Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Prototype-guided cross-task knowledge distillation*

Deng LI[1], Peng LI[2], Aming WU[3], Yahong HAN[†‡1]

[1]*College of Intelligence and Computing, Tianjin University, Tianjin 300350, China*

[2]*Songshan Laboratory, Zhengzhou 450000, China*

[3]*School of Electronic Engineering, Xidian University, Xi'an 710401, China*

[†]E-mail: yahong@tju.edu.cn

Received May 12, 2024; Revision accepted Sept. 18, 2024; Crosschecked May 7, 2025

Abstract: Recently, large-scale pretrained models have revealed their benefits in various tasks. However, due to the enormous computation complexity and storage demands, it is challenging to apply large-scale models to real scenarios. Existing knowledge distillation methods require mainly the teacher model and the student model to share the same label space, which restricts their application in real scenarios. To alleviate the constraint of different label spaces, we propose a prototype-guided cross-task knowledge distillation (ProC-KD) method to migrate the intrinsic local-level object knowledge of the teacher network to various task scenarios. First, to better learn the generalized knowledge in cross-task scenarios, we present a prototype learning module to learn the invariant intrinsic local representation of objects from the teacher network. Second, for diverse downstream tasks, a task-adaptive feature augmentation module is proposed to enhance the student network features with the learned generalization prototype representations and guide the learning of the student network to improve its generalization ability. Experimental results on various visual tasks demonstrate the effectiveness of our approach for cross-task knowledge distillation scenarios.

Key words: Knowledge distillation; Cross-task; Prototype learning
https://doi.org/10.1631/FITEE.2400383 **CLC number:** TP391

1 Introduction

Recently, the Transformer network (Vaswani et al., 2017) has achieved great advances in some visual tasks, for example, image classification (Dosovitskiy et al., 2021; Liu Z et al., 2021; Touvron et al., 2021), object detection (Carion et al., 2020; Zhu XZ et al., 2021; Zhou et al., 2023), image segmentation (Ye LW et al., 2019; Jain et al., 2023), and visual language joint learning (Chen YC et al., 2020; Li LJ et al., 2020; Fu et al., 2023).

Based on the self-attention mechanism, Transformer networks can process complete input sequences and possess the advantage of parallelization. Therefore, these networks are usually used to obtain the pretrained model from large-scale datasets (Deng J et al., 2009). Currently, fine-tuning is the common strategy for the utilization of pretrained models in cross-task learning scenarios. After learning the generalized feature representation from large-scale datasets, fine-tuning is performed on the downstream task with the small dataset to boost the performance of the downstream task model. However, applying these large-scale models to practical application scenarios with limited resources (e.g., mobile devices) has become a big challenge due to their enormous computation complexity and huge storage needs.

To solve the above model application issue, some model compression and acceleration technologies have been proposed, for example, parameter pruning (Molchanov et al., 2017; Zhu MH and Gupta,

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (Nos. 62376186 and 61932009)

ORCID: Yahong HAN, https://orcid.org/0000-0003-2768-1398

2018), model quantization (Wu JX et al., 2016), and knowledge distillation (KD) (Hinton et al., 2015). Particularly, KD is a valid approach for model compression, which transfers the knowledge from a large deep neural network into a small one (Hinton et al., 2015).

Different from other model compression approaches, KD can decrease the number of parameters and boost the performance of small models on downstream tasks, irrespective of the architectural differences between the teacher network and the student network. It has been successful in a variety of tasks, such as computer vision (Hinton et al., 2015; Romero et al., 2015; Yim et al., 2017; Müller et al., 2019; Park et al., 2019; Gou et al., 2021), natural language processing (Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2020), and speech recognition (Chebotar and Waters, 2016; Kurata and Saon, 2020; Yoon et al., 2021).

However, these KD approaches require mainly the teacher network and the student network to perform the same task; for example, the teacher network and the student network share the same label space, which limits their application in real scenarios, such as downstream tasks in different label spaces as shown in Fig. 1a. By transferring the knowledge from the teacher network to downstream tasks with different label spaces, the cross-task KD method expands the application of the teacher model to a variety of downstream tasks. The existing same-task KD method works mainly to transfer the final prediction logit or the hidden layer knowledge, which is the global-level knowledge alignment and cannot be applied to cross-task KD directly. An earlier work on cross-task KD (Ye HJ et al., 2020) aligns the high-order comparison relationship between models in a local manner; however, this method lags in the representation power of the invariant intrinsic object and is a two-stage distillation method.

Under the scenario of cross-task KD, the intrinsic object characteristics can give beneficial guidance to train the student network; for example, the leg shape features of a cow and a horse are similar, while the cow belongs to the teacher network dataset and the horse belongs to the student network. Considering the complexity and variability of real-scenario tasks and the generalization capability of large-scale pretrained models, we propose a prototype-guided cross-task knowledge distillation (ProC-KD) approach to migrate local intrinsic knowledge of a large-scale teacher network to various task scenarios as shown in Fig. 1b. Our method obtains the downstream-task model with a one-stage training process that does not require fine-tuning. Specifically, our proposed approach consists of two integrated modules: the prototype-based representation learning module and the feature augmentation module. The prototype-based representation learning module is carefully designed to capture intrinsic feature information from the hidden layer of the large teacher network. Next, we feed the learned prototype representation into the feature augmentation

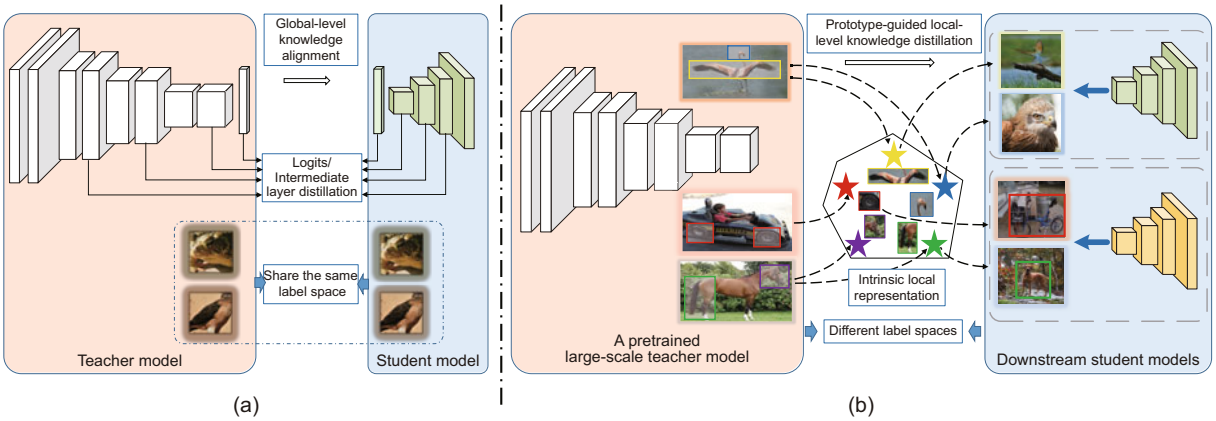

Fig. 1 Comparison between the same-task knowledge distillation method and the proposed prototype-guided cross-task knowledge distillation method: (a) the same-task knowledge distillation method, in which the large teacher network and the small student network share the same label space; (b) the proposed prototype-guided cross-task knowledge distillation method, in which the large teacher network and the small student network have different label spaces

module. The feature augmentation module enriches the student model feature that is more related to the prototype representation, while suppressing unrelated features. To give guidance on the training of the student model with the learned generalized prototype representation, a consistency loss is designed to obtain the maximum agreement between the prototype augmented output and the student network output.

In the experiments, we first verify the efficacy of our approach on various cross-task KD tasks. Then, we evaluate our approach on standard KD tasks. Our contributions are concluded as follows:

1. We propose a prototype-guided KD approach to migrate the intrinsic knowledge from a large-scale model to different small cross-task models without fine-tuning on the downstream task and improve the student model generalization ability.

2. We propose a prototype-based representation learning module and a feature augmentation module to learn the invariant intrinsic knowledge from the large-scale teacher network and enhance the student network feature with the attention mechanism, respectively.

3. We verify our method on both cross-task and same-task KD on various visual tasks. The experimental results show the validity and generality of our approach.

2 Related works

2.1 KD approach

KD compresses the mode by transferring knowledge from a larger network to a smaller one. It can be classified into three types, namely, response-based KD (Hinton et al., 2015; Müller et al., 2019), feature-based KD (Romero et al., 2015), and relation-based KD (Yim et al., 2017; Park et al., 2019).

The response-based KD directly mimics the neural responses of the output layer. Ba and Caruana (2014) and Hinton et al. (2015) proposed the mimicking of the knowledge by learning the probability distribution via soft labels. However, their methods need to obtain the class probability distribution. An effective method is to distill the feature knowledge or the relationship knowledge from the large teacher model. The goal of feature-based KD is to align the feature representation of the student network with the teacher network. FitNets (Romero et al., 2015) initially introduces intermediate representation learning, in which hints are defined as the outputs of the middle layer of the teacher network and are used to boost the learning of the student network. Inspired by Romero et al. (2015), various feature-based KD approaches (Zagoruyko and Komodakis, 2017; Passalis and Tefas, 2018; Chen DF et al., 2021) have been proposed. The relation-based KD approach probes the relationships among different intermediate layer features (Yim et al., 2017) or data in the dataset (Park et al., 2019).

Different from existing works, we explore the scenario of learning the intrinsic local-level features and reusing the knowledge of large-scale models for different downstream tasks in a cross-task manner.

2.2 Prototype learning

Prototype learning aims to learn a set of prototypes from the source set, enabling this set of prototypes to retain the maximum amount of information contained in the target set, while ensuring that all elements within the prototype set have minimal overlapping information. Deng JK et al. (2021) introduced a variational prototype learning (VPL) method, which represents each category as a distribution in the latent space to align samples with prototypes. An approach named classifiers for prototypes and reciprocals (CPR) (Hur et al., 2023) links each prototype with the corresponding known class features while pushing the reciprocal components away from these prototypes, thereby learning class-wise prototypes in the potential unknown feature space. Some methods (Snell et al., 2017; Liu JL et al., 2020; Li G et al., 2021) extract more representative prototypes by aggregating similar feature vectors for few-shot learning. Wei et al. (2023) proposed an online prototype updating method with adaptive momentum for aligning prototypes in online continual learning tasks. Here, we propose an implicit prototype learning approach to learn invariant essential representations of objects from the generalized features of a teacher model, thereby enhancing the performance of cross-task KD.

3 Main approach

Our approach is to distill the knowledge in the large-scale network to different downstream small

networks. The label space of the teacher network is different from that of the student network, which is called cross-task KD. Existing same-task KD methods mainly directly mimic the final prediction or the middle layers of the teacher model, thereby transferring the global features, and they are task-specific. The local intrinsic representations can greatly benefit the cross-task student model in understanding the novel dataset of the downstream task. As shown in Fig. 2, to enhance the generalization performance of the downstream model, the ProC-KD method is proposed to migrate the invariant intrinsic knowledge from the large-scale teacher network to the small student network. The prototype-based representation learning module and the feature augmentation module are the key modules of our model.

3.1 Prototype-based representation learning

Compared to previous methods that directly mimic the final predicted logit or the intermediate layers, our approach is to design a module to learn the intrinsic representation. Recent studies (Snell et al., 2017; Liu JL et al., 2020) have demonstrated that constructing prototype learning in models can help solve novel dataset problems. The category-specific information can be captured by prototype learning. Thus, we propose a prototype-guided module for the teacher–student distillation architecture to learn the generalized representations with the guidance of prototypes.

Fig. 3a shows an illustration of the prototype-based representation learning module. The forward process is first to align the prototypes with the input features, then reconstruct the prototype-related feature with the attention mechanism, and finally aggregate the reconstructed attention features with the input features. The whole process can be divided into three subprocesses, which are alignment, attention, and aggregation.

Specifically, before feeding forward the two-dimensional (2D) hidden layer feature extracted by the Transformer-based model, we reshape it to $\boldsymbol{F} \in \mathbb{R}^{D\times H\times W}$, where D, H, and W indicate the feature dimension, height, and width, respectively. We define the prototypes as $\boldsymbol{P}$ ($\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_n]$, $\boldsymbol{p}_i \in \mathbb{R}^D$). Here, n refers to the number of pre-set prototypes. In the alignment subprocess, both the defined prototype tensor and the input feature tensor are expanded to a dimension of $n \times D \times (W \times H)$, and we align them with the residual operation $\boldsymbol{F} - \boldsymbol{P}$. In the attention subprocess, we calculate the feature descriptors $\boldsymbol{V}$ based on the attention maps, which

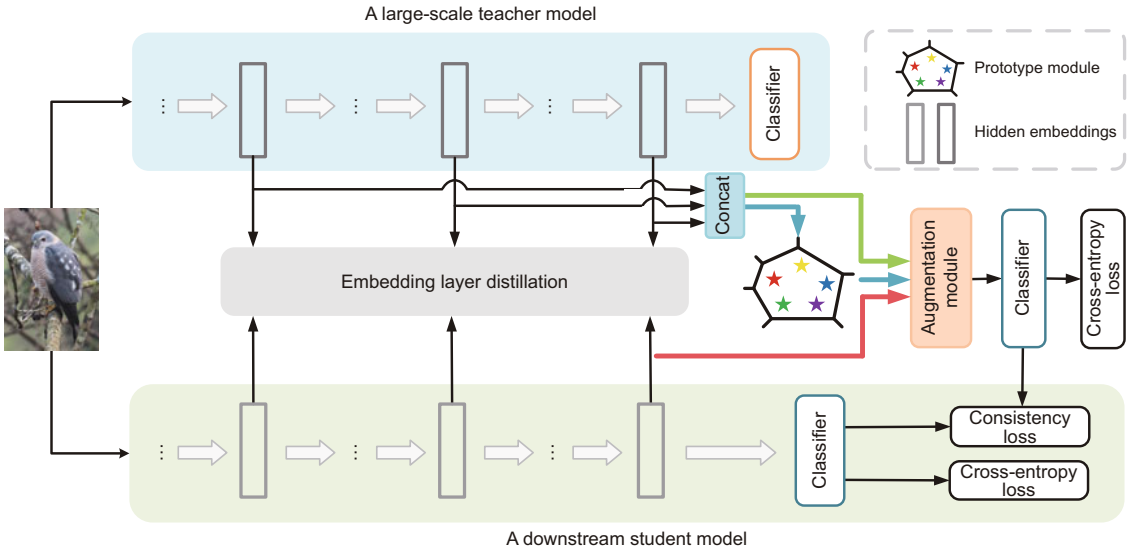

Fig. 2 Illustration of our proposed ProC-KD framework. ProC-KD includes an embedding layer distillation module, a prototype-based representation learning module, and a feature augmentation module. The blue arrow in the framework represents generalized representation learning based on prototypes. The green and red arrows indicate that the prototypes are used to enhance the features extracted from the teacher network and the student network, respectively. References to color refer to the online version of this figure

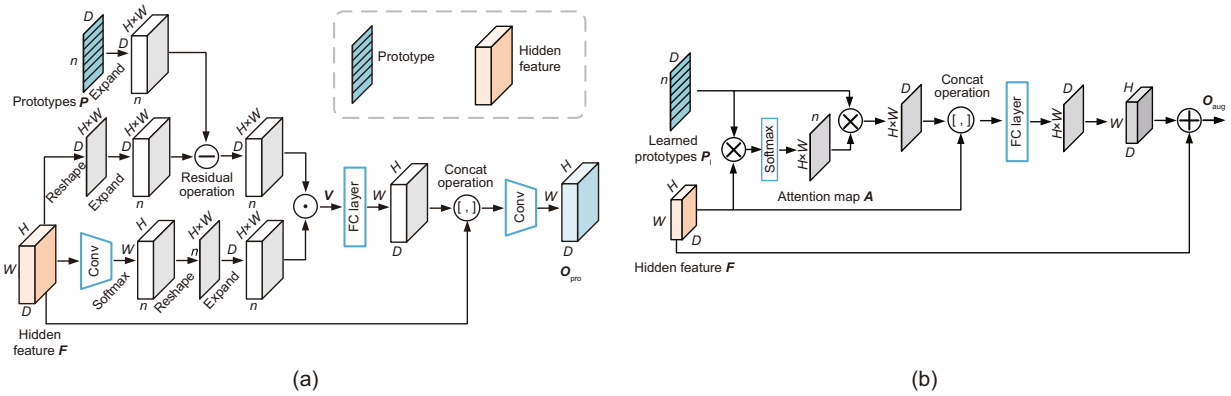

Fig. 3 Schematic of the prototype-based representation learning module (a) and the feature augmentation module (b). Conv indicates convolution operation, and FC layer is the fully connected layer. $\ominus$, $\odot$, $\otimes$, $\oplus$, and [,] indicate the residual operation, element-wise multiplication, matrix multiplication, element-wise addition, and concatenation operation, respectively

can be formulated as follows:

$$\boldsymbol{V}_i = \sum_{j=1}^{WH} \frac{\mathrm{e}^{\boldsymbol{L}_{ji}}}{\sum_{i=1}^{n} \mathrm{e}^{\boldsymbol{L}_{ji}}} \left(\boldsymbol{F}_j - \boldsymbol{p}_i\right), \tag{1}$$

where $\boldsymbol{L}_{ji}$ denotes the $(ji)^{\mathrm{th}}$ feature value within the dimension $n \times (W \times H)$, which is derived from the convolution operation on the hidden feature $\boldsymbol{F}$. $\boldsymbol{F}_j$ denotes the j^{th} feature of the expanded hidden features.

In the aggregation subprocess, we first concatenate the feature descriptors $\boldsymbol{V}$ and the input features and then transform the result with a nonlinear transformation block f. It can be presented as follows:

$$\boldsymbol{O}_{\mathrm{pro}} = f\left(\mathrm{concat}\left[\boldsymbol{F}, \boldsymbol{V}_{\mathrm{r}}\boldsymbol{W}_{\mathrm{p}} + \boldsymbol{b}_{\mathrm{p}}\right]\right), \tag{2}$$

where $\boldsymbol{V}_{\mathrm{r}}$ is the reshaped feature descriptor of $\boldsymbol{V}$, and $\boldsymbol{W}_{\mathrm{p}}$ and $\boldsymbol{b}_{\mathrm{p}}$ indicate the weight and bias of the fully connected layer, respectively. Moreover, concat[,] indicates the concatenation operation. The shapes of the output $\boldsymbol{O}_{\mathrm{pro}}$ and the input $\boldsymbol{F}$ are the same.

Through the above processes, the generalized representation of the prototypes could be learned from the input features of the large-scale teacher model. Upon such generalized prototypes, we seek to enhance the student features with these prototypes.

3.2 Feature augmentation with prototypes

To strengthen the generalization ability of student networks for different downstream tasks, the learned generalized prototypes are used to enhance the feature in the feature augmentation module, as shown in the right part of Fig. 2. The main idea is to enrich the feature that is more related to the prototype representation while suppressing the unrelated features.

Fig. 3b shows the illustration of the feature augmentation module. The forward process is first to pay attention to the feature correlated with the prototype representation, and then enhance the input feature with the prototype-related feature. The whole feature augmentation process can also be broken into two subprocesses, namely attention and augmentation.

Concretely, for the learned prototypes $\boldsymbol{P}_{\mathrm{l}} \in \mathbb{R}^{n \times D}$ and the hidden features $\boldsymbol{F} \in \mathbb{R}^{t \times D}$ in the attention subprocess, we first encode the prototypes and input features, respectively. Attention map $\boldsymbol{A}$ is obtained by calculating the softmax of the cross-product between the learned prototypes $\boldsymbol{P}_{\mathrm{l}}$ and the encoded feature $\boldsymbol{F}_{\mathrm{e}}$, which can be expressed as follows:

$$\boldsymbol{A} = \mathrm{softmax}\left(\boldsymbol{F}_{\mathrm{e}}\boldsymbol{P}_{\mathrm{l}}^{\mathrm{T}}\right). \tag{3}$$

Then the attention feature is obtained by calculating the cross product between the attention map and the prototypes.

In the augmentation subprocess, we concatenate the attention feature with the encoded input features and then apply a fully connected layer to transform the shape of the result. The fully connected layer employs the rectified linear unit (ReLU) as its activation function. The original input hidden layer feature is enhanced with the prototype-related feature through element-wise sum operation. This whole process can

be written as follows:

$$\boldsymbol{O}_{\text{aug}} = \text{ReLU}(\Phi\,(\text{concat}\,[\boldsymbol{F}_{\text{e}}, \boldsymbol{A}\boldsymbol{P}_{\text{l}}]) + \boldsymbol{F}_{\text{r}}), \quad (4)$$

where $\boldsymbol{O}_{\text{aug}}$ is the output, Φ indicates the function of the fully connected layer, and $\boldsymbol{F}_{\text{r}}$ indicates the reshaped feature of $\boldsymbol{F}$. Finally, the enhanced feature is inputted to the classifier shared with the student model to predict the categories.

3.3 Cross-task KD

In this paper, KD of the student model and teacher model in different label spaces is defined as cross-task KD. We propose a prototype-based representation learning module and a feature augmentation module to guide the student network training in cross-task KD scenarios and improve its generalization ability. Here, we also design some loss functions to constrain the training of cross-task KD. Following Jiao et al. (2020), we distill the knowledge of hidden state features from the large-scale teacher model for the embedding layer KD. Assuming that we are distilling the knowledge from an m-layer teacher model to an n-layer small student model, we need to select n out of m layers from the large teacher network. The loss function for hidden layer KD can be expressed as follows:

$$L_{\text{emb}} = \sum_{i=1}^{n} \text{MSE}(\boldsymbol{F}_i^{\text{S}}\boldsymbol{W}_{\text{h}},\ \boldsymbol{F}_i^{\text{T}}), \quad (5)$$

where $\boldsymbol{F}^{\text{S}} \in \mathbb{R}^{l\times d'}$ is the student hidden feature and $\boldsymbol{F}^{\text{T}} \in \mathbb{R}^{l\times d}$ is the teacher hidden feature, l denotes the sequence length of the hidden features from both the student and teacher models, and d and d' denote the hidden embedding sizes of the teacher model and the student model, respectively. $\boldsymbol{W}_{\text{h}} \in \mathbb{R}^{d'\times d}$ is a learnable transformation weight matrix, which transforms the hidden layer features of the student model into the same dimensions as the features of the teacher model. MSE$(\cdot)$ indicates the mean squared error function.

We also define a consistency loss L_{con} and a classification loss L_{procls} for prototype learning. The consistency loss is obtained by calculating the Kullback–Leibler (KL) divergence between the logit y_{con} from the prototype augmentation module and the predicted logit y_{stu} from the student model, $L_{\text{con}} = \mathcal{H}(y_{\text{con}}, y_{\text{stu}})$. The classification loss is obtained with the softmax cross-entropy loss between y_{con} and the label y. Thus, the loss function of the prototype learning module can be defined as follows:

$$L_{\text{pro}} = L_{\text{con}} + L_{\text{procls}}. \quad (6)$$

In addition to the embedding layer feature distillation loss function and the prototype learning loss function, we define the student model loss function as L_{stu}. The joint training loss function for our cross-task KD can be expressed as follows:

$$L_{\text{total}} = \lambda_{\text{emb}}L_{\text{emb}} + \lambda_{\text{pro}}L_{\text{pro}} + \lambda_{\text{stu}}L_{\text{stu}}, \quad (7)$$

where λ_{emb}, λ_{pro}, and λ_{stu} are the weights of the embedding layer feature distillation loss, prototype learning loss, and student model loss, respectively.

4 Experiments

To evaluate the general effectiveness of our method, we conducted experiments on both cross-task KD and standard same-task KD settings. The experiments were carried out on image classification and object detection for each setting. In this paper, the KD scheme was set as offline distillation, which means that the weights of the teacher model are fixed.

4.1 Cross-task KD

4.1.1 Image classification

We carried out our experiments on the Transformer-based model in three downstream tasks, including standard image classification, long-tailed image classification, and cross-domain image classification. Here, the teacher networks were trained on ImageNet-1K (Deng J et al., 2009).

1. Datasets

CIFAR-100 (Rebuffi et al., 2017) consists of 50 000 and 10 000 images for training and validation, respectively. It contains 100 categories and 600 images per category. Following Cao et al. (2019) and Cui et al. (2019), we created the long-tailed CIFAR-100 by reducing the number of training samples for each category but with the verification set unchanged. We also defined the imbalance ratio β to represent the ratio of sample sizes between the most- and least-frequent categories; this can be formulated as $\beta = N_{\max}/N_{\min}$. The number of samples decays exponentially between classes. The imbalance ratio was set as 10 in our experiments. Office–Home

dataset (Venkateswara et al., 2017) has been built to test domain adaptation methods for image classification. It contains four different domains and a total of 15 500 images; the four domains are named Clip Art (Cl), Artistic images (Ar), Real-World images (Rw), and Product images (Pr). Each domain in this dataset contains 65 categories, and the images are from office or home scenarios. We set the Rw images to be the training set and the other domains to be the test sets in our experiments.

2. Implementation details

The well-known vision Transformer (ViT) (Dosovitskiy et al., 2021) and Swin Transformer (Liu Z et al., 2021) models were used in our experiments. For ViT, we set the teacher network as a 12-layer ViT-B and the student network as six-, four-, and two-layer small ViT models. The indices of hidden layers selected for distillation in the teacher model were {2, 4, 6, 8, 10, 12}, {3, 6, 9, 12}, and {6, 12}. Both the attention map knowledge and hidden layer feature knowledge were distilled. We set the hyperparameters λ_{pro} and λ_{stu} in Eq. (7) to be 1.0 and set λ_{emb} to be 0.3. In the training phase, the embedding features were selected to concatenate and then inputted into the prototype-based representation learning module and the feature augmentation module. We set the number of prototypes as 72 and the optimizer as AdamW with a learning rate of 5e-4 and a weight decay of 0.05. The size of the input image was 224×224 and the batch size was set to 32 for each graphics processing unit (GPU).

For the Swin Transformer, the teacher model was a 24-layer Swin-L, and the student models were smaller Swin Transformer models of 12 and 4 layers. We set the training optimizer as AdamW with a learning rate of 5e-4 and a weight decay of 0.05. The training batch had 64 images.

The experiments were run on eight NVIDIA Tesla V100 GPUs with 32 GB VRAM. We used the NVIDIA Collective Communications Library (NCCL) for multi-node parallel training. We also reduced multi-GPU communication overhead with gradient accumulation.

3. Results and analysis

The experimental results on the standard, long-tailed, and cross-domain image classification tasks are shown in Table 1. We reimplemented relation knowledge distillation (RKD) in our experimental setting. Following TinyBERT (Jiao et al., 2020), FBKD is a feature-based KD method for the Transformer-based model, which distills the knowledge from the embedding layers and attention maps to the student model. We reimplemented the hidden layer KD method in the PKD approach (Miles and Mikolajczyk, 2024) for comparison.

(1) Image classification. Compared with the baseline method FBKD, for ViT, our ProC-KD improves the classification accuracy by 1.51% (from 86.16% to 87.46%), 0.79% (from 83.75% to 84.41%), and 0.96% (from 73.50% to 74.21%) on the six-, four-, and two-layer small ViT models, respectively. For the Swin Transformer, our ProC-KD improves the classification accuracy by 0.69% (from 83.63% to 84.21%) and 4.17% (from 73.08% to 76.13%) on 12- and 4-layer small Swin Transformer models, respectively. This demonstrates that our ProC-KD method can promote the prototypes to learn the generalized representation and improve the generalized ability of the student model in cross-task image classification KD scenarios.

(2) Long-tailed image classification. Table 1 shows that our ProC-KD improves the performance upon the baseline approach FBKD by 7.74% (from 72.69% to 78.32%), 5.20% (from 69.83% to 73.46%), and 0.75% (from 58.43% to 58.87%) on six-, four-, and two-layer small ViT models, respectively, on the long-tailed CIFAR-100. For the Swin Transformer, our method improves the performance by 17.98% (from 57.83% to 68.23%) on the 12-layer student Swin Transformer model and has a comparable performance to the second-best level on the four-layer student Swin Transformer model. This demonstrates that our ProC-KD can boost the generalization performance in long-tailed image classification tasks.

(3) Cross-domain image classification. The cross-domain image classification experiment was conducted on the Office–Home dataset. Compared with the FBKD baseline, ProC-KD improves the performance by 1.75% (from 75.51% to 76.83%) on domain shift Rw→Pr and by 2.35% (from 40.02% to 40.96%) on the hardest domain shift Rw→Cl for the six-layer ViT student model. Our method achieves the best results on Swin Transformer compared with other KD methods. This demonstrates that distilling knowledge from the large-scale network can boost the performance of the small network in the cross-domain scenario, and our ProC-KD can further improve the generalization ability.

Table 1 Mean accuracy on three cross-task image classification knowledge distillation tasks

Teacher	Method	Number of parameters	Mean accuracy (%)				
			Standard	Long-tailed	Cross-domain		
			CIFAR-100	LT-CIFAR	Rw→Ar	Rw→Cl	Rw→Pr
ViT-B (86M)	Student model	43M	78.84	55.83	20.85	16.91	34.85
	RKD (Park et al., 2019)	43M	87.13	76.52	60.61	39.04	76.19
	ABLoss (Heo et al., 2019b)	43M	81.11	76.73	57.31	40.64	73.50
	OFD (Heo et al., 2019a)	43M	82.83	76.84	60.19	40.49	75.19
	FBKD (Jiao et al., 2020)	43M	86.16	72.69	60.28	40.02	75.51
	PKD (Miles and Mikolajczyk, 2024)	43M	86.37	75.81	59.17	40.16	75.22
	ProC-KD (ours)	43M	**87.46**	**78.32**	**61.41**	**40.96**	**76.83**
ViT-B (86M)	Student model	29M	73.48	49.05	18.62	15.30	31.94
	FBKD (Jiao et al., 2020)	29M	83.75	69.83	19.53	16.17	35.41
	OFD (Park et al., 2019)	29M	77.78	72.20	**43.15**	**36.91**	63.87
	PKD (Miles and Mikolajczyk, 2024)	29M	79.69	71.84	36.55	27.93	57.85
	ProC-KD (ours)	29M	**84.41**	**73.46**	42.69	32.58	**64.65**
ViT-B (86M)	Student model	15M	68.65	45.98	17.51	15.21	29.35
	FBKD (Jiao et al., 2020)	15M	73.50	58.43	19.37	16.20	32.28
	OFD (Park et al., 2019)	15M	60.83	53.15	24.80	**21.40**	**46.83**
	PKD (Miles and Mikolajczyk, 2024)	15M	73.08	58.72	24.85	20.55	44.38
	ProC-KD (ours)	15M	**74.21**	**58.87**	**25.67**	21.17	45.51
Swin-L (197M)	Student model	110M	78.90	41.48	26.87	20.51	43.32
	RKD (Park et al., 2019)	110M	83.99	58.94	27.71	21.73	46.32
	ABLoss (Heo et al., 2019b)	110M	83.26	67.08	36.88	23.01	52.64
	OFD (Heo et al., 2019a)	110M	80.99	47.91	40.49	26.81	59.20
	FBKD (Jiao et al., 2020)	110M	83.63	57.83	41.29	29.03	63.40
	AttentionProbe (Wang JH et al., 2022)	110M	80.78	66.34	38.03	26.87	58.05
	ProC-KD (ours)	110M	**84.21**	**68.23**	**42.16**	**30.24**	**64.27**
Swin-L (197M)	Student model	44M	74.89	55.82	25.01	19.95	40.14
	ABLoss (Heo et al., 2019b)	44M	75.23	**58.03**	34.33	20.84	51.43
	OFD (Heo et al., 2019a)	44M	75.21	52.00	28.75	19.49	44.25
	FBKD (Jiao et al., 2020)	44M	73.08	45.94	26.28	19.16	40.96
	AttentionProbe (Wang JH et al., 2022)	44M	75.28	56.71	32.98	24.97	49.47
	ProC-KD (ours)	44M	**76.13**	56.50	**34.45**	**25.53**	**55.32**

Cross-domain image classification task is performed on the Office–Home dataset. LT-CIFAR indicates long-tailed CIFAR-100. M means million. The value in the bracket in the first column means the number of parameters of the teacher network. The best results are in bold, and the second-best results are underlined

(4) Visualization analysis. Fig. 4 shows the attention map visualization of the baseline FBKD and our ProC-KD. As can be seen, compared with the FBKD baseline, the attention map of our method ProC-KD focuses more on objects in both the shallow and deep layers. It indicates that our ProC-KD method can promote student network learning in both the shallow and deep layers. Fig. 5 shows the visualization of the 72 prototypes of the CIFAR-100 dataset on the six-layer ViT model. Fig. 5a is the distance matrix of the initial prototypes, Fig. 5b is the distance matrix of the learned prototypes, and Fig. 5c is the t-distributed stochastic neighbor embedding (t-SNE) visualization of the learned prototypes. The visualization results show that the distances between the initial prototypes are relatively small, concentrating around 0.3, and that they are tightly clustered in the feature space. In contrast, the distance of the prototype after learning is relatively divergent, and the training process enables the prototypes to be better distributed in the feature space to capture the diversity and complexity of the data.

4.1.2 Object detection

In addition to the image classification tasks, we evaluated our approach on standard object detection and domain-adaptive object detection tasks. We only took the source domain as the training domain and then tested the student model on the target

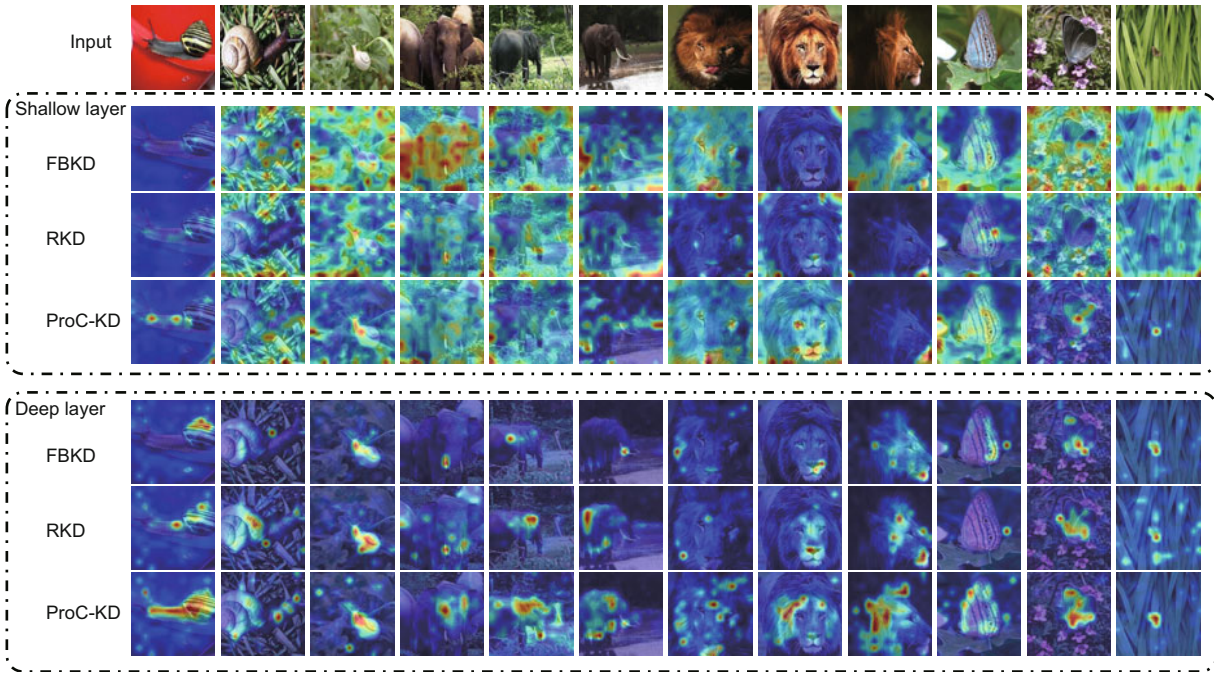

Fig. 4 Comparison of attention maps by using the Transformer interpretability method (Chefer et al., 2021). Here, the second and the last layers of the ViT are selected as the shallow layer and the deep layer, respectively

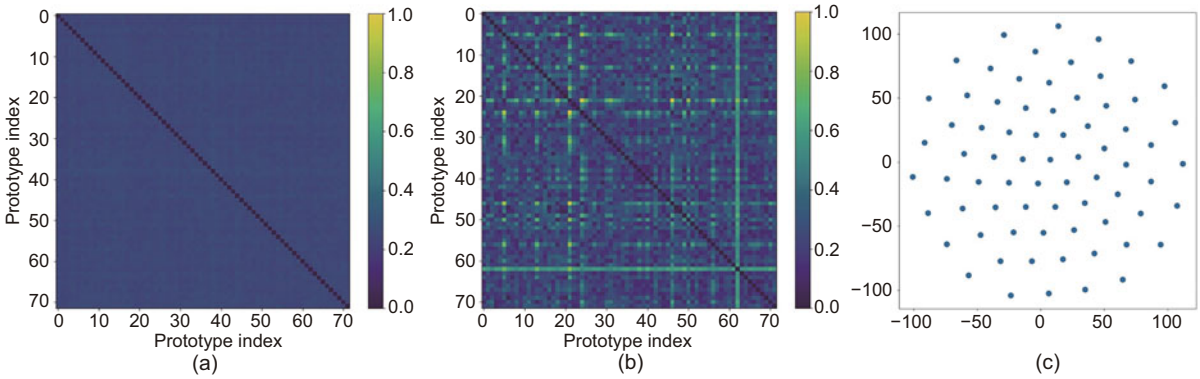

Fig. 5 Visualization of prototypes on the CIFAR-100 dataset of the cross-task knowledge distillation task: (a) distance matrix of the initial prototypes; (b) distance matrix of the learned prototypes; (c) t-SNE of the learned prototypes

domain directly for domain-adaptive object detection. Here, we trained the teacher model on the COCO (Lin et al., 2014) dataset, and the weights of the teacher model were fixed.

1. Datasets

The Cityscapes (Cordts et al., 2016) is an urban street dataset with eight categories of objects. It contains about 3000 and 500 images for training and validation, respectively. It is often used for object detection and image segmentation tasks. Foggy Cityscapes (Sakaridis et al., 2018) is a dataset generated by synthesizing different degrees of fog on Cityscapes (Cordts et al., 2016). Thus, the quantity of images in the training and validation sets is the same as that of Cityscapes. Daytime-sunny, Dusk-rainy, and Night-rainy (Wu AM et al., 2021) are three street-scene datasets under different weather environments collected from the BDD-100k dataset. In our experiments, about 28 000 images from Daytime-sunny were selected as the training set, and 2500 and 3500 images from Night-rainy and Dusk-rainy were selected as the test set for the two scenarios, respectively.

2. Implementation details

For the cross-task KD experiment of standard object detection and cross-domain object detection, we set the teacher network as the Cascade Mask-RCNN with the backbone of 24-layer Swin-Base (Liu Z et al., 2021) model trained on COCO (Lin et al., 2014), and we set the student network as the Cascade Mask-RCNN with the backbone of 12-layer Swin-Tiny. The fourth-layer feature from the feature pyramid network (FPN) in the teacher network was inputted to the prototype-based representation learning module and the feature augmentation module for generalized representation learning. The coefficients λ_{pro} and λ_{emb} were both set to 1.0. The optimizer used for training was the stochastic gradient descent (SGD) optimizer with a learning rate of 0.01 and a parameter decay of 1e-4. We set the training batch size to be 2 per GPU in the cross-task object detection KD.

3. Results and analysis of Cityscapes and Foggy Cityscapes

Table 2 shows the detection results on Cityscapes and Foggy Cityscapes. Here, we reimplemented the methods of FBKD (Jiao et al., 2020), CWD (Shu et al., 2021), MGD (Yang et al., 2022b), and SKD (Zhang LF and Ma, 2023) in our experimental setting. FBKD is a KD method that follows TinyBERT (Jiao et al., 2020). It shows that our method boosts the performance under the cross-task KD scenario significantly. For Cityscapes, compared to the second-best method CWD (Shu et al., 2021), our method improves the mean average precision (mAP) performance by 3.07%. For Foggy Cityscapes, compared to the second-best method CWD (Shu et al., 2021), our method boosts the mAP by 3.07%. The results demonstrate that the generalized prototype representation is helpful for learning of the student network in the context of object detection.

4. Results and analysis of domain-adaptive object detection

We evaluated our method on the domain-adaptive object detection scenario. Here, the student model was trained on the Daytime-sunny domain and tested on the Night-rainy domain and the Dusk-rainy domain. The teacher networks were all trained on the COCO dataset, and the weight was frozen during distillation training. As shown in Table 3, for Daytime-sunny→Night-rainy, compared with the CWD (Shu et al., 2021) baseline, our ProC-KD method boosts the mAP by 4.44%. For Daytime-sunny→Dusk-rainy, our ProC-KD method improves the mAP by 1.34%. The reason that the performance of our ProC-KD is poorer than that of the CWD on the object of the motorcycle may be that the quantity of the ground truth of the motorcycle is quite small. It contains only 49 ground-truth annotations of motorcycles in the Daytime-sunny→Night-rainy test set and only 110 ground-truth annotations in the Daytime-sunny→Dusk-rainy test set.

5. Visualization analysis

Table 2 Detection results of cross-task knowledge distillation on object detection for the Cityscapes and Foggy Cityscapes datasets

Method	Detection accuracy on Cityscapes (%)								mAP (%)
	Bicycle	Bus	Car	Motorcycle	Person	Rider	Train	Truck	
Student model	47.4	46.1	68.4	27.0	33.9	35.6	32.1	31.7	40.3
CWD (Shu et al., 2021)	<u>57.9</u>	<u>62.1</u>	**77.5**	**41.4**	<u>51.1</u>	<u>49.4</u>	<u>51.5</u>	**52.3**	<u>55.4</u>
MGD (Yang et al., 2022b)	52.6	55.2	<u>73.8</u>	31.0	45.7	47.0	42.4	38.5	48.3
ProC-KD (ours)	**58.7**	**66.7**	**77.5**	<u>39.2</u>	**53.4**	**55.2**	**56.3**	<u>50.0</u>	**57.1**
Method	Detection accuracy on Foggy Cityscapes (%)								mAP (%)
	Bicycle	Bus	Car	Motorcycle	Person	Rider	Train	Truck	
Student model	31.8	42.3	63.0	27.3	40.8	40.3	11.6	27.8	35.6
FBKD (Jiao et al., 2020)	49.0	53.9	68.6	40.6	52.0	54.1	35.7	37.5	48.9
CWD (Shu et al., 2021)	<u>50.9</u>	<u>57.1</u>	<u>71.9</u>	<u>43.2</u>	<u>53.3</u>	<u>55.8</u>	<u>46.8</u>	37.5	<u>52.1</u>
MGD (Yang et al., 2022b)	47.3	49.4	66.3	32.3	47.8	28.5	31.3	**41.0**	43.0
SKD (Zhang LF and Ma, 2023)	45.4	53.7	69.3	41.4	52.1	51.2	42.6	36.0	49.0
ProC-KD (ours)	**51.5**	**57.7**	**73.1**	**44.2**	**53.8**	**57.9**	**51.2**	<u>40.3</u>	**53.7**

Teacher networks are trained on the COCO and the weight is frozen during distillation training. The best results are in bold, and the second-best results are underlined

Table 3 Detection results of cross-task knowledge distillation on domain-adaptive object detection of Daytime-sunny→Night-rainy and Daytime-sunny→Dusk-rainy

Method	Detection accuracy on Daytime-sunny→Night-rainy (%)							mAP (%)
	Bicycle	Bus	Car	Motorcycle	Person	Rider	Truck	
Student model	24.3	9.1	33.8	1.1	12.3	9.1	16.1	15.1
FBKD (Jiao et al., 2020)	35.7	17.0	47.1	**9.8**	22.7	13.9	31.7	25.4
CWD (Shu et al., 2021)	38.6	17.1	**49.4**	9.7	24.4	15.6	34.4	27.0
MGD (Yang et al., 2022b)	32.6	10.6	42.5	1.4	21.4	9.9	27.8	20.9
SKD (Zhang LF and Ma, 2023)	36.9	14.0	46.4	6.6	21.6	13.0	31.6	24.3
ProC-KD (ours)	**40.9**	**18.3**	**49.4**	8.6	**26.1**	**18.2**	**35.7**	**28.2**

Method	Detection accuracy on Daytime-sunny→Dusk-rainy (%)							mAP (%)
	Bicycle	Bus	Car	Motorcycle	Person	Rider	Truck	
Student model	40.6	14.9	66.0	11.5	25.8	15.2	39.7	30.5
FBKD (Jiao et al., 2020)	48.2	33.0	73.1	21.5	42.2	28.7	53.7	42.9
CWD (Shu et al., 2021)	49.9	34.8	**73.9**	**24.0**	43.9	**32.0**	**54.7**	44.7
MGD (Yang et al., 2022b)	45.1	26.8	72.0	10.8	39.9	22.5	49.1	38.0
SKD (Zhang LF and Ma, 2023)	48.1	30.3	73.4	20.9	41.5	26.7	52.1	41.9
ProC-KD (ours)	**52.6**	**36.6**	73.3	21.6	**46.5**	31.6	54.6	**45.3**

Teacher models are all trained on the COCO dataset, and the weight is frozen during distillation training. The best results are in bold, and the second-best results are underlined

Fig. 6 Qualitative results on Cityscapes. The first, second, and third rows represent the ground truth, the results of the CWD method, and the results of our ProC-KD method, respectively. Compared with the CWD baseline, our ProC-KD method could detect objects more accurately, for example, the bus, bicycle, truck, and person

The visualization results of the detection results on the Cityscapes and Foggy Cityscapes datasets are shown in Figs. 6 and 7, respectively. Here, the first row is the ground truth, the second row shows the results of the baseline method CWD (Shu et al., 2021), and the third row shows the detection results of ProC-KD. We can see that our method ProC-KD could detect objects more precisely in normal and foggy scenes compared with the second-best method CWD.

The visualization results of domain-adaptive object detection on Daytime-sunny→Night-rainy and Daytime-sunny→Dusk-rainy are shown in Figs. 8 and 9, respectively. Here, we take CWD (Shu et al., 2021) as the baseline. The first row is the ground truth, the second row shows the results of the baseline method CWD (Shu et al., 2021), and the third row shows the results of our ProC-KD. The visualization results show that our ProC-KD could localize and recognize objects in both Night-rainy and Dusk-rainy images more accurately compared with the second-best method CWD.

Fig. 7 Qualitative results on Foggy Cityscapes. The first, second, and third rows represent the ground truth, the results of the CWD method, and the results of our ProC-KD method, respectively. Compared with the CWD baseline, our ProC-KD method could detect objects more accurately in the foggy scene, for example, the rider, bicycle, bus, and car

Fig. 8 Qualitative results of domain-adaptive object detection on Daytime-sunny→Night-rainy. The first, second, and third rows represent the ground truth, the results of the CWD method, and the results of our ProC-KD method, respectively. Compared with the CWD baseline, our ProC-KD method could detect objects more accurately in the Night-rainy scene, for example, the person, bus, truck, and car

4.2 Same-task KD

Our method is verified in the standard KD setting, in which the teacher network and the student network share the same label space.

4.2.1 Image classification

In this part, we verify our method on the image classification model of convolutional neural network (CNN) structure. The experiments were conducted with the wide residual network (Wide-ResNet). By changing the depth and width of the student model, we can obtain different student models, thus verifying the adaptability of the method to networks of different scales. The dataset used in the experiments was CIFAR-100 (Rebuffi et al., 2017). Following ReFilled (Ye HJ et al., 2020), all teacher models were set as Wide-ResNet with a depth of 40 and width of 2 in these experiments.

The comparison results between our ProC-KD and state-of-the-art (SOTA) distillation methods with different student models are shown in Table 4. Similar to ReFilled (Ye HJ et al., 2020), we tested the model after training convergence on the training

Fig. 9 Qualitative results of domain-adaptive object detection on Daytime-sunny→Dusk-rainy. The first, second, and third rows represent the ground truth, the results of the CWD method, and the results of our ProC-KD method, respectively. Compared with the CWD baseline, our ProC-KD method could detect objects more accurately in the Dusk-rainy scene, for example, the person, bus, car, truck, and bicycle

Table 4 Classification results for the standard image classification knowledge distillation scenario

Method	Accuracy (%)		
	(40, 1)	(16, 2)	(16, 1)
Student	68.97	70.15	65.44
KD (Hinton et al., 2015)	70.46	71.87	66.54
FitNets (Romero et al., 2015)	68.66	70.89	65.38
VID-I (Ahn et al., 2019)	71.51	73.31	66.32
RKD (Park et al., 2019)	72.18	72.56	65.22
ReFilled (Ye HJ et al., 2020)	72.72	74.01	67.56
DKD (Zhao et al., 2022)	<u>74.81</u>	<u>76.24</u>	67.46
MASCKD (Gou et al., 2023)	–	–	67.26
BookKD (Zhu SL et al., 2023)	–	–	<u>69.29</u>
ProC-KD (ours)	**75.14**	**76.43**	**69.36**

The teacher network and student network share the same label space of the CIFAR-100 dataset. The first and second values in the bracket are the depth and width, respectively. "–" means that the values of the compared methods are not provided in the literature. The best results are in bold, and the second-best results are underlined

set. Table 4 shows that our method achieves the best accuracy in three student models with different structures compared with other KD methods.

The visualization results using t-SNE (van der Maaten and Weinberger, 2012) for embedding features of 10 randomly selected classes are shown in Fig. 10. It shows that for the embedding features of 10 categories sampled randomly, the embedding representation of our method is more discriminative.

4.2.2 Object detection

Our prototype-guided KD method was also applied to the standard object detection KD. The teacher model in the experiments was set as Cascade Mask-RCNN with ResNeXt101 backbone, and the student model was Faster-RCNN with the backbones of ResNet-18 and ResNet-50. Different from the cross-task KD experiments, here, the training dataset of the pretrained teacher network and the distillation process were both performed on the COCO dataset.

Table 5 shows the comparison results with SOTA methods on object detection. Our method has a performance comparable to those of other KD methods in different intersection over union (IoU) thresholds and with different object sizes. In particular, we achieve a 1.78% improvement over the second-best method on AP_L. Here, AP_L refers to the detection accuracy of large-sized objects. Table 6 illustrates the results for various lightweight detectors on the COCO dataset, and it shows that our proposed method outperforms the SKD approach on the detector with the backbones of ResNet-18 and ResNet-50. These results demonstrate the robustness of our methodology across different compression ratio detectors.

Fig. 11 shows the error analysis of precision–recall curves of all-area objects, large-sized objects,

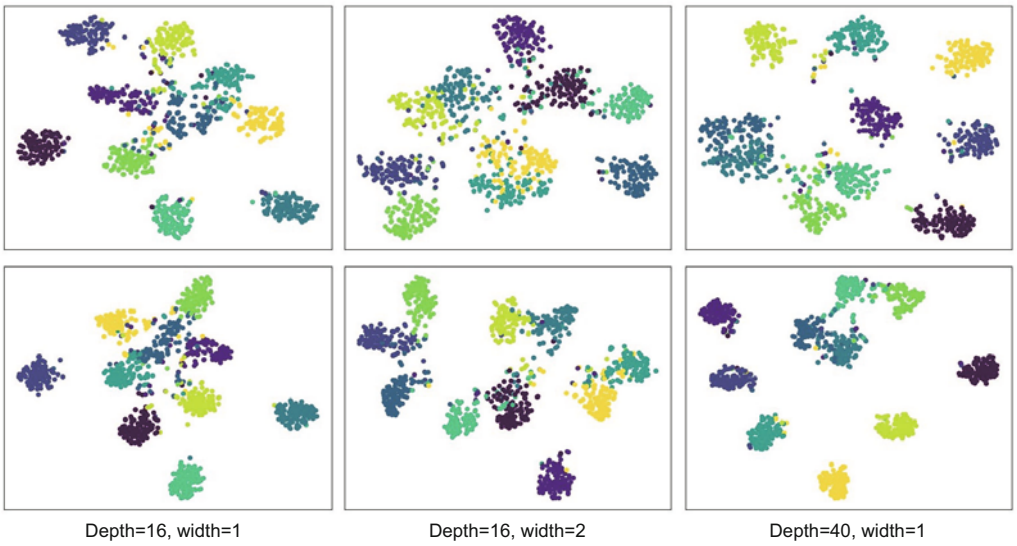

Fig. 10 t-SNE of the baseline (upper) and our (bottom) methods over 10 classes randomly sampled from the CIFAR-100 dataset

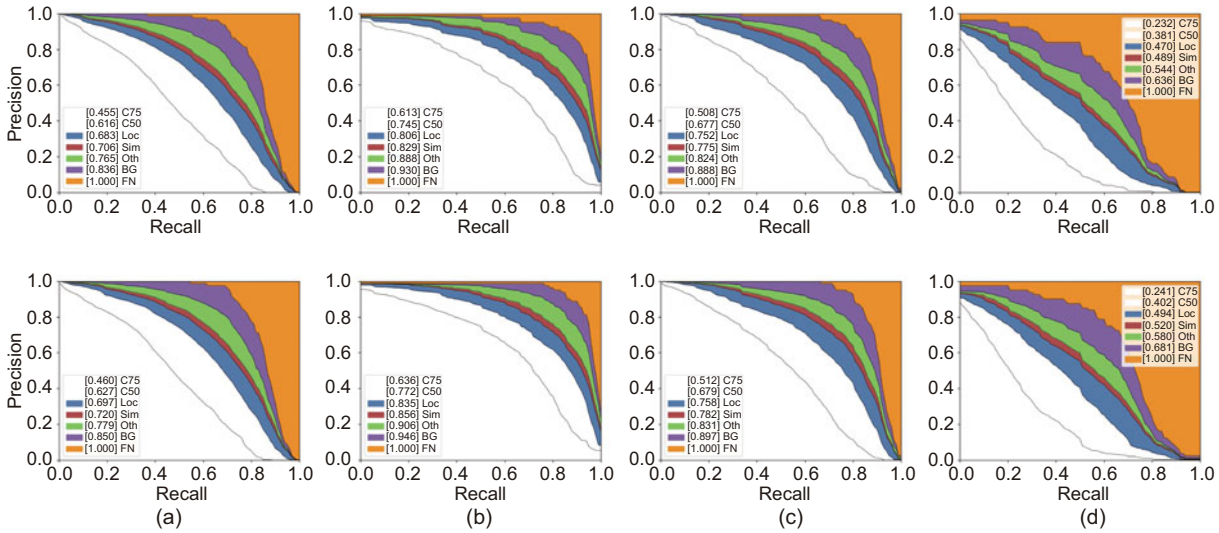

Fig. 11 Error analysis of precision–recall curves of all-area objects (a), large-sized objects (b), medium-sized objects (c), and small-sized objects (d) on the COCO dataset. The top row shows the results of the baseline CWD, and the bottom row shows the results of our ProC-KD. Here, C75 indicates the results at a 0.75 IoU threshold, C50 indicates the results at a 0.50 IoU threshold, Loc indicates the results after ignoring localization errors, Sim indicates the results obtained by ignoring false positives from similar classes within the same supercategory, Oth indicates the results after ignoring all category confusions, BG indicates the results after ignoring all false positives, and FN indicates the results after ignoring all false negatives

medium-sized objects, and small-sized objects under different conditions on the COCO dataset. The top row shows the detection results of the baseline method CWD (Shu et al., 2021), and the bottom row shows the detection results of our ProC-KD. We can see that our ProC-KD achieves better performance on different IoU thresholds for all different-sized objects. Compared with the baseline method, our method improves performance by 0.029 and 0.024 on large- and small-sized objects, respectively, ignoring the localization errors. This indicates that our method can provide more precise classification information. Our ProC-KD also outperforms the baseline CWD method by an average of 0.014 on

Table 5 Detection results on the standard object detection knowledge distillation scenario

Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Teacher	44.3	62.7	48.4	25.4	48.4	58.1
Student (ResNet50)	38.4	59.0	42.0	21.5	42.1	50.3
Chen GB et al. (2017)'s	38.7	59.0	42.1	22.0	41.9	51.0
Wang T et al. (2019)'s	39.1	59.8	42.8	22.2	42.9	51.1
Heo et al. (2019a)'s	38.9	60.1	42.6	21.8	42.7	50.7
CWD (Shu et al., 2021)	41.7	62.0	45.5	23.3	45.5	55.5
FGD (Yang et al., 2022a)	42.0	–	–	**23.8**	**46.4**	55.5
MGD (Yang et al., 2022b)	**42.1**	–	–	23.7	**46.4**	56.1
SKD (Zhang LF and Ma, 2023)	41.5	62.2	45.1	23.5	45.0	55.3
AKD (Zhang Y et al., 2023)	42.0	62.3	45.7	23.6	45.9	55.7
ProC-KD (ours)	**42.1**	**62.7**	**46.0**	23.5	45.8	**57.1**

The teacher network and student network share the same label space of the COCO dataset. "–" means that the values of the compared methods are not provided in the literature. AP_{50} and AP_{75} refer to the detection accuracy at the 0.50 and 0.75 IoU threshold, respectively. AP_S, AP_M, and AP_L refer to the detection accuracy of small-, medium-, and large-sized objects, respectively. The best results are in bold, and the second-best results are underlined

Table 6 Quantitative results of diverse lightweight detectors on the COCO dataset

Backbone	Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-18	Student	34.6	55.0	37.1	19.3	36.9	45.9
	SKD	37.0	57.2	39.7	19.9	39.7	50.3
	ProC-KD	**37.5**	**57.4**	**40.5**	**20.0**	**40.9**	**50.7**
ResNet-50	Student	38.4	59.0	42.0	21.5	42.1	50.3
	SKD	41.5	62.2	45.1	**23.5**	45.0	55.3
	ProC-KD	**42.1**	**62.7**	**46.0**	**23.5**	**45.8**	**57.1**

ResNeXt101 is the backbone of the teacher model. The best results are in bold

all-area objects after ignoring localization errors, ignoring similar classes from the same supercategory, ignoring all category confusions, and ignoring all false positives, which demonstrates a better location and recognition ability of our method.

4.3 Ablation studies

We conducted ablation studies on the design elements, the number of prototypes, and the weight of the loss function in the proposed prototype-guided KD.

4.3.1 Design elements

We study the effects of the design elements on the Foggy Cityscapes. The ablation results in Table 7 show the following results: (1) compared with the baseline, a 7.16% (from 48.9% to 52.4%) mAP increasement can be obtained with the prototype learning module, indicating that the prototype representation is beneficial to the learning of the student network; (2) the combination of the prototype-based representation learning module and feature augmentation module leads to a significant mAP improvement, which is 2.48% (from 52.4% to 53.7%). The reason may be that the feature augmentation module enriches the feature that is more related to the object.

4.3.2 Number of prototypes

We ablated the number of prototypes on the long-tailed image classification task with the ViT model. The 12-layer base version of the ViT model was set as the teacher network, and a six-layer ViT model was set as the student network. We set the numbers of prototypes as 24, 48, 72, and 96. Here, we only changed the number of prototypes and kept other network settings unchanged to evaluate the training of KD. Table 8 shows that the performance increases as the number of prototypes increases, and the best accuracy obtained is 78.32% when the number of prototypes is 72. It indicates that the small number of prototypes could not learn the generalized representation sufficiently. In other experiments, we set the number of prototypes in prototype-guided KD methods as 72.

4.3.3 Hyperparameters

We conducted ablation studies of hyperparameters in Eq. (7) on Foggy Cityscapes. As shown in Table 9, we set the loss weights λ_{emb}, λ_{pro}, and λ_{stu} with different values and obtained the object detection results. The ablation study results of the loss weights demonstrate the effectiveness of our prototype learning method in object detection KD. When λ_{emb} and λ_{pro} are disabled and λ_{stu} is set to 1.0, it represents the student model. When λ_{pro} is disabled and λ_{emb} and λ_{stu} are both set to 1.0, it represents the FBKD method. Our proposed method introduces λ_{pro}, and the results demonstrate a significant improvement of our method over the FBKD approach.

5 Conclusions

To solve the issue of applying a large-scale model to different downstream tasks, a prototype-guided cross-task KD method ProC-KD is proposed, wherein the label spaces of the teacher network and the student network are inconsistent. Specifically, the prototype-based representation learning module

Table 7 Ablation study results of design elements on Foggy Cityscapes

Method	PL	FA	Detection accuracy (%)								mAP(%)
			Bicycle	Bus	Car	Motorcycle	Person	Rider	Train	Truck	
Baseline	✗	✗	49.0	53.9	68.6	40.6	52.0	54.1	35.7	37.5	48.9
Ours	✓	✗	50.5	55.3	72.3	**45.4**	53.4	56.5	47.7	38.1	52.4
Ours	✓	✓	**51.5**	**57.7**	**73.1**	44.2	**53.8**	**57.9**	**51.2**	**40.3**	**53.7**

PL: prototype-based representation learning module; FA: feature augmentation module. The best results are in bold

Table 8 Ablation study results of our ProC-KD method with varying numbers of prototypes on the long-tailed CIFAR-100 dataset

Number	Accuracy (%)
24	77.75
48	78.11
72	78.32
96	78.28

Table 9 Ablation study of loss function hyperparameters on Foggy Cityscapes

Number	λ_{emb}	λ_{pro}	λ_{stu}	mAP (%)
Student			1.0	35.6
0	1.0		1.0	48.9
1	0.3	1.0	1.0	49.5
2	1.0	0.3	1.0	52.5
3	1.0	1.0	0.3	51.2
4	0.5	1.0	1.0	50.7
5	1.0	0.5	1.0	52.5
6	1.0	1.0	0.5	52.0
7	0.8	1.0	1.0	52.1
8	1.0	0.8	1.0	52.9
9	1.0	1.0	0.8	52.8
10	1.0	1.0	1.0	53.7

is trained to capture the invariant intrinsic local-level representations of objects, leveraging the robust capability of the teacher network. Then, the learned prototypes are used to augment the student network features to improve the generalization ability of the student network. We conduct experiments on image classification and object detection tasks, and the quantitative and qualitative results demonstrate the effectiveness of our ProC-KD for cross-task KD.

Contributors

Deng LI and Yahong HAN conceived the method. Deng LI and Aming WU designed and implemented the ProC-KD model. Deng LI and Peng LI processed the data. Deng LI drafted the paper. Peng LI, Aming WU, and Yahong HAN helped organize the paper. Yahong HAN supervised the project. Deng LI and Yahong HAN revised and finalized the paper.

Conflict of interest

Yahong HAN is a corresponding expert of *Frontiers of Information Technology & Electronic Engineering*, and he was not involved with the peer review process of this paper. All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are openly available in mmrazor (https://github.com/open-mmlab/mmrazor) and mdistiller (https://github.com/megvii-research/mdistiller). The other data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Ahn S, Hu SX, Damianou A, et al., 2019. Variational information distillation for knowledge transfer. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9163-9171. https://doi.org/10.1109/CVPR.2019.00938

Ba LJ, Caruana R, 2014. Do deep nets really need to be deep? Proc 27th Int Conf on Neural Information Processing Systems, p.2654-2662.

Cao KD, Wei CL, Gaidon A, et al., 2019. Learning imbalanced datasets with label-distribution-aware margin loss. Proc 33rd Int Conf on Neural Information Processing Systems, Article 140.

Carion N, Massa F, Synnaeve G, et al., 2020. End-to-end object detection with transformers. Proc 16th European Conf on Computer Vision, p.213-229. https://doi.org/10.1007/978-3-030-58452-8_13

Chebotar Y, Waters A, 2016. Distilling knowledge from ensembles of neural networks for speech recognition. Proc 17th Annual Conf of the Int Speech Communication Association, p.3439-3443.

Chefer H, Gur S, Wolf L, 2021. Transformer interpretability beyond attention visualization. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.782-791. https://doi.org/10.1109/CVPR46437.2021.00084

Chen DF, Mei JP, Zhang Y, et al., 2021. Cross-layer distillation with semantic calibration. Proc 35th AAAI Conf on Artificial Intelligence, p.7028-7036. https://doi.org/10.1609/aaai.v35i8.16865

Chen GB, Choi W, Yu X, et al., 2017. Learning efficient object detection models with knowledge distillation. Proc 31st Int Conf on Neural Information Processing Systems, p.742-751.

Chen YC, Li LJ, Yu LC, et al., 2020. UNITER: UNiversal Image-TExt Representation learning. Proc 16th European Conf on Computer Vision, p.104-120. https://doi.org/10.1007/978-3-030-58577-8_7

Cordts M, Omran M, Ramos S, et al., 2016. The Cityscapes dataset for semantic urban scene understanding. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.3213-3223. https://doi.org/10.1109/CVPR.2016.350

Cui Y, Jia ML, Lin TY, et al., 2019. Class-balanced loss based on effective number of samples. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.9268-9277. https://doi.org/10.1109/CVPR.2019.00949

Deng J, Dong W, Socher R, et al., 2009. ImageNet: a large-scale hierarchical image database. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.248-255. https://doi.org/10.1109/CVPR.2009.5206848

Deng JK, Guo J, Yang J, et al., 2021. Variational prototype learning for deep face recognition. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11906-11915. https://doi.org/10.1109/CVPR46437.2021.01173

Dosovitskiy A, Beyer L, Kolesnikov A, et al., 2021. An image is worth 16 × 16 words: transformers for image recognition at scale. Proc 9th Int Conf on Learning Representations.

Fu TJ, Li LJ, Gan Z, et al., 2023. An empirical study of end-to-end video-language transformers with masked visual modeling. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.22898-22909. https://doi.org/10.1109/CVPR52729.2023.02193

Gou JP, Yu BS, Maybank SJ, et al., 2021. Knowledge distillation: a survey. *Int J Comput Vis*, 129(6):1789-1819. https://doi.org/10.1007/s11263-021-01453-z

Gou JP, Sun LY, Yu BS, et al., 2023. Multilevel attention-based sample correlations for knowledge distillation. *IEEE Trans Ind Inform*, 19(5):7099-7109. https://doi.org/10.1109/TII.2022.3209672

Heo B, Kim J, Yun S, et al., 2019a. A comprehensive overhaul of feature distillation. Proc IEEE/CVF Int Conf on Computer Vision, p.1921-1930. https://doi.org/10.1109/ICCV.2019.00201

Heo B, Lee M, Yun S, et al., 2019b. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. Proc 33rd AAAI Conf on Artificial Intelligence, p.3779-3787. https://doi.org/10.1609/aaai.v33i01.33013779

Hinton G, Vinyals O, Dean J, 2015. Distilling the knowledge in a neural network. https://arxiv.org/abs/1503.02531

Hur S, Shin I, Park K, et al., 2023. Learning classifiers of prototypes and reciprocal points for universal domain adaptation. Proc IEEE/CVF Winter Conf on Applications of Computer Vision, p.531-540. https://doi.org/10.1109/WACV56688.2023.00060

Jain J, Li JC, Chiu MT, et al., 2023. OneFormer: one transformer to rule universal image segmentation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.2989-2998. https://doi.org/10.1109/CVPR52729.2023.00292

Jiao XQ, Yin YC, Shang LF, et al., 2020. TinyBERT: distilling BERT for natural language understanding. Proc Findings of the Association for Computational Linguistics, p.4163-4174. https://doi.org/10.18653/v1/2020.findings-emnlp.372

Kurata G, Saon G, 2020. Knowledge distillation from offline to streaming RNN transducer for end-to-end speech recognition. Proc 21st Annual Conf of the Int Speech Communication Association, p.2117-2121.

Li G, Jampani V, Sevilla-Lara L, et al., 2021. Adaptive prototype learning and allocation for few-shot segmentation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.8334-8343. https://doi.org/10.1109/CVPR46437.2021.00823

Li LJ, Chen YC, Cheng Y, et al., 2020. HERO: hierarchical encoder for video+language omni-representation pre-training. Proc Conf on Empirical Methods in Natural Language Processing, p.2046-2065. https://doi.org/10.18653/v1/2020.emnlp-main.161

Lin TY, Maire M, Belongie S, et al., 2014. Microsoft COCO: common objects in context. Proc 13th European Conf on Computer Vision, p.740-755. https://doi.org/10.1007/978-3-319-10602-1_48

Liu JL, Song L, Qin YQ, 2020. Prototype rectification for few-shot learning. Proc 16th European Conf on Computer Vision, p.741-756. https://doi.org/10.1007/978-3-030-58452-8_43

Liu Z, Lin YT, Cao Y, et al., 2021. Swin Transformer: hierarchical vision transformer using shifted windows. Proc IEEE/CVF Int Conf on Computer Vision, p.10012-10022. https://doi.org/10.1109/ICCV48922.2021.00986

Miles R, Mikolajczyk K, 2024. Understanding the role of the projector in knowledge distillation. Proc 38th AAAI Conf on Artificial Intelligence, p.4233-4241. https://doi.org/10.1609/aaai.v38i5.28219

Molchanov P, Tyree S, Karras T, et al., 2017. Pruning convolutional neural networks for resource efficient inference. Proc 5th Int Conf on Learning Representations.

Müller R, Kornblith S, Hinton G, 2019. When does label smoothing help? Proc 33rd Int Conf on Neural Information Processing Systems, Article 422.

Park W, Kim D, Lu Y, et al., 2019. Relational knowledge distillation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.3967-3976. https://doi.org/10.1109/CVPR.2019.00409

Passalis N, Tefas A, 2018. Learning deep representations with probabilistic knowledge transfer. Proc 15th European Conf on Computer Vision, p.283-299. https://doi.org/10.1007/978-3-030-01252-6_17

Rebuffi SA, Bilen H, Vedaldi A, 2017. Learning multiple visual domains with residual adapters. Proc 31st Int Conf on Neural Information Processing Systems, p.506-516.

Romero A, Ballas N, Kahou SE, et al., 2015. FitNets: hints for thin deep nets. Proc 3rd Int Conf on Learning Representations.

Sakaridis C, Dai DX, Van Gool L, 2018. Semantic foggy scene understanding with synthetic data. *Int J Comput Vis*, 126(9):973-992. https://doi.org/10.1007/s11263-018-1072-8

Sanh V, Debut L, Chaumond J, et al., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. https://arxiv.org/abs/1910.01108

Shu CY, Liu YF, Gao JF, et al., 2021. Channel-wise knowledge distillation for dense prediction. Proc IEEE/CVF Int Conf on Computer Vision, p.5311-5320. https://doi.org/10.1109/ICCV48922.2021.00526

Snell J, Swersky K, Zemel R, 2017. Prototypical networks for few-shot learning. Proc 31st Int Conf on Neural Information Processing Systems, p.4080-4090.

Sun SQ, Cheng Y, Gan Z, et al., 2019. Patient knowledge distillation for BERT model compression. Proc Conf on Empirical Methods in Natural Language Processing and the 9th Int Joint Conf on Natural Language Processing, p.4322-4331. https://doi.org/10.18653/v1/D19-1441

Touvron H, Cord M, Douze M, et al., 2021. Training data-efficient image transformers & distillation through attention. Proc 38th Int Conf on Machine Learning, p.10347-10357.

van der Maaten L, Weinberger K, 2012. Stochastic triplet embedding. Proc IEEE Int Workshop on Machine Learning for Signal Processing, p.1-6. https://doi.org/10.1109/MLSP.2012.6349720

Vaswani A, Shazeer N, Parmar N, et al., 2017. Attention is all you need. Proc 31st Int Conf on Neural Information Processing Systems, p.6000-6010.

Venkateswara H, Eusebio J, Chakraborty S, et al., 2017. Deep hashing network for unsupervised domain adaptation. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5018-5027. https://doi.org/10.1109/CVPR.2017.572

Wang JH, Cao MD, Shi SW, et al., 2022. Attention probe: vision transformer distillation in the wild. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.2220-2224. https://doi.org/10.1109/ICASSP43922.2022.9747484

Wang T, Yuan L, Zhang XP, et al., 2019. Distilling object detectors with fine-grained feature imitation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4933-4942. https://doi.org/10.1109/CVPR.2019.00507

Wei YJ, Ye JX, Huang ZZ, et al., 2023. Online prototype learning for online continual learning. Proc IEEE/CVF Int Conf on Computer Vision, p.18764-18774. https://doi.org/10.1109/ICCV51070.2023.01720

Wu AM, Liu R, Han YH, et al., 2021. Vector-decomposed disentanglement for domain-invariant object detection. Proc IEEE/CVF Int Conf on Computer Vision, p.9342-9351. https://doi.org/10.1109/ICCV48922.2021.00921

Wu JX, Leng C, Wang YH, et al., 2016. Quantized convolutional neural networks for mobile devices. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4820-4828. https://doi.org/10.1109/CVPR.2016.521

Yang ZD, Li Z, Jiang XH, et al., 2022a. Focal and global knowledge distillation for detectors. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.4643-4652. https://doi.org/10.1109/CVPR52688.2022.00460

Yang ZD, Li Z, Shao MQ, et al., 2022b. Masked generative distillation. Proc 17th European Conf on Computer Vision, p.53-69. https://doi.org/10.1007/978-3-031-20083-0_4

Ye HJ, Lu S, Zhan DC, 2020. Distilling cross-task knowledge via relationship matching. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.12396-12405. https://doi.org/10.1109/CVPR42600.2020.01241

Ye LW, Rochan M, Liu Z, et al., 2019. Cross-modal self-attention network for referring image segmentation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.10502-10511. https://doi.org/10.1109/CVPR.2019.01075

Yim J, Joo D, Bae J, et al., 2017. A gift from knowledge distillation: fast optimization, network minimization and transfer learning. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.7130-7138. https://doi.org/10.1109/CVPR.2017.754

Yoon JW, Lee H, Kim HY, et al., 2021. TutorNet: towards flexible knowledge distillation for end-to-end speech recognition. *IEEE/ACM Trans Audio Speech Lang Process*, 29:1626-1638. https://doi.org/10.1109/TASLP.2021.3071662

Zagoruyko S, Komodakis N, 2017. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. Proc 5th Int Conf on Learning Representations.

Zhang LF, Ma KS, 2023. Structured knowledge distillation for accurate and efficient object detection. *IEEE Trans Patt Anal Mach Intell*, 45(12):15706-15724. https://doi.org/10.1109/TPAMI.2023.3300470

Zhang Y, Chen WH, Lu YC, et al., 2023. Avatar knowledge distillation: self-ensemble teacher paradigm with uncertainty. Proc 31st ACM Int Conf on Multimedia, p.5272-5280.

Zhao BR, Cui Q, Song RJ, et al., 2022. Decoupled knowledge distillation. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.11953-11962. https://doi.org/10.1109/CVPR52688.2022.01165

Zhou C, Zhang YN, Chen JX, et al., 2023. OcTr: octree-based transformer for 3D object detection. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.5166-5175. https://doi.org/10.1109/CVPR52729.2023.00500

Zhu MH, Gupta S, 2018. To prune, or not to prune: exploring the efficacy of pruning for model compression. Proc 6th Int Conf on Learning Representations.

Zhu SL, Shang RH, Tang K, et al., 2023. BookKD: a novel knowledge distillation for reducing distillation costs by decoupling knowledge generation and learning. *Knowl-Based Syst*, 279:110916. https://doi.org/10.1016/j.knosys.2023.110916

Zhu XZ, Su WJ, Lu LW, et al., 2021. Deformable DETR: deformable transformers for end-to-end object detection. Proc 9th Int Conf on Learning Representations.