Class Attention Transfer Based Knowledge Distillation

Ziyao Guo[1], Haonan Yan[1,2,*], Hui Li[1,*], Xiaodong Lin[2]
[1]Xidian University, [2]University of Guelph
gzyaftermath@outlook.com

Abstract

Previous knowledge distillation methods have shown their impressive performance on model compression tasks, however, it is hard to explain how the knowledge they transferred helps to improve the performance of the student network. In this work, we focus on proposing a knowledge distillation method that has both high interpretability and competitive performance. We first revisit the structure of mainstream CNN models and reveal that possessing the capacity of identifying class discriminative regions of input is critical for CNN to perform classification. Furthermore, we demonstrate that this capacity can be obtained and enhanced by transferring class activation maps. Based on our findings, we propose class attention transfer based knowledge distillation (CAT-KD). Different from previous KD methods, we explore and present several properties of the knowledge transferred by our method, which not only improve the interpretability of CAT-KD but also contribute to a better understanding of CNN. While having high interpretability, CAT-KD achieves state-of-the-art performance on multiple benchmarks. Code is available at: https://github.com/GzyAftermath/CAT-KD.

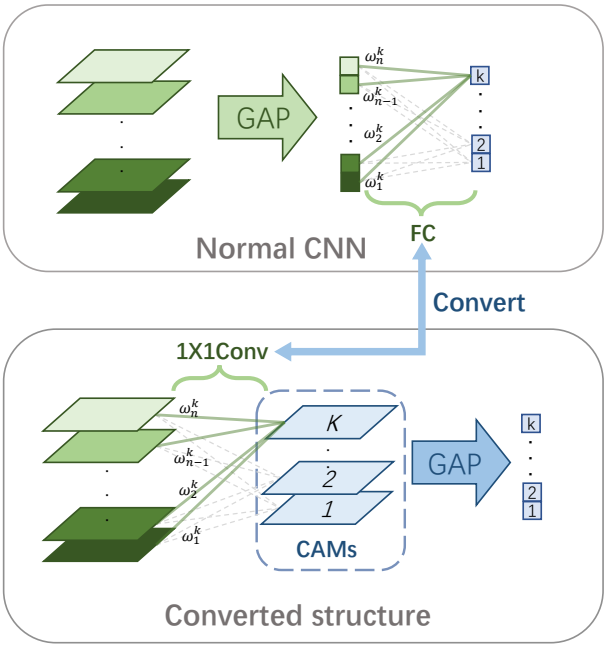

Figure 1. Illustration of the converted structure. After converting the FC layer into a convolutional layer with 1×1 kernel and moving the position of the global average pooling layer, CAMs can be obtained during the forward propagation.

1. Introduction

Knowledge distillation (KD) transfers knowledge distilled from the bigger teacher network to the smaller student network, aiming to improve the performance of the student network. Depending on the type of the transferred knowledge, previous KD methods can be divided into three categories: based on transferring logits [3, 6, 11, 16, 33], features [2, 10, 17–19, 23, 24, 28], and attention [29]. Although KD methods that are based on transferring logits and features have shown their promising performance [2, 33], it is hard to explain how the knowledge they transferred helps to improve the performance of the student network, due to the uninterpretability of logits and features. Relatively, the principle of attention-based KD methods is more intuitive: it aims at telling the student network which part of the input should it focus on during the classification, which is realized by forcing the student network to mimic the transferred attention maps during training. However, though previous work AT [29] has validated the effectiveness of transferring attention, it does not present what role attention plays during the classification. This makes it hard to explain why telling the trained model where should it focus could improve its performance on the classification mission. Besides, the performance of the previous attention-based KD method [29] is less competitive compared with the methods that are based on transferring logits and features [2, 33]. In this work, we focus on proposing an attention-based KD method that has higher interpretability and better performance.

*Corresponding author

Figure 2. Visualization of CAMs corresponding to categories with Top 4 prediction scores for the given image. The predicted categories and their scores are reported in the picture.

We start our work by exploring what role attention plays during classification. After revisiting the structure of the mainstream models, we find that with a little conversion (illustrated in Figure 1), class activation map (CAM) [34], a kind of class attention map which indicates the discriminative regions of input for a specific category, can be obtained during the classification. Without changing the parameters and outputs, the classification process of the converted model can be viewed in two steps: (1) the model exploits its capacity to identify class discriminative regions of input and generate CAM for each category contained in the classification mission, (2) the model outputs the prediction score of each category by computing the average activation of the corresponding CAM. Considering that the converted model makes predictions by simply comparing the average activation of CAMs, possessing the capacity to identify class discriminative regions of input is critical for CNN to perform classification. The question is: can we enhance this capacity by offering hints about class discriminative regions of input during training? To answer this question, we propose class attention transfer (CAT).

During CAT, the trained model is not required to predict the category of input, it is only forced to mimic the transferred CAMs, which are normalized to ensure they only contain hints about class discriminative regions of input. Through experiments with CAT, we reveal that transferring only CAMs can train a model with high accuracy on the classification task, reflecting the trained model obtains the capacity to identify class discriminative regions of input. Besides, the performance of the trained model is influenced by the accuracy of the model offering the transferred CAMs. This further demonstrates that the capacity of identifying class discriminative regions can be enhanced by transferring more *precise* CAMs.

Based on our findings, we propose class attention transfer based knowledge distillation (CAT-KD), aiming to enable the student network to achieve better performance by improving its capacity of identifying class discriminative regions. Different from previous KD methods transferring *dark knowledge*, we present why transferring CAMs to the trained model can improve its performance on the classification task. Moreover, through experiments with CAT, we reveal several interesting properties of transferring CAMs, which not only help to improve the performance and interpretability of CAT-KD but also contribute to a better understanding of CNN. While having high interpretability, CAT-KD achieves state-of-the-art performance on multiple benchmarks. Overall, the main contributions of our work are shown below:

- We propose class attention transfer and use it to demonstrate that the capacity of identifying class discriminative regions of input, which is critical for CNN to perform classification, can be obtained and enhanced by transferring CAMs.

- We present several interesting properties of transferring CAMs, which contribute to a better understanding of CNN.

- We apply CAT to knowledge distillation and name it CAT-KD. While having high Interpretability, CAT-KD achieves state-of-the-art performance on multiple benchmarks.

2. Background

The concept of knowledge distillation was proposed in [11]. As a transfer learning method, KD aims to improve the performance of the smaller student network by transferring the *dark knowledge* distilled from the bigger teacher network. Previous KD methods can be divided into three types: distillation from logits [3, 6, 11, 16, 33], features [2, 10, 17–19, 23, 24, 28] and attention [29].

To our knowledge, AT [29] is the only KD method based on transferring attention, which defines attention map as the spatial map indicating the area of input that the model focus on most. In practice, they obtain attention maps by calculating the sum of feature maps while their values are absolutized. However, AT did not present what role *attention* plays during the classification and why transferring attention maps defined in this way can improve the performance of the student network.

Previous works related to class attention originate from [34], where the authors propose to utilize high-level feature maps and the parameters of the fully connected layer to generate attention map for a specific category, which is named

class activation map (CAM). According to [34], class discriminative regions of input are highlighted in the corresponding CAM. To facilitate understanding, we visualize several CAMs in Figure 2. The following works have successfully applied CAM in various weakly supervised visual tasks [14, 27, 31]. Besides, there are also many works focus on generalizing CAM [1, 21, 26] and improving the performance of models by exploiting the information contained in CAM during training [7, 25].

Previous works have not presented what role attention plays during classification and why transferring attention maps can improve the trained model's performance on the classification mission. In this paper, we focus on figuring out this question and try to propose an attention-based KD method that has both high interpretability and competitive performance.

3. Our Method

In this section, we first analyze the structure of the mainstream CNN models and reveal that possessing the capacity of identifying class discriminative regions is critical for CNN to perform classification. Then we further propose class attention transfer to prove that this capacity can be obtained and enhanced by transferring CAMs. Finally, we apply CAT to knowledge distillation.

3.1. Revisit the structure of CNN

In image classification tasks, mainstream models usually use CNN to extract features, the resulting high-level feature maps are then globally pooled and fed to a simple fully connected layer to perform classification [8, 9, 12]. Let $\mathbf{F} = [F_1, F_2, ..., F_C] \in \mathbb{R}^{C\times W\times H}$ represents the feature maps generated by the last convolutional layer, where C, W, and H indicate channel dimension, width, and height respectively. And $f_j(x, y)$ denotes the activation of $\mathbf{F}$ in j channel at spatial location (x, y), while GAP is the global average pooling layer. Then the process of calculating logits for normal CNN models can be written as:

$$\begin{aligned} L_i &= \sum_{1\le j\le C} \omega_j^i \times GAP(F_j) \\ &= \frac{1}{W\times H}\sum_{x,y}\sum_{1\le j\le C} \omega_j^i \times f_j(x, y), \end{aligned} \tag{1}$$

where L_i denotes the logit of i-th class, ω_j^i is the weight of the fully connected layer (FC layer) corresponding to class i for $GAP(F_j)$. According to [34], we can obtain the CAM corresponding to category i by:

$$CAM_i(x, y) = \sum_{1\le j\le C} \omega_j^i \times f_j(x, y). \tag{2}$$

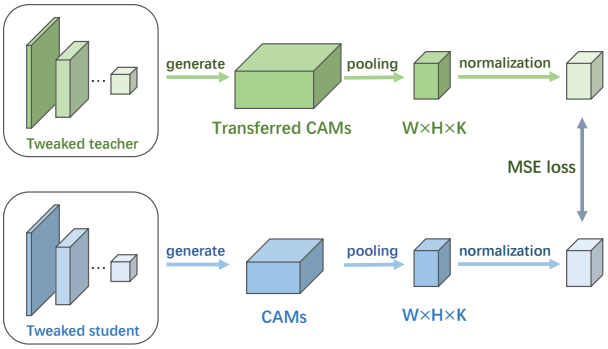

Figure 3. Illustration of CAT. During CAT, the structure of teacher and student are converted to our style (Figure 1).

According to Equation (1) and Equation (2), the calculation of L_i can be written in another form:

$$\begin{aligned} L_i &= \frac{1}{W\times H}\sum_{x,y} CAM_i(x, y) \\ &= GAP(CAM_i). \end{aligned} \tag{3}$$

As reflected in Equation (3), logits can be obtained by computing the average activation of CAMs. Inspired by it, as illustrated in Figure 1, we convert the FC layer into a 1×1 convolutional layer and move the position of the GAP layer. Then $\bar{L}_i$, the logit of i-th class generated by the converted model, can be obtained by:

$$\begin{aligned} \bar{L}_i &= GAP(Conv_i(\mathbf{F})) \\ &= \frac{1}{W\times H}\sum_{x,y}\Big(\sum_{1\le j\le C} \omega_j^i \times f_j(x, y)\Big) \\ &= GAP(CAM_i), \end{aligned} \tag{4}$$

where $Conv_i$ denotes the converted 1×1 convolution kernel that used to separate features corresponding to i-th class from $\mathbf{F}$, and ω_j^i is its weight of j channel. As reflected in Eqn(3) and Eqn(4), the conversion does not change the value of its prediction score (i.e., logits). And class activation maps can be obtained during the classification of the converted model.

As reflected in Eqn(4), the classification process of the converted model can be viewed in two steps: (1) the model exploits its capacity to identify class discriminative regions of input and generate CAMs, (2) the model outputs prediction score of each category by computing the average activation of the corresponding CAM. Considering that the model makes predictions by simply comparing the average activation of CAMs, possessing the capacity to identify class discriminative regions of input is critical for CNN to perform classification. To examine if this capacity can be obtained and enhanced by offering hints indicating class discriminative regions of input to the trained model, we propose class attention transfer.

3.2. Class Attention Transfer

The purpose of CAT is to examine if a model can obtain the capacity to identify class discriminative regions of input by transferring **only** CAMs. Thus, during CAT, the trained model is not required to perform classification, and any information related to the category of the training set data (e.g., ground-truth labels and logits) is not released to the trained model. In practice, we utilize a pre-trained model with the converted structure to generate the transferred CAMs. The illustration of the process of CAT is shown in Figure 3, while the formal description is shown below.

For a given input, let $\mathbf{A} \in \mathbb{R}^{K\times W\times H}$ denotes the CAMs generated by the converted structure, where K is the number of categories contained in the classification task, W and H denote the width and height of the generated CAM respectively. $A_i \in \mathbb{R}^{W\times H}$ represents the i channel of $\mathbf{A}$, which is the CAM corresponding to category i. And S, T denote student and teacher correspondingly. Besides, we use the average pooling function ϕ to reduce the resolution of the transferred CAMs, to improve the performance of CAT (Section 4.2). Then CAT's loss function can be defined as:

$$\mathcal{L}_{CAT} = \sum_{1\le i\le K} \frac{1}{K} \| \frac{\phi(A_i^T)}{\|\phi(A_i^T)\|_2} - \frac{\phi(A_i^S)}{\|\phi(A_i^S)\|_2} \|_2^2. \quad (5)$$

As can be seen, we perform l_2 normalization on $\phi(A_i^T)$ and $\phi(A_i^S)$ (l_1 normalization can be used as well), to ensure that information related to the category of input is not released to the trained model during CAT, considering that the average activation of CAM indicates the prediction score (Equation (3)). Besides, note that here we transfer CAMs of all categories, which is based on our finding that CAMs of all categories both contain beneficial information for CAT (Section 4.2).

Our core findings through the experiments with CAT are presented as follows, while the corresponding experimental verification and their detailed analysis can be found in Section 4.2.

- The capacity to identify class discriminative regions of input can be obtained and enhanced by transferring CAMs.
- CAMs of all categories both contain beneficial information for CAT.
- Transferring smaller CAMs performs better.
- For CAT, the critical information contained in the transferred CAMs is the spatial location of the regions with high activation in them rather than their specific value.

3.3. CAT-KD

After validating the effectiveness of CAT, we apply CAT to knowledge distillation and name it CAT-KD. The loss function of CAT-KD is:

$$\mathcal{L}_{KD} = \mathcal{L}_{CE} + \beta\mathcal{L}_{CAT}, \quad (6)$$

where $\mathcal{L}_{CE}$ denotes the standard cross-entropy loss, and β is the factor used to balance the CE loss and CAT loss.

Different from previous KD methods, we present how the *knowledge* transferred by CAT-KD helps to improve the performance of the student network: by improving its capacity of identifying class discriminative regions. Besides, through experiments with CAT, we analyze and reveal several properties of the $knowledge$ transferred by our method. This further enhances the interpretability of CAT-KD.

4. Experiments

4.1. Datasets and Implementation Details

Datasets. In the following section we explore CAT and CAT-KD mainly on two image classification datasets:

(1) CIFAR-100 [13] comprise 32×32 pixel images of 100 categories, the training and validate sets contain 50K and 10K images.

(2) ImageNet [5] is a large-scale dataset for the classification of 1K categories, containing 1.2 million training and 50K validation images.

Implementation details. Our implementation for CIFAR-100 and ImageNet strictly follows [2, 33]. Specifically, for CIFAR-100, we train all models for 240 epochs with batch size 64 using SGD. The initial learning rate is 0.05 (0.01 for ShuffleNet [15, 32] and MobileNet [20]), divided by 10 at 150, 180, and 210 epochs. For ImageNet, we train models for 100 epochs with batch size 512. The initial learning rate is 0.2 and divided by 10 for every 30 epochs. We experiment with various representative CNN network: VGG [22], ResNet [9], WideResNet [30], MobileNet [20], and ShuffleNet [15, 32].

For fairness, all the results of previous methods are either reported in previous papers [2,33] (we keep our training setting the same as theirs) or obtained using codes released by the author with our training setting. All results on CIFAR-100 are the average over 5 trials, while that on ImageNet is the average over 3 trials.

For all experiments reported in Section 4.2 and Section 4.3, without special specification, we pool the transferred CAMs into 2×2 during CAT and CAT-KD. More implementation details such as the settings of β are attached in the appendix due to the page limits.

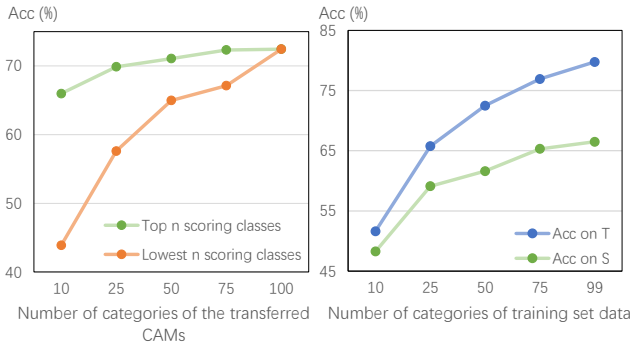

Figure 4. Accuracy of models trained with CAT on CIFAR-100. **Left**: Only CAMs of certain categories are transferred, which are selected by two strategies: (1) select categories with top n prediction scores, (2) select categories with the lowest n prediction scores. **Right**: The training set is reduced to contain data of partial categories only. T: test set of CIFAR-100. S: a subset of T which only contains data of classes that are not contained in the training set.

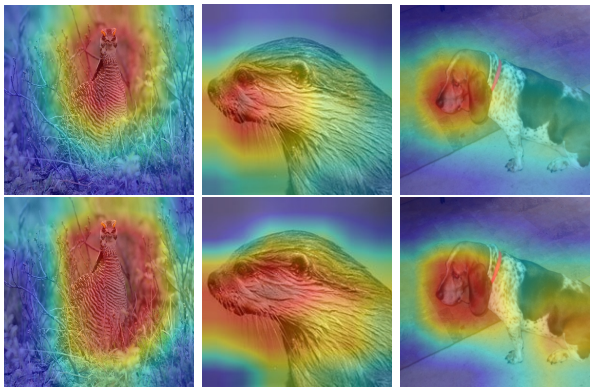

Figure 5. We set a pre-trained ResNet50 as CAMs producer to train another ResNet50 from scratch with CAT, CAMs are pooled into 2×2 during the transfer. The first row shows the visualization of the CAMs generated by the producer, while the CAMs visualized in the second row come from the trained model.

4.2. Exploration of CAT

In this section, we explore several properties of class attention transfer, which not only help to improve the performance and interpretability of CAT-KD but also contributes to a better understanding of CNN. Note that any information related to the category of the training set (e.g., ground-truth labels and logits) is **not** utilized in the experiments reported in this section.

The capacity of identifying class discriminative regions can be obtained and enhanced by transferring CAMs. As revealed in Section 3.1, being able to identify class discriminative regions of input is critical for CNN to perform classification. Thus, the intensity of this capacity can be evaluated by the model's performance on the classification mission. We perform CAT on ShuffleNetV1, where the transferred CAMs are produced by different models with various accuracy. As the results reported in Table 1, transferring only CAMs can train a model with high accuracy on the classification mission, proving the capacity of identifying class discriminative regions can be obtained by transferring CAMs. Besides, the performance of the trained model is influenced by the accuracy of the model producing the transferred CAMs, indicating that this capacity can be enhanced by transferring more *precise* CAMs.

CAMs of all categories both contain beneficial information for CAT. For a given input, we can use the method of CAM [34] to generate class activation maps for any categories contained in the classification mission. However, though a few non-target categories may share certain similarities (e.g., shape and patterns) with the target category, most of them are completely irrelevant to the input from a human understanding. However, our experiments show that class activation maps of all categories both contain beneficial information for CAT.

We first perform CAT on CIFAR-100 where only CAMs of certain categories are transferred. We designed two strategies to select the categories of the transferred CAMs: (1) select categories with the lowest n prediction scores. (2) select categories with top n prediction scores (the empirical assumption here we make is that the categories with higher prediction scores have more similarities with the target category). As the results reported in Figure 4 (left), while CAMs of classes with higher prediction scores bring more improvement, others are also beneficial for CAT. Besides, we further perform CAT on the reduced CIFAR-100, where CAMs of all classes are transferred but the training set is reduced to contain data of only partial categories. Then the trained model is evaluated on the complete test set and a subset of it which only contains data of classes that are not contained in the training set. As the results reported in Figure 4 (right), interestingly, the trained model achieves high accuracy on the subset, indicating that **transferring CAMs enables the trained model to classify the categories that are not contained in the training set**. This further proves that non-target CAMs contain beneficial information for CAT even if their categories seem to be irrelevant to the input from a human perspective.

Transferring smaller CAMs performs better. Intuitively, larger CAM contains more detailed hints about the spatial location of the class discriminative regions, then transferring larger CAMs should perform better. However, insufficient accuracy of the model will result in deviations

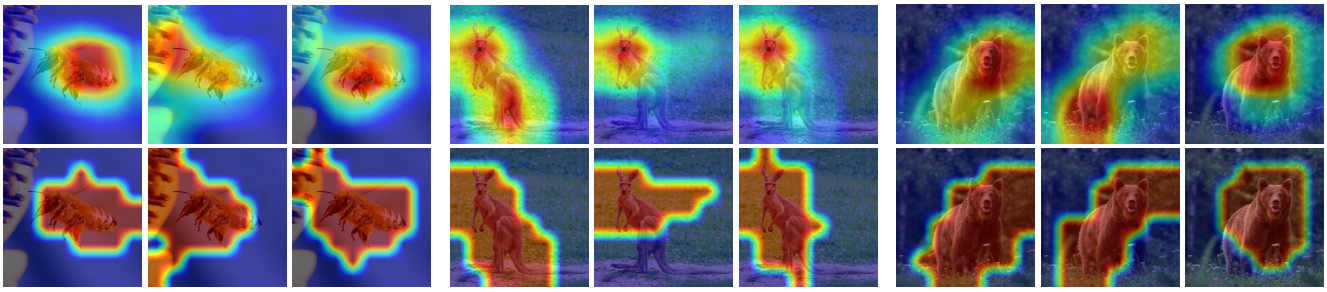

Figure 6. The first row shows the visualization of the CAMs corresponding to the top 3 predicted categories, while the following row shows the visualization of them after binarization.

CAM Producer	ResNet56	ResNet110	ResNet50
Acc	72.34	74.31	79.34
CAT	72.47	74.42	76.17

Table 1. Accuracy (%) of ShuffleNetV1 trained with CAT on CIFAR-100. The transferred CAMs are produced by different models with various accuracy.

Baseline	Model	ResNet32×4	ResNet8×4	ResNet20
	Acc	79.42	72.5	69.06
CAT	8×8	79.65	67.92	66.21
	4×4	**79.84**	71.61	66.43
	2×2	79.71	**72.45**	**66.84**

Table 2. Accuracy (%) of various models trained with CAT on the CIFAR-100 test set. During CAT, CAMs are pooled into various sizes. The transferred CAMs are produced by ResNet32×4.

between the highlighted areas in its generated CAM and the actual class discriminative regions of the image (which can be observed in Figure 2). Besides, different models differ in their capacity to identify class discriminative regions, which will lead to subtle differences in the generated CAMs. Therefore transferring CAMs with a larger size does not necessarily improve the performance of CAT. Through experiments, we found that performing average pooling on the transferred CAMs, which will expand the highlighted areas of CAMs and reduce the bias between CAMs generated by different models, could alleviate the above issues. As the results reported in Table 2, though pooling blurs the details, transferring smaller CAMs always performs better. Besides, since the pooling operation expands the highlighted areas of CAM, which will make it encompass larger class discriminative regions, transferring pooled CAMs will force the trained model to pay attention to more discriminative regions, which can be observed in Figure 5. In practice, we pool the transferred CAMs into a smaller size to improve the performance of CAT and CAT-KD (normally 2×2).

Baseline	Model	ResNet32×4	ResNet50
	Acc	79.42	79.34
CAT	CAMs	79.71	80.45
	Binarized CAMs	79.35	79.65

Table 3. Results of transferring binarized CAMs. The transferred CAMs are produced by ResNet32×4.

Teacher	ResNet32×4		
Acc	77.51	79.42	81.36
ReviewKD [2]	76.42	<u>77.45</u>	<u>77.91</u>
DKD [33]	<u>76.58</u>	76.45	77.29
CAT-KD	76.36	78.26	78.84
Δ	-0.22	+0.81	+0.93

Table 4. Comparison with two SOTA methods. The student network is ShuffleNetV1. Δ represents the gap between CAT-KD and the best-performing method among ReviewKD and DKD (marked with underline).

The exact value of the transferred CAMs is not important. To demonstrate that the role CAMs play in CAT is offering hints about the spatial location of the class discriminative regions of input, we binarize the values of the transferred CAMs to 0 and 1, using their average values as the thresholds. The regions of CAM with values above the threshold are considered as being highlighted, indicating the class discriminative regions of input. Thus, we set the values of these regions to 1 to keep them activated after the binarization. Other regions with values below the threshold are considered unhighlighted, and their values are set to 0. As shown in Figure 6, though the specific values of CAMs are lost during the binarization process, the binarized CAMs still contain hints about the spatial location of the class discriminative regions. Note that the threshold can also be specified in other ways (e.g., median).

As the results reported in Table 3, although the class discriminative regions obtained by our rudimentary binarization method are not precise, the accuracy of the resulting model dropped by less than one percent, proving that the

Distillation Mechanism	Teacher	ResNet32×4	WRN40-2	ResNet32×4	ResNet50	VGG13
	Acc	79.42	75.61	79.42	79.34	74.64
	Student	ShuffleNetV1	ShuffleNetV1	ShuffleNetV2	MobileNetV2	MobileNetV2
	Acc	70.5	70.5	71.82	64.6	64.6
Logits	KD [11]	74.07	74.83	74.45	67.35	67.37
	DKD [33]	76.45	76.7	77.07	70.35	69.71
Features	CRD [23]	75.11	76.05	75.65	69.11	69.73
	OFD [10]	75.98	75.85	76.82	69.04	69.48
	FitNet [19]	73.59	73.73	73.54	63.16	64.14
	RKD [17]	72.28	72.21	73.21	64.43	64.52
	ReviewKD [2]	77.45	77.14	77.78	69.89	**70.37**
Attention	AT [29]	71.73	73.32	72.73	58.58	59.4
	CAT-KD	**78.26**	**77.35**	**78.41**	**71.36**	69.13
	↑	+6.53	+4.03	+5.68	+12.78	+9.73

Table 5. Results on CIFAR-100. Teachers and students have different architectures. ↑ represents the performance improvement of CAT-KD compared with AT.

Distillation Mechanism	Teacher	ResNet56	ResNet110	ResNet32×4	WRN-40-2	WRN-40-2	VGG13
	Acc	72.34	74.31	79.42	75.61	75.61	74.64
	Student	ResNet20	ResNet32	ResNet8×4	WRN-16-2	WRN-40-1	VGG8
	Acc	69.06	71.14	72.5	73.26	71.98	70.36
Logits	KD [11]	70.66	73.08	73.33	74.92	73.54	72.98
	DKD [33]	**71.97**	**74.11**	76.32	**76.24**	74.81	74.68
Features	CRD [23]	71.16	73.48	75.51	75.48	74.14	73.94
	OFD [10]	70.98	73.23	74.95	75.24	74.33	73.95
	FitNet [19]	69.21	71.06	73.5	73.58	72.24	71.02
	RKD [17]	69.61	71.82	71.9	73.35	72.22	71.48
	ReviewKD [2]	71.89	73.89	75.63	76.12	**75.09**	**74.84**
Attention	AT [29]	70.55	72.31	73.44	74.08	72.77	71.43
	CAT-KD	71.62	73.62	**76.91**	75.6	74.82	74.65
	↑	+1.07	+1.31	+3.47	+1.52	+2.05	+3.22

Table 6. Results on CIFAR-100. Teachers and students have the same architecture. ↑ represents the performance improvement of CAT-KD compared with AT.

critical information CAMs contained for CAT is the spatial location of class discriminative regions rather than its exact value. This strongly demonstrates that our method is based on transferring attention.

4.3. Evaluation of CAT-KD

Consistent with previous works [2, 23, 33], we compare the performance of CAT-KD with several representative KD methods. Moreover, we further evaluate our method from two aspects: transferability and efficiency.

Results on CIFAR-100. Table 5 reports the results on CIFAR-100 with the teachers and students having different architectures. Table 6 shows the results where teachers and students have architectures of the same style. Notably, our method outperforms the other attention-based method AT [29] with a large margin (1.07% $\sim$ 12.78%). Moreover, CAT-KD achieves comparable or even better performance compared with feature-based distillation method [2] which requires additional networks and multiple-layer information. Besides, consistent with CAT, the performance of CAT-KD is affected by the accuracy of the teacher: CAMs produced by teachers with lower accuracy contain more incorrect hints about the class discriminative regions of input. To verify this, we further evaluate the impact of the accuracy of the teacher on our method. As the results reported in Table 4, CAT-KD is relatively less effective when the teacher is weak. Thus, as can be observed in Table 6, the performance of CAT-KD is not the best when the teacher is weak.

Results on ImageNet. Table 7 and Table 8 report the top-1 and top-5 accuracy of image classification on ImageNet. Though the performance of CAT-KD is restricted by the weakness of the teacher network in this setting, our method still outperforms most KD methods.

			Features			Logits		Attention	
	Teacher	Student	OFD [10]	CRD [23]	ReviewKD [2]	KD [11]	DKD [33]	AT [29]	CAT-KD
Top-1	73.31	69.75	70.81	71.17	<u>71.61</u>	70.66	**71.7**	70.69	71.26
Top-5	91.41	89.07	89.98	90.13	**90.51**	89.88	90.41	90.01	<u>90.45</u>

Table 7. Results on ImageNet. In this group, we set ResNet34 as the teacher and ResNet18 as the student. The method with the second-best performance is marked with an underline.

			Features			Logits		Attention	
	Teacher	Student	OFD [10]	CRD [23]	ReviewKD [2]	KD [11]	DKD [33]	AT [29]	CAT-KD
Top-1	76.16	68.87	71.25	71.37	**72.56**	68.58	72.05	69.56	<u>72.24</u>
Top-5	92.86	88.76	90.34	90.41	91.00	88.98	<u>91.05</u>	89.33	**91.13**

Table 8. Results on ImageNet. In this group, we set ResNet50 as the teacher and MobileNet as the student. The method with the second-best performance is marked with an underline.

Teacher	ResNet32×4		ResNet50	
Student	ShuffleNetV1		MobileNetV2	
Dataset	STL	TI	STL	TI
Baseline	69.05	36.54	64.39	30.85
KD [11]	66.61	32.56	67.81	32.37
DKD [33]	70.73	36.77	71.05	36.48
CRD [23]	70.68	37.85	71.46	38.75
ReviewKD [2]	71.46	38.46	66.16	32.65
AT [29]	71.36	37.36	65.1	29.13
CAT-KD	**74.43**	**40.73**	**73.2**	**39.87**

Table 9. Comparison on transferring representations learned from CIFAR-100 to STL-10 (STL) and Tiny-ImageNet (TI).

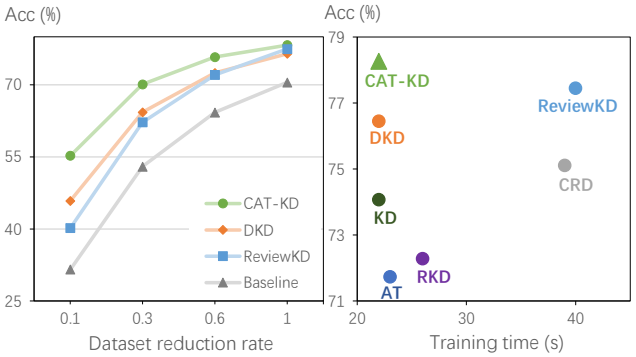

Figure 7. We set ResNet32×4 as the teacher and ShuffleNetV1 as the student. Left: accuracy of students trained with various methods on CIFAR-100, where the training set is reduced at various ratios. Right: comparison of accuracy and training time (per epoch) on CIFAR-100.

Transferability. We perform experiments to compare the transferability of representations to evaluate the generalizability of the *knowledge* transferred by various methods. We use ShuffleNetV1 and MobileNetV2 as the frozen representations extractors, which are either trained from scratch on CIFAR-100 [13] or distilled from ResNet32×4 and ResNet50 with various KD methods. Then linear probing tasks are performed on STL-10 [4] and Tiny-ImageNet [5] to quantify their transferability. As the results reported in Table 9, CAT-KD outperforms other methods by a large margin, indicating the outstanding generalizability of the *knowledge* transferred by our method.

Efficiency. We first compare the performance of multiple KD methods on CIFAR-100, where the training set is reduced at various ratios, to evaluate their dependence on the amount of training data. As the results reported in Figure 7 (left), CAT-KD is minimally affected by the decrease in the amount of training data, proving the outstanding distillation efficiency of our method. Besides, we further compare the training cost and performance of various KD methods. As reflected in the results reported in Figure 7 (right), CAT-KD has the highest training efficiency. Since CAT-KD does not require extra parameters, its computational cost is almost the same as logits-based methods. Relatively, feature-based methods require much more computational resources because most of them need additional auxiliary networks to distill features.

5. Conclusion

In this paper, we propose CAT-KD which has both high interpretability and competitive performance. More importantly, we demonstrate that the capacity of identifying class discriminative regions of input can be obtained and enhanced by transferring CAMs. Furthermore, we present several interesting properties of transferring CAMs, which contribute to a better understanding of CNN. We hope our findings will help future research on the interpretability of CNN and knowledge distillation.

Acknowledgement. We thank the reviewers for their constructive feedback. Part of Hui Li's work is supported by the National Natural Science Foundation of China (61932015), Shaanxi Innovation Team project (2018TD-007), Higher Education Discipline Innovation 111 project (B16037). Part of Haonan Yan's work is done when he visits the University of Guelph.

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. 3

[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. 1, 2, 4, 6, 7, 8

[3] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. *international conference on computer vision*, 2019. 1, 2

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 8

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *computer vision and pattern recognition*, 2009. 4, 8

[6] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Animashree Anandkumar. Born again neural networks. *international conference on machine learning*, 2018. 1, 2

[7] Hao Guo, Kang Zheng, Xiaochuan Fan, Hongkai Yu, and Song Wang. Visual attention consistency under image transforms for multi-label image classification. *computer vision and pattern recognition*, 2019. 3

[8] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. *computer vision and pattern recognition*, 2016. 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv: Computer Vision and Pattern Recognition*, 2015. 3, 4

[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. 1, 2, 7, 8

[11] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv: Machine Learning*, 2015. 1, 2, 7, 8

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *computer vision and pattern recognition*, 2016. 3

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4, 8

[14] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. *computer vision and pattern recognition*, 2018. 3

[15] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. *european conference on computer vision*, 2018. 4

[16] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. *national conference on artificial intelligence*, 2019. 1, 2

[17] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. *computer vision and pattern recognition*, 2019. 1, 2, 7

[18] Baoyun Peng, Xiao Jin, Dongsheng Li, Shunfeng Zhou, Yichao Wu, Jiaheng Liu, Zhaoning Zhang, and Yu Liu. Correlation congruence for knowledge distillation. *international conference on computer vision*, 2019. 1, 2

[19] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv: Learning*, 2014. 1, 2, 7

[20] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *computer vision and pattern recognition*, 2018. 4

[21] 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. 3

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *computer vision and pattern recognition*, 2014. 4

[23] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 1, 2, 7, 8

[24] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. *arXiv: Computer Vision and Pattern Recognition*, 2019. 1, 2

[25] Chaofei Wang, Jiayu Xiao, Yizeng Han, Qisen Yang, Shiji Song, and Gao Huang. Towards learning spatially discriminative feature representations. *international conference on computer vision*, 2021. 3

[26] 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. 3

[27] Seunghan Yang, YoonHyung Kim, Youngeun Kim, and Changick Kim. Combinational class activation maps for weakly supervised object localization. *workshop on applications of computer vision*, 2019. 3

[28] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. *computer vision and pattern recognition*, 2017. 1, 2

[29] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. *Learning*, 2016. 1, 2, 7, 8

[30] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *british machine vision conference*, 2016. 4

[31] Xiaopeng Zhang, Jiashi Feng, Hongkai Xiong, and Qi Tian. Zigzag learning for weakly supervised object detection. *computer vision and pattern recognition*, 2018. 3

[32] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *computer vision and pattern recognition*, 2017. 4

[33] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. *arXiv preprint arXiv:2203.08679*, 2022. 1, 2, 4, 6, 7, 8

[34] 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, 3, 5