A. Appendix

A.1. Cross-entropy loss and CAT loss

As we have presented in the paper, the capacity of identifying class discriminative regions is critical for CNN models to perform classification. This capacity can be obtained in two approaches: (1) train models from scratch using cross-entropy loss, and (2) transfer CAMs to the trained model. However, this capacity of the models trained with the first approach is relatively restricted, since during the raw training only hard labels of the training data are offered. For the second approach, though offering hints about the class discriminative regions of input will make it easier for the trained model to obtain this capacity, its performance is also restricted by the accuracy of the model producing the transferred CAMs, because CAMs generated by the model with insufficient accuracy contain incorrect hints for the class discriminative regions of input.

As the results reported in Table A.1 and Table A.2, when the CAM producer is stronger than the trained model, only transferring CAMs can let the trained model achieve better performance compared with trained from scratch, since the transferred CAMs are more *correct* than the one that the trained model itself could generate. In contrast, when the CAM producer is weaker than the trained model, transferring CAM is not that effective: its performance is worse than using only the cross-entropy loss function during training. To sum up, (1) compared with the case where only cross-entropy loss function is used, using CAT loss function can further improve the performance of the trained model, (2) using cross-entropy loss function guarantees the performance of the trained model when the CAMs producer is relatively weak. Thus, to ensure the performance of CAT-KD, we need to utilize both cross-entropy loss function and CAT loss function, and balance them correctly.

A.2. Guidance for balancing CE loss and CAT loss

As we have discussed in Appendix A.1, properly combining the CAT loss and cross-entropy loss is of great importance for the performance of CAT-KD. As depicted by Eqn (6) in the paper, we use the factor β to balance CAT loss and cross-entropy loss. Here we present a guide for tuning β from our perspective. As can be observed in Table A.1 and Table A.2, the transferred CAMs bring more improvement when the teacher is much stronger than the student, while they might not be that beneficial when the capacity of the teacher and student is similar. Thus, the optimal value of β should be positively correlated with the capacity of the teacher, but negatively correlated with the capacity of the student. The relevant experimental verification is reported in Table A.3.

CAM producer	ResNet56	ResNet110	ResNet32×4
Acc	72.34	74.31	79.42
Trained Model	ResNet110	ResNet110	ResNet110
Acc	74.31	74.31	74.31
CAT	71.86	74.54	78.13

Table A.1. Accuracy (%) of ResNet110 trained with CAT on CIFAR-100 validation set, where the transferred CAMs are produced by various networks.

CAM producer	ResNet56	ResNet50	ResNet32×4
Acc	72.34	79.34	79.42
Trained Model	ResNet32×4	ResNet32×4	ResNet32×4
Acc	79.42	79.42	79.42
CAT	72.56	78.96	79.65

Table A.2. Accuracy (%) of ResNet32×4 trained with CAT on CIFAR-100 validation set, where the transferred CAMs are produced by various networks.

	Teacher	ResNet56	WRN-40-2	ResNet32×4
	Acc	72.34	75.61	79.42
β	10	74.08	74.96	74.57
	50	**76.28**	76.83	76.87
	100	75.84	**77.31**	77.42
	300	74.78	76.71	77.86
	600	74.63	76.43	**78.26**

Table A.3. Accuracy (%) of the model trained by CAT-KD on CIFAR-100 with various β and different teacher. The student network is ShuffleNetV1.

A.3. Normalization in CAT-KD

During CAT, we perform l_2 normalization on the transferred CAMs to ensure information indicating the category of the input is not released to the trained model. However, this process is not necessary for CAT-KD. As can be observed in Table A.4 and Table A.5, when the teacher and student have different architecture, performing normalization is beneficial for CAT-KD. However, it will become harmful when the teacher and student have similar architectures. A reasonable explanation is that the *dark knowledge* contained in logits, which will be released to the student model if the normalization is not performed, is relatively more beneficial for the student networks that have similar structure to the teacher. This coincides with the phenomenon that logit-based KD methods perform relatively better when the teacher and student have similar structures, which can be observed in Table 5 and Table 6 reported in the paper. Thus, for CAT-KD, normalization is performed when the student and teacher have different structures, while others are not.

Teacher	ResNet110	WRN-40-2	ResNet32×4
Acc	74.31	75.61	79.42
Student	ResNet32	WRN-16-2	ResNet8×4
Acc	71.14	73.26	72.5
(a)	**73.62**	**75.6**	**76.91**
(b)	73.45	75.46	76.29

Table A.4. Accuracy (%) of students trained with CAT-KD on CIFAR-100, where students and teachers have similar structure. (a): normalization is performed on the transferred CAMs during CAT-KD. (b): without performing normalization.

Teacher	ResNet50	WRN-40-2	ResNet32×4
Acc	79.34	75.61	79.42
Student	MobileNetV2	ShuffleNetV1	ShuffleNetV1
Acc	64.6	70.5	70.5
(a)	70.86	77.24	77.78
(b)	**71.36**	**77.35**	**78.26**

Table A.5. Accuracy (%) of students trained with CAT-KD on CIFAR-100, where students and teachers have different structure. (a): normalization is performed on the transferred CAMs during CAT-KD. (b): without performing normalization.

A.4. Extensions

To facilitate future works related to CAT and CAT-KD, here we offer several extensive experiment results.

Transfer CAMs generated by other methods. Following [7], many works propose to generate CAM in other ways [1, 4, 5]. Although these methods always consume much more resources, their generated target class's CAM also correctly highlights the class discriminative regions. To examine if CAT is still effective when the transferred CAMs are generated in these generalized ways, we perform CAT on CIFAR-10 and use GradCAM [4] to generate the transferred CAMs. The trained model's accuracy is only among 10%-15%, indicating transferring GradCAM [4] barely works. We think this is because CAMs of non-target classes generated by the generalized ways [1, 4, 5] do not contain useful information for CAT, though the visualization of their target class's CAM may look better than that of [7].

Coefficients in CAT loss. As we have revealed in Section 4.2, transferring CAMs of categories with higher prediction scores will bring more improvement for the trained model. Then an intuitive idea is that the trained model should focus more on mimicking the CAMs of categories with higher prediction scores. However, through experiments, we find that preferentially transferring CAMs of categories with higher prediction scores brings little benefit for CAT and CAT-KD, while it will increase the complexity and cost of the implementation of our method. Thus, as reported in Eqn (5), we consider transferring CAMs of all categories equally important and give them the same coefficient $1/k$.

A.5. More implementation details.

For all experiments reported in Section 4, without special specifications, the transferred CAMs are pooled into 2×2 during CAT and CAT-KD. For the experiments reported in Section 4.2, since there do not exist comparisons with other methods, we change the batch size to 128 to accelerate the training, while other settings are the same as those reported in Section 4.1.

Setup. All experiments are performed on an Ubuntu 16.04.1 LTS 64-bit server, with one Intel(R) Xeon(R) Silver 4214 CPU, 128GB RAM. For experiments on CIFAR-100, we utilize one RTX 2080 Ti GPU with 11GB dedicated memory. For experiments on ImageNet, we utilize four RTX 2080 Ti GPUs.

Visualization. All visualizations presented in the paper are generated by ResNet50, which has 76.16% Acc on ImageNet.

CAM's original resolution. For CIFAR-100, the resolution of CAM generated by all the models involved in this paper is 8×8 except ShuffleNet (4×4), ResNet50 (4×4), VGG (4×4), and MobileNet (2×2). For ImageNet, their original resolution is 7×7.

Figure 4. For the experiment reported in Figure 4 (right), the training set is reduced to only contain data of n categories. The reserved categories are the first n categories in the CIFAR-100 default category order.

Table 3. Binarization is performed on the transferred CAMs before they are normalized.

Table 4. We employ TrivialAugment [3] to obtain the strong teacher ResNet32×4, which has 81.36% accuracy on CIFAR-100 validation set. The results of DKD [6] and ReviewKD [2] are obtained using author-released code. For fairness, the hyper-parameters of CAT-KD, DKD, and ReviewKD are not changed with the accuracy of the teachers.

Table 9. We first use the code released by DKD [6] to obtain student models trained with various distillation methods. For the implementation of linear probing experiments, STL-10 and TinyImageNet share an identical setup. More specifically, we train linear fully connected (FC) layers of models for 40 epochs with batch size 128 using SGD. The initial learning rate is 0.1, divided by 10 at 10, 20, and 30 epochs.

Figure 7. For the experiments reported in Figure 7

(left), the training data of each category is reduced by the same proportion. The reduced data is selected in the CIFAR-100 default order. For the results reported in Figure 7 (right), we evaluate the training time (per epoch) of various KD methods, where one RTX 2080 Ti GPU with 11GB dedicated memory is used.

References

[1] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *workshop on applications of computer vision*, 2018. 2

[2] 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. 2

[3] Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. *international conference on computer vision*, 2021. 2

[4] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 2016. 2

[5] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. *computer vision and pattern recognition*, 2020. 2

[6] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. *arXiv preprint arXiv:2203.08679*, 2022. 2

[7] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 2