teacher and student models are depicted with magenta and cyan colors, respectively, and they are almost indistinguishable in our SimKD. Best viewed in color.

B.6. Visualization

We adopt ResNet-8x4 as the student model and ResNet-32x4 as the teacher model for visualization experiments.

Ten randomly selected classes in the main submission includes "road", "bee" , "lawn_mower", "bottle", "shrew", "bridge", "man", "mouse", "sweet_pepper" and "cattle". We further visualize all 100 classes on CIFAR-100 with t-SNE in Figure S.2. The visualization results show that with the help of a simple ℓ_2 loss, the extracted features from teacher and student models become almost indistinguishable in SimKD, which ensures the student features to be correctly classified with the reused teacher classifier later.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019. 2

[2] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7028–7036, 2021. 1, 2

[3] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *International Conference on Computer Vision*, pages 3513–3521, 2019. 1

[4] Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. In *Advances in Neural Information Processing Systems*, 2020. 1

[5] Gongfan Fang, Jie Song, Xinchao Wang, Chengchao Shen, Xingen Wang, and Mingli Song. Contrastive model inversion for data-free knolwedge distillation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2374–2380, 2021. 1

[6] 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

[7] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. 2, 5

[8] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical Report*, 2009. 1

[9] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN archi-

tecture design. In *Proceedings of the European Conference on Computer Vision*, pages 122–138, 2018. 1

[10] Paul Micaelli and Amos J. Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems*, pages 9547–9557, 2019. 1

[11] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019. 1

[12] 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. 2

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[14] Mark Sandler, Andrew G. 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. 1

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1

[16] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020. 1, 2

[17] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *International Conference on Computer Vision*, pages 1365–1374, 2019. 2

[18] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 5

[19] Jing Yang, Brais Martínez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2021. 2

[20] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016. 1

[21] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. 2

[22] Hailin Zhang, Defang Chen, and Can Wang. Confidence-aware multi-teacher knowledge distillation. *arXiv preprint arXiv:2201.00007*, 2021. 1

[23] 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. 1