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

Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, Shi-Min Hu

BNRist, Department of Computer Science and Technology

Tsinghua University, Beijing 100084

{gmh20, liu-zn17}@mails.tsinghua.edu.cn {taijiang, shimin}@tsinghua.edu.cn

Abstract

Attention mechanisms, especially self-attention, play an increasingly important role in deep feature representation in 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 long-range dependency within a single sample. However, self-attention has a quadratic complexity and ignores potential correlation between different samples. This paper proposes a novel attention mechanism which we call *external attention*, based on two external, small, learnable, and 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 samples. Extensive experiments on image classification, semantic segmentation, image generation, point cloud classification and point cloud segmentation tasks reveal that our method provides comparable or superior performance to the self-attention mechanism and some of its variants, with much lower computational and memory costs.

1 Introduction

Given 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 focuses on refining the representation at each position by aggregating features from all other locations in a single sample, which leads to quadratic computational complexity in the number of locations in the sample. Thus, some variants attempt to approximate self-attention to 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 potential 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 a 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 Figure 1c)). As shown in Figure 1a), to compute self-attention, we first calculate an attention map by computing the affinities between self query vectors and self key vectors, then generate 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 linear computational complexity w.r.t. 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. 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.

Though 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]. Figure 2 demonstrates a typical encoder-decoder like architecture replacing self-attention with our external attention for an image semantic segmentation task. We have conducted extensive experiments for such basic visual tasks as classification, semantic segmentation and generation, with different input modalities (images and point cloud). The results reveal that our method achieves comparable or better results than the original self-attention mechanism and some of its variants, at much lower computational cost.

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 [13, 14, 15, 16, 3, 17, 18]. Hu et al. proposed SENet [15], showing that the attention mechanism can reduce noise and improve classification performance. After that, many other papers 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. Xie et al. proposed A-SCN [20] for point cloud processing.

2.2 Self-attention in visual tasks

Self-attention is a special case of attention, and many papers [3, 17, 4, 11], 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 first considers row attention and then column attention 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.

Unlike self-attention which obtains the 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 using the back-propagation algorithm in an end-to-end way instead of using an iterative clustering algorithm.

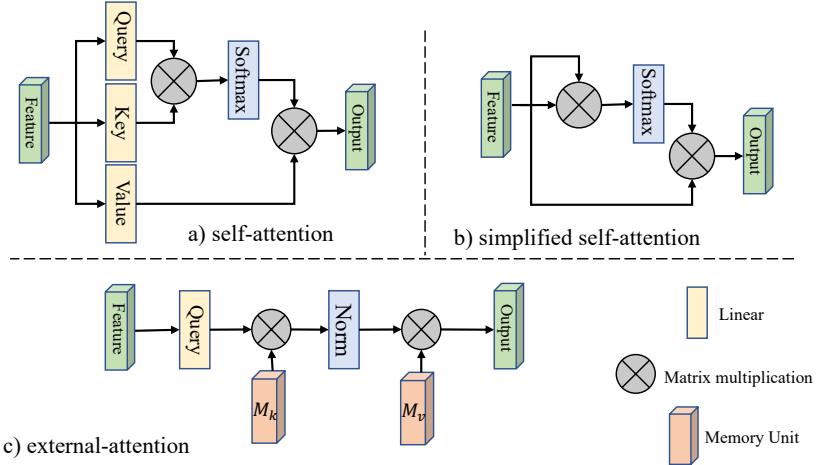


Figure 1: Self-attention versus external-attention

2.3 Transformer in visual tasks

Transformer-based models have had great success in natural language processing [1, 21, 16, 2, 22, 23, 24]. Recently, they have also demonstrated huge potential for visual tasks. Carion et al. [25] presented an end-to-end detection transformer that takes CNN features as input and generates bounding boxes with a transformer. Chen et al. [26] proposed iGPT for image generation based on use of 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.

Subsequently, transformer methods have been successfully applied to many visual tasks, including image classification [27, 12], object detection [28], lower-level vision [29], semantic segmentation [30], tracking [31], video instance segmentation [32], image generation [33], multimodal learning [34], object re-identification [35], image captioning [36], and point cloud learning [37]. Readers are referred to recent surveys [38, 39] for a more comprehensive review of use of transformer methods for visual tasks.

3 Method

In this section, we firstly analyze the original self-attention mechanism. Then we will detail our External Attention, which is a novel way to define attention. It can be implemented easily by only using two linear layers and two normalization layers as shown in Listing1.

3.1 External Attention

We first revisit the self-attention mechanism (see Figure 1a)). Given an input feature map $F \in \mathbb{R}^{N \times d}$, where N is the number of pixels 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:

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

$$F_{out} = AV \quad (2)$$

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

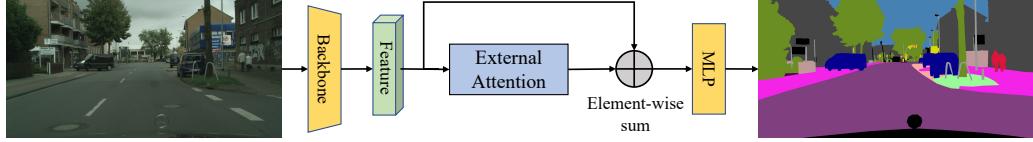


Figure 2: The architecture of EANet for semantic segmentation using our proposed external attention.

A popular simplified variation (Figure 1b)) of self-attention simplifies the QKV matrices, and directly calculates the attention map by using the input feature F as:

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

$$F_{out} = AF \quad (4)$$

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 if simplified, the high computational complexity of $\mathcal{O}(dN^2)$ 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 works [18] utilize self-attention on patches rather than pixels to reduce the computational complexity.

When visualizing the attention map, we find that most pixels are closely related to just a few other pixels, and an N -to- N attention matrix may be redundant. Thus, the refined features can be fulfilled by using the needed values. Thus, we propose as an alternative an *external attention* module, which computes attention between the input pixels and an external memory unit $M \in \mathbb{R}^{S \times d}$, as:

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

$$F_{out} = AM \quad (6)$$

Unlike self-attention, $\alpha_{i,j}$ in Equation (5) is the similarity between the i -th pixel and the j -th rows of M . Here, M is a learnable parameter independent of the input, which acts as a memory for the whole training dataset. A is the attention map inferred from 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. The overall computation of external attention is now

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

$$F_{out} = AM_v \quad (8)$$

The computational complexity of external attention is $\mathcal{O}(dSN)$. 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.

Listing 1: Python 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]
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)
```

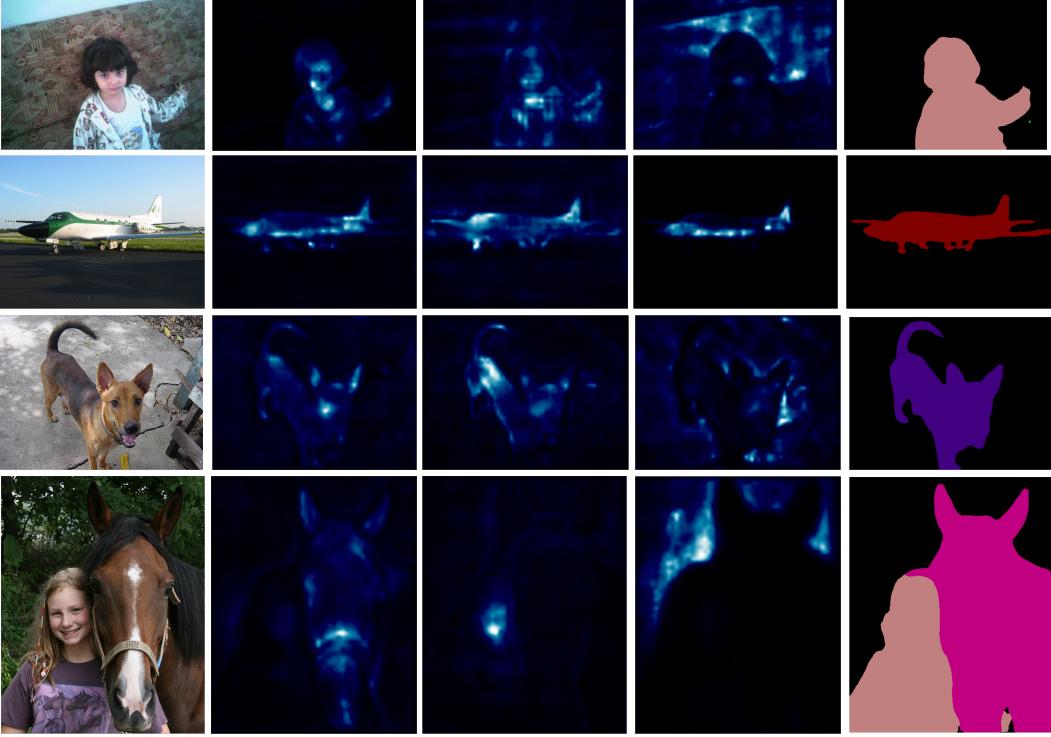


Figure 3: attention map.

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 [37], which separately normalizes columns and rows. The double-normalization is formulated as:

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

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

$$\alpha_{i,j} = \frac{\hat{\alpha}_{i,j}}{\sum_k \hat{\alpha}_{i,k}} \quad (11)$$

4 Experiments

We have conducted experiments on image classification, semantic 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 [40] and Pytorch [41] deep learning frameworks.

4.1 Image classification

ImageNet-1K[42] is a widely-used dataset for image classification. We replaced the T2T-Transformer blocks in T2T-ViT [12] with our external attention. To make a fair comparison, the other hyperparameter settings were the same as for T2T-ViT. Experimental results in Table 1 show that external attention achieves comparable results to multi-head attention and performer [43].

Table 1: Experiments on ImageNet based on T2T-ViT. Top1 means top1 accuracy

Method	T2T-Transformer	Top1
T2T-ViT-7	Performer	71.7%
T2T-ViT-14	Performer	81.5%
T2T-ViT-14	Transformer	81.7%
T2T-ViT-7	Our Attention	71.7%
T2T-ViT-14	Our Attention	81.7%

Table 2: Comparison to state-of-the-art methods on the PASCAL VOC test set w/o COCO pretraining.

Method	Backbone	mIoU(%)
PSPNet [44]	ResNet-101	82.6
DFN [45]	ResNet-101	82.7
EncNet [46]	ResNet-101	82.9
SANet [47]	ResNet-101	83.2
DANet [4]	ResNet-101	82.6
CFNet [48]	ResNet-101	84.2
SpyGR [49]	ResNet-101	84.2
EANet(Ours)	ResNet-101	84.0

Table 3: Comparison to state-of-the-art methods on the ADE20K val set.

Method	Backbone	mIoU(%)
PSPNet [44]	ResNet-101	43.29
PSPNet [44]	ResNet-152	43.51
PSANet [50]	ResNet-101	43.77
EncNet [46]	ResNet-101	44.65
CFNet [48]	ResNet-101	44.89
PSPNet [44]	ResNet-269	44.94
OCNet [17]	ResNet-101	45.04
ANN [51]	ResNet-101	45.24
DANet [4]	ResNet-101	45.26
OCRNet [7]	ResNet-101	45.28
EANet(Ours)	ResNet-101	45.33

Table 4: Comparison to state-of-the-art methods on the cityscapes val set; results quoted are taken from [52].

Method	Backbone	mIoU(%)
EncNet [46]	ResNet-101	78.7
APCNet [53]	ResNet-101	79.9
ANN [51]	ResNet-101	80.3
DMNet [54]	ResNet-101	80.7
GCNet [55]	ResNet-101	80.7
PSANet [50]	ResNet-101	80.9
EMANet [6]	ResNet-101	81.0
PSPNet [44]	ResNet-101	81.0
DANet [4]	ResNet-101	82.0
EANet(Ours)	ResNet-101	81.7

4.2 Semantic segmentation

In this experiment, we adopted the semantic segmentation architecture in Figure 2 and used the Pascal VOC dataset [56], the ADE20K dataset [57] and the cityscapes dataset [58].

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; the latter was pre-trained on ImageNet-1K. A poly-learning rate policy was adopted in the training stage. 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; results are shown in Table 2.

ADE20K is a more challenging dataset with 150 classes, with 20K, 2K, and 3K images for training, validation, and testing. We adopted dilated ResNet-101 with an output stride of 8 as the backbone. The experimental configurations are same as for mmsegmentation, training ADE20K for 160k iterations. Results in Table 3 show that our method provides better results 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. The experimental configurations are same as for mmsegmentation, training ADE20K with 80k iterations. Results in Table 4 show that our method achieves comparable results on the cityscapes val set.

Table 5: Compared with some GAN methods on cifar-10 dataset.

Method	FID	IS
DCGAN [59]	49.03	6.638
LSGAN [60]	66.686	5.577
WGAN-GP [61]	25.852	7.458
ProjGAN [62]	33.830	7.539
SAGAN [47]	14.498	8.626
EAGAN(Ours)	14.105	8.630

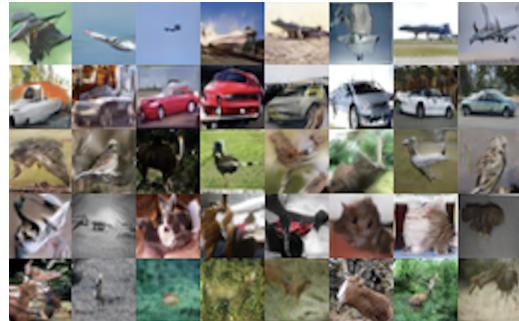


Figure 4: Generated images using our method on cifar-10.

Table 6: Comparison to state-of-the-art methods using the ModelNet40 classification dataset. Accuracy means overall accuracy. All results quoted are taken from the cited papers. P = points, N = normals.

Method	input	#points	Accuracy
PointNet [63]	P	1k	89.2%
A-SCN [20]	P	1k	89.8%
SO-Net [64]	P, N	2k	90.9%
Kd-Net [65]	P	32k	91.8%
PointNet++ [66]	P	1k	90.7%
PointNet++ [66]	P, N	5k	91.9%
PointGrid [67]	P	1k	92.0%
PCNN [68]	P	1k	92.3%
PointWeb [69]	P	1k	92.3%
PointCNN [70]	P	1k	92.5%
PointConv [71]	P, N	1k	92.5%
A-CNN [72]	P, N	1k	92.6%
P2Sequence [73]	P	1k	92.6%
KPConv [74]	P	7k	92.9%
DGCNN [75]	P	1k	92.9%
RS-CNN [76]	P	1k	92.9%
PointASNL [77]	P	1k	92.9%
PCT [37]	P	1k	93.2%
EAT(Ours)	P	1k	93.4%

4.3 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 are based on the popular PyTorch-StudioGAN repo [78]. The hyper-parameters use the default configuration for SAGAN. We choose Frechet Inception Distance (FID) [79] and Inception Score (IS) [80] as our evaluation metric. Some generated images are shown in Figure 4 and quantitative results in Table 5 reveal that external attention provides better performance than SAGAN and some other GANs.

4.4 Point cloud classification

ModelNet40 [81] is a popular benchmark for 3D shape classification, containing 12,311 CAD models in 40 categories. There are 9,843 training samples and 2,468 test samples. EAT replaces all self-attention modules in PCT [37]. We sampled 1024 points on each shape and augmented the input with random translation, anisotropic scaling, and dropout, following PCT [37]. Table 6 indicates that our method outperforms all other methods, including some other attention-based methods like PCT. The

Table 7: Comparison using the ShaperNet part segmentation dataset. pIoU means part-average intersection-over-nnion. All results quoted are taken from the cited papers.

Method	pIoU	air-plane	bag	cap	car	chair	ear-phone	guitar	knife	lamp	laptop	motor-bike	mug	pistol	rocket	skate-board	table
PointNet [63]	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 [65]	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 [64]	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++ [66]	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 [68]	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 [75]	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 [73]	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 [71]	85.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PointCNN [70]	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 [77]	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 [76]	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 [37]	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
Our	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

Table 8: Computational requirements compared to self-attention and its variants. MACs: Multiply-accumulate operations.

Method	SA [16]	DA [4]	A^2 [83]	APC [53]	DM [84]	ACF [85]	Ham [8]	EA(our)
Params	1.00M	4.82M	1.01M	2.03M	3.00M	0.75M	0.50M	0.33M
MACs	292G	79.5G	25.7G	17.6G	35.1G	79.5G	17.6G	5.4G

results highlight that the proposed method provides an outstanding backbone for both 2D and 3D vision.

4.5 Point cloud segmentation

We conducted a point cloud segmentation experiment on the ShapeNet part dataset [82]. There are 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 [37]. EAT achieves the best results on this dataset, as reported in Table 7.

4.6 Computational requirements

The linear complexity with respect to size of inputs brings about a significant advantage in efficiency. We compared external attention to standard self-attention (SA) [16] and several of its variants in terms of numbers of parameters and inference operations under the input size of $1 \times 512 \times 128 \times 128$, and report the results in Table 8. External attention requires only a third of parameters needed by self-attention and its speed is 50 times faster. Compared to the best variant, external attention is still 3 times faster, and much more more light-weight and efficient.

5 Conclusions

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 a more representative feature for the input while trading off computational cost. We hope external attention will inspire practical applications and researches into its use in other domains such as NLP.

References

- [1] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *International Conference on Learning Representations*, 2015. [Online].

Available: <http://arxiv.org/abs/1409.0473>

- [2] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: pre-training of deep bidirectional transformers for language understanding,” in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186. [Online]. Available: <https://doi.org/10.18653/v1/n19-1423>
- [3] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [4] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, “Dual attention network for scene segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 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,” 2019.
- [6] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, “Expectation-maximization attention networks for semantic segmentation,” in *International Conference on Computer Vision*, 2019.
- [7] Y. Yuan, X. Chen, and J. Wang, “Object-contextual representations for semantic segmentation,” 2020.
- [8] Z. Geng, M.-H. Guo, H. Chen, X. Li, K. Wei, and Z. Lin, “Is attention better than matrix decomposition?” in *International Conference on Learning Representations*, 2021.
- [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, 2006.
- [11] H. Zhang, I. J. Goodfellow, D. N. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 7354–7363. [Online]. Available: <http://proceedings.mlr.press/v97/zhang19d.html>
- [12] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. Tay, J. Feng, and S. Yan, “Tokens-to-token vit: Training vision transformers from scratch on imagenet,” 2021.
- [13] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, “Recurrent models of visual attention,” 2014.
- [14] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” 2016.
- [15] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2018, pp. 7132–7141. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008. [Online]. Available: <http://papers.nips.cc/paper/7181-attention-is-all-you-need>
- [17] Y. Yuan and J. Wang, “Ocnet: Object context network for scene parsing,” 2019.
- [18] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recognition at scale,” *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: <https://arxiv.org/abs/2010.11929>
- [19] H. Hu, J. Gu, Z. Zhang, J. Dai, and Y. Wei, “Relation networks for object detection,” 2018.
- [20] S. Xie, S. Liu, Z. Chen, and Z. Tu, “Attentional shapecontextnet for point cloud recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

- [21] Z. Lin, M. Feng, C. N. dos Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, “A structured self-attentive sentence embedding,” in *International Conference on Learning Representations*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=BJC_jUqxe
- [22] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” in *Advances in Neural Information Processing Systems*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 5754–5764. [Online]. Available: <http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding>
- [23] 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 *Association for Computational Linguistics*, A. Korhonen, D. R. Traum, and L. Màrquez, Eds. Association for Computational Linguistics, 2019, pp. 2978–2988. [Online]. Available: <https://doi.org/10.18653/v1/p19-1285>
- [24] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “Biobert: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020. [Online]. Available: <https://doi.org/10.1093/bioinformatics/btz682>
- [25] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End object detection with transformers,” *CoRR*, vol. abs/2005.12872, 2020. [Online]. Available: <https://arxiv.org/abs/2005.12872>
- [26] M. Chen, A. Radford, R. Child, J. Wu, H. Jun, P. Dhariwal, D. Luan, and I. Sutskever, “Generative pretraining from pixels,” 2020.
- [27] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” 2021.
- [28] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” 2021.
- [29] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, “Pre-trained image processing transformer,” 2020.
- [30] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, “Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers,” 2020.
- [31] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, “Transformer tracking,” in *CVPR*, 2021.
- [32] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia, “End-to-end video instance segmentation with transformers,” 2020.
- [33] Y. Jiang, S. Chang, and Z. Wang, “Transgan: Two transformers can make one strong gan,” 2021.
- [34] R. Hu and A. Singh, “Transformer is all you need: Multimodal multitask learning with a unified transformer,” 2021.
- [35] S. He, H. Luo, P. Wang, F. Wang, H. Li, and W. Jiang, “Transreid: Transformer-based object re-identification,” 2021.
- [36] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, “Cptr: Full transformer network for image captioning,” 2021.
- [37] 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.
- [38] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, “A survey on visual transformer,” 2021.
- [39] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, “Transformers in vision: A survey,” 2021.
- [40] 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,” *Information Sciences*, vol. 63, no. 222103, pp. 1–21, 2020.
- [41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” 2019.

- [42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [43] K. M. Choromanski, V. Likhoshesterov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller, “Rethinking attention with performers,” in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=Ua6zuk0WRH>
- [44] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *CVPR*, 2017.
- [45] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Learning a discriminative feature network for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [46] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, “Context encoding for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [47] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. Ben Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong, “Squeeze-and-attention networks for semantic segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [48] H. Zhang, H. Zhang, C. Wang, and J. Xie, “Co-occurrent features in semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [49] X. Li, Y. Yang, Q. Zhao, T. Shen, Z. Lin, and H. Liu, “Spatial pyramid based graph reasoning for semantic segmentation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [50] H. Zhao, Y. Zhang, S. Liu, J. Shi, C. C. Loy, D. Lin, and J. Jia, “PSANet: Point-wise spatial attention network for scene parsing,” in *ECCV*, 2018.
- [51] Z. Zhu, M. Xu, S. Bai, T. Huang, and X. Bai, “Asymmetric non-local neural networks for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [52] M. Contributors, “MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark,” <https://github.com/open-mmlab/mmsegmentation>, 2020.
- [53] J. He, Z. Deng, L. Zhou, Y. Wang, and Y. Qiao, “Adaptive pyramid context network for semantic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [54] J. He, Z. Deng, and Y. Qiao, “Dynamic multi-scale filters for semantic segmentation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
- [55] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, “Gcnet: Non-local networks meet squeeze-excitation networks and beyond,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [56] 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.
- [57] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ADE20K dataset,” *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, 2019.
- [58] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” 2016.
- [59] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2016.
- [60] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, “Least squares generative adversarial networks,” *arXiv preprint arXiv:1611.04076*, 2016.
- [61] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of wasserstein gans,” 2017.

- [62] T. Miyato and M. Koyama, “cgans with projection discriminator,” 2018.
- [63] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “Pointnet: Deep learning on point sets for 3d classification and segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2017, pp. 77–85. [Online]. Available: <https://doi.org/10.1109/CVPR.2017.16>
- [64] J. Li, B. M. Chen, and G. H. Lee, “So-net: Self-organizing network for point cloud analysis,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2018, pp. 9397–9406. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Li_SO-Net_Self-Organizing_Network_CVPR_2018_paper.html
- [65] R. Klokov and V. S. Lempitsky, “Escape from cells: Deep kd-networks for the recognition of 3d point cloud models,” in *IEEE International Conference on Computer Vision*. IEEE Computer Society, 2017, pp. 863–872. [Online]. Available: <https://doi.org/10.1109/ICCV.2017.99>
- [66] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108. [Online]. Available: <http://papers.nips.cc/paper/7095-pointnet-deep-hierarchical-feature-learning-on-point-sets-in-a-metric-space>
- [67] T. Le and Y. Duan, “Pointgrid: A deep network for 3d shape understanding,” in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2018, pp. 9204–9214. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Le_PointGrid_A_Deep_CVPR_2018_paper.html
- [68] M. Atzmon, H. Maron, and Y. Lipman, “Point convolutional neural networks by extension operators,” *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 71:1–71:12, 2018. [Online]. Available: <https://doi.org/10.1145/3197517.3201301>
- [69] H. Zhao, L. Jiang, C. Fu, and J. Jia, “Pointweb: Enhancing local neighborhood features for point cloud processing,” in *IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 2019, pp. 5565–5573. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Zhao_PointWeb_Enhancing_Local_Neighborhood_Features_for_Point_Cloud_Processing_CVPR_2019_paper.html
- [70] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “PointCNN: Convolution on x-transformed points,” in *Advances in Neural Information Processing Systems*, 2018, pp. 828–838. [Online]. Available: <http://papers.nips.cc/paper/7362-pointcnn-convolution-on-x-transformed-points>
- [71] W. Wu, Z. Qi, and F. Li, “PointConv: Deep convolutional networks on 3d point clouds,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9621–9630.
- [72] A. Komarichev, Z. Zhong, and J. Hua, “A-CNN: annularly convolutional neural networks on point clouds,” in *IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 2019, pp. 7421–7430. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Komarichev_A-CNN_Annularly_Convolutional_Neural_Networks_on_Point_Clouds_CVPR_2019_paper.html
- [73] 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 *AAAI Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 8778–8785. [Online]. Available: <https://doi.org/10.1609/aaai.v33i01.33018778>
- [74] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, “Kpconv: Flexible and deformable convolution for point clouds,” in *IEEE/CVF International Conference on Computer Vision*. IEEE, 2019, pp. 6410–6419. [Online]. Available: <https://doi.org/10.1109/ICCV.2019.00651>
- [75] 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 Transactions on Graphics*, vol. 38, no. 5, pp. 146:1–146:12, 2019. [Online]. Available: <https://doi.org/10.1145/3326362>
- [76] Y. Liu, B. Fan, S. Xiang, and C. Pan, “Relation-shape convolutional neural network for point cloud analysis,” in *IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation / IEEE, 2019, pp. 8895–8904. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_Relation-Shape_Convolutional_Neural_Network_for_Point_Cloud_Analysis_CVPR_2019_paper.html

- [77] X. Yan, C. Zheng, Z. Li, S. Wang, and S. Cui, “PointASNL: Robust point clouds processing using nonlocal neural networks with adaptive sampling,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, 2020, pp. 5588–5597. [Online]. Available: <https://doi.org/10.1109/CVPR42600.2020.00563>
- [78] M. Kang and J. Park, “ContraGAN: Contrastive Learning for Conditional Image Generation,” 2020.
- [79] 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,” 2018.
- [80] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” 2016.
- [81] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *CVPR*. IEEE Computer Society, 2015, pp. 1912–1920.
- [82] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. J. Guibas, “A scalable active framework for region annotation in 3d shape collections,” *ACM Trans. Graph.*, vol. 35, no. 6, pp. 210:1–210:12, 2016.
- [83] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, “ A^2 -nets: Double attention networks,” *CoRR*, vol. abs/1810.11579, 2018.
- [84] J. He, Z. Deng, and Y. Qiao, “Dynamic multi-scale filters for semantic segmentation,” in *ICCV*. IEEE, 2019, pp. 3561–3571.
- [85] H. Zhang, H. Zhang, C. Wang, and J. Xie, “Co-occurrent features in semantic segmentation,” in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 548–557.