

Received 21 February 2024, accepted 8 April 2024, date of publication 11 April 2024, date of current version 19 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3387859

## RESEARCH ARTICLE

# Bridging the Knowledge Gap via Transformer-Based Multi-Layer Correlation Learning

HUN-BEOM BAK AND SEUNG-HWAN BAE<sup>id</sup>, (Member, IEEE)

Vision and Learning Laboratory, Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea

Corresponding author: Seung-Hwan Bae (shbae@inha.ac.kr)

This work was supported by Inha University Research Grant.

**ABSTRACT** We tackle a multi-layer knowledge distillation problem between deep models with heterogeneous architectures. The main challenges of that are the mismatches of the feature maps in terms of the resolution or semantic levels. To resolve this, we propose a novel transformer-based multi-layer correlation knowledge distillation (TMC-KD) method in order to bridge the knowledge gap between a pair of networks. Our method aims to narrow the relational knowledge gaps between teacher and student models by minimizing the local and global feature correlations. Based on extensive comparisons with the recent KD methods on classification and detection tasks, we prove the effectiveness and usefulness of our TMC-KD method.

**INDEX TERMS** Correlation learning, image classification, knowledge distillation, model compression, object detection, transformer-based learning.

## I. INTRODUCTION

Over the decades, deep neural networks show promising performance on many down streaming vision tasks such as image classification and object detection. Recently, there are many efforts to apply the powerful deep models for small or embedded devices which have limited hardware resources. To achieve this, one of the common approaches is to reduce model size while preserving its learned knowledge at most using pruning [1], quantization [2], and knowledge distillation (KD) [3]. In particular, KD methods less suffer from accuracy degradation and complex training than others.

The vanilla KD method developed by [3] allows smaller student models to mimic the representation of a larger teacher model. This can be achieved by aligning the output responses (*e.g.* logits and predictions) of both models. However, the transferring of the output knowledge of the teacher often achieves marginal improvement only due to the limited distillation of representations in the mid-level layers. Therefore,

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose<sup>id</sup>.

there are attempts to align the intermediate knowledge between teacher and student networks for transferring more knowledge. For instance, FitNets [4] adds the hint training procedure for distillation of the selected intermediate layers of a teacher. SemCKD [5] distills the correlation features using the attention learning. Inspired by these works, our work is also based on the distillation of the mid-level features as well as output features.

In this work, we assume that relational knowledge within feature maps of the teacher network is beneficial and should be transferred to the student network. Unfortunately, the most recent KD works with the knowledge of the intermediate layers [5] less pay attention to this point. In some works using the relational structure knowledge, it is the essential knowledge needed to be transferred to the student model for improving robustness or accuracy. For achieving this, feature distance-wise or angle-wise losses are exploited in [6]. Inter-channel correlation [7] of features is learned to capture feature intrinsic distributions of a teacher model. However, cross correlations between multiple features of teacher and student models are not leveraged well as done in the multi-layer KD methods. Therefore, our work aims to transfer the

relational knowledge of a teacher model and align multi-layer feature maps of different models.

In order to achieve that, a powerful KD model which can capture both correlations is required, and we present a novel transformer-based multi-layer correlation learning for knowledge distillation (TMC-KD). One of the main issues of applying the transformer for KD is how to encode the multi-level feature tensors with different dimensionality. To resolve this, we present a multi-layer feature converter (MLC) that can transform the different-level features into a series of encoded features. Based on the multi-head attention learning of the transformer, we can then produce the decoded features by feeding the serially encoded features to the transformer.

In order to align the knowledge level between teacher and student networks across mid-level layers, we learn the layer-wise matched local correlation with the similarity between the teacher-student decoded features. Then, we minimize the local semantic gap between the internal layers with the learned local correlation. Moreover, we reduce their global knowledge gap by minimizing the self-correlation discrepancy of the whole decoded features.

To prove the effectiveness, we compare our TMC-KD with the recent KD methods on CIFAR-100 and ImageNet datasets. Our TMC-KD method offers greater accuracy improvements than other KD methods on both sets for most student models regardless of its architecture. In addition, we provide the ablation study to show the usefulness of each method.

To sum up, our contributions are

- We propose a novel transformer-based multi-layer correlation learning for the relational knowledge distillation across intermediate layers.
- We design the multi-layer feature converter to transform multi-level features into sequentially-encoded ones and use them as the inputs of the transformer.
- We present global and local correlation learning for bridging their local and global knowledge gaps.

In Section II, we present related works to the knowledge distillation. In Section III, we discuss our TMC-KD method composed of a multi-layer feature converter, local semantic learning, and global relational learning. Section IV provides experimental set-up and results. The conclusion is made in Section V.

## II. RELATED WORKS

### A. MULTI-LAYER KNOWLEDGE DISTILLATION

As one of the pioneer works, Hinton et.al. [3] proposes a simple knowledge distillation by minimizing the distance between teacher and student outputs. Distilling only model outputs is effective, but it shows some insufficient results for many tasks. Using multiple teachers [8] and teacher assistants [9], [10] improve the generalization, robustness, and accuracy of the student model. Applying curriculum learning [11] and generating virtual distribution [12] also improve student models. For distilling more teacher

knowledge within other layers, some works present KD methods using intermediate-layer feature maps. The existing works can be categorized into based-on the local correlation learning [13], [14], [15], [16], [17] between teacher-student features and relational correlation learning [6], [7], [18] of the model itself. In the former works, they use the local correlation between the matched layer features of the teacher and student models. FitNets [4] minimize between L2 distance of teacher-student intermediate features. Attention-guided KD methods [16], [19], [20] for object classes are introduced for transferring more knowledge of the crucial regions. VID [21] maximizes the mutual information between teacher-student intermediate features. SimKD [15] reuses a pre-trained teacher classifier.

One common limitation of these methods requires the prior knowledge of the target layers to be distilled within teacher and student models. To overcome this, some KD methods [22], [23] solve the layer assignment problem by using the attention mechanism. In specific, SemCKD [5] and ASM [13] calculate the correlations across intermediate layers.

On the other hand, in the latter approach, a student model tries to learn the relational representation of a teacher. [24] defines the flow of solution procedure (FSP) matrix to distill the flow knowledge between sequential layers. RKD [6] learns the relational knowledge of data samples in terms of the angles and distances. In [18], a student is learned to mimic the activation patterns of similar training samples for a teacher. ICKD [7] computes the inter-channel correlation by capturing feature diversity and homology. LSL [25] defines the inter-class and inter-layered Gram matrices to evaluate the diversity and discrimination of feature maps.

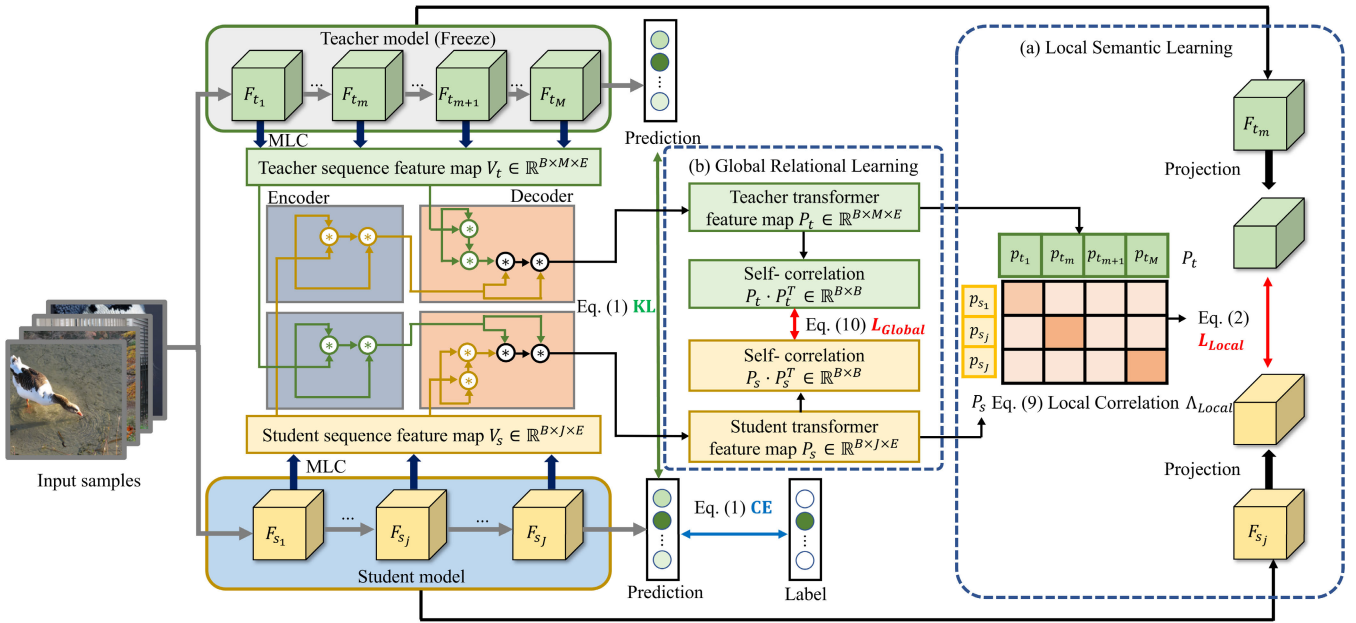
Since both approaches show promising results and can be complementary to each other, we present a transformer-based KD method to combine both approaches. This allows us to transfer important knowledge related to internal feature relationship within a model and cross-correlations between models to the student, resulting in improved performance for the target student.

### B. TRANSFORMER-BASED KNOWLEDGE DISTILLATION

Even though a transformer [26] came up with for natural language processing, its variant models achieved remarkable performance for other vision tasks [27], [28]. In addition, the KD methods [29], [30] which distill a bunch of internal knowledge of a teacher transformer have been developed. A target-aware transformer [14] transfers the spatial semantic knowledge of a teacher via one-to-all spatial matching. In this work, we exploit a transformer for learning global and local correlation between many-to-many matching layers between different models rather than using it as a teacher itself [29], [30].

## III. METHOD

We first discuss the preliminary for KD and multi-layer based KD methods. We then explain our TMC-KD



**FIGURE 1.** The overall architecture of a transformer-based multi-layer knowledge distillation (TMC-KD) mainly consisting of (a) local semantic learning and (b) global relational learning parts is described in Sec. III-B. Local semantic learning minimizes the discrepancy between intermediate-layer features using the learned local correlation, but global semantic learning reduces the gaps of decoded features from a transformer using self-attention.

method that can reduce local semantic and global relation gaps between models using our multi-layer feature converter.

## A. PRELIMINARY

### 1) KNOWLEDGE DISTILLATION

We denote  $f_t$  and  $f_s$  as teacher and student models, respectively. In general, the pre-trained  $f_t$  with more parameters is superior to  $f_s$  with less one. Then, the goal of the knowledge distillation is to improve the  $f_s$  by transferring the core knowledge of  $f_t$  on a dataset  $D = \{(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})\}_{i=1}^N$  with  $N$  samples. Here, for the image classification task, each sample consists of an image  $\mathbf{x}^{(i)}$  and its label  $\mathbf{y}^{(i)}$ . In general, in Hinton et. al. KD [3], the knowledge gaps between  $f_t$  and  $f_s$  are reduced by minimizing the cross entropy (CE) between the student predicted label  $\sigma(\mathbf{z}_s^i)$  from the softmax layer  $\sigma(\cdot)$  with an input of the logits  $\mathbf{z}_s^i = f_s(\mathbf{x}^{(i)})$  and the one-hot encoded target label  $\mathbf{y}^{(i)}$  for the image  $\mathbf{x}^{(i)}$ . In addition, the Kullback-Leibler (KL) divergence between the teacher and student predicted probabilities  $\sigma(\mathbf{z}_t^i)$  and  $\sigma(\mathbf{z}_s^i)$  is added as the total KD loss as follows:

$$L_{KD} = \sum_{i=1}^N (CE(\sigma(\mathbf{z}_s^i), \mathbf{y}^{(i)}) + \tau^2 KL(\sigma(\mathbf{z}_s^i/\tau), \sigma(\mathbf{z}_t^i/\tau))) \quad (1)$$

where  $\tau$  is a temperature factor and controls the softness of the outputs. Since the KD method aims at reducing the output predictions of both models, the multi-layer KD methods introduce the additional losses to reduce the discrepancy of the mid-level features.

### 2) MULTI-LAYER KNOWLEDGE DISTILLATION

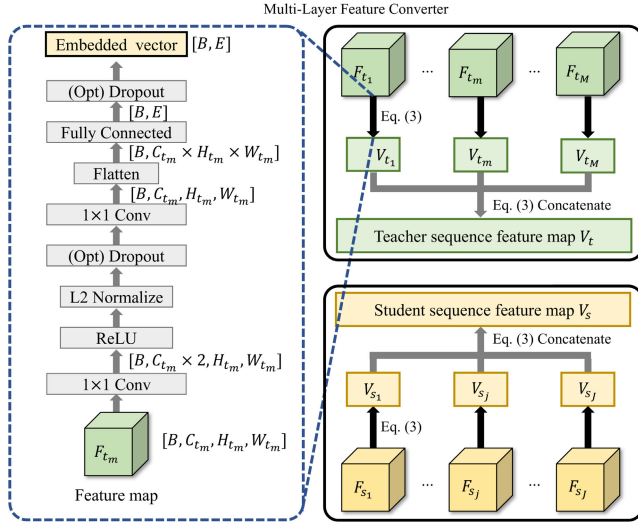
Since the dimension of a feature map at each layer for a model is usually different, we denote  $F_t^m \in \mathbb{R}^{B \times C_t^m \times H_t^m \times W_t^m}$  and  $F_s^j \in \mathbb{R}^{B \times C_s^j \times H_s^j \times W_s^j}$  as the  $m$ -th and  $j$ -th feature maps for the teacher  $t$  and student  $s$ , where  $H$  and  $W$  are the height and width of the feature map, and  $B$  and  $C$  are the cardinality of the batches and channels.  $M$  and  $J$  indicate the number of teacher and student layers. Then, the multi-layer KD [4], [5] reduce the feature gaps across layers between both by minimizing the total mean square error as:

$$L_{Local} = \sum_{j=1}^J \sum_{m=1}^M \Lambda_{Local}^{j,m} \left\| \phi_s(F_s^j) - \phi_t(F_t^m) \right\|_2^2 \quad (2)$$

where  $\phi_s(\cdot)$  and  $\phi_t(\cdot)$  are transformation functions to match the channel number and feature resolution between  $F_s^j$  and  $F_t^m$ . Different from papers [4] the multi-layer KD methods [5] avoid the usage of the prior knowledge to be an associated pair by learning the local correlation  $\Lambda_{Local}^{j,m}$ . For instance, SemCKD [5] computes the correlation  $\Lambda_{Local}^{j,m}$  between  $m$ -th teacher and  $j$ -th student model features with the query-key attention learning.

## B. TRANSFORMER-BASED MULTI-LAYER CORRELATION LEARNING

The limitation of the the multi-layer KD methods discussed above is that they do not leverage the relational knowledge between the feature maps within a teacher network. However, in most cases, there is some relationship between multi-layer features since the output feature at the preceding layer is used as an input of the succeeding layer. We conjecture that the



**FIGURE 2.** The structure of multi-layer feature converter (MLC). The parameter size of the Conv and FC layers can be tuned by the dimensionality of an input feature. Thus, it can produce the output encoded feature  $V_{t_m}$  and  $V_{s_j}$  with the same dimensionality.

feature relational knowledge within a teacher model should be transferred to the student. In particular, we strive to transfer the strong global relation among features of the teacher to the student, and present a transformer-based multi-layer KD to achieve this. More concretely, we encode multi-layer features of teacher and student models to the sequential features for compensating feature dimension mismatches. Then, we generate the decoded features of each model by exploiting the encoded features as keys or queries alternately. Subsequently, we learn local correlation between the decoded features and use it for local and global relational learning. In the local semantic learning, we use the local correlation as  $\Lambda_{Local}^{j,m}$  of the Eq. (2), and minimize the decoded feature discrepancy. On the other hand, the global relation learning allows the student to mimic the global representation across all feature maps within the teacher model.

### 1) MULTI-LAYER FEATURE CONVERTER

Due to the mismatches of intermediate features between teacher and student models, we design a multi-layer feature converter (MLC) before feeding each feature to the transformer. As shown in Fig. 2, our MLC consists of two  $1 \times 1$  Conv, ReLU, normalization layer, and fully-connected layer. By applying each MLC  $\psi(\cdot)$  to each  $F_{s_j}$  and  $F_{t_m}$ , we can produce the encoded features  $V_{t_m} \in \mathbb{R}^{B \times E}$  and  $V_{s_j} \in \mathbb{R}^{B \times E}$  with the same dimensionality  $E$ :  $V_{t_m} = \psi_{t_m}(F_{t_m})$  and  $V_{s_j} = \psi_{s_j}(F_{s_j})$ . We then concatenate features of all the layers  $\{V_{t_m}\}_{m=1}^M$  and  $\{V_{s_j}\}_{j=1}^J$  to obtain the sequentially encoded features  $V_t$  and  $V_s$  using

$$\begin{aligned} V_t &= \text{Concat}(V_{t_1}, \dots, V_{t_m}, \dots, V_{t_M}) \\ V_s &= \text{Concat}(V_{s_1}, \dots, V_{s_j}, \dots, V_{s_J}) \end{aligned} \quad (3)$$

Then, we use the  $V_t$  and  $V_s$  as the inputs of the transformer as described in the next section.

### 2) LOCAL SEMANTIC LEARNING

Because a transformer [26] is a powerful way to learn global feature correlation as mentioned, we use it for our multi-layer KD. Followed by the implementation [26], we design an encoder  $Enc$  and a decoder  $Dec$  of a stack of  $N_E = 6$  identical layers. By feeding the sequentially-encoded features  $V_t$  and  $V_s$  in Eq. (3) to the transformer, we can produce the decoded features  $P_t \in \mathbb{R}^{B \times M \times E}$  and  $P_s \in \mathbb{R}^{B \times J \times E}$  as:

$$\begin{aligned} P_t &= \text{Dec}(\text{Enc}(V_s), V_t) \\ P_s &= \text{Dec}(\text{Enc}(V_t), V_s) \end{aligned} \quad (4)$$

For describing the encoding process in  $Enc$ , we denote  $W_Q \in \mathbb{R}^{E \times E}$  and  $W_K \in \mathbb{R}^{E \times E}$  as the query and key weight matrices. Then, we learn the global correlation  $C_{Global}$  within  $V_t$  or  $V_s$  by the self-attention mechanism with matrix multiplication (\*) as

$$C_{Global}(V_q) = \frac{(W_Q * V_q) * (W_K * V_q)^T}{\sqrt{E}} \quad (5)$$

where  $V_q$  can be  $V_t$  or  $V_s$ . We then learn  $h$ -th head attention features  $H_{Enc}^h$  of  $N_H = 8$  multiple attention heads by applying the global correlation  $C_{Global}$ :

$$H_{Enc}^h(C_{Global}, V_q, W_V) = C_{Global} * (W_V * V_q) \quad (6)$$

where  $W_V \in \mathbb{R}^{E \times E}$  is the learned weight matrix for  $V_q$ . The multi-head attention consists of concatenating heads, additional weight, residual connection, and layer normalization. Then, we obtain  $e$ -th encoder outputs  $F_{Enc}^e$  by applying two feed-forward networks and a single activation function, and layer normalization. Each encoder layer output  $F_{Enc}^e$  is fed into the next encoder layer subsequently. Then, we represent the output of the last encoder layer as  $F_{Enc}^{N_E}$ .

In the decoder  $Dec$ ,  $V_p$ , which can be  $V_t$  or  $V_s$ , is fed into the self-attention, and its output  $F_{Self}^{V_p}$  and encoder output  $F_{Enc}^{N_E}$  are fed into cross-attention. Using multi-head attention with Eq. (5) and (6), we produce the enhanced feature  $F_{Self}^{V_p}$  for  $V_p$ . Then, we compute the cross-attention  $C_{Cross}$  with output of encoder  $F_{Enc}^{N_E}$  and  $F_{Self}^{V_p}$ :

$$C_{Cross}(F_{Self}^{V_p}, F_{Enc}^{N_E}) = \frac{(W_Q * F_{Self}^{V_p}) * (W_K * F_{Enc}^{N_E})^T}{\sqrt{E}} \quad (7)$$

The cross attention-applied feature map  $H_{Cross}^h$  is given as:

$$H_{Cross}^h(C_{Cross}, W_V, F_{Self}^{V_p}) = C_{Cross} * (W_V * F_{Self}^{V_p}) \quad (8)$$

Similar to the encoder, we feed each  $H_{Cross}^h$  to the next decoder for  $N_E - 1$  steps, and denote  $P_t$  or  $P_s$  as the outputs of the last decoder for the teacher and student.

Basically, we can minimize the local semantic gap between models by Eq. (2). However, in our TMC-KD, we use the decoded features  $P_t$  and  $P_s$  for evaluating  $\Lambda_{Local}^{j,m}$ . To this end, we first slice  $P_t \in \mathbb{R}^{B \times M \times E}$  and  $P_s \in \mathbb{R}^{B \times J \times E}$  to

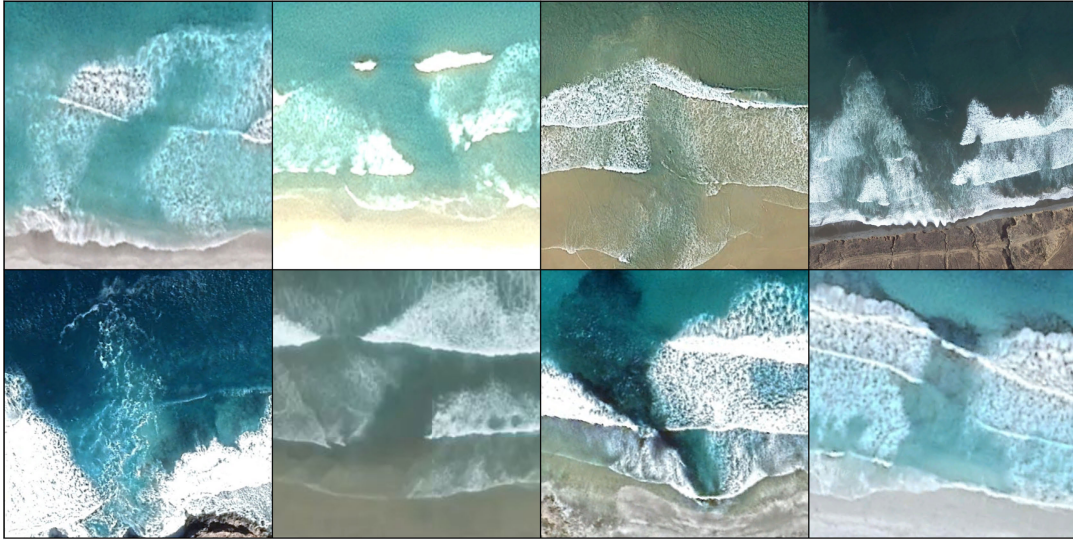


FIGURE 3. Examples for rip current dataset .

---

**Algorithm 1** Transformer-Based Multi-Layer Correlation Knowledge Distillation (TMC-KD)

---

**Input:** Training dataset  $D$ , a pre-trained teacher  $f_t$  with parameter  $\theta_t$ , a student  $f_s$  with parameter  $\theta_s$

**Output:** A trained  $f_s$

- 1: **while**  $\theta_s$  is not converged **do**
  - 2:   Extract feature maps  $F_{t_m}$  and  $F_{s_j}$  by feeding a mini-batch sampled from  $D$  to  $f_t$  and  $f_s$ .
  - 3:   Compute  $L_{KD}$  by Eq. (1)
  - 4:   // **Multi-layer feature converter**
  - 5:   Convert  $F_{t_m}$  and  $F_{s_j}$  to  $V_{t_m}$  and  $V_{s_j}$  using  $\psi(\cdot)$ .
  - 6:   Compute  $V_t$  and  $V_s$  by Eq. (3).
  - 7:   // **Local semantic learning**
  - 8:   Compute  $P_t$  and  $P_s$  by Eq. (4).
  - 9:   Slicing  $P_t$  and  $P_s$  to obtain  $\{p_{t_m}\}_{m=1}^M$  and  $\{p_{s_j}\}_{j=1}^J$ .
  - 10:   Compute  $\Lambda_{Local}^{j,m}$  by Eq. (9).
  - 11:   Compute  $L_{Local}$  by Eq. (2).
  - 12:   // **Global relational learning**
  - 13:   Compute self-correlations of  $P_t$  and  $P_s$ .
  - 14:   Compute  $L_{Global}$  by Eq. (10).
  - 15:   Compute  $L_{Total}$  by Eq. (11).
  - 16:   Update  $\theta_s$  by minimizing Eq. (11).
  - 17: **end while**
- 

$\{p_{t_m} \in \mathbb{R}^{B \times E}\}_{m=1}^M$  and  $\{p_{s_j} \in \mathbb{R}^{B \times E}\}_{j=1}^J$ . We then learn the local correlation  $\Lambda_{Local}^{j,m}$  in the form of the softmax with the dot-product attention as

$$\Lambda_{Local}^{j,m} = \frac{\exp(p_{t_m} \cdot p_{s_j}^T)}{\sum_{i=1}^M \sum_{j=1}^J \exp(p_{t_m} \cdot p_{s_j}^T)}. \quad (9)$$

### 3) GLOBAL RELATIONAL LEARNING

We make the student learn the global feature dependency within the teacher model as well as the global feature

correlation between them. Since both decoded  $P_t$  and  $P_s$  are the refined-attention features from the global and local correlation learning Eq. (4), they could include the global relational information and use them for this. To transfer this knowledge, we define  $L_{Global}$  with the self-correlation of each decode feature as

$$L_{Global} = \left\| P_t \cdot P_t^T - P_s \cdot P_s^T \right\|_2^2 \quad (10)$$

By minimizing Eq. (10), we make the student mimic the global representation of the teacher. Finally, we define a total TMC-KD loss including local and global losses from Eq. (2) and Eq. (10):

$$L_{Total} = L_{KD} + \zeta L_{Global} + \beta L_{Local} \quad (11)$$

where  $\beta$  and  $\zeta$  are balancing parameters to be tuned experimentally.

In the Figure 5, we study sensitivity of hyper-parameter  $\zeta$  and  $\beta$ . Our training process is described in Algorithm 1.

## IV. EXPERIMENTS

In this section, we have evaluated our TMC-KD by comparing the recent KD methods. We also provide the ablation study and sensitive analysis for proving our method.

### A. DATASET

We exploit the CIFAR-100 [31] and ImageNet [32] datasets for classification. We report the accuracy of methods in terms of Top-1 accuracy. So, a higher Top-1 score indicates better results.

The CIFAR-100 has 100 classes and each class consists of 500 training samples and 100 test samples. All samples of CIFAR-100 have  $32 \times 32$  resolution. The ImageNet has 1.2 M images and 1,000 object classes. All the images are resized to  $224 \times 224$  during training and testing.

We have evaluated our TMC-KD for a more challenging object detection problem, and compared the KD performance

**TABLE 1.** Comparison results with the recent multi-layer KD methods on CIFAR-100. \* and  $\diamond$  results are in [5] and [14], respectively. The best results are marked with bold.

Teacher	ResNet-32x4	ResNet-32x4	VGG-13	ResNet-32x4	WRN-40-2	VGG-13	ResNet-32x4
	79.42	79.42	74.64	79.42	75.61	74.64	79.42
Student	VGG-8	VGG-13	ShuffleNetV2	ShuffleNetV2	MobileNetV2	VGG-8	ResNet-8x4
	70.46 $\pm$ 0.29	74.82 $\pm$ 0.22	72.60 $\pm$ 0.12	72.60 $\pm$ 0.12	65.43 $\pm$ 0.29	70.46 $\pm$ 0.29	73.09 $\pm$ 0.30
<b>KD (NIPS-14) [3]</b> *	72.73 $\pm$ 0.15	77.17 $\pm$ 0.11	75.60 $\pm$ 0.21	75.49 $\pm$ 0.24	68.70 $\pm$ 0.22	73.38 $\pm$ 0.05	74.42 $\pm$ 0.05
<b>FitNet (ICLR-15) [4]</b> *	72.91 $\pm$ 0.18	77.06 $\pm$ 0.14	75.44 $\pm$ 0.11	75.82 $\pm$ 0.22	68.64 $\pm$ 0.12	73.63 $\pm$ 0.11	74.32 $\pm$ 0.08
<b>AT (ICLR-17) [19]</b> *	71.90 $\pm$ 0.13	77.23 $\pm$ 0.19	75.41 $\pm$ 0.10	75.91 $\pm$ 0.14	68.79 $\pm$ 0.13	73.51 $\pm$ 0.08	75.07 $\pm$ 0.03
<b>SP (CVPR-19) [18]</b> *	73.12 $\pm$ 0.10	77.72 $\pm$ 0.33	75.54 $\pm$ 0.18	75.77 $\pm$ 0.08	68.48 $\pm$ 0.36	73.53 $\pm$ 0.23	74.29 $\pm$ 0.07
<b>VID (CVPR-19) [21]</b> *	73.19 $\pm$ 0.23	77.45 $\pm$ 0.13	75.22 $\pm$ 0.07	75.55 $\pm$ 0.18	68.37 $\pm$ 0.24	73.63 $\pm$ 0.07	74.55 $\pm$ 0.10
<b>HKD (CVPR-20) [33]</b> *	72.63 $\pm$ 0.12	76.76 $\pm$ 0.13	76.24 $\pm$ 0.09	76.64 $\pm$ 0.05	69.23 $\pm$ 0.16	73.06 $\pm$ 0.24	74.86 $\pm$ 0.21
<b>SemCKD (AAAI-21) [5]</b> *	75.27 $\pm$ 0.13	79.43 $\pm$ 0.02	76.39 $\pm$ 0.12	77.62 $\pm$ 0.32	69.61 $\pm$ 0.05	74.43 $\pm$ 0.25	76.23 $\pm$ 0.04
<b>TaT (CVPR-22) [14]</b> $\diamond$	N/A	N/A	N/A	N/A	N/A	74.35	75.54
<b>TMC-KD (ours)</b>	<b>76.23</b>	<b>79.83</b>	<b>76.91</b>	<b>77.88</b>	<b>70.03</b>	<b>75.10</b>	<b>76.63</b>

with other KD methods. For this comparison, we use the Rip currents detection dataset [34]. The dataset contains 1,600 annotated images with rip currents and 700 images without rip currents. For training, we use 1,200 images with rip currents and 700 images without rip currents. For evaluation, we use 400 images with rip currents. Since the appearances of the rip currents are various, it is very challenging for the target student detector to learn the generalized features of the rip currents. Therefore, we use a teacher detector with the stronger backbone (R-101) for this KD comparison.

## B. IMPLEMENTATION DETAILS

The architecture of our TMC-KD mainly consists of the multi-layer converter and the transformer. The details of the MLC structure is described in Sec. III-B. We set a batch size  $B$  to 64 or 256 for the CIFAR-100 or ImageNet. The embedding size  $E$  is tuned to 16. When implementing the transformer, we follow the implementation of the original version [26]. Therefore, the transformer is composed of the stack of 6 encoders and 6 decoders. We also use 8 parallel attention heads. For the teacher and student models, we use the various CNN models such as ResNet [35], VGG [36], ShuffleNet [37], WRN [38], MobileNet [39] with different model combinations.

In the KD loss Eq. (1), we set the temperature factor  $\tau$  to 4. For finding optimal  $\beta$  and  $\zeta$  used in the total KD loss Eq. (11), we perform the sensitive analysis in Sec. IV-F, and set  $\beta$  and  $\zeta$  to 50 and 0.1. For KD training, we use the SGD optimizer with Nesterov momentum. For CIFAR-100, the initial learning rate is 0.01 for the variants of MobileNets and ShuffleNets. Otherwise, it is set to 0.05. We train models during 240 epochs and decay the learning rates by 0.1 times at the 150, 180, and 210 epochs. For ImageNet, we train models during 100 epochs. We set the initial learning rate to 0.1, and decay it by a factor of 0.1 at the 30, 60, and 90 epochs. We implement all the KD methods by using the same HW/SW: Intel-Xeon@2.40GHz, Titan-V, and PyTorch (v1.10).

For implementing our TMC-KD knowledge distillation, we use the MMRazor. For the KD between teacher and

**TABLE 2.** Comparison with the relation-based KD methods on CIFAR-100. The PKT (ECCV-18), RKD (CVPR-19), IRG (CVPR-19), CC (ICCV-19) and CRD (ICLR-21) results marked with \* are in [5]. The ICKD (ICCV-21) result marked with † is in [7].

Teacher	ResNet-32x4	WRN-40-2	ResNet-32x4
	79.42	75.61	79.42
Student	ResNet-8x4	MobileNetV2	VGG-8
	70.46 $\pm$ 0.29	65.43 $\pm$ 0.29	73.09 $\pm$ 0.30
<b>PKT [43]</b> *	73.11 $\pm$ 0.21	68.68 $\pm$ 0.29	74.61 $\pm$ 0.25
<b>RKD [6]</b> *	72.49 $\pm$ 0.08	68.71 $\pm$ 0.20	74.36 $\pm$ 0.23
<b>IRG [44]</b> *	72.57 $\pm$ 0.20	68.83 $\pm$ 0.18	74.67 $\pm$ 0.15
<b>CC [45]</b> *	72.63 $\pm$ 0.30	68.68 $\pm$ 0.14	74.50 $\pm$ 0.13
<b>CRD [46]</b> *	73.54 $\pm$ 0.19	69.98 $\pm$ 0.27	75.59 $\pm$ 0.07
<b>ICKD [7]</b> †	75.48	N/A	N/A
<b>TMC-KD</b>	<b>76.63</b>	<b>70.03</b>	<b>76.23</b>

student detectors, we compare the outputs of the feature pyramid networks (FPNs) since our method focuses on the multi-layer KD. More specifically, we use the outputs of FPN as inputs of multi-layer converter (MLC) and use the output of MLC as inputs of transformer for global and local correlation.

For more comparisons, we perform the KD between both detectors using CWD [40], FBKD [41], and PKD [42]. In the CWD-based implementation, we minimize the Kullback-Leibler divergence between teacher and student activation maps from FPNs. In the FBKD, we extract both spatial and channel attentions from teacher and student detectors. We then minimize each mean square error between teacher and student attention maps. In the PKD, we normalize outputs of FPNs and minimize the mean square error between teacher and student normalized features. While other methods perform KD between same-level layers, our TMC-KD compares features among all FPN layers using the local correlation Eq.(9).

## C. COMPARISON ON CIFAR-100 AND IMAGENET

We compared our TMC-KD with KD [3] and multi-layer KD methods: FitNets [4], AT [19], SP [18], VID [21], HKD [33], SemCKD [5], ICKD [7], and TaT [14]. We also compared the relational knowledge KD methods: PKT [43], RKD [6], IRG [44], CC [45], CRD [46], and ICKD [7].

**TABLE 3.** Results on knowledge distillation on rip current detection. The bold highlights the best results and the underline indicates the second best results. ‡ represents our implementation result.

Method	AP@0.5:0.95	AP@0.5	AP@0.75	AR@0.5:0.95
Teacher (R-101) ‡	42.9	<b>93.0</b>	29.6	50.2
Student (R-18) ‡	39.9	88.1	25.6	49.0
CWD (ICCV-21) [40] ‡	43.3	91.0	32.5	51.0
FBKD (ICLR-21) [41] ‡	43.6	91.7	33.7	<b>51.9</b>
PKD (Neurips-22) [42] ‡	<u>43.7</u>	<u>92.3</u>	<b>35.5</b>	51.0
TMC-KD (ours) ‡	<b>44.2</b>	91.9	34.4	<u>51.7</u>

In Table 1, we provide the comparison results with multi-layer KD methods on CIFAR-100. As shown, our TMC-KD achieves the best results for all different teacher-student combinations. Compared to scores of the student models, our TMC-KD improves 4.74 scores in average. In particular, for the similar architectures of teacher and student models (*e.g.* VGG-13 and VGG-8), TMC-KD provides 4.09 accuracy gain on average, but 4.99 gain for the heterogeneous architecture setup (*e.g.* ResNet-32 × 4 and VGG-13). TMC-KD provides more gains between heterogeneous models where the exact correlation learning between multi-layers is necessary.

Table 2 shows the results of the relation KD methods. In this comparison, our TMC-KD is superior to other methods. These comparison results show that exploiting both inter-layer and intra-layer knowledge is very effective for KD.

For more comparison, we evaluate our method on ImageNet as in Table 4. For reproducing results, we use the officially released code of [5] from the KD to SemCKD implementation, but use the code of [14] for ICKD and TaT implementation. For the ResNet-18, our TMC-KD achieves the better accuracy except for TaT. For the ShuffleNetV2 × 0.5, our TMC-KD achieves the best performance in this heterogeneous setting.<sup>1</sup> From the results, we show that our method can work well on the large-scale classification task.

Moreover, we provide the qualitative comparison in Fig. 6. We visualize the saliency region for classification by using Grad-CAM [47], and compare ours with other KD methods. We visualize feature maps from a convolution layer before the last fully-connected layer. Even some regions are not discriminative in other methods, our TMC-KD provides clearer saliency even for those regions. Compared to the results of the teacher models, our TMC-KD produces almost similar saliency responses. The more results can be found in the appendix A. These qualitative results support that our TMC-KD can achieve the better accuracy for the quantitative comparison.

To compare complexity with other knowledge distillation methods, we measure average training and inference time per epoch on the ImageNet. We use the ResNet-34 and ResNet-18 networks as a teacher and a student. The training speed of TMC is slower than other methods due to the multi-layer KD using the attached MLC and transformer. However, the inference speed of our TMC-KD is similar to others since the attached modules are not exploited during inference. This

**TABLE 4.** Comparison with other KD methods on the ImageNet. Bold indicates the best Top-1 accuracy. \* results are in [5], and ‡ results are from our re-implementation.

Teacher	ResNet-34	
	73.31	
Student	ResNet-18	ShuffleV2x0.5
	69.67	54.73
KD (NIPS-14) [3]	70.62 *	50.42 ‡
FitNet (ICLR-15) [4]	70.31 *	53.36 ‡
AT (ICLR-17) [19]	70.30 *	54.49 ‡
SP (CVPR-19) [18]	69.99 *	54.42 ‡
VID (CVPR-19) [21]	70.30 *	54.49 ‡
SemCKD (AAAI-21) [5]	70.87 *	54.59 ‡
ICKD (ICCV-21) [7]	68.35 ‡	48.70 ‡
TaT (CVPR-22) [14]	<b>71.74 ‡</b>	N/A
TMC-KD	71.43	<b>54.72</b>

**TABLE 5.** Average training and inference speed of different KD methods per epoch on ImageNet. All results are evaluated by our re-implementation and measured on the same H/W environment.

Method	Average training time per epoch (hour)	Average inference time per epoch (second)
KD [3]	0.417	90.198
FitNet [4]	0.428	90.245
AT [19]	0.426	94.340
SP [18]	0.419	90.708
VID [21]	0.462	90.612
SemCKD [5]	1.027	92.251
ICKD [7]	0.435	90.359
TMC-KD	1.261	90.886

**TABLE 6.** Results on different components of TMC-KD.

	$L_{KD}$	$L_{Local}$	$L_{Global}$	Top-1 Acc.	Top-5 Acc.
(a)	○	×	×	72.59	91.84
(b)	○	○	×	75.96	93.40
(c)	○	○	○	76.23	93.47
(d)	SemCKD converter			74.95	92.98

implies that our TMC-KD does not impose extra costs on the target student during the inference stage.

#### D. COMPARISON ON RIP CURRENT DETECTION

For more comparisons, we apply the KD to object detection. Despite the development of high-performance detectors [48], [49], [50], [51] in recent years, we use a simple Faster R-CNN [52] detector for implementation and comparison. We perform the KD between both detectors using CWD [40],

<sup>1</sup>TaT does not provide the code for learning the ShuffleNetV2 × 0.5.

**Algorithm 2** Our PyTorch Code for Implementing the Multi-layer Feature Converter (MLC)

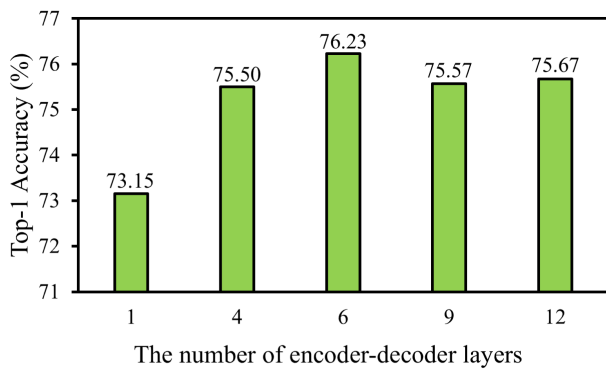
```

import torch.nn as nn

class MLC(nn.Module):
    def __init__(self, in_chn, in_w, dim_out=16, drop=0):
        super().__init__()
        ## in_chn : The number of input feature map channels
        ## in_w : The width size of input feature map
        ## dim_out : Dimension of output
        ## drop : Rate for Dropout

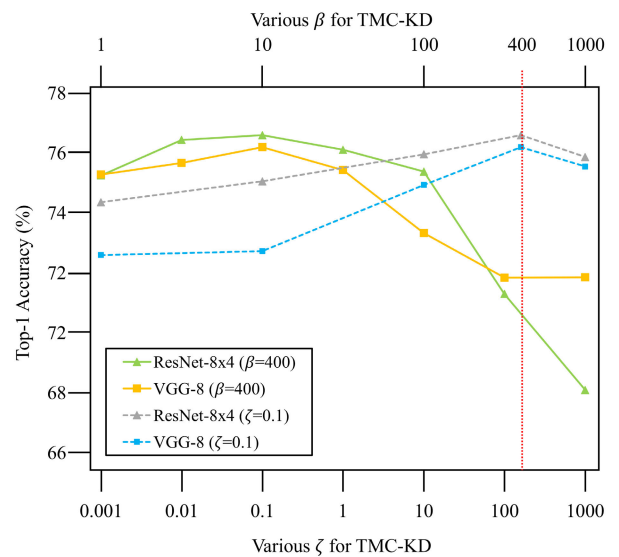
        hidden_chn = in_chn * 2
        hidden_dim = in_chn * in_w * in_w
        self.conv1 = nn.Conv2d(in_chn, hidden_chn, 1)
        self.bn1 = nn.BatchNorm2d(hidden_chn)
        self.act = nn.ReLU(inplace=True)
        self.conv2 = nn.Conv2d(hidden_chn, in_chn, 1)
        self.Linear = nn.Linear(hidden_dim, dim_out)
        self.drop = nn.Dropout(drop)

    def forward(self, x):
        B = x.shape[0] # B : Batch size
        x = self.conv1(x)
        x = self.act(x)
        x = self.bn1(x)
        x = self.drop(x)
        x = self.conv2(x)
        x = self.Linear(x.reshape(B, -1)) ## Flatten and fully connection
        x = self.drop(x)
        return x
    
```



**FIGURE 4.** Sensitivity analysis by changing the cardinality of the encoder and decoder layers.



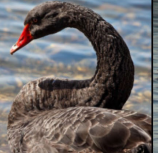



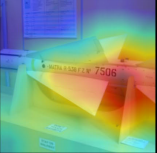
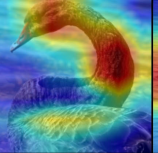
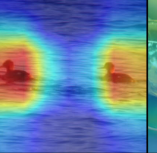
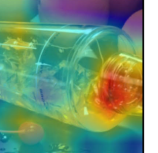

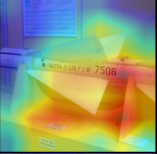
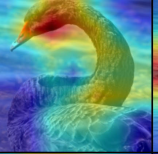
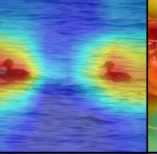
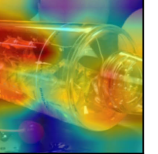

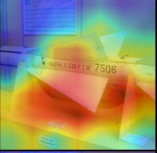
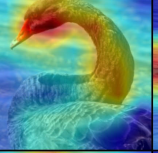
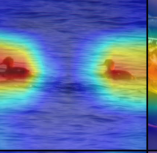
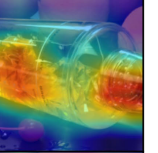

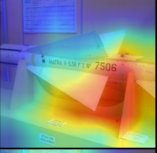
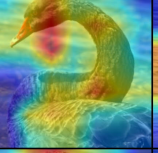
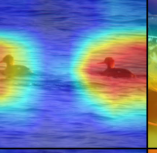
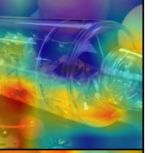
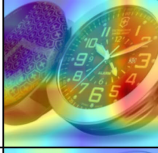
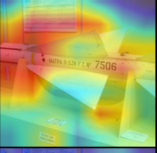
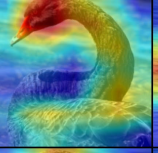
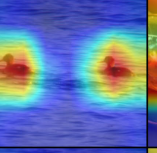
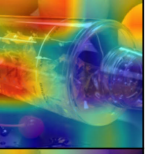

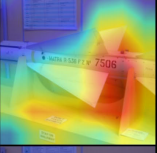
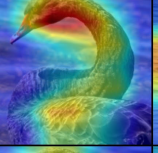
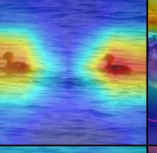
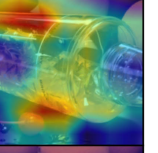
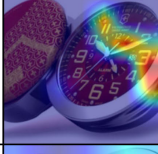
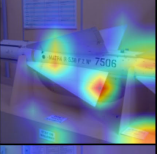
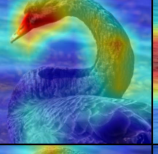
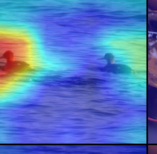


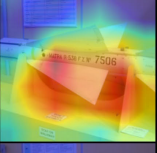
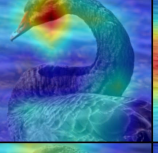
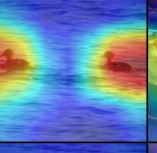
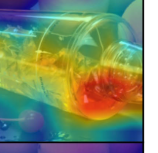
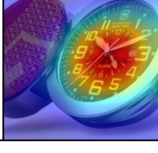
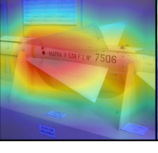
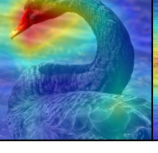
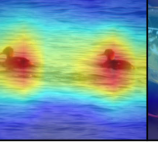
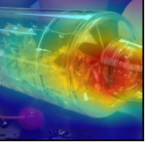
FBKD [41], and PKD [42]. In the CWD-based implementation, we minimize the Kullback-Leibler divergence between teacher and student activation maps from FPNs. In the FBKD, we extract both spatial and channel attentions from teacher and student detectors. We then minimize each mean square error between teacher and student attention maps. In the PKD, we normalize outputs of FPNs and minimize the mean









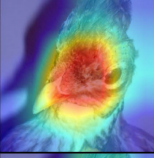
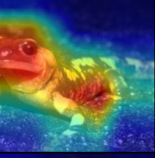
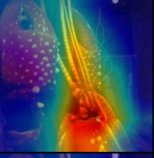
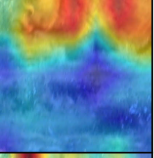

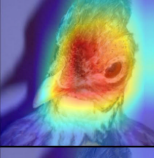
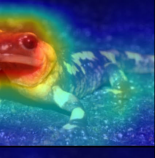
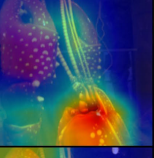
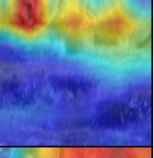


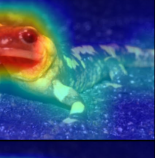
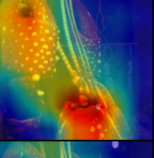
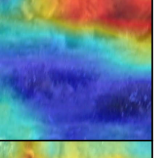

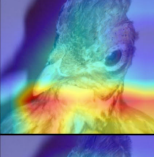
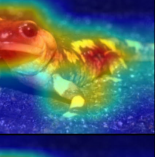
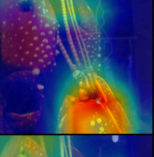
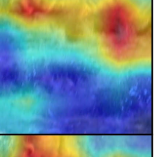

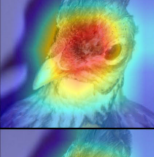
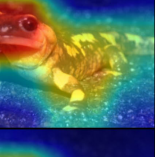
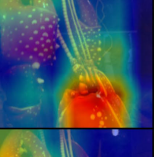
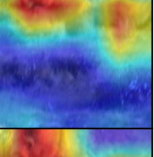

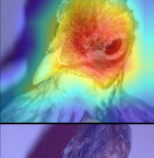
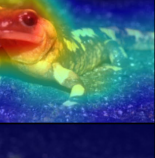
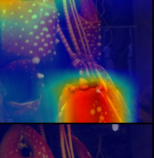
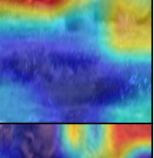
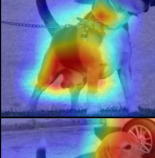

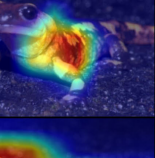

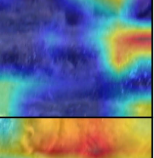
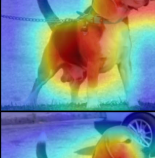
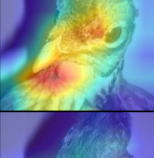
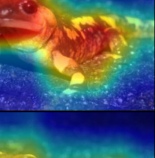
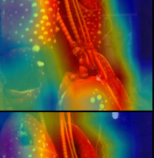
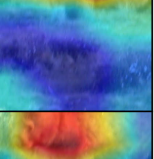

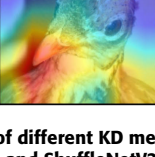
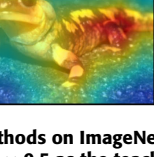
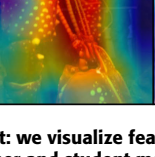
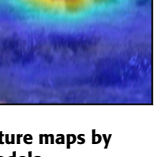
**FIGURE 5.** Sensitivity analysis of our TMC-KD for  $\zeta$  (bottom axis) and  $\beta$  (top axis).

square error between teacher and student normalized features. While other methods perform KD between same-level layers,



Label	Analog clock	Missile	Black swan	Mergus serrator	Cocktail shaker
Original Image					
KD					
FitNet					
AT					
SP					
VID					
SemCKD					
ICKD					
TMC-KD (Ours)					
Teacher					

**FIGURE 6.** Qualitative comparison of different KD methods on ImageNet: we visualize feature maps by using Grad-CAM. We use ResNet-34 and ShuffleNetV2  $\times$  0.5 as the teacher and student models. We represent the strength of the saliency with different colors (red indicates the stronger response).

Label	Beagle	Prairie chicken	European fire salamander	Crayfish	Wombat
Original Image					
KD					
FitNet					
AT					
SP					
VID					
SemCKD					
ICKD					
TMC-KD (Ours)					
Teacher					

**FIGURE 7.** Qualitative comparison of different KD methods on ImageNet: we visualize feature maps by using Grad-CAM. We use ResNet-34 and ShuffleNetV2  $\times$  0.5 as the teacher and student models. We represent the strength of the saliency with different colors (red indicates the stronger response).

Sample	Rip-1705	Rip-904	Rip-863	Rip-81
Ground Truth				
Teacher				
Student				
CWD				
FBKD				
PKD				
TMC-KD (Ours)				

**FIGURE 8.** Qualitative comparison of different KD methods on the Rip current dataset. We visualize detection results with red box. For this comparison, we use Faster R-CNNs with ResNet-101 and ResNet-18 as the teacher and student detectors.

our TMC-KD compares features among all FPN layers using the local correlation Eq.(9). For this comparison, we evaluate detectors using the COCO style metrics: average precision (AP) and recall (AR). We evaluate AP at IoU  $\in [0.5 : 0.05 : 0.95]$  (AP@0.5:0.95), at IoU 0.5 (AP@0.5)

and at IoU 0.75 (AP@0.75). We evaluate AR at IoU  $\in [0.5 : 0.05 : 0.95]$  (AR@0.5:0.95).

In Table 3, we provide the comparison results for KD methods on the rip current detection. Compared to other KD methods, our TMC-KD achieves best meanAP 44.2%.

---

**Algorithm 3** Our Code for Implementing the global Correlation  $C_{Global}$  and the Self-attention  $H_{Global}$

---

```

import math
import torch
import torch.nn.functional as F

def Self_Attention(W_Q, W_K, W_V, V_P):
    ## W_P : Weight for query
    ## W_K : Weight for key
    ## W_V : Weight for value
    ## V_P : Input feature
    E = V_P.shape[2]

    ## Matrix multiplication
    Q = F.linear(V_P, W_Q)
    K = F.linear(V_P, W_K)
    V = F.linear(V_P, W_V)

    ## Eq. (5) in the main text
    Q = Q / math.sqrt(E)
    C_Global = torch.bmm(Q, K.transpose(-2, -1))
    C_Global = F.softmax(C_Global, dim=-1)

    ## Eq. (6) in the main text
    H_Global = torch.bmm(C_Global, V)

return H_Global

```

---

By applying our TMC-KD, we greatly improve the mean AP by 4.3 point. On average recall, TMC-KD achieved the second highest score 51.7. While other KD methods ignore the correlation between different-level feature maps, TMC-KD considers both local and global correlations. Therefore, we achieve the high scores on both AP and AR metrics. Moreover, we provide the qualitative comparison in Fig. 8.

We visualize the detection results and compare ours with other KD methods. On “Rip-1705”, “Rip-904” and “Rip-863” samples, TMC-KD detects rip currents successfully, whereas other KD methods produce some missing or inaccurate detection results of the rip currents.

### E. ABLATION STUDY

We evaluate the effects of each method applying for TMC-KD by measuring Top-1 and Top-5 accuracy. For this study, we use the ResNet-32  $\times$  4 and VGG-8 as teacher and student models, respectively. Then, we use (a) the baseline with the KD method using [3]. We then add our method one-by-one into the baseline: (b) with the multi-layer local loss Eq. (2), (c) with the global relation loss Eq. (10). To show the effect of our MLC, we also implement (d) that uses SemCKD instead of using our MLC. In this case, we use the feature pooling and  $1 \times 1$  convolution layers described in SemCKD for matching the spatial resolution and channel number between different layers. Compared to the baseline, (c) using all our

methods provides the 3.64 Top-1 accuracy and 1.63 Top-5 accuracy gain. By adding the local and global losses, we can improve the accuracy by 75.96 and 76.23 Top-1 accuracies. In addition, we can improve Top-5 accuracy by 93.40 and 93.47. When comparing (b) and (d), replacing our MLC with the SemCKD-based feature converter degrades Top-1 and Top-5 accuracies by 74.95 and 92.98. This because our MLC generates the more stronger sequential features which are input of the transformer. These results indicate that all our methods are beneficial of improving the multi-layer KD training.

### F. SENSITIVITY ANALYSIS

We investigate the sensitivity of our TMC-KD by varying the values of the important hyper-parameters. We use ResNet-32  $\times$  4 and VGG-8 as a teacher and student models, and evaluate them on CIFAR-100.

#### 1) THE SIZE OF TRANSFORMER

We change the number of stacked layers used in the encoder and decoder from 1 to 12, and report the results in Figure 4. We achieve the best score when using 6 layers. Using too many layers degrades the accuracy even due to the large discrepancy between decoded features. We expect that transformers with many layers could be rather over-fitted due to the small sample size of the CIFAR-100. On the other hand,

---

**Algorithm 4** Our Code for Implementing the local Correlation  $\Lambda$  and the Local Semantic Loss  $L_{Local}$

---

```

import torch.nn as nn
import torch.nn.functional as F
from einops import rearrange

class Local_Loss(nn.Module):
    def __init__(self):
        super().__init__()
        self.crit = nn.MSELoss(reduction='none')

    def forward(self, f_s, f_t, P_t, P_s):
        ## f_t: List of the feature maps for projected teacher
        ## f_s: List of the feature maps for projected student
        ## P_t : Decoder output for teacher
        ## P_s : Decoder output for student

        # Compute local correlation (Lambda)
        ## Eq. (9) in the main text
        ## Rearrange dimension: From [B, M, E] to [B, E, M]
        temp_P_t = rearrange(P_t, 'B M E -> B E M')
        Lambda = F.softmax(torch.bmm(P_s, temp_P_t), dim=-1)

        # Compute Eq. (2) in the main text
        B, J, M = Lambda.shape
        ## B : Batch size
        ## J : The number of student feature maps
        ## M : The number of teacher feature maps

        loss_i = torch.zeros(B, J, M)
        for j in range(J):
            for m in range(M):
                loss_i[:, j, m] = self.crit(f_s[j][m], f_t[j][m]).reshape(B, -1).
                mean(-1) local_loss = (Lambda * loss_i ).sum() / (B*J)

        return local_loss

```

---

a transformer with few layers is likely to be insufficient to extract the exact correlation features.

## 2) HYPER-PARAMETER $\zeta$ AND $\beta$

We change the values of  $\zeta$  and  $\beta$  which are used for balancing between losses in Eq. (11). For  $\zeta$ , we change the score from 0.001 to 1000 by multiplying 10. To investigate the effect of the architecture difference, we fix a teacher model with ResNet-32  $\times$  4, but use ResNet-8  $\times$  4 and VGG-8 for homogeneous and heterogeneous setup as a student model. Figure 5 shows the results. We achieve the best scores to 76.23% and 76.63% for VGG-8 and ResNet-8  $\times$  4 when using  $\zeta = 0.1$ . However, the accuracy difference between 0.01 and 1 is rather marginal. For  $\beta$ , we change the score from 1 to 1000 by multiplying 10 (including 400 tuned in SemCKD [5]). We also achieve the best scores to 76.23% and 76.63% for VGG-8 and ResNet-8  $\times$  4 when using  $\beta = 400$ .

The accuracy of both student models tends to be enhanced as increases  $\beta$  before  $\beta = 400$ .

## V. CONCLUSION

For multi-layer KD, we propose a novel transformer-based multiple layer correlation KD (TMC-KD) method. Our TMC-KD can bridge the knowledge gap between different models via global and local correlation learning. For learning both correlations between intermediate layers of different architectures, we design a multi-layer feature converter and exploit it to transform multi-layer features to serially-connected encoded features. By using the decoded features and attention tensors from a transformer, we can minimize the discrepancy between models in terms of local and global semantic relations. The comparison results with recent KD methods prove the effectiveness of our method. In image classification, our TMC-KD provides about 5% accuracy

---

**Algorithm 5** Our Code for Implementing the Global Relational Loss  $L_{Global}$

---

```

import torch
import torch.nn as nn
import torch.nn.functional as F

class Global_Loss(nn.Module):
    def __init__(self):
        super().__init__()

    def forward(self, P_s, P_t):
        B = P_s.shape[0] # Batch size
        temp_t = P_t.reshape(B, -1)
        temp_s = P_s.reshape(B, -1)

        self_cor_t = torch.matmul(temp_t, temp_t.t())
        self_cor_s = torch.matmul(temp_s, temp_s.t())
        global_loss = F.mse_loss(self_cor_s, self_cor_t, reduction='mean')

    return global_loss

```

---

gain on average. It outperforms other methods for the knowledge distillation between heterogeneous architectures on the CIFAR-100. On ImageNet, TMC-KD provides the best accuracy 54.72 accuracy. For more comparison, we also evaluate our TMC-KD and other KD methods on the rip current detection set. In this comparison, our TMC-KD achieves the best mAP of 44.2 % surpassing the performance of other methods. From the extensive ablation study, we show the effects of multi-layer feature converter, local and global correlation learning.

The training speed of TMC-KD is slower than other KD methods due to the additional complexity of the transformer. To reduce the complexity of the transformer, our future work could combine the deformable attention techniques [48], [53]. We believe that our work could be a solid guideline of multi-layer KD.

## APPENDIX A QUALITATIVE COMPARISON

We visualize discriminative regions by using Grad-CAM [47] for qualitative comparisons. We use images on ImageNet [32]. As shown in Figure 6 and 7, our TMC-KD shows our method produces almost similar discriminative regions as the teacher model. Compared to other recent KD methods, our method clearly shows the more distinctive saliency regions. In particular, in some sample images (e.g. *Mergus serrator*, a beagle, and a European fire salamander), our TMC-KD shows the better results than the oracle teacher model.

## APPENDIX B THE MAIN CODES FOR OUR METHOD IMPLEMENTATION

We provide our code for implementing the proposed methods described in Sec. III-B of our manuscript: the

multi-layer feature converter (MLC), local semantic learning, and global relational learning. Basically, we implement our code using the PyTorch [54]. The Algorithm 2 describes the implementation of the MLC. The MLC class constructor requires the number of input channels, the size of input width, and an embedding size  $E$  for outputs. During training, we set the rate for the dropout to 0.

In Algorithm 3 and 4, we provide the code for implementing local semantic learning. Algorithm 3 shows the  $C_{Global}$  and self-attention implementation described in Eq. (5) and (6) of the paper. Given the weights for the query, key, and value for the series of the MLC converted features  $V_P$ , we can compute the global correlation  $C_{Global}$  and the self-attention  $H_{Global}^h$  using the multiple attention heads. Algorithm 4 shows how to compute the local correlation and the local KD loss  $L_{Local}$  between multiple layers. To evaluate  $\Lambda$ , we first perform the matrix multiplication between the teacher and student feature maps as  $P_t$  and  $P_s$ , and normalize it using the softmax. We then compute the local loss  $L_{Local}$  by evaluating the feature L2 distance applied to the correlation  $\Lambda$ .

The Algorithm 5 shows the implementation of the global loss in Eq. (10). By using the L2 distance between the self-attentions of the decode features  $P_t$  and  $P_s$ , we evaluate the global loss.

## REFERENCES

- [1] S. Hanson and L. Pratt, "Comparing biases for minimal network construction with back-propagation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, 1988, pp. 177–185.
- [2] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4820–4828.

- [3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS Deep Learn. Represent. Learn. Workshop*, 2015, pp. 1–9. [Online]. Available: <http://arxiv.org/abs/1503.02531>
- [4] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–13.
- [5] D. Chen, J.-P. Mei, Y. Zhang, C. Wang, Z. Wang, Y. Feng, and C. Chen, "Cross-layer distillation with semantic calibration," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 8, 2021, pp. 7028–7036.
- [6] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3962–3971.
- [7] L. Liu, Q. Huang, S. Lin, H. Xie, B. Wang, X. Chang, and X. Liang, "Exploring inter-channel correlation for diversity-preserved knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8251–8260.
- [8] A. Amirkhani, A. Khosravian, M. Masih-Tehrani, and H. Kashiani, "Robust semantic segmentation with multi-teacher knowledge distillation," *IEEE Access*, vol. 9, pp. 119049–119066, 2021.
- [9] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 4, 2020, pp. 5191–5198.
- [10] W. Son, J. Na, J. Choi, and W. Hwang, "Densely guided knowledge distillation using multiple teacher assistants," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9375–9384.
- [11] A. Banitalebi-Dehkordi, A. Amirkhani, and A. Mohammadinasab, "EBCDet: Energy-based curriculum for robust domain adaptive object detection," *IEEE Access*, vol. 11, pp. 77810–77825, 2023.
- [12] S. Kim, "A virtual knowledge distillation via conditional GAN," *IEEE Access*, vol. 10, pp. 34766–34778, 2022.
- [13] D. Chen, H. Tan, L. Lan, X. Zhang, T. Liang, and Z. Luo, "Frustratingly easy knowledge distillation via attentive similarity matching," in *Proc. 26th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2022, pp. 2357–2363.
- [14] S. Lin, H. Xie, B. Wang, K. Yu, X. Chang, X. Liang, and G. Wang, "Knowledge distillation via the target-aware transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10905–10914.
- [15] D. Chen, J.-P. Mei, H. Zhang, C. Wang, Y. Feng, and C. Chen, "Knowledge distillation with the reused teacher classifier," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11923–11932.
- [16] Z. Guo, H. Yan, H. Li, and X. Lin, "Class attention transfer based knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 11868–11877.
- [17] S.-G. Park and D.-J. Kang, "Knowledge distillation with feature self attention," *IEEE Access*, vol. 11, pp. 34554–34562, 2023.
- [18] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1365–1374.
- [19] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–13.
- [20] Y. Lee, N. Ahn, J. H. Heo, S. Y. Jo, and S.-J. Kang, "Teaching where to see: Knowledge distillation-based attentive information transfer in vehicle maker classification," *IEEE Access*, vol. 7, pp. 86412–86420, 2019.
- [21] S. Ahn, S. X. Hu, A. Damianou, N. D. Lawrence, and Z. Dai, "Variational information distillation for knowledge transfer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9155–9163.
- [22] M. Ji, B. Heo, and S. Park, "Show, attend and distill: Knowledge distillation via attention-based feature matching," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 7945–7952.
- [23] Q. Tang, Y. Zhang, X. Xu, J. Wang, and Y. Guo, "Input-dependent dynamical channel association for knowledge distillation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.
- [24] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7130–7138.
- [25] H.-T. Li, S.-C. Lin, C.-Y. Chen, and C.-K. Chiang, "Layer-level knowledge distillation for deep neural network learning," *Appl. Sci.*, vol. 9, no. 10, p. 1966, May 2019. [Online]. Available: <https://www.mdpi.com/2076-3417/9/10/1966>
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.
- [27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.
- [28] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–22.
- [29] G. Aguilar, Y. Ling, Y. Zhang, B. Yao, X. Fan, and C. Guo, "Knowledge distillation from internal representations," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7350–7357.
- [30] V. Sampath, I. Maurtua, J. J. A. Martín, A. Iriondo, I. Lluvia, and A. Rivera, "Vision transformer based knowledge distillation for fasteners defect detection," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Jul. 2022, pp. 1–6.
- [31] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 0, 2009.
- [32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [33] N. Passalis, M. Tzelepi, and A. Tefas, "Heterogeneous knowledge distillation using information flow modeling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2336–2345.
- [34] A. de Silva, I. Mori, G. Dusek, J. Davis, and A. Pang, "Automated rip current detection with region based convolutional neural networks," *Coastal Eng.*, vol. 166, Jun. 2021, Art. no. 103859.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015, pp. 1–14. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [37] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient CNN architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 116–131.
- [38] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*. BMVA Press, 2016, pp. 87.1–87.12, doi: [10.5244/c.30.87](https://doi.org/10.5244/c.30.87).
- [39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- [40] C. Shu, Y. Liu, J. Gao, Z. Yan, and C. Shen, "Channel-wise knowledge distillation for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 5291–5300.
- [41] L. Zhang and K. Ma, "Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–14.
- [42] W. Cao, Y. Zhang, J. Gao, A. Cheng, K. Cheng, and J. Cheng, "PKD: General distillation framework for object detectors via Pearson correlation coefficient," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 15394–15406.
- [43] N. Passalis and A. Tefas, "Learning deep representations with probabilistic knowledge transfer," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 268–284.
- [44] Y. Liu, J. Cao, B. Li, C. Yuan, W. Hu, Y. Li, and Y. Duan, "Knowledge distillation via instance relationship graph," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7089–7097.

- [45] B. Peng, X. Jin, D. Li, S. Zhou, Y. Wu, J. Liu, Z. Zhang, and Y. Liu, "Correlation congruence for knowledge distillation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5006–5015.
- [46] Y. Tian, D. Krishnan, and P. Isola, "Contrastive representation distillation," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2019, pp. 1–14.
- [47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
- [48] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–16.
- [49] S.-H. Bae, "Deformable part region learning for object detection," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, 2022, pp. 95–103.
- [50] S.-H. Lee and S.-H. Bae, "AFI-GAN: Improving feature interpolation of feature pyramid networks via adversarial training for object detection," *Pattern Recognit.*, vol. 138, Jun. 2023, Art. no. 109365.
- [51] S.-H. Bae, "Deformable part region learning and feature aggregation tree representation for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10817–10834, 2023, doi: 10.1109/TPAMI.2023.3268864.
- [52] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [53] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4784–4793.
- [54] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 8026–8037.



**HUN-BEOM BAK** received the B.S. degree in physics from Incheon National University in 2020, and the M.S. degree in electrical and computer engineering from Inha University in 2024. He is currently a Research Engineer with Innovation Development Department, DANUSYS, South Korea. His current research interests include model compression, knowledge distillation, and data-free knowledge distillation.



**SEUNG-HWAN BAE** (Member, IEEE) received the B.S. degree in information and communication engineering from Chungbuk National University, in 2009, and the M.S. and Ph.D. degrees in information and communications from Gwangju Institute of Science and Technology (GIST), in 2010 and 2015, respectively. He was a Senior Researcher with the Electronics and Telecommunications Research Institute (ETRI), South Korea, from 2015 to 2017. He was an Assistant Professor with the Department of Computer Science and Engineering, Incheon National University, South Korea, from 2017 to 2020. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Inha University. His research interests include object tracking, object detection, generative model learning, continual learning, and on-device ML.

• • •