BLACK-BOX KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

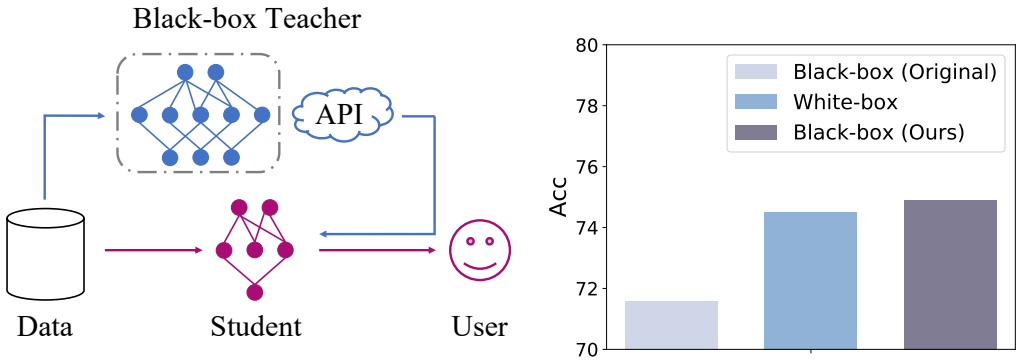
Knowledge Distillation (KD) aims at distilling the knowledge from the large teacher model to a light-weight student model. Enhancing model efficiency effectively, mainstream methods often rely on the assumption that the teacher model is *white-box* (*i.e.*, visible during distillation). However, this assumption does not always hold due to commercial, privacy, or safety concerns, which hinders these strong methods from being applied. Towards this dilemma, in this paper, we consider **black-box knowledge distillation**, an interesting yet challenging problem which aims at distilling teacher knowledge when merely the teacher predictions are accessible (*i.e.*, the teacher model is invisible). Some early KD methods can be directly applied to black-box knowledge distillation, but the performance appears to be unsatisfactory. In this paper, we propose a simple yet effective approach, which makes better utilization of teacher predictions with **prediction augmentation** and **multi-level prediction alignment**. Through this framework, the student model learns from more diverse teacher predictions. Meanwhile, the prediction alignment is not only conducted at the *instance* level, but also at the *batch* and *class* level, through which the student model learns instance prediction, input correlation, and category correlation simultaneously. Extensive experiment results validate that our method enjoys consistently higher performance than previous black-box methods, and even reaches competitive performance with mainstream white-box methods. We promise to release our code and models to ensure reproducibility.

1 INTRODUCTION

The last few decades have witnessed the prosperity of deep learning in computer vision tasks, such as image classification Krizhevsky et al. (2012); Simonyan & Zisserman (2015); He et al. (2016); Dosovitskiy et al. (2021), object detection Ren et al. (2015), and segmentation Shelhamer et al. (2016); Zhao et al. (2017). However, due to their overwhelming large model size, many deep models rely heavily on computation and storage resources, which makes it nearly impossible to deploy them in some practical scenarios, such as mobile devices. Towards this dilemma, Knowledge Distillation Hinton et al. (2015) (KD) was introduced to reduce model capacity. Concretely, the KD framework consists of one teacher model (large) and one student model (small). The main objective of KD is to distill the knowledge in the teacher model to the light-weight student model, which is ready to be deployed. Various KD methods Romero et al. (2015); Park et al. (2019); Tian et al. (2020); Heo et al. (2019a); Chen et al. (2021) have been proposed and proved to be effective.

Among them, the earliest method Hinton et al. (2015) distills knowledge by reducing the divergence of predictions between the teacher and student model, where distillation is implemented merely on the logit level. Towards better utilization of teacher knowledge, recent researches Romero et al. (2015); Chen et al. (2021) shed light on the intermediate layers in the teacher model, conducting distillation by matching feature distributions among the teacher and student model. These feature-level KD methods boast superior performance than the original logit-level method. Up till now, the majority of mainstream KD methods are feature-level ones.

Though achieving great successes, these feature-level methods often assume that the teacher model is in a white-box (*i.e.*, visible during the whole process of knowledge distillation). Such an assumption enables them to distill feature knowledge in the teacher model, but it is not always valid in real-world applications. Due to commercial, privacy, and safety concerns, some big models are



(a) Problem Setup (b) Method Performance

Figure 1: **Problem Setup and Method Performance.** **(a)** In black-box knowledge distillation, the large teacher model (blue) is invisible, with only an API available. The user can query data through the API to gain the corresponding model predictions, and then utilize them to distill the teacher knowledge to the light-weight student model (purple). **(b)** Our proposed method surpasses original method in black-box scenario. The performance is also competitive over previous white-box methods that utilize intermediate features to distill knowledge.

provided in black-box, where users can only have access to the model predictions. The feature-level KD methods become invalid in these scenarios since the features in the teacher model are invisible.

In this paper, we consider this interesting yet challenging problem, **black-box knowledge distillation**. Figure 1(a) illustrates black-box knowledge distillation, where merely a model API is provided to generate predictions. The user can query the API, gain corresponding predictions, and utilize them to distill the knowledge to the student model. For clarity, we compare black-box knowledge distillation with traditional white-box knowledge distillation in Table 1. In white-box knowledge distillation, since the whole teacher model is accessible, the features in intermediate layers can be extracted for distillation, which gives birth to previous feature-level methods. However, in black-box knowledge distillation, the teacher model is unaccessible. Towards this setting, considering that the features in teacher models are invisible, we state that attention should be paid to another side of the coin, distilling knowledge with predictions (logits) alone. The previous logit-level KD methods can be applied naturally, but the performance is unsatisfactory.

Table 1: **Comparison of different settings.** Compared with traditional white-box knowledge distillation, the teacher model is invisible in black-box knowledge distillation.

Setting	Student Model	Teacher Model	Teacher Prediction
White-box KD	✓	✓	✓
Black-box KD	✓	✗	✓

To make better use of the teacher predictions, in this paper, we propose a simple yet effective approach to black-box knowledge distillation, which absorbs more information from teacher models via **prediction augmentation** and **multi-level alignment**. Concretely, we apply augmentations to model predictions and reduce the divergence of predictions between the teacher and student at the instance, batch, and class level. Through the multi-level alignment, the student model absorbs knowledge from the teacher model not only in *instance*-level prediction, but in *batch*-level input correlation and *class*-level category correlation as well, boosting knowledge distillation from the teacher model to the student model with merely teacher predictions.

Extensive experiment results on mainstream benchmarks validate that our method surpasses the previous black-box KD methods. In addition, our method even reaches competitive performance over previous white-box methods, in both homogenous and heterogeneous network knowledge distillation settings. For instance, as shown in Figure 1(b), our method outperforms the original KD

method remarkably in black-box scenarios. Meanwhile, our method even performs slightly better than previous while-box methods, proving that our method excels at utilizing teacher predictions.

2 Related Work

Proposed in Hinton et al. (2015), Knowledge Distillation (KD) defines a new model compression framework. It consists of one large teacher model and one light-weight student model and aims at distilling (transferring) the knowledge in the teacher model to the student model. Concretely, it forces the student model to mimic the teacher outputs by minimizing the divergence between the predictions from the teacher and student model. Towards the over-confidence / miscalibration phenomenon Guo et al. (2017) in neural networks, temperature rescaling is applied to alleviate the influence. In our method, we also implement prediction augmentations by incorporating multiple temperatures.

Upon proposal, various methods have been proposed for knowledge distillation. These methods fall into two lines of work: 1) logit-level methods Cho & Hariharan (2019); Furlanello et al. (2018); Mirzadeh et al. (2020); Yang et al. (2019); Zhang et al. (2018b) and 2) feature-level methods Heo et al. (2019a;b); Huang & Wang (2017); Kim et al. (2018); Park et al. (2019); Peng et al. (2019); Romero et al. (2015); Tian et al. (2020); Tung & Mori (2019); Yim et al. (2017); Zagoruyko & Komodakis (2017).

Table 2: **Comparison of different settings.** Compared with previous logit-level and feature-level KD methods, our method conducts prediction alignment in instance-level, batch-level and category-level simultaneously.

Setting	Instance-level Alignment	Batch-level Alignment	Category-level Alignment
Logit-level KD (Previous)	✓	✗	✗
Feature-level KD	✓	✓	✗
Logit-level KD (Ours)	✓	✓	✓

Logit-Level KD Logit-level KD methods distill knowledge merely with output logits. For instance, the earliest KD method is a logit-level method. Other logit-level methods boost knowledge distillation by introducing a mutual-learning paradigm Zhang et al. (2018b) or additional teacher assistant module Mirzadeh et al. (2020). The logit-level methods appear to be straightforward and are ready to be applied to any scenario, no matter black-box or white-box. However, their performance is often inferior to feature-level methods.

Feature-Level KD To further boost knowledge distillation, another line of works, feature-level KD, are proposed to conduct distillation on intermediate features. Concretely, some of them Heo et al. (2019a;b); Romero et al. (2015) mitigate the divergence between features in the teacher and student model, which enforces the student model to imitate the teacher model at feature level. Other methods Park et al. (2019); Tian et al. (2020); Tung & Mori (2019) also convey teacher knowledge by distilling the input correlation. We note that feature-level KD methods are unable to tackle black-box scenarios and compare them with our method in Table 2. Different from these feature-level methods that transfer input correlation via intermediate features, our black-box distillation method learns input correlation by logit outputs. Our method also absorbs class correlation, which previous works rarely pay attention to.

3 Methodology

To smooth the presentation, we start from preliminaries. Then we introduce our approach to black-box knowledge distillation.

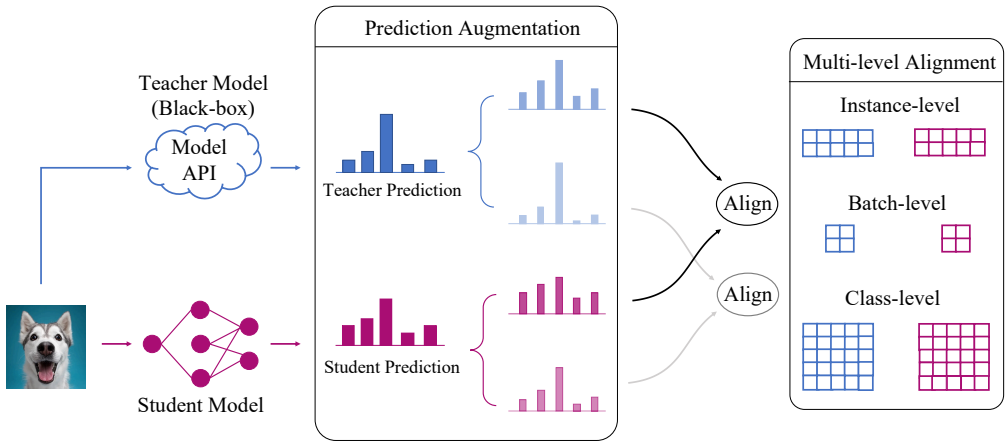


Figure 2: **Method Overview.** In black-box knowledge distillation, the teacher model is invisible, with merely a model API available. After obtaining the teacher and student predictions, we conduct **prediction augmentation**, converting them to multiple outputs with different temperatures respectively. The augmented predictions are matched respectively through **multi-level alignment**, which consists of instance-level, batch-level, and class-level alignment. We take batch size $B = 2$ and class number $C = 5$ as an example to demostrate our multi-level alignment. *(Best viewed in color)*

3.1 PRELIMINARIES

Knowledge Distillation We start from the original Knowledge Distillation (KD) method, which was proposed in Hinton et al. (2015). To illustrate the procedure of KD, we consider C-way classification task and denote the logit output of a single input as $z \in \mathbb{R}^C$, then the class probability is

$$p_j = \frac{e^{z_j/T}}{\sum_{c=1}^{C} e^{z_c/T}}, \tag{1}$$

where p_j and z_j is the probability value on the j-th class. We compute the Softmax value and T is the temperature scaling hyper-parameter Guo et al. (2017). In knowledge distillation, T is often larger than 1.0, which alleviates the over-confidence phenomenon in neural network Guo et al. (2017). When $T = 1.0$, the output will shrink to vanilla Softmax output.

The objective of KD is to distill the knowledge from the large teacher model to the light-weight student model. With rescaled outputs, the original KD method implements distillation by minimizing the KL divergence between the outputs from the teacher and student model,

$$L_{KD} = KL(p^{tea}||p^{stu}) = \sum_{j=1}^{C} p_j^{tea} log(\frac{p_j^{tea}}{p_j^{stu}}), \tag{2}$$

where L_{KD} is the knowledge distillation loss, p_j^{tea} and p_j^{stu} indicates the probability value on the j-th category of the teacher and student output, respectively.

The original KD method, minimizing the divergence on logit outputs, serves as the most fundamental baseline in KD research. Meanwhile, when confronting black-box scenarios, this method can be applied directly, but the performance is unsatisfactory. In this paper, we strive to seek a stronger method for black-box KD.

3.2 BLACK-BOX KNOWLEDGE DISTILLATION

In this section, we will introduce our approach to black-box knowledge distillation. Here, we consider the output of a batch of data instead of a single data. We denote the logit output as $z \in \mathbb{R}^{B\times C}$, where B is the batch size and C means C-way classification. Our method has two core components: 1) prediction augmentation and 2) multi-level alignment.

3.2.1 PREDICTION AUGMENTATION

To gain richer knowledge from predictions, we propose a prediction augmentation mechanism, through which we can expand a single output to multiple ones. Concretely, we conduct prediction augmentation through temperature rescaling,

$$p_{i,j,k} = \frac{e^{z_{i,j}/T_k}}{\sum_{c=1}^{K} e^{z_{i,c}/T_k}}, \tag{3}$$

where $p_{i,j,k}$ is the probability value of the i-th input on the j-th category, with temperature hyper-parameter T_k. In our mechanism, $T_0, T_1, ..., T_K$ forms a pool with K temperatures, which enables us to augment one prediction to K diverse outputs.

As shown in Figure 2, take $K = 2$ as an instance, outputs from the teacher and student model are augmented respectively. Through the prediction augmentation mechanism, we convert one prediction to K outputs that are diverse in probability sharpness.

3.2.2 MULTI-LEVEL ALIGNMENT

With augmented predictions, as shown in Figure 2, we propose to align the teacher output and the corresponding student output (according to the temperature) one by one. Instead of the original logit alignment through KL divergence, we propose a novel multi-level alignment, which includes 1) instance-level, 2) batch-level, and 3) class-level alignment.

Instance-level Alignment We inherit the original mechanism in KD to implement instance-level alignment in our method. Concretely, as shown below, we minimize the KL divergence between augmented predictions from the teacher and student model one by one,

$$L_{ins} = \sum_{i=1}^{N}\sum_{k=1}^{K} KL(p_{i,k}^{tea}||p_{i,k}^{stu}) = \sum_{i=1}^{N}\sum_{k=1}^{K}\sum_{j=1}^{C} p_{i,j,k}^{tea} log(\frac{p_{i,j,k}^{tea}}{p_{i,j,k}^{stu}}), \tag{4}$$

where L_{ins} means the instance-level alignment loss, $p_{i,j,k}^{tea}$ and $p_{i,j,k}^{stu}$ indicate the teacher and student outputs on the i-th instance, j-th category, that are augmented by T_k. The instance-level alignment forces the student model to mimic the teacher predictions on each instance, which plays the most fundamental role in knowledge distillation. When compared with the vanilla KD Hinton et al. (2015) method, the core difference of our alignment is that we adopt prediction augmentation by temperature rescaling, which transfers more diverse knowledge from the teacher model to the student model.

Batch-level Alignment Instead of aligning predictions at merely instance level, we propose to conduct batch-level alignment by input correlation, the relation between two inputs, which is modeled via features in previous work. In our method, we take logit predictions to quantify it. Specifically, we compute the Gram Matrix on the model predictions as follows,

$$G^k = p_k p_k^T, G_{ab}^k = \sum_{j=1}^{C} p_{a,j,k} \cdot p_{b,j,k}, \tag{5}$$

where G^k is a $B \times B$ matrix, and p_k indicates the predictions obtained via T_k. We can derive that G_{ab}^k models the probability that the a-th and the b-th input are classified in the same category, which indicates the relationship between them.

Then we compute the input correlation matrix G_k according to different T_k, with the teacher and student predictions respectively. Our objective is to mitigate the divergence between them, thus

$$L_{batch} = \frac{1}{B}\sum_{k=1}^{K} ||G_{tea}^k - G_{stu}^k||_2^2, \tag{6}$$

where L_{batch} serves as the batch-level alignment loss, G_{tea}^k and G_{stu}^k are the input correlation matrix computed by teacher and student predictions with temperature T_k, respectively. Similarly, with instance-level alignment, we take all augmented predictions and conduct alignment accordingly.

Algorithm 1: Pseudo code of DKD in a PyTorch-like style.

```
# z_stu, z_tea: student, teacher logit outputs, B x C
# T = [T_1, T_2, ..., T_K]: one set of K different temperatures
# l_ins, l_batch, l_class: three parts of alignment loss
# l_total: total loss
l_total = 0
for t in T do
    p_stu = F.softmax(z_stu / t) # B x C
    p_tea = F.softmax(z_tea / t) # B x C
    l_ins = F.kl_div(p_tea, p_stu)
    G_stu = torch.mm(p_stu, p_stu.t()) # B x B
    G_tea = torch.mm(p_tea, p_tea.t()) # B x B
    l_batch = ((G_stu - G_tea) ** 2).sum() / B
    M_stu = torch.mm(p_stu.t(), p_stu) # C x C
    M_tea = torch.mm(p_tea.t(), p_tea) # C x C
    l_class = ((M_stu - M_tea) ** 2).sum() / C
    l_total += (l_ins + l_batch + l_class)
end
```

Class-level Alignment The last part of our method lies in class-level alignment. We state that the model predictions can depict the relationship between categories, *i.e.*, if one class is very similar to the ground-truth class, the model is prone to be reluctant between them, forming two high peaks in predictions. Such a category correlation can be modeled by predictions of a batch of data as follows,

$$M^k = p_k^T p_k, M_{ab}^k = \sum_{i=1}^{N} p_{i,a,k} \cdot p_{i,b,k}, \tag{7}$$

where M^k is a $C \times C$ matrix, p_k indicates the predictions obtained via T_k, and M_{ab}^k presents the probability that the inputs in this batch are classified to the a-th category and the b-th category simultaneously, which quantifies the relationship between the two classes.

After quantifying the category correlation, we can enforce the student model to absorb this part of knowledge from the teacher model by the following loss,

$$L_{class} = \frac{1}{C} \sum_{k=1}^{K} ||M_{tea}^k - M_{stu}^k||_2^2 \tag{8}$$

where L_{class} serves as the class-level alignment loss, M_{tea}^k and M_{stu}^k are the category correlation matrix computed by teacher and student predictions with temperature hyper-parameter T_k. Augmented predictions with multiple temperatures alleviate the over-confidence phenomenon in neural networks, which is crucial in modeling the category correlation.

Multi-level Alignemt Now we have designed the mechanism for instance-level, batch-level, and class-level alignment and can formulate our multi-level alignment loss as follows,

$$L_{total} = L_{ins} + L_{batch} + L_{class}. \tag{9}$$

By integrating three parts of loss together, our method enforces the student model to imitate the teacher model not only in instance-level predicitons, but in batch-level input correlation and class-level category correlation as well. We provide the pseudo code in Algorithm 1.

4 Experiments

4.1 Datasets and Settings

In our experiments, we evaluate the performance of our method on image classification and objection detection respectively.

Datasets We take three widely researched datasets, CIFAR-100 Krizhevsky et al. (2009), and ImageNet Russakovsky et al. (2015) for image classification, and MS-COCO Lin et al. (2014) for object detection. More detailed descriptions are included in supplementary materials.

Settings We focus on black-box knowledge distillation with two different settings in our experiment section. 1) Homogenous architecture where the teacher and student model are in the same type of architecture (*e.g.* ResNet56 and ResNet20), and 2) Heterogeneous architecture where the two models are different in architecture (*e.g.* ResNet32x4 and ShuffleNet-V1). We include various neural network architectures in our experiment, including ResNet He et al. (2016), WRN Zagoruyko & Komodakis (2016), VGG Simonyan & Zisserman (2015), ShuffleNet-V1 Zhang et al. (2018a)/V2 Ma et al. (2018) and MobileNetV2 Sandler et al. (2018).

4.2 Experimental Results

In our experiments, we evaluate the performance of our method in the black-box knowledge distillation scenario, where the features in teacher models are all inaccessible. We also report the performance of other white-box methods Romero et al. (2015); Park et al. (2019); Tian et al. (2020); Heo et al. (2019a); Chen et al. (2021). We note that such a comparison is unfair since for these methods, all the intermediate layers and features in the teacher model are available.

Table 3: **Results on CIFAR-100, Homogenous Architecture.** Top-1 accuracy is adopted as the evaluation metric. The teacher model and student model are in homogenous architecture and their original performance is reported respectively.

Method	Teacher / Student	ResNet56 72.34 ResNet20 69.06	ResNet110 74.31 ResNet32 71.14	ResNet32×4 79.42 ResNet8×4 72.50	WRN-40-2 75.61 WRN-16-2 73.26	WRN-40-2 75.61 WRN-40-1 71.98	VGG13 74.64 VGG8 70.36	Avg
White-box	FitNet Romero et al. (2015)	69.21	71.06	73.50	73.58	72.24	71.02	71.77
	RKD Park et al. (2019)	69.61	71.82	71.90	73.35	72.22	71.48	71.73
	CRD Tian et al. (2020)	71.16	73.48	75.51	75.48	74.14	73.94	73.95
	OFD Heo et al. (2019a)	70.98	73.23	74.95	75.24	74.33	73.95	73.78
	ReviewKD Chen et al. (2021)	71.89	73.89	75.63	76.12	75.09	74.84	74.58
Black-box	KD Hinton et al. (2015)	70.66	73.08	73.33	74.92	73.54	72.98	73.09
	DML Zhang et al. (2018b)	69.52	72.03	72.12	73.58	72.68	71.79	71.95
	TAKD Mirzadeh et al. (2020)	70.83	73.37	73.81	75.12	73.78	73.23	73.36
	Ours	**72.19**	**74.11**	**77.08**	**76.63**	**75.35**	**75.18**	**75.09**

Table 4: **Results on CIFAR-100, Heterogeneous Architecture.** Top-1 accuracy is adopted as the evaluation metric. The teacher model and student model are in heterogeneous architecture and their original performance is reported respectively.

Method	Teacher / Student	ResNet32×4 79.42 ShuffleNet-V1 70.50	WRN-40-2 75.61 ShuffleNet-V1 70.50	VGG13 74.64 MobileNet-V2 64.60	ResNet50 79.34 MobileNet-V2 64.60	ResNet32×4 79.42 ShuffleNet-V2 71.82	Avg
White-box	FitNet Romero et al. (2015)	73.59	73.73	64.14	63.16	73.54	69.63
	RKD Park et al. (2019)	72.28	72.21	64.52	64.43	73.21	69.33
	CRD Tian et al. (2020)	75.11	76.05	69.73	69.11	75.65	73.13
	OFD Heo et al. (2019a)	75.98	75.85	69.48	69.04	76.82	73.43
	ReviewKD Chen et al. (2021)	**77.45**	77.14	70.37	69.89	77.78	74.53
Black-box	KD Hinton et al. (2015)	74.07	74.83	67.37	67.35	74.45	71.60
	DML Zhang et al. (2018b)	72.89	72.76	65.63	65.71	73.45	70.09
	TAKD Mirzadeh et al. (2020)	74.53	75.34	67.91	68.02	74.82	72.12
	Ours	77.18	**77.44**	**70.57**	**71.04**	**78.44**	**74.93**

CIFAR-100 We evaluate our method on CIFAR-100 and compare it with previous methods. For knowledge distillation where the teacher and student model are in homogenous architecture, as shown in Table 3, our method performs best among black-box methods, showing obvious improvements over the original student model and the vanilla KD Hinton et al. (2015) method. Moreover, our accuracy is slightly better than the white-box knowledge distillation methods. We note that it

validates the strong effectiveness of our method, since it only takes output predictions to surpass all the methods that absorb abundant knowledge from intermediate features.

When it comes to the situation where the teacher and student model are heterogeneous in architecture, the results in Table 4 demonstrate that our method shows a remarkable advantage over the previous black-box KD methods, enhancing the lightweight student model effectively. In addition, our method also shows competitive performance over the white-box methods.

Table 5: **Results on ImageNet.** Top-1 and Top-5 accuracy is adopted as the evaluation metric. The original accuracies of the teacher and student model are also reported.

		Top-1	Top-5	Top-1	Top-5
Method	Teacher	ResNet34		ResNet50	
		73.31	91.42	76.16	92.86
	Student	ResNet18		MobileNet-V2	
		69.75	89.07	68.87	88.76
White-box	AT Zagoruyko & Komodakis (2017)	70.69	90.01	69.56	89.33
	OFD Heo et al. (2019a)	70.81	89.98	71.25	90.34
	CRD Tian et al. (2020)	71.17	90.13	71.37	90.41
	ReviewKD Chen et al. (2021)	71.61	90.51	72.56	91.00
Black-box	KD Hinton et al. (2015)	70.66	89.88	68.58	88.98
	DML Zhang et al. (2018b)	70.82	90.02	71.35	90.31
	TAKD Mirzadeh et al. (2020)	70.78	90.16	70.82	90.01
	Ours	**71.62**	**90.55**	**73.01**	**91.42**

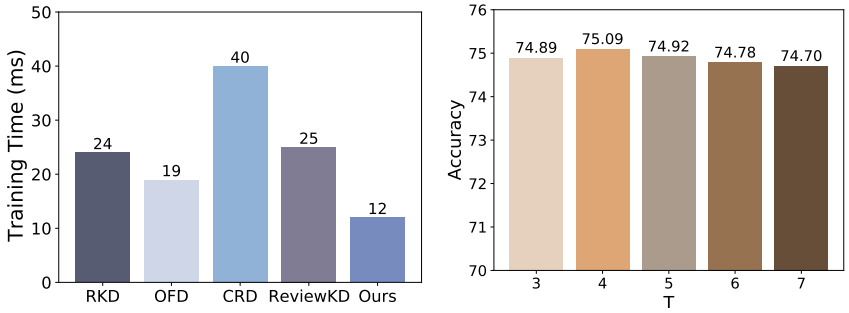
ImageNet We plug our method into black-box knowledge distillation on ImageNet, with teacher and student models in homogenous or heterogeneous architecture. We compare our method with previous methods that can tackle black-box scenarios, as well as present the performance of previous white-box methods. We report both Top-1 and Top-5 accuracy in Table 5.

The results demonstrate that no matter the teacher and student models are homogenous (ResNet34 and ResNet18) or heterogeneous (ResNet50 and MobileNet-V2), our method consistently outperforms previous KD methods in black-box scenarios. In addition, our method still shows competitive performance over white-box methods on such a large-scale and complicated dataset.

MS-COCO We extend our experiments to objection detection, another fundamental computer vision task. We take Faster-RCNN Ren et al. (2015)-FPN Lin et al. (2017) as the backbone, and AP, AP_{50}, and AP_{75} as the evaluation metric. The results in Table 6 validate that our method is steadily superior to mainstream KD methods and enjoys strong performance over previous white-box methods.

Table 6: **Results on MS-COCO.** We take Faster-RCNNRen et al. (2015)-FPNLin et al. (2017) as the backbone, and AP, AP_{50}, and AP_{75} as the evaluation metric. The original accuracies of the teacher and student model are also reported.

		AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Method	Teacher	ResNet101			ResNet101			ResNet50		
		42.04	62.48	45.88	42.04	62.48	45.88	40.22	61.02	43.81
	Student	ResNet18			ResNet50			MobileNetV2		
		33.26	53.61	35.26	37.93	58.84	41.05	29.47	48.87	30.90
White-box	FitNet Romero et al. (2015)	34.13	54.16	36.71	38.76	59.62	41.80	30.20	49.80	31.69
	FGFI Wang et al. (2019)	35.44	55.51	38.17	39.44	60.27	43.04	31.16	50.68	32.92
	ReviewKD Chen et al. (2021)	36.75	56.72	34.00	**40.36**	60.97	44.08	33.71	53.15	36.13
Black-box	KD Hinton et al. (2015)	33.97	54.66	36.62	38.35	59.41	41.71	30.13	50.28	31.35
	TAKD Mirzadeh et al. (2020)	34.59	55.35	37.12	39.01	60.32	43.10	31.26	51.03	33.46
	DKD **?**	35.05	56.60	37.54	39.25	60.90	42.73	32.34	53.77	34.01
	Ours	**37.03**	**57.68**	**40.01**	40.15	**61.67**	**44.57**	**34.83**	**55.01**	**36.82**



(a) Training Time (b) Hyperparameter Sensitivity

Figure 3: **Training Time and Hyperparameter Sensitivity. (a)** Training time for each batch of data of different KD methods. Our method takes the shortest training time among them. **(b)** Our method performs steadily over different T hyperparameter. Both experiments are conducted on CIFAR-100, with teacher and student models in homogenous architecture.

4.3 Analyses

Ablation study We investigate the contributions of each component in our method, instance-level alignment, batch-level alignment, class-level alignment, and prediction augmentation. In Table 7, when merely instance-level alignment is adopted, the method shrinks to the original KD Hinton et al. (2015) method, while with all the four components, our method performs better than all the other variants, proving that each part of our method is indispensable.

Table 7: **Ablation Study.** The experients are conducted on CIFAR-100, with ResNet32x4 as the teacher model, ResNet8x4 as the student model, and Top-1 accuracy as the evaluation metric.

Instance-level Alignment	Batch-level Alignment	Class-level Alignment	Prediction Augmentation	Acc
✓	✗	✗	✗	73.33
✓	✓	✗	✗	74.58
✓	✓	✓	✗	76.26
✓	✓	✓	✓	**77.08**

Training Speed We compare the training speed of various KD methods by assessing the training time of each batch of data, on CIFAR-100, with teacher and student models in homogenous architecture. We can observe from Figure 3(a) that our method takes the shortest training time among previous methods. We conjecture that the reason is that our method takes merely the logit outputs to conduct knowledge distillation, while previous methods need more time and computational costs to distill the feature knowledge in intermediate layers.

Hyperparameter Sensitivity In our experiments, we set the median of temperatures as $T = 4.0$. Here, we conduct hyperparameter sensitivity on T. Following the our experiment settings, we take $K = 5$ temperatures with median $T = [3.0, 4.0, 5.0, 6.0, 7.0]$ and evaluate the model on CIFAR-100 with teacher and student models in homogeneous architecture respectively. The results are shown in Figure 3(b). Our method performs stably under different T hyperparameters.

5 Conclusion

In this paper, we consider black-box knowledge distillation, an interesting yet challenging problem where the teacher model is in black-box, leaving merely the logit outputs accessible. Towards this problem, we propose a novel approach to make better utilization of teacher outputs via prediction augmentations and multi-level alignment that consists of instance-level, batch-level, and class-level alignment. Extensive experiment results prove the effectiveness of our method.

REFERENCES

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *IEEE International Conference on Computer Vision*, 2019.

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. In *International Conference on Learning Representations*, 2021.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *IEEE International Conference on Computer Vision*, 2019a.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI Conference on Artificial Intelligence*, 2019b.

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

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

Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, 2018.

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: Practical guidelines for efficient cnn architecture design. In *European Conference on Computer Vision*, 2018.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *AAAI Conference on Artificial Intelligence*, 2020.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *International Conference on Learning Representations*, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobilenetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

K. Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, May 2015.

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

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *IEEE International Conference on Computer Vision*, 2019.

Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *International Conference on Learning Representations*, 2017.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, 2018a.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018b.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.