Beyond Self-Attention: External Attention Using Two Linear Layers for Visual Tasks

Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu, *Senior Member, IEEE*

Abstract—Attention mechanisms, especially self-attention, have played an increasingly important role in deep feature representation for visual tasks. Self-attention updates the feature at each position by computing a weighted sum of features using pair-wise affinities across all positions to capture the long-range dependency within a single sample. However, self-attention has quadratic complexity and ignores potential correlation between different samples. This article proposes a novel attention mechanism which we call *external attention*, based on two external, small, learnable, shared memories, which can be implemented easily by simply using two cascaded linear layers and two normalization layers; it conveniently replaces self-attention in existing popular architectures. External attention has linear complexity and implicitly considers the correlations between all data samples. We further incorporate the multi-head mechanism into external attention to provide an all-MLP architecture, *external attention MLP* (EAMLP), for image classification. Extensive experiments on image classification, object detection, semantic segmentation, instance segmentation, image generation, and point cloud analysis reveal that our method provides results comparable or superior to the self-attention mechanism and some of its variants, with much lower computational and memory costs.

Index Terms—Deep learning, computer vision, attention, transformer, multi-layer perceptrons

1 Introduction

Due to its ability to capture long-range dependencies, the self-attention mechanism helps to improve performance in various natural language processing [1], [2] and computer vision [3], [4] tasks. Self-attention works by refining the representation at each position via aggregating features from all other locations in a single sample, which leads to quadratic computational complexity in the number of locations in a sample. Thus, some variants attempt to approximate self-attention at a lower computational cost [5], [6], [7], [8].

Furthermore, self-attention concentrates on the self-affinities between different locations within a single sample, and ignores potential correlations with other samples. It is easy to see that incorporating correlations between different samples can help to contribute to a better feature representation. For instance, features belonging to the same category but distributed across different samples should be treated consistently in the semantic segmentation task, and a similar observation applies in image classification and various other visual tasks.

This paper proposes a novel lightweight attention mechanism which we call *external attention* (see Fig. 1c)). As shown in Fig. 1a), computing self-attention requires first calculating an attention map by computing the affinities between self query vectors and self key vectors, then generating a new feature map by weighting the self value vectors with this attention map. External attention works differently. We first calculate the attention map by computing the affinities between the self query vectors and an external learnable *key* memory, and then produce a refined feature map by multiplying this attention map by another external learnable *value* memory.

In practice, the two memories are implemented with linear layers, and can thus be optimized by back-propagation in an end-to-end manner. They are independent of individual samples and shared across the entire dataset, which plays a strong regularization role and improves the generalization capability of the attention mechanism. The key to the lightweight nature of external attention is that the number of elements in the memories is much smaller than the number in the input feature, yielding a computational complexity linear in the number of elements in the input. The external memories are designed to learn the most discriminative features across the whole dataset, capturing the most informative parts, as well as excluding interfering information from other samples. A similar idea can be found in sparse coding [9] or dictionary learning [10]. Unlike those methods, however, we neither try to reconstruct the input features nor apply any explicit sparse regularization to the attention map.

Although the proposed external attention approach is simple, it is effective for various visual tasks. Due to its simplicity, it can be easily incorporated into existing popular self-attention based architectures, such as DANet [4], SAGAN [11] and T2T-Transformer [12]. Fig. 3 demonstrates a typical architecture replacing self-attention with our external attention for an image semantic segmentation task. We have conducted extensive experiments on such basic visual tasks as classification, object detection, semantic segmentation, instance segmentation and generation, with different

• *The authors are with the BNRist, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China. E-mail: {gmh20, liu-zn17}@mails.tsinghua.edu.cn, {taijiang, shimin}@tsinghua.edu.cn.*

Manuscript received 1 June 2021; revised 1 December 2021; accepted 25 September 2022. Date of publication 5 October 2022; date of current version 3 April 2023.
This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0112902 and in part by the Natural Science Foundation of China under Grant 62220106003.
(Corresponding author: Shi-Min Hu.)
Recommended for acceptance by J. Hoffman.
Digital Object Identifier no. 10.1109/TPAMI.2022.3211006

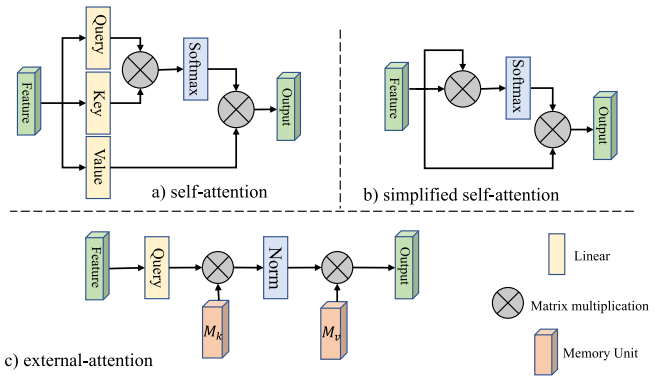


Fig. 1. Self-attention versus external-attention.

input modalities (images and point clouds). The results reveal that our method achieves results comparable to or better than the original self-attention mechanism and some of its variants, at much lower computational cost.

To learn different aspects of attentions for the same input, we explore the multi-head mechanism. We find that the current multi-head mechanism used by default is not optimal way for both self-attention and external attention. We simply modify the multi-head mechanism and make all heads interact together, achieving a better performance than current manner for both self-attention and external attention. Based on this finding, we incorporate a novel multi-head mechanism into external attention to boost its capability. Benefiting from the proposed multi-head external attention, we have designed a novel all-MLP architecture named EAMLP, which is comparable to CNNs and the original Transformers for the image classification task.

The main contributions of this paper are summarized below:

- A novel attention mechanism, external attention, with $O(n)$ complexity; it can replace self-attention in existing architectures. It can mine potential relationships across the whole dataset, affording a strongly regularizing role, and improving the generalization capability of the attention mechanism.
- We explore the multi-head mechanism and find a better multi-head mechanism than original multi-head mechanism for both self-attention and external attention. Based on this finding, we design a multi-head external attention, which benefits us to build an all MLP architecture; it achieves a top1 accuracy of 79.5% on the ImageNet-1K dataset.
- Extensive experiments utilize external attention for image classification, object detection, semantic segmentation, instance segmentation, image generation, point cloud classification, and point cloud segmentation. In scenarios where computational effort must be kept low, it achieves better results than the original self-attention mechanism and some of its variants.

2 Related Work

Since a comprehensive review of the attention mechanism is beyond the scope of this paper, we only discuss the most closely related literature in the vision realm.

2.1 The attention mechanism in visual tasks

The attention mechanism can be viewed as a mechanism for reallocating resources according to the importance of activation. It plays an important role in the human visual system. There has been vigorous development of this field in the last decade [3], [13], [14], [15], [16], [17], [18]. Hu et al. proposed SENet [15], showing that the attention mechanism can reduce noise and improve classification performance. Subsequently, many other papers have applied it to visual tasks. Wang et al. presented non-local networks [3] for video understanding, Hu et al. [19] used attention in object detection, Fu et al. proposed DANet [4] for semantic segmentation, Zhang et al. [11] demonstrated the effectiveness of the attention mechanism in image generation, and Xie et al. proposed A-SCN [20] for point cloud processing. Readers are referred to recent survey [21] for a more comprehensive review of the use of attention methods for visual tasks.

2.2 Self-attention in visual tasks

Self-attention is a special case of attention, and many papers [3], [4], [11], [17], [22], have considered the self-attention mechanism for vision. The core idea of self-attention is calculating the affinity between features to capture long-range dependencies. However, as the size of the feature map increases, the computing and memory overheads increase quadratically. To reduce computational and memory costs, Huang et al. [5] proposed criss-cross attention, which considers row attention and column attention in turn to capture the global context. Li et al. [6] adopted expectation maximization (EM) clustering to optimize self-attention. Yuan et al. [7] proposed use of object-contextual vectors to process attention; however, it depends on semantic labels. Geng et al. [8] show that matrix decomposition is a better way to model the global context in semantic segmentation and image generation. Other works [23], [24] also explore extracting local information by using the self-attention mechanism.

Unlike self-attention which obtains an attention map by computing affinities between self queries and self keys, our external attention computes the relation between self queries and a much smaller learnable key memory, which captures the global context of the dataset. External attention does not rely on semantic information and can be optimized by the back-propagation algorithm in an end-to-end way instead of requiring an iterative algorithm.

2.3 Transformer in visual tasks

Transformer-based models have had great success in natural language processing [1], [2], [16], [25], [26], [27], [28]. Recently, they have also demonstrated huge potential for visual tasks. Carion et al. [29] presented an end-to-end detection transformer that takes CNN features as input and generates bounding boxes with a transformer. Dosovitskiy [18] proposed ViT, based on patch encoding and a transformer, showing that with sufficient training data, a transformer provides better performance than a traditional CNN. Chen et al. [30] proposed iGPT for image generation based on use of a transformer.

Subsequently, transformer methods have been successfully applied to many visual tasks, including image classification [12], [31], [32], [33], object detection [34], lower-level vision [35], semantic segmentation [36], tracking [37], video instance segmentation [38], image generation [39], multimodal learning [40],

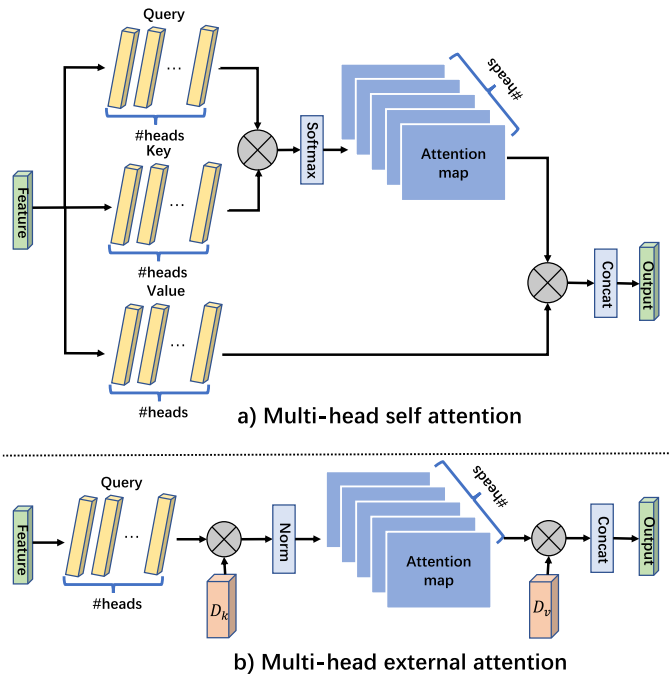


Fig. 2. Multi-head self-attention and multi-head external-attention.

TABLE 1
Ablation Study on PASCAL VOC val Set

Method	Backbone	Norm	#S	OS	mIoU(%)
FCN	ResNet-50	-	-	16	75.7
FCN + SA	ResNet-50	Softmax	-	16	76.2
FCN + SA	ResNet-50	DoubleNorm	-	16	76.6
FCN + EA	ResNet-50	DoubleNorm	8	16	77.1
FCN + EA	ResNet-50	DoubleNorm	32	16	77.2
FCN + EA	ResNet-50	Softmax	64	16	75.3
FCN + EA	ResNet-50	DoubleNorm	64	16	77.4
FCN + EA	ResNet-50	DoubleNorm	64	8	77.8
FCN + EA	ResNet-50	DoubleNorm	256	16	77.0
FCN + EA	ResNet-101	DoubleNorm	64	16	**78.3**

Norm: Normalization method in attention. #S: number of elements in memory units. OS: output stride of backbone. FCN [48]: fully convolutional network. SA: self-attention. EA: external-attention. DoubleNorm: normalization depicted as Eq. (9).

object re-identification [41], image captioning [42], point cloud learning [43] and self-supervised learning [44]. Readers are referred to recent surveys [45], [46] for a more comprehensive review of the use of transformer methods for visual tasks.

3 Methodology

In this section, we start by analyzing the original self-attention mechanism. Then we detail our novel way to define attention: external attention. It can be implemented easily by only using two linear layers and two normalization layers, as later shown in Algorithm 1.

3.1 Self-Attention and External Attention

We first revisit the self-attention mechanism (see Fig. 1a)). Given an input feature map $F \in \mathbb{R}^{N\times d}$, where N is the number of elements (or pixels in images) and d is the number of feature dimensions, self-attention linearly projects the input to a query matrix $Q \in \mathbb{R}^{N\times d'}$, a key matrix $K \in \mathbb{R}^{N\times d'}$, and a value matrix $V \in \mathbb{R}^{N\times d}$ [16]. Then self-attention can be formulated as:

$$A = (\alpha)_{i,j} = \text{softmax}(QK^T), \tag{1}$$

$$F_{out} = AV, \tag{2}$$

where $A \in \mathbb{R}^{N\times N}$ is the attention matrix and $\alpha_{i,j}$ is the pair-wise affinity between (similarity of) the ith and jth elements.

A common simplified variation (Fig. 1b)) of self-attention directly calculates an attention map from the input feature F using:

$$A = \text{softmax}(FF^T), \tag{3}$$

$$F_{out} = AF. \tag{4}$$

Here, the attention map is obtained by computing pixel-wise similarity in the feature space, and the output is the refined feature representation of the input.

However, even when simplified, the high computational complexity of $O(dN^2)$ presents a significant drawback to use of self-attention. The quadratic complexity in the number of input pixels makes direct application of self-attention to images infeasible. Therefore, previous work [18] utilizes self-attention on patches rather than pixels to reduce the computational effort.

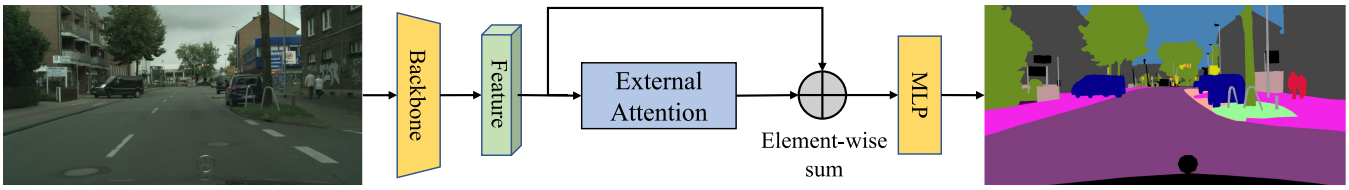


Fig. 3. EANet architecture for semantic segmentation using our proposed external attention.

TABLE 2
Ablation Study on Different Multi-Head Mechanism on ImageNet Dataset

Method	Attention	Multi-head mechanism	#Params(M)	#Throughput	#Memory(G)	Acc(%)
T2T-ViT-7	Self-attention	Single head interaction	4.3	2000	2.17	71.7
T2T-ViT-7	Self-attention	All heads interaction	4.3	1524	2.32	72.3
T2T-ViT-7	External attention	Single head interaction	6.6	1524	2.14	63.7
T2T-ViT-7	External attention	All heads interaction	5.9	1684	2.14	66.8

Acc(%) means Top-1 Accuracy. Throughput is measured by using GeForce RTX 3090 graphics card.Memory denotes inference memory cost with batch size 32.

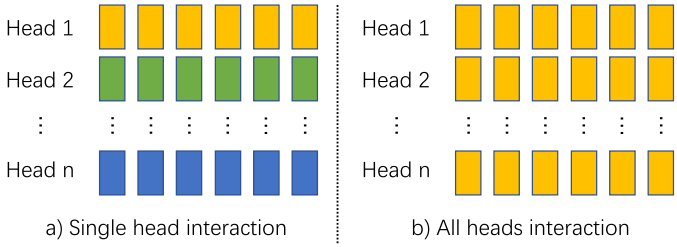


Fig. 4. Different multi-head mechanism. Interaction occurs between blocks which have same color.

Self-attention can be viewed as using a linear combination of self values to refine the input feature. However, it is far from obvious that we really need $N \times N$ self attention matrix and an N element self value matrix in this linear combination. Furthermore, self-attention only considers the relation between elements within a data sample and ignores potential relationships between elements in different samples, potentially limiting the ability and flexibility of self-attention.

Thus, we propose a novel attention module named *external attention*, which computes attention between the input pixels and an external memory unit $M \in \mathbb{R}^{S \times d}$, via:

$$A = (\alpha)_{i,j} = \text{Norm}(FM^T), \tag{5}$$

$$F_{out} = AM. \tag{6}$$

Unlike self-attention, $\alpha_{i,j}$ in Equation (5) is the similarity between the ith feature and the jth row of M, where M is a learnable parameter independent of the input, which acts as a memory of the whole training dataset. A is the attention map inferred from this learned dataset-level prior knowledge; it is normalized in a similar way to self-attention (see Section 3.2). Finally, we update the input features from M by the similarities in A.

In practice, we use two different memory units M_k and M_v as the key and value, to increase the capability of the network. This slightly changes the computation of external attention to

$$A = \text{Norm}(FM_k^T), \tag{7}$$

$$F_{out} = AM_v. \tag{8}$$

We tried three different initialization methods to initialize external memory unit M_k and M_v. The first manner is random initialization which is a common way to initialize networks. The second way is initialize the external attention by using Xavier initialization [47]. The last method is to initialize $M_k = M_v$ which is motivated by Eq. (5). Meanwhile, We conducted experiments to compare above three initialization methods. The final results show that there are no significant difference between the above three methods. We adopt random initialization method by default.

The computational complexity of external attention is $O(dSN)$; as d and S are hyper-parameters, the proposed algorithm is linear in the number of pixels. In fact, we find that a small S, e.g., 64, works well in experiments. Thus, external attention is much more efficient than self-attention, allowing its direct application to large-scale inputs. We also note that the computation load of external attention is roughly equivalent to a 1×1 convolution.

3.2 Normalization

Softmax is employed in self-attention to normalize the attention map so that $\sum_j \alpha_{i,j} = 1$. However, the attention map is calculated by matrix multiplication. Unlike cosine similarity, the attention map is sensitive to the scale of the input features. To avoid this problem , we opt for the double-normalization proposed in [43], which separately normalizes columns and rows. This double-normalization is formulated as:

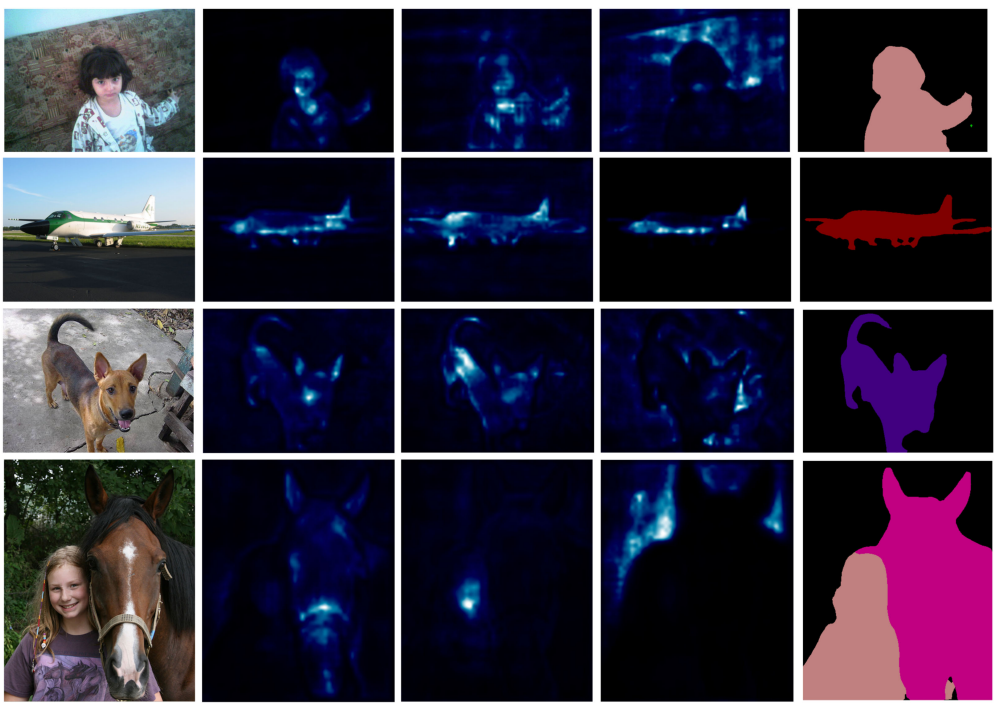


Fig. 5. Attention map and segmentation results on Pascal VOC test set. Left to right: input images, attention maps w.r.t. three selected entries in the external memory, segmentation results.

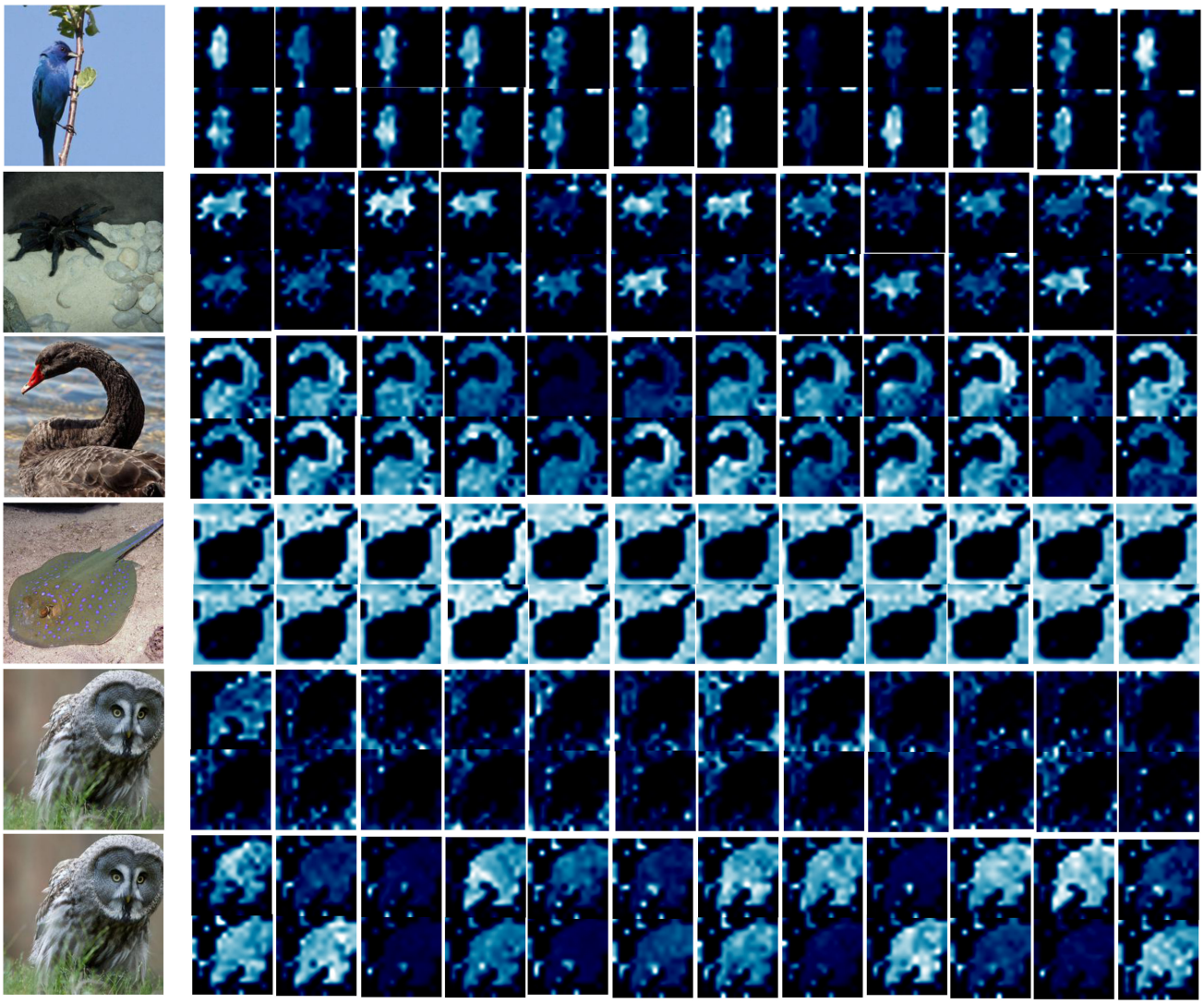


Fig. 6. Multi-head attention map in the last layer of EAMLP-14 on ImageNet val set. Left: Input image Others: 24 head attention map in the last layer of EAMLP-14 for the ImageNet val set. Last two rows: attention of two different rows of M_k to the image patches.

$$(\tilde{\alpha})_{i,j} = FM_k^T \tag{9}$$

$$\hat{\alpha}_{i,j} = \exp(\tilde{\alpha}_{i,j}) / \sum_k \exp(\tilde{\alpha}_{k,j}) \tag{10}$$

$$\alpha_{i,j} = \hat{\alpha}_{i,j} / \sum_k \hat{\alpha}_{i,k} \tag{11}$$

A python-style pseudo-code for external attention is listed in Algorithm 1.

Algorithm 1. Pseudo-Code for External Attention

```
# Input: F, an array with shape [B, N, C]
(batch size, pixels, channels)
# Parameter: M_k, a linear layer without bias
# Parameter: M_v, a linear layer without bias
# Output: out, an array with shape [B, N, C]
F = query_linear(F) # shape=(B, N, C)
attn = M_k(F)    # shape=(B, N, M)
attn = softmax(attn, dim=1)
attn = l1_norm(attn, dim=2)
out = M_v(attn)    # shape=(B, N, C)
```

Algorithm 2. Pseudo-code for Multi-Head External Attention

```
# Input: F, an array of shape [B, N, C_in]
 (batch size, pixels, channels)
# Parameter: M_k, a linear layer
# Parameter: M_v, a linear layer
# Parameter: H, number of heads
# Output: out, an array of shape [B, N, C_in]
F = query_linear(F) # shape=(B, N, C)
F = F.view(B, N, H, C // H)
F = F.permute(0, 2, 1, 3)
attn = M_k(F)  # shape=(B, H, N, M)
attn = softmax(attn, dim=2)
attn = l1_norm(attn, dim=3)
out = M_v(attn) # shape=(B, H, N, C // H)
out = out.permute(0, 2, 1, 3)
out = out.view(B, N, C)
out = W_o(out) # shape=(B, N, C_in)
```

3.3 Multi-head external attention

In Transformer [16], self-attention is computed many times on different input channels, which is called multi-head self-

TABLE 3
Experiments on ImageNet

Method	T2T-Transformer	T2T-Backbone	Input size	#Heads	#Memory units	#Params(M)	#Throughput	Top1(%)
T2T-ViT-7	Performer	Transformer	224 x 224	1	-	4.3	2133.3	67.4
T2T-ViT-7	Performer	Transformer	224 x 224	4	-	4.3	2000.0	71.7
T2T-ViT-14	Performer	Transformer	224 x 224	6	-	21.5	969.7	81.5
T2T-ViT-14	Transformer	Transformer	224 x 224	6	-	21.5	800.0	81.7
T2T-ViT-19	Performer	Transformer	224 x 224	6	-	39.2	666.7	81.9
T2T-ViT-7	EA	Transformer	224 x 224	4	-	4.2	1777.8	71.9
T2T-ViT-14	EA	Transformer	224 x 224	6	-	21.5	941.2	81.7
T2T-ViT-7	Performer	MEA	224 x 224	1	256	4.3	2133.3	63.2
T2T-ViT-7	Performer	MEA	224 x 224	4	256	5.9	1684.2	66.8
T2T-ViT-7	Performer	MEA	224 x 224	16	64	6.2	1684.2	68.6
T2T-ViT-7	Performer	MEA	384 x 384	16	64	6.3	615.4	70.9
T2T-ViT-7	Performer	MEA	224 x 224	16	128	6.3	1454.5	69.9
T2T-ViT-7	Performer	MEA	224 x 224	32	32	9.9	1280.0	70.5
T2T-ViT-14	Performer	MEA	224 x 224	24	64	29.9	744.2	78.7
T2T-ViT-19	Performer	MEA	224 x 224	24	64	54.6	470.6	79.3
MLP-7	MLP	MLP	224 x 224	-	-			Failed
EAMLP-7	EA	MEA	224 x 224	16	64	6.1	1523.8	68.9
EAMLP-BN-7	EA	MEA(BN)	224 x 224	16	64	6.1	1523.8	70.0
EAMLP-7	EA	MEA	384 x 384	16	64	6.2	542.4	71.7
EAMLP-14	EA	MEA	224 x 224	24	64	29.9	711.1	78.9
EAMLP-BN-14	EA	MEA(BN)	224 x 224	24	64			Failed
EAMLP-19	EA	MEA	224 x 224	24	64	54.6	463.8	79.5
EAMLP-BN-19	EA	MEA(BN)	224 x 224	24	64			Failed

Top1: top1 accuracy. EA: external-attention. MEA: multi-head external attention. EAMLP: proposed all MLP architecture. Failed: Unable to converge. EAMLP-BN: replace LN by BN in T2T-ViT backbone's MLP blocks(not external attention blocks). Throughput is measured by using GeForce RTX 3090 graphics card.

attention. Multi-head self-attention can capture different relations between tokens, improving upon the capacity of single head attention.

In fact, there are two manners to build multi-head mechanism. As shown in Fig. 4, the first way is adopting attention for different heads independently like multi-head self-attention. By this way, tokens between different heads cannot interact through multi-head attention mechanism. Another way is that all heads are equal and can interact with each other. We conducted experiments to explore the pros and cons of the two methods. Results in Table 2 show that the second way achieves a better performance. It is worth noting that the current multi-head self-attention is using the first method by default which saves the amount of calculation and memory, but causes a decrease in performance.

For external attention, we can make all tokens interact together by using shared M_k and M_v which can both improve final performance and reduce the amount of parameters and calculations. Algorithm 2 and Fig. 2 display the detail of multi-head external attention.

Multi-head external attention can be written as:

$$h_i = \text{ExternalAttention}(F_i, M_k, M_v), \tag{12}$$

$$F_{out} = \text{MultiHead}(F, M_k, M_v) \tag{13}$$

$$= \text{Concat}(h_1, \ldots, h_H)W_o, \tag{14}$$

where h_i is the ith head, H is the number of heads and W_o is a linear transformation matrix making the dimensions of input and output consistent. $M_k \in \mathbb{R}^{S\times d}$ and $M_v \in \mathbb{R}^{S\times d}$ are the shared memory units for different heads.

The flexibility of this architecture also allows us to balance between the number of head H and number of elements S in shared memory units. For instance, we can multiply H by k while dividing S by k.

4 Experiments

We have conducted experiments on image classification, object detection, semantic segmentation, instance segmentation, image generation, point cloud classification, and point cloud segmentation tasks to assess the effectiveness of our proposed external attention approach. All experiments were implemented with Jittor [101] and Pytorch [102] deep learning frameworks.

4.1 Ablation study

To validate the proposed modules in our full model, we conducted experiments on the PASCAL VOC segmentation dataset [103]. Fig. 3 depicts the architecture used for ablation study, which takes the FCN [48] as the feature backbone. The batch size and total number of iterations were set to 12 and 30,000 respectively. We focus on the number of memory units, self attention versus external attention, the backbone, the normalization method, and output stride of the backbone. As shown in Table 1, we can observe external attention provides better accuracy than self attention on the Pascal VOC dataset. Choosing a suitable number of memory units is important to quality of results. The normalization method can produce a huge positive effect on external attention and make an improvement on self-attention.

4.2 Visual analysis

Attention maps using external attention for segmentation (see Fig. 3) and multi-head external attention for classification (see Section 4.3) are shown in Figs. 5 and 6, respectively. We randomly select a row M_k^i from a memory unit M_k in a layer. Then the attention maps are depicted by calculating the attention of M_k^i to the input feature. We observe that the learned attention maps focus on meaningful objects or

TABLE 4
Comparison to the State-of-the-art Transformer Method on ImnageNet Dataset [49]

Method	#Param(M)	GFLOPs	Acc(%)
PVTv2_B0 [50]	3.4	0.6	70.5
PVTv2_B0_EA(Our)	3.9	0.9	71.8
ResNet18 [51]	11.7	1.8	69.8
DeiT-Tiny/16 [31]	5.7	1.3	72.2
PVTv1-Tiny [52]	13.2	1.9	75.1
PVTv2_B1 [50]	13.1	2.1	78.7
PVTv2_B1_EA(Our)	15.2	3.1	79.1
ResNeXt50-32x4d [53]	25.0	4.3	77.6
RegNetY-4G [54]	21.0	4.0	80.0
DeiT-Small/16 [31]	22.1	4.6	79.9
T2T-ViT-14 [12]	21.5	6.1	81.7
PVTv1-Small [52]	24.5	3.8	79.8
TNT-S [55]	23.8	5.2	81.3
Swin-T [32]	29.0	4.5	81.3
CvT-13 [56]	20.0	4.5	81.6
CoaT-Lite Small [57]	20.0	4.0	81.9
Twins-SVT-S [58]	24.0	2.8	81.7
PVTv2_B2 [50]	25.4	4.0	82.0
PVTv2_B2_EA(Our)	27.0	6.0	81.7
ResNet101 [51]	44.7	7.9	77.4
ResNeXt101-32x4d [53]	44.2	8.0	78.8
RegNetY-8G [54]	39.0	8.0	81.7
T2T-ViT-19 [12]	39.2	9.8	82.4
PVTv1-Medium [52]	44.2	6.7	81.2
CvT-21 [56]	32.0	7.1	82.5
PVTv2_B3 [50]	45.2	6.9	83.2
PVTv2_B3_EA(Our)	46.9	10.4	83.0

Table follows PVTv2 [50]. Acc(%) means Top-1 Accuracy. #Param denotes the number of parameters. GFLOPs is calculated under 224 x 224 input. PVTv2_EA means we replace all attention layer with multi-haed external attention in PVTv2 architecture.

background for segmentation task as in Fig. 5. The last two rows in Fig. 6 suggest that different rows of M_k pay attention to different regions. Each head of multi-head external attention can activate regions of interest to different extents, as shown in Fig. 6, improving the representation ability of external attention.

4.3 Image classification

ImageNet-1K[104] is a widely-used dataset for image classification. We replaced the Performer [105] and multi-head self-attention blocks in T2T-ViT [12] with external attention and multi-head external attention. For fairness, other hyper-parameter settings were the same as T2T-ViT. Experimental results in Table 3 show that external attention achieves a better result than Performer [105] and about 2% point lower than multi-head attention. We find multi-head mechanism is necessary to both self-attention and external attention. We also attempt the strategy proposed by MoCo V3 [44] to replace LayerNorm(LN) [106] by BatchNorm(BN) [107] in the T2T-ViT backbone's MLP blocks(not external attention blocks). We observe a 1% improvement on our EAMLP-7. However, it produces failed cases in our big model EAMLP-14 and EAMLP-19.

Furthermore, we also replace all the attention module with multi-head external attention in PVTv2 [50] architecture which is named PVTv2_EA. We use PVTv2_EA to compare against the latest methods. Result in Table 4 shows that our method achieves comparable performance to other methods. Besides, we also adopt PVTv2_EA as our pre-trained backbone to conduct object detection and instance segmentation experiment. Tables 5 and 6 display that our method is on par with common CNN-based and transformer-based methods.

TABLE 6
Object Detection Experiments on COCO val2017 Dataset

Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}
ResNet50 [51]	ATSS [59]	43.5	61.9	47.0
Swin-T [32]		47.2	66.5	51.3
PVTv2-B2 [50]		49.9	69.1	54.1
PVTv2_B2_EA(Our)		48.6	69.1	52.9
ResNet50 [51]	GFL [60]	44.5	63.0	48.3
Swin-T [32]		47.6	66.8	51.7
PVTv2-B2 [50]		50.2	69.4	54.7
PVTv2_B2_EA(Our)		48.6	67.6	52.8
ResNet50 [51]	Sparse R-CNN [61]	44.5	63.4	48.2
Swin-T [32]		47.9	67.3	52.3
PVTv2-B2 [50]		50.1	69.5	54.9
PVTv2_B2_EA(Our)		48.9	68.4	53.4

All models are pretrained on ImageNet-1K dataset. Table follows [50].

4.4 Object detection and instance segmentation

The MS COCO dataset [108] is a popular benchmark for object detection and instance segmentation. It contains more than 200,000 images with over 500,000 annotated object instances from 80 categories.

MMDetection [62] is a widely-used toolkit for object detection and instance segmentation. We conducted our object detection and instance segmentation experiments using MMDetection with a RestNet-50 backbone, applied to

TABLE 5
Object Detection and Instance Segmentation Experiments on COCO Val2017 Dataset

Backbone	RetinaNet 1×							Mask R-CNN 1×						
	#P (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	#P (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
PVTv2-B0 [50]	13.0	37.2	57.2	39.5	23.1	40.4	49.7	23.5	38.2	60.5	40.7	36.2	57.8	38.6
PVTv2_B0_EA(Our)	13.3	37.2	57.1	39.3	22.1	40.2	49.6	23.8	38.0	60.0	41.0	35.7	57.1	38.2
ResNet18 [51]	21.3	31.8	49.6	33.6	16.3	34.3	43.2	31.2	34.0	54.0	36.7	31.2	51.0	32.7
PVTv1-Tiny [52]	23.0	36.7	56.9	38.9	22.6	38.8	50.0	32.9	36.7	59.2	39.3	35.1	56.7	37.3
PVTv2-B1 [50]	23.8	41.2	61.9	43.9	25.4	44.5	54.3	33.7	41.8	64.3	45.9	38.8	61.2	41.6
PVTv2_B1_EA(Our)	24.9	40.4	61.0	43.1	24.0	44.1	53.5	34.9	41.4	63.6	45.2	38.3	60.8	41.1
ResNet50 [51]	37.7	36.3	55.3	38.6	19.3	40.0	48.8	44.2	38.0	58.6	41.4	34.4	55.1	36.7
PVTv1-Small [52]	34.2	40.4	61.3	43.0	25.0	42.9	55.7	44.1	40.4	62.9	43.8	37.8	60.1	40.3
PVTv2-B2 [50]	35.1	44.6	65.6	47.6	27.4	48.8	58.6	45.0	45.3	67.1	49.6	41.2	64.2	44.4
PVTv2_B2_EA(Our)	36.8	43.7	64.6	46.9	26.8	47.5	57.5	46.7	44.4	66.1	48.8	40.3	63.2	43.6

All models are pretrained on ImageNet-1K dataset.#P means parameter number. AP^b and AP^m denote bounding box AP and mask AP respectively. Table follows [50].

TABLE 7
Experiments on COCO Object Detection Dataset

Method	Backbone	Box AP
Faster RCNN [63]	ResNet-50	37.4
Faster RCNN + 1EA	ResNet-50	38.5
Mask RCNN [64]	ResNet-50	38.2
Mask RCNN + 1EA	ResNet-50	39.0
RetinaNet [65]	ResNet-50	36.5
RetinaNet + 1EA	ResNet-50	37.4
Cascade RCNN [66]	ResNet-50	40.3
Cascade RCNN + 1EA	ResNet-50	41.4
Cascade Mask RCNN [66]	ResNet-50	41.2
Cascade Mask RCNN + 1EA	ResNet-50	42.2

Results quoted are taken from [62]. Box AP: Box Average Precision.

TABLE 8
Experiments on COCO Instance Segmentation Dataset

Method	Backbone	Mask AP
Mask RCNN [64]	ResNet-50	34.7
Mask RCNN + 1EA	ResNet-50	35.4
Cascade RCNN [66]	ResNet-50	35.9
Cascade RCNN + 1EA	ResNet-50	36.7

Results quoted are taken from [62]. Mask AP: Mask Average Precision.

TABLE 9
Comparison to State-of-the-art Methods on the PASCAL VOC Test set w/o COCO Pretraining

Method	Backbone	mIoU(%)
PSPNet [67]	ResNet-101	82.6
DFN [68]	ResNet-101	82.7
EncNet [69]	ResNet-101	82.9
SANet [70]	ResNet-101	83.2
DANet [4]	ResNet-101	82.6
CFNet [71]	ResNet-101	84.2
SpyGR [72]	ResNet-101	84.2
EANet (Ours)	ResNet-101	84.0

TABLE 10
Comparison to State-of-the-art Methods on the ADE20K val Set

Method	Backbone	mIoU(%)
PSPNet [67]	ResNet-101	43.29
PSPNet [67]	ResNet-152	43.51
PSANet [73]	ResNet-101	43.77
EncNet [69]	ResNet-101	44.65
CFNet [71]	ResNet-101	44.89
PSPNet [67]	ResNet-269	44.94
OCNet [17]	ResNet-101	45.04
ANN [74]	ResNet-101	45.24
DANet [4]	ResNet-101	45.26
OCRNet [7]	ResNet-101	45.28
CCNet [5]	ResNet-101	45.76
EANet (Ours)	ResNet-101	45.33

the COCO dataset. We only added our external attention at the end of Resnet stage 4. Results in Tables 7 and 8 show that external attention brings about 1% improvement in accuracy for both object detection and instance segmentation tasks.

4.5 Semantic segmentation

In this experiment, we adopt the semantic segmentation architecture in Fig. 3, referring to it as EANet, and applied it to the Pascal VOC [103], ADE20K [109] and cityscapes [110] datasets.

Pascal VOC contains 10,582 images for training, 1,449 images for validation and 1,456 images for testing. It has 20 foreground object classes and a background class for segmentation. We used dilated ResNet-101 with an output stride of 8 as the backbone for all compared methods; it was pre-trained on ImageNet-1K. A poly-learning rate policy

TABLE 11
Comparison to State-of-the-art Methods on the Cityscapes Val Set; Results Quoted are Taken From [75]

Method	Backbone	mIoU(%)
EncNet [69]	ResNet-101	78.7
APCNet [76]	ResNet-101	79.9
ANN [74]	ResNet-101	80.3
DMNet [77]	ResNet-101	80.7
GCNet [78]	ResNet-101	80.7
PSANet [73]	ResNet-101	80.9
EMANet [6]	ResNet-101	81.0
PSPNet [67]	ResNet-101	81.0
DANet [4]	ResNet-101	82.0
EANet (Ours)	ResNet-101	81.7

Fig. 7. Images generated using our method on cifar-10.

TABLE 12
Comparison to GAN Methods on Cifar-10 Dataset

Method	FID	IS
DCGAN [79]	49.030	6.638
LSGAN [80]	66.686	5.577
WGAN-GP [81]	25.852	7.458
ProjGAN [82]	33.830	7.539
SAGAN [70]	14.498	8.626
EAGAN (Ours)	**14.105**	**8.630**

TABLE 13
Comparison to GAN Methods on Tiny-ImageNet Dataset

Method	FID	IS
DCGAN [79]	91.625	5.640
LSGAN [80]	90.008	5.381
ProjGAN [82]	89.175	6.224
SAGAN [70]	51.414	8.342
EAGAN (Ours)	**48.374**	**8.673**

TABLE 14
Comparison Using the ShaperNet Part Segmentation Dataset

Method	pIoU	airplane	bag	cap	car	chair	earphone	guitar	knife	lamp	laptop	motorbike	mug	pistol	rocket	skateboard	table
PointNet [83]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
Kd-Net [84]	82.3	80.1	74.6	74.3	70.3	88.6	73.5	90.2	87.2	81.0	94.9	57.4	86.7	78.1	51.8	69.9	80.3
SO-Net [85]	84.9	82.8	77.8	88.0	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
PointNet++ [86]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
PCNN [87]	85.1	82.4	80.1	85.5	79.5	90.8	73.2	91.3	86.0	85.0	95.7	73.2	94.8	83.3	51.0	75.0	81.8
DGCNN [88]	85.2	84.0	83.4	86.7	77.8	90.6	74.7	91.2	87.5	82.8	95.7	66.3	94.9	81.1	63.5	74.5	82.6
P2Sequence [89]	85.2	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	95.7	70.8	94.6	79.3	58.1	75.2	82.8
PointConv [90]	85.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PointCNN [91]	86.1	84.1	**86.5**	86.0	80.8	90.6	79.7	**92.3**	88.4	85.3	96.1	**77.2**	95.2	84.2	**64.2**	**80.0**	83.0
PointASNL [92]	86.1	84.1	84.7	87.9	79.7	**92.2**	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
RS-CNN [93]	86.2	83.5	84.8	88.8	79.6	91.2	**81.1**	91.6	88.4	86.0	96.0	73.7	94.1	83.4	60.5	77.7	83.6
PCT [43]	86.4	85.0	82.4	89.0	81.2	91.9	71.5	91.3	88.1	**86.3**	95.8	64.6	**95.8**	83.6	62.2	77.6	**83.7**
Ours	**86.5**	**85.1**	85.7	**90.3**	**81.6**	91.4	75.9	92.1	**88.7**	85.7	**96.2**	74.8	95.7	**84.3**	60.2	76.2	83.5

pIoU: part-average intersection-over-union. Results quoted are taken from cited papers.

TABLE 15
Computational Requirements Compared to Self-Attention and its Variants

Method	SA [16]	DA [4]	A^2 [94]	APC [76]	DM [95]	ACF [96]	Ham [8]	EA (ours)
Params	1.00M	4.82M	1.01M	2.03M	3.00M	0.75M	**0.50M**	0.55M
MACs	292G	79.5G	25.7G	17.6G	35.1G	79.5G	17.6G	**9.2G**

MACs: Multiply-accumulate operations.

was adopted during training. The initial learning rate, batch size and input size were set to 0.009, 16 and 513×513. We first trained for 45k iterations on the training set and then fine-tuned for 15k iterations on the trainval set. Finally we used multi-scale and flip tests on the test set. Visual results are shown in Fig. 5 and quantitative results are given in Table 9: our method can achieve comparable performance to the state-of-the-art methods.

ADE20K is a more challenging dataset with 150 classes, and 20K, 2K, and 3K images for training, validation, and testing, respectively. We adopted dilated ResNet-101 with an output stride of 8 as the backbone. The experimental configuration was the same as for mmsegmentation [75], training ADE20K for 160k iterations. Results in Table 10 show that our method outperforms others on the ADE20K val set.

Cityscapes contains 5,000 high quality pixel-level finely annotated labels in 19 semantic classes for urban scene understanding. Each image is 1024×2048 pixels. It is divided into 2975, 500 and 1525 images for training, validation and testing. (It also contains 20,000 coarsely annotated images, which we did not use in our experiments). We adopted dilated ResNet-101 with an output stride of 8 as the backbone for all methods. The experimental configurations was again the same as for mmsegmentation, training cityscapes with 80k iterations. Results in Table 11 show that our method achieves comparable results to state-of-the-art method, i.e, DANet [4], on the cityscapes val set.

4.6 Image generation

Self-attention is commonly used in image generation, a representative approach being SAGAN [11]. We replaced the self-attention mechanism in SAGAN by our external attention approach in both the generator and discriminator to obtain our EAGAN model. All experiments were based on the popular PyTorch-StudioGAN repo [111]. The hyper-parameters use the default configuration for SAGAN. We used Frechet Inception Distance (FID) [112] and Inception Score (IS) [113] as our evaluation metric. Some generated images are shown in Fig. 7 and quantitative results are given in Tables 12 and 13: external attention provides better results than SAGAN and some other GANs.

TABLE 16
Comparison to State-of-the-art Methods on ModelNet40 Classification Dataset

Method	input	#points	Accuracy
PointNet [83]	P	1k	89.2%
A-SCN [20]	P	1k	89.8%
SO-Net [85]	P, N	2k	90.9%
Kd-Net [84]	P	32k	91.8%
PointNet++ [86]	P	1k	90.7%
PointNet++ [86]	P, N	5k	91.9%
PointGrid [97]	P	1k	92.0%
PCNN [87]	P	1k	92.3%
PointWeb [98]	P	1k	92.3%
PointCNN [91]	P	1k	92.5%
PointConv [90]	P, N	1k	92.5%
A-CNN [99]	P, N	1k	92.6%
P2Sequence [89]	P	1k	92.6%
KPConv [100]	P	7k	92.9%
DGCNN [88]	P	1k	92.9%
RS-CNN [93]	P	1k	92.9%
PointASNL [92]	P	1k	92.9%
PCT [43]	P	1k	93.2%
EAT (Ours)	P	1k	**93.4%**

Accuracy: overall accuracy. All results quoted are taken from the cited papers. P = points, N = normals.

4.7 Point cloud classification

ModelNet40 [114] is a popular benchmark for 3D shape classification, containing 12,311 CAD models in 40 categories. It has 9,843 training samples and 2,468 test samples. Our EAT model replaces all self-attention modules in PCT [43]. We sampled 1024 points on each shape and augmented the input with random translation, anisotropic scaling, and dropout, following PCT [43]. Table 16 indicates that our method outperforms all others, including other attention-based methods like PCT. Our proposed method provides an outstanding backbone for both 2D and 3D vision.

4.8 Point cloud segmentation

We conducted a point cloud segmentation experiment on the ShapeNet part dataset [115]. It has 14,006 3D models in the training set and 2,874 in the evaluation set. Each shape is segmented into parts, with 16 object categories and 50 part labels in total. We followed the experimental setting in PCT [43]. EAT achieved the best results on this dataset, as indicated in Table 14.

4.9 Computational requirements

The linear complexity with respect to the size of the input brings about a significant advantage in efficiency. We compared external attention (EA) module to standard self-attention (SA) [16] and several of its variants in terms of numbers of parameters and inference operations for an input size of $1 \times 512 \times 128 \times 128$, giving the results in Table 15. External attention requires only half of the parameters needed by self-attention and is 32 times faster. Compared to the best variant, external attention is still about twice as fast.

5 Conclusion

This paper has presented external attention, a novel lightweight yet effective attention mechanism useful for various visual tasks. The two external memory units adopted in external attention can be viewed as dictionaries for the whole dataset and are capable of learning more representative features for the input while reducing computational cost. We hope external attention will inspire practical applications and research into its use in other domains such as NLP.

Acknowledgments

We would like to thank Xiang-Li Li for his kind help in experiments, Jun-Xiong Cai for helpful discussions, and Prof. Ralph R. Martin for his insightful suggestions and great help in writing.

References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2015, *arXiv:1409.0473*.

[2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[3] X. Wang, R. B. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[4] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3146–3154.

[5] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 603–612.

[6] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9166–9175.

[7] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 173–190.

[8] Z. Geng, M.-H. Guo, H. Chen, X. Li, K. Wei, and Z. Lin, "Is attention better than matrix decomposition?," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=1FvkSpWosOl

[9] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[10] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[11] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

[12] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on imagenet," in *Proc. IEEE/CVF Int. Conf. Comput.*, 2021, pp. 558–567.

[13] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2204–2212.

[14] K. Xu et al., "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[15] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[16] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[17] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen and J Wang, "OCNet: Object context for semantic segmentation," *Int. J. Comput. Vis.*, vol. 129 no. 8, pp. 2375–2398, 2021.

[18] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[19] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, "Relation networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3588–3597.

[20] S. Xie, S. Liu, Z. Chen, and Z. Tu, "Attentional shapecontextnet for point cloud recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4606–4615.

[21] M.-H. Guo et al., "Attention mechanisms in computer vision: A survey," *Comput. Vis. Media*, vol. 8, no. 3, pp. 329–367, 2022.

[22] I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens, "Attention augmented convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3285–3294.

[23] N. Parmar, P. Ramachandran, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 68–80.

[24] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10073–10082.

[25] Z. Lin et al., "A structured self-attentive sentence embedding," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=BJC_jUqx

[26] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 5754–5764.

[27] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. Assoc. Comput. Linguistics*, 2019, pp. 2978–2988.

[28] J. Lee et al., "BioBERT: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[29] 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.*, 2020, pp. 213–229.

[30] M. Chen et al., "Generative pretraining from pixels," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1691–1703.

[31] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in*Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[32] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10012–10022.

[33] H. Fan et al., "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 6824–6835.

[34] 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. Representations*, 2021.

[35] H. Chen et al., "Pre-trained image processing transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition.*, 2021, pp. 12299–12310.

[36] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition.*, 2021, pp. 6881–6890.

[37] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8126–8135.

[38] Y. Wang et al., "End-to-end video instance segmentation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition.*, 2021, pp. 8741–8750.

[39] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two pure transformers can make one strong gan, and that can scale up," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 14745–14758, 2021.

[40] R. Hu and A. Singh, "Unit: Multimodal multitask learning with a unified transformer," in *Proc. in IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1439–1449.

[41] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, "TransReID: Transformer-based object re-identification," in *Proc. in IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 15013–15022.

[42] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "CPTR: Full transformer network for image captioning," 2021, *arXiv:2101.10804*.

[43] M. Guo, J. Cai, Z. Liu, T. Mu, R. R. Martin, and S. Hu, "PCT: Point cloud transformer," *Comput. Vis. Media*, vol. 7, no. 2, pp. 187–199, 2021.

[44] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 9640–9649.

[45] K. Han et al., "A survey on visual transformer," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, doi: 10.1109/TPAMI.2022.3152247.

[46] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM Comput. Surveys*, vol. 54, no. 10s, 2021, Art. no. 200.

[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

[48] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[49] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[50] W. Wang et al., "PVTV2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, 2022.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[52] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 568–578.

[53] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.

[54] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10 428–10 436.

[55] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 15908–15919, 2021.

[56] H. Wu et al., "CVT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 22–31.

[57] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9981–9990.

[58] X. Chu et al., "Twins: Revisiting the design of spatial attention in vision transformers," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 34, pp. 9355–9366, 2021.

[59] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition.*, 2020, pp. 9759–9768.

[60] X. Li et al., "Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection," *Proc. Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 21 002–21 012, 2020.

[61] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognition.*, 2021, pp. 14454–14463.

[62] K. Chen et al., "Mmdetection: Open mmlab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.

[63] S. Ren, K. He, R. B. 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.

[64] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[65] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.

[66] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1483–1498, May 2021.

[67] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239.

[68] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Learning a discriminative feature network for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1857–1866.

[69] H. Zhang et al., "Context encoding for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7151–7160.

[70] Z. Zhong et al., "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13062–13071.

[71] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 548–557.

[72] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, "Spatial pyramid based graph reasoning for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8947–8956.

[73] H. Zhao et al., "PSANet: Point-wise spatial attention network for scene parsing," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 270–286.

[74] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, "Asymmetric non-local neural networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 593–602.

[75] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," 2020. [Online]. Available: https://github.com/open-mmlab/mmsegmentation

[76] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, "Adaptive pyramid context network for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7519–7528.

[77] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3561–3571.

[78] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 1971–1980.

[79] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2016, *arXiv:1511.06434*.

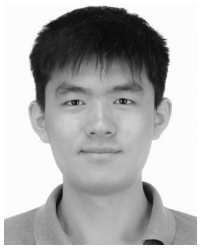
[80] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2813–2821.

[81] X. Wei, Z. Liu, L. Wang, and B. Gong, "Improving the improved training of wasserstein GANs," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=SJx9GQb0-

[82] T. Miyato and M. Koyama, "cGANs with projection discriminator," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ByS1VpgRZ

[83] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.
[84] R. Klokov and V. S. Lempitsky, "Escape from cells: Deep KD-networks for the recognition of 3d point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 863–872.
[85] J. Li, B. M. Chen, and G. H. Lee, "So-net: Self-organizing network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9397–9406.
[86] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
[87] M. Atzmon, H. Maron, and Y. Lipman, "Point convolutional neural networks by extension operators," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 71:1–71:12, 2018.
[88] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 146:1–146:12, 2019.
[89] X. Liu, Z. Han, Y. Liu, and M. Zwicker, "Point2Sequence: Learning the shape representation of 3D point clouds with an attention-based sequence to sequence network," in *Proc. 33rd AAAI Conf. Artif. Intell. 31st Innov. Appl. Artif. Intell. Conf. 9th AAAI Symp. Educ. Adv. Artif. Intell.*, 2019, pp. 8778–8785.
[90] W. Wu, Z. Qi, and F. Li, "PointConv: Deep convolutional networks on 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9621–9630.
[91] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, "PointCNN: Convolution on X-transformed points," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 828–838.
[92] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, "PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5588–5597.
[93] Y. Liu, B. Fan, S. Xiang, and C. Pan, "Relation-shape convolutional neural network for point cloud analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8895–8904.
[94] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A^2-Nets: Double attention networks," in *Proc. Adv. Neural Inform. Process. Syst.*, pp. 350–359, 2018.
[95] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3561–3571.
[96] H. Zhang, H. Zhang, C. Wang, and J. Xie, "Co-occurrent features in semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 548–557.
[97] T. Le and Y. Duan, "PointGrid: A deep network for 3D shape understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9204–9214.
[98] H. Zhao, L. Jiang, C. Fu, and J. Jia, "PointWeb: Enhancing local neighborhood features for point cloud processing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 5565–5573.
[99] A. Komarichev, Z. Zhong, and J. Hua, "A-CNN: Annularly convolutional neural networks on point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7421–7430.
[100] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6410–6419.
[101] S.-M. Hu, D. Liang, G.-Y. Yang, G.-W. Yang, and W.-Y. Zhou, "Jittor: A novel deep learning framework with meta-operators and unified graph execution," *Sci. China Inf. Sci.*, vol. 63, no. 222103, pp. 1–21, 2020.
[102] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 8024–8035.
[103] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
[104] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[105] K. Choromanski et al., "Rethinking attention with performers," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=Ua6zuk0WRH
[106] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[107] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[108] T. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[109] B. Zhou et al., "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019.
[110] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3213–3223.
[111] M. Kang and J. Park, "ContraGAN: Contrastive learning for conditional image generation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21357–21369.
[112] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
[113] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2226–2234.
[114] Z. Wu et al., "3D shapenets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1912–1920.
[115] L. Yi et al., "A scalable active framework for region annotation in 3D shape collections," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 210:1–210:12, 2016.

Meng-Hao Guo received the bachelor's degree from Xidian University. He is currently working toward the PhD degree supervised by prof. Shi-Min Hu in the Department of Computer Science and Technology with Tsinghua University, Beijing, China. His research interests include computer vision, computer graphics, and machine learning.



Zheng-Ning Liu received the bachelor's degree in computer science from Tsinghua University, in 2017. He is currently working toward the PhD degree in computer science with Tsinghua University. His research interests include 3D computer vision, 3D reconstruction, and computer graphics.

Tai-Jiang Mu received the BS and PhD degrees in computer science, in 2011 and 2016, respectively. He is currently an assistant researcher with Tsinghua University. His research interests include computer vision, robotics and computer graphics.

Shi-Min Hu (Senior Member, IEEE) received the PhD degree from Zhejiang University, in 1996. He is currently a professor in Computer Science with Tsinghua University. His research interests include geometry processing, image & video processing, rendering, computer animation, and CAD. He has published more than 100 papers in journals and refereed conferences. He is Editor-in-Chief of Computational Visual Media, and on the editorial boards of several journals, including Computer Aided Design and Computer & Graphics.