Patch-based Separable Transformer for Visual Recognition

Shuyang Sun, Xiaoyu Yue, Hengshuang Zhao, Philip H.S. Torr, Song Bai

Abstract—The computational complexity of transformers limits it to be widely deployed onto frameworks for visual recognition. Recent work [9] significantly accelerates the network processing speed by reducing the resolution at the beginning of the network, however, it is still hard to be directly generalized onto other downstream tasks *e.g.* object detection and segmentation like CNN. In this paper, we present a transformer-based architecture retaining both the local and global interactions within the network, and can be transferable to other downstream tasks. The proposed architecture reforms the original full spatial self-attention into pixel-wise local attention and patch-wise global attention. Such factorization saves the computational cost while retaining the information of different granularities, which helps generate multi-scale features required by different tasks. By exploiting the factorized attention, we construct a Separable Transformer (SeT) for visual modeling. Experimental results show that SeT outperforms the previous state-of-the-art transformer-based approaches and its CNN counterparts on three major tasks including image classification, object detection and instance segmentation.

Index Terms—Transformer, Image Classification, Object Detection, Instance Segmentation

1 Introduction

Convolutional Neural Networks (CNNs) [14], [25], [36] have been dominating nearly all vision tasks since the emergence of AlexNet [25]. However, CNN has been proven to be less effective when applied on some other sequential data like natural language, where the attention-based Transformers [42] serve as the most popular feature extractor. The attention model like Transformer [42], due to its data-adaptive nature, conceptually have a larger capacity and preserve the translational equivariance [31], [52]. These properties imply the potential of attention-based models as a better engine for vision tasks.

Existing attention-based networks [9], [19], [31], [52] cannot be practically transferred onto downstream tasks due to the computational and memory inefficiency [19], [52] or the lack of multi-scale features [9], [41]. These attention-based models can be roughly divided into two kinds, (1) token-based global attention [9], [41] and (2) convolution-like local attention [19], [31], [52]. To save the computational and memory cost, Vision Transformer (ViT) [9] as a typical representation of the token-based approach, proposes to summarize all pixels of independent patches into global visual tokens before feeding them into the Transformer encoder. However, as human vision knows when and how to capture the information at the highest resolution, an ideal visual processor should be capable to analyze both the local and global information at the same time. Visual modeling without interacting with the local information is counter-intuitive. As a consequence, the lack of local information modeling in ViT results in data-inefficiency and limited transferability to downstream tasks that call for multi-scale features.

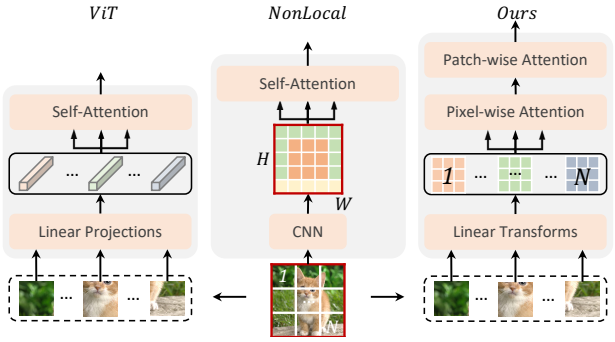

Fig. 1. Left: Vision Transformer (ViT) [9] directly projects all pixels of each patch into global tokens and applies self-attention afterwards. Middle: the full spatial attention (*a.k.a.* Non-local [44] block) takes all pixels of an image as inputs so that it will have huge memory and computational cost when the resolution is high. Right: our SeT factorizes the full spatial attention into (1) a pixel-wise attention that only interacts with local pixels of each patch and (2) a patch-wise attention that reasons the global relationship between patches.

As for the local attention, although existing convolution-free networks [19], [31], [52] based on the pyramid architecture are data-efficient and could (conceptually) be transferable to downstream tasks, all these works only propose local attention within their building blocks, which limits the model in capturing long-range global visual patterns. Besides, they are still memory-expensive, or too slow due to the incompatibility with the modern hardware accelerator, or both. Such inefficiency also prevents them to be widely deployed for other downstream vision tasks that may need to process images of way larger resolutions.

The aim of this work is to construct a practical transformer-based network with strong performance and transferability to down-stream tasks so that it can be used as an alternative to conventional CNN. To this end, the

- *Shuyang Sun, Hengshuang Zhao and Philip H.S. Torr are with the Department of Engineering Science, University of Oxford. E-mail: kevinsun@robots.ox.ac.uk*
- *Song Bai is with ByteDance AI Lab.*

Manuscript received 29th July 2021.

network needs to take advantages of both local and global interactions. As shown in Figure 1, similar to ViT (Figure 1 left), we divide the entire image into N patches. However, instead of projecting all pixels of each patch into global tokens, we first query the pixels of each patch locally with their close neighbors using the *pixel-wise* multi-head attention. After the local pixel-wise interaction, the spatial information of each patch will be gathered, so that the *patch-wise* attention can use them to infer the correlation among patches. The whole process can be also regarded as a spatial factorization of the original full spatial attention (*a.k.a.* Non-local [44] block in Figure 1 middle). Therefore we name it as Separable Transformer (SeT). Such factorization largely reduces the memory cost compared to the full spatial attention so that both local and global attention can be practically applied even on large feature maps. In summary, the contribution of this paper is three-fold:

1) By factorizing the full spatial attention into local pixel-wise attention and patch-wise attention, SeT is the first transformer-based architecture retaining both local and global interactions throughout the network.
2) SeT achieves state-of-the-art performance compared with existing transformer-based architectures. Besides, SeT can be well transferred onto other downstream tasks like object detection and instance segmentation.
3) We also reveal that the it is the global interaction that makes the token-based transformers to be unfriendly towards convergence. By enhancing the locality of the transformer, SeT can be more data-efficient compared to the token-based approaches *e.g.* ViT [9] and DeiT [41].

2 Related Work

Convolutional neural networks. Conventional CNNs have dominated the field of computer vision. After the great success of AlexNet [25] for image classification, lots of following powerful convolutional architectures are developed for image recognition [14], [21], [30], [40]. They serve as the basic backbones for various downstream computer vision tasks, such as semantic segmentation [1], [53] and object detection [11], [12], [13], [33].

Transformers and self-attention mechanisms for visual recognition. With the success of Memory Networks [38] and Transformers [42] for natural language modeling, lots of works in the field of computer vision attempted to migrate similar self-attention mechanism as an independent block into CNNs for image classification [2], [4], [20], [45], object detection [3], [18], [37] and video prediction [23], [44] *etc*.

Recent works tried to replace all convolutional layers in neural networks with local attention layers to build up self-attention based networks [5], [15], [18], [31], [50], [52]. Though these works are proven to be successful in improving performance while reducing the network complexity in terms of FLOPS and the number of parameters. The memory cost and runtime latency of these models are still very higher than the conventional CNNs, which prevents them to be widely deployed for practical use. To mitigate this problem, the Vision Transformer (ViT) [9] chose to largely reduce the image resolution and only retain global information while processing. The success of such radical change on image modeling is surprising but the downside is also obvious. Compared with other existing models, ViT requires much more training data for generalization and the results are still worse than the state-of-the-art CNN counterparts *e.g.* [30], [40] when both models are well-tuned on ImageNet. Some concurrent works like Swin Transformer [28], PVT [43] and MViT [10] use either local attention or global attention for feature extraction. Another concurrent work ViL [50] propose both global and local interactions within its structure, which is the most similar work to ours. However, its local attention is a convolution-like self-attention with dense sliding window like [19], [31], [52], which will lead to very high computational latency as we discussed above. Unlike the listed works above, as a transformer-based network, our model can do both local and global attention simultaneously as we cut down the computational cost via spatial grouping.

Task-specific transformers. In addition to the transformer based architectures designed for image recognition, there are also task-specific transformer-based architectures as the transferability of the current transformer is still limited. Specifically, DETR [3] and its variant Deformable DETR [55], UP-DETR [8], Sparse RCNN [39] are designed specifically for object detection and instance segmentation. What proposed in this paper is independent to these works as we intend to build up a versatile transformer-based network that can be directly transferred onto other down-stream tasks as the engine for extracting features.

Kernel factorization methods. Recent works design different efficient CNN architectures by factorizing the convolutions on the channel [17], [35], [46], [51], or spatiotemporal dimension [47]. Our patch-based separable attention also takes advantage of the grouping concept, but it is applied to spatial dimension instead. Going beyond the spatial grouping, we also introduce to enlarge the reference scope of the key and value so that pixels within the pixel-wise attention can have a larger field of view and learn to model long-range interactions.

3 Patch-base Separable Transformer

The patch-based Separable Transformer (SeT) is primarily composed of two sub-modules, the pixel-wise attention for extracting local features, and the patch-wise attention for reasoning the global interactions.

3.1 Overview

SeT can be regarded as a variant of the traditional Transformer Encoder. The basic building block of SeT is exhibited in Figure 2. Given an input image $x \in \mathbb{R}^{C\times H\times W}$, the whole image is first divided into N ***query patches*** with no overlapping in between. The patch size of each patch is $K \times K$, so that $N = \frac{HW}{K^2}$. Note that if HW cannot be exactly divided by K^2, then we will apply zero padding to the input and mask it out like [3], [31] in the lateral self-attention modules. Apart from these non-overlapping patches, we further crop N ***reference patches*** with a larger patch size $(K + 2O) \times (K + 2O)$ and step size K, so that each pixel within the query patch, especially those at the patch boundary, can refer to a wider neighborhood like convolution. The prepared query patches and reference patches will be parallelly sent into SeT for processing.

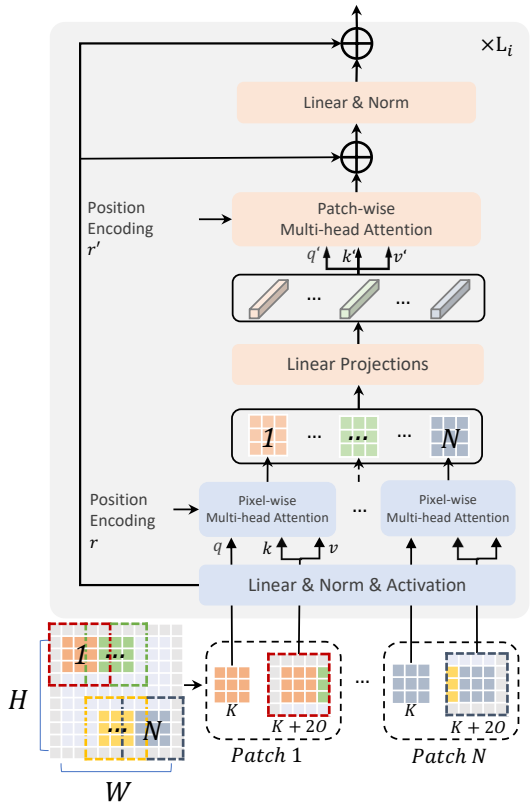

Fig. 2. Overview of SeT. L_i is the number of blocks of stage i. Here we show an example when $K = 3, O = 1$.

Before being sent to the self-attention modules, all pixels on the feature map will be first transformed by a linear operation. The linear transformation will not change the spatial size of each patch, instead, it maps the C channels of each pixel into $D = C/B$, which can be viewed as a bottleneck design with $B\times$ channel reduction like [14]. The transformed features will be then fed into the pixel-wise separable attention. Following the *query-key-value* formulation of the original multi-head self-attention, here in the pixel-wise attention, features of the query patch serve as the query and those of the reference patch serve as the key and value. We adopt the grouping concept in [42], [46] which applies independent groups of weights to generate attention maps. The output of the pixel-wise attention are still patches with local features corresponding to a specific part of input. Therefore a linear projection is needed to squeeze the spatial dimension of each patch into 1 so that each patch is gathered as a visual token. These visual tokens will serve as the ingredients of the patch-wise separable attention for global reasoning. In this way, both the local and global information are modelled by SeT.

Following [42], skip connections are applied from the input to the output of the separable attention module. We simplify the original feed-forward layer into a single linear transformation followed with normalization after the identity mapping. The transformed feature will be added to the input of the block to generate the final output of the SeT building block.

3.2 Pixel-wise Separable Attention

Pixel-wise separable attention make pixels of each patch interact with their close neighbors. All pixels $x^p \in \mathbb{R}^{K^2\times D}$ of a specific patch p share a same neighborhood $x_n^p \in \mathbb{R}^{(K+2O)^2\times D}$. Given three learnable linear transformation with weights W_q, W_k, W_v, we obtain three variables q, k, v representing the query, key and value of the self-attention. Note that $q \in \mathbb{R}^{K^2\times D}$ is generated using contents in the query patch x^p while the $k, v \in \mathbb{R}^{(K+2O)^2\times D}$ are linear transformations of the content of the reference patch x_n^p. Following [2], [31], [37], a learnable relative positional embedding $r \in \mathbb{R}^{(K+2O)^2\times D}$ is applied to capture the location information, in this way, the attention affinity matrix M can be calculated as:

$$\begin{aligned} q &= x^p W_q, \\ k &= x_n^p W_k, \\ v &= x_n^p W_v, \\ M &= q \cdot k^T + q \cdot r^T, \end{aligned} \tag{1}$$

where $M \in \mathbb{R}^{K^2\times(K+2O)^2}$ represents the correspondence between every pixel in the query patch to pixels in the reference patch. Following the common practice, we further apply a softmax normalization on the matrix by:

$$M_{i,j} = \frac{1}{\sqrt{D}} \frac{e^{M_{i,j}}}{\sum_l e^{M_{i,l}}}. \tag{2}$$

A normalization factor $\frac{1}{\sqrt{D}}$ is set here to prevent the output distribution of softmax to be one-hot. The matrix M represents the affinity matrix between pixels of the query patch and the reference patch. We can aggregate the values in v using M by:

$$y^p = M \cdot v, \tag{3}$$

where $y^p \in \mathbb{R}^{K^2\times D}$ represents the output of each head of the pixel-wise separable attention. Following [42], the above process can run in multiple heads in parallel. The output of each head will be stacked together as the final output.

3.3 Patch-wise Separable Attention

Patch-wise separable attention intends to update all pixels within each patch with the inferred correlation with other patches. Given $y = \{y^1, ..., y^N\} \in \mathbb{R}^{N\times K^2\times D}$, we first normalize y with a BatchNorm [22] layer, then apply a linear transformation to all K^2 pixels of the patch that maps K^2 dimension into 1, which is identical to a weighted average pooling operation. Denote the down-sampled output as $y_d \in \mathbb{R}^{N\times D}$, and reshape y to $NK^2 \times D$. Similar to the formulation of the pixel-wise separable attention, we denote the linear transformations and their weights for the patch-wise separable attention as $q' \in \mathbb{R}^{NK^2\times D}$ and $k', v' \in \mathbb{R}^{N\times D}$ and $W_{q'}, W_{k'}, W_{v'}$, then the attention matrix can be formulated as:

$$\begin{aligned} q' &= yW_{q'} \\ k' &= y_d W_{k'} \\ v' &= y_d W_{v'} \\ M' &= q' \cdot (k')^T + q' \cdot (r')^T, \end{aligned} \tag{4}$$

TABLE 1
General specification for SeT family. Building blocks of SeT are shown in brackets.

Stage	Output Resolution	Block Settings
	$\frac{H}{4} \times \frac{W}{4}$	7×7, 64, Conv, stride 2 3×3 Max Pool, stride 2
Stage$_1$	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} C = 64 \times B \\ D = 64 \\ G = 4 \end{bmatrix} \times L_1$
Stage$_2$	$\frac{H}{8} \times \frac{W}{8}$	2×2 Avg Pool, stride 2 $\begin{bmatrix} C = 128 \times B \\ D = 128 \\ G = 8 \end{bmatrix} \times L_2$
Stage$_3$	$\frac{H}{16} \times \frac{W}{16}$	2×2 Avg Pool, stride 2 $\begin{bmatrix} C = 256 \times B \\ D = 256 \\ G = 8 \end{bmatrix} \times L_3$
Stage$_4$	$\frac{H}{32} \times \frac{W}{32}$	2×2 Avg Pool, stride 2 $\begin{bmatrix} C = 512 \times B \\ D = 512 \\ G = 8 \end{bmatrix} \times L_4$
	1×1	1000, Linear

where $r' \in \mathbb{R}^{N \times D}$ represents the absolute positional embedding for all patches, and $M' \in \mathbb{R}^{NK^2 \times N}$ denotes the affinity matrix between patches. Then the output of the patch-wise separable attention can be calculated as:

$$\begin{aligned} M'_{i,j} &= \frac{1}{\sqrt{D}} \frac{e^{M'_{i,j}}}{\sum_l e^{M'_{i,l}}} \\ z &= M' \cdot v', \end{aligned} \tag{5}$$

where $z \in \mathbb{R}^{NK^2 \times D}$ is the final output of the patch-wise attention. Following [42], we apply skip connections from the input x^p to the output of the attention. Note that the pixel-wise and patch-wise separable attention can be binded together or applied independently. When both attention are applied together, the final output of the entire attention module is z. If the global separable attention is not applied, the final output will turn to be y. In Section 4, we will compare both effect of applying the two separable attention modules.

3.4 Computational Cost of Separable Attention

The factorization above reduces the computational cost compared with the conventional full spatial attention [37], [44]. Given the same input to the standard full spatial attention, which takes the *all2all* interaction between every two pixels on a feature map, its computational cost $\mathcal{C}_{full}$ can be calculated as:

$$\mathcal{C}_{full} = \mathcal{O}(CHWHW), \tag{6}$$

where the square of $H \times W$ will result in huge memory and computing cost when the spatial size is high.

As for the computational cost of SeT, the computational complexity $\mathcal{C}_{SeT}$ is:

$$\begin{aligned} \mathcal{C}_{SeT} &= \mathcal{O}(CK^2(K+2O)^2 + CK^2N^2) \\ &= \mathcal{O}(CK^2(K+2O)^2 + C\frac{H^2W^2}{K^2}) \end{aligned} \tag{7}$$

TABLE 2
Hyper-parameters for models of SeT family.

Model	Number of Blocks $[L_1, L_2, L_3, L_4]$	B	K	O
SeT-8	$[2, 2, 2, 2]$	2	8	2
SeT-16	$[3, 4, 6, 3]$	4	8	2
SeT-33	$[3, 4, 23, 3]$	4	8	2

To make it clear, we calculate $\frac{\mathcal{C}_{SeT}}{\mathcal{C}_{full}}$ for better comparison:

$$\begin{aligned} \frac{\mathcal{C}_{SeT}}{\mathcal{C}_{full}} &= \frac{CK^2(K+2O)^2 + C\frac{H^2W^2}{K^2}}{CH^2W^2} \\ \frac{\mathcal{C}_{SeT}}{\mathcal{C}_{full}} &= \frac{1}{K^2} + \frac{K^2(K+2O)^2}{H^2W^2} \\ \frac{\mathcal{C}_{SeT}}{\mathcal{C}_{full}} &\approx \frac{K^2(K+2O)^2}{H^2W^2}. \end{aligned} \tag{8}$$

For simplicity, we ignore the fractional $\frac{1}{K^2}$ as we set $K > 3$ in practice. Since normally a patch is just a small portion of the entire feature map, *s.t.* $(K+2O)^2 < HW$, $\mathcal{C}_{SeT} < \mathcal{C}_{full}$. SeT will save more computational cost when the feature map resolution is high. The huge reduction in computation and memory cost makes it possible to transfer attention onto larger feature maps.

3.5 Network Structure

To construct an entire network using the above basic block, we first divide the whole network into four stages according to the resolution of their feature maps. Note that the network is constructed with the following hyper-parameters:

- B is the bottleneck factor
- G represents the number of heads (groups) of the multi-head attention.
- C represents the number of channels for the input and output of each block.
- L_1, L_2, L_3, L_4 are the numbers of blocks for stage $1, 2, 3, 4$ respectively.
- $K, K + 2O$ are the patch sizes of query patch and reference patch as illustrated above.

We build up three models named SeT-8, set-16 and SeT-33 according to the number of the building blocks. Based on the experimental results in Section 4, for the first two stages of SeT, we only use pixel-wise attention in the building blocks, while for the last two stages, both pixel-wise local attention and patch-wise global attention are applied.

Apart from the attention model, we also construct a hybrid model by replacing all separable attention components of the building block with a 3×3 convolutions at the first two stages of the network.

4 Experiments for Image Classification

4.1 Implementation Details on ImageNet

We first conduct experiments for image classification on the ImageNet 2012 classification dataset [34] that includes 1000 classes. There are 1.2 million images for training and 50 thousands images for validation. Unlike ViT [9], which is pre-trained on other datasets *e.g.* ImageNet-21k, we only train our network on the training set of ImageNet. We train

TABLE 3
Experimental results on ImageNet. [1] represents our enhanced re-implementation using training tricks *e.g.* Mixup and AutoAug shown in the second last row of Table 5.

Model	Params (M)	FLOPS (G)	throughput (img/s)	Top-1 Acc (%)
CNN Architectures				
ResNet-50 [14]	25.5	4.1	1226	76.2
ResNet-101 [14]	44.4	7.8	753	77.4
ResNet-152 [14]	60.0	11.6	526	78.3
ResNet-50++[1]	25.5	4.1	1226	78.6
ResNet-101++[1]	44.4	7.8	753	80.4
RegNetY-4G++ [41]	20.6	4.0	1156.7	80.0
RegNetY-8G++ [41]	39.2	8.0	591.6	81.7
Transformer Architectures				
ViT-B↑384 [9]	86.4	55.4	85.9	77.9
DeiT-Ti [41]	5.7	1.6	2536.5	72.2
DeiT-S [41]	22.1	4.6	940.4	79.8
DeiT-B [41]	86.6	17.6	292.3	81.8
LRNet-18 [19]	14.4	2.5	-	74.6
LRNet-50 [19]	23.3	4.3	-	77.3
LRNet-101 [19]	42.0	8.0	-	78.5
Swin-Ti [28]	29	4.5	755	81.3
Swin-S [28]	50	8.7	437	83.0
MViT-B-maxpool [10]	37	7.8	-	82.5
MViT-B [10]	37	7.8	-	83.0
PVT-T [43]	13.2	1.9	-	75.1
PVT-S [43]	24.5	3.8	-	79.8
PVT-M [43]	44.2	6.7	-	81.2
SAResNet-26 [31]	13.7	2.7	-	74.8
SAResNet-50 [31]	18.0	3.6	-	77.6
SeT-8	9.7	1.8	1336	78.1
SeT-16	24.7	4.4	683	81.7
SeT-33	40.3	7.7	401	83.3

the network using a batch size of 2048. The learning rate is first warmed up for 5 epochs from 0 to 0.6 and will be decreased to 0 at the end of the training process following a cosine annealing schedule [29]. Previous transformer-based backbones like [9], [41] heavily rely on the data augmentation and regularization, as the learnt weights are data-dependant. However, we find our network can be well-tuned with much fewer bells and whistles. Details about this will be discussed in the following sections.

4.2 Experimental results on ImageNet

We show the experimental results on ImageNet in Table 3, which includes results for both CNN architectures *e.g.* ResNet [14], SEResNet [20] and RegNet [30] and transformer-based architectures *e.g.* ViT [9], DeiT [41], Local Relation Networks (LRNet) [19] and Stand-Alone Networks (SAResNet) [31]. Note that we categorize the attention models like [19], [31] as transformer-based architectures because the whole backbone of attention model is identical to a transformer encoder.

SeT *vs.* Token-based Transformers. We first compare SeT to the token-based Vision Transformer (ViT), which radically reduces the image resolution at the beginning of the network. When both networks are trained from scratch only training on the training set of ImageNet, SeT-33 can outperform ViT-B by 4.9% with only about half of its number of parameters and FLOPS. The remarkable improvement on ImageNet demonstrates the importance of retaining the local features for image recognition.

TABLE 4
Experimental results on ImageNet when applying factorization on DeiT.

Model	Params(M)	FLOPS(G)	Top-1 Acc (%)
DeiT-S [41]	22.1	4.6	79.8
SeT-ViT-S	23.9	4.2	80.4

Another token-based vision transformer, DeiT, is also listed in Table 3 for comparison. We note that DeiT is *concurrent* to our work. The basic structure of DeiT is identical to what proposed in ViT. The biggest difference between them is that DeiT can achieve competitive performance on ImageNet without using any additional datasets for pre-training. DeiT manages to do this by successfully finding extensive data augmentations and regularization techniques. When compare SeT with DeiT, we observe that SeT-8 can surpass DeiT-Tiny by a significant 5.9% in terms of top-1 accuracy. As for small models like set-16, it can outperform DeiT-S by 1.7% with much fewer data augmentations and training tricks, which indicates that SeT is much more data-efficient than the DeiT. SeT-33 as a larger model in the SeT family can achieve comparable performance to DeiT with 56.3% fewer FLOPS and 53.1% fewer parameters. As for some other concurrent Transformers like PVT [43] and Swin Transformer [28], SeT can also show its superior performance. For example, SeT-8 can outperform PVT-T by a significant 3%, and SeT-16 can outperform PVT-S and Swin-T by 1.9% and 0.4% respectively.

SeT *vs.* Local-aware Transformers. SeT is better than existing local-aware models. Apart from token-based methods, we further compare SeT with other local-aware attention backbones *e.g.* LRNet [19] and SAResNet [31]. LRNet [19] is designed for inferring local relationship between pixels with attentive mechanism. As shown in Table 3, SeT outperforms LRNet when compared under similar model capacity. As for the comparison with SAResNet, SeT-8 can surpass SAResNet26 by 3.5% with about 33.3% fewer FLOPS. The comparison with local-aware attention-based models demonstrates the significance of the global attention in SeT.

SeT *vs.* CNN architectures. In this part, we observe that SeT can be comparable with the state-of-the-art CNN architectures when FLOPS $\leq$ 8G. We compare SeT with CNN baselines including ResNet [14], Squeeze-Excite (SE) ResNet [20] and RegNetY (*w/* SE). In addition to the results of their original paper, we also compare with their better-tuned versions (followed by ++ in Table 3) reported in [37], [41] for fair comparison. Note that the tricks like data augmentation and regularization applied on these baselines can lead to significant improvement. For example, the top-1 accuracy of SEResNet50++ is 2.5% higher than its original version. The proposed SeT achieves $1.4\% - 1.9\%$ superior to SEResNet. As for the comparison with RegNetY, the top-1 accuracy between these two models are very close.

4.3 SeT is data-efficient compared to ViT/DeiT

As discussed above, SeT is better than ViT in term of accuracy. Here we further show that SeT is more data-efficient compared with the token-based transformers *e.g.* ViT and DeiT. Table 5 exhibits the data augmentations and

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3231725

TABLE 5
Training tricks used for improving DeiT and SeT. SeT can outperform DeiT with much fewer data augmentations and regularization techniques.

Model	Optimizer	RandAug	AutoAug	MixUp	CutMix	Color Jitter	RandErase	DropPath	Repeated Aug	Label Smooth	Activation Func.	Epochs	Top-1
DeiT-S	AdamW	✓	✗	✓	✓	✓	✓	✓	✓	✓	GELU	300	79.8
ViT-B	SGD	✗	✓	✓	✗	✗	✗	✓	✗	✓	GELU	300	71.5
SeT-16	SGD	✗	✗	✗	✗	✗	✗	✓	✗	✓	ReLU	350	79.2
		✗	✗	✗	✗	✗	✗	✓	✗	✓	SiLU	350	79.6
		✗	✓	✓	✗	✗	✗	✓	✗	✓	SiLU	350	80.5
	AdamW	✓	✗	✓	✓	✓	✓	✓	✗	✓	SiLU	300	81.7

TABLE 6
✓and ✗in the brackets represent the existence of the module in stage [1, 2, 3, 4]. If both the pixel-wise and patch-wise attentions are not applied, then a 3×3 convolution is placed as alternative, making it to be a hybrid model. Models without postfix are trained for 120 epochs without bells and whistles. ++ indicates that the model is trained under the training regime shown in the last row in Table 5.

Model	Params (M)	FLOPs (G)	Pixel-wise	Patch-wise	Top-1 Acc (%)
SeT-16	25.5	4.1	[✗,✗,✗,✗]	[✗,✗,✗,✗]	76.9
	24.8	4.2	[✗,✗,✓,✓]	[✗,✗,✓,✓]	78.1 (+1.2)
	24.2	4.1	[✓,✓,✓,✓]	[✗,✗,✓,✓]	78.2 (+1.3)
	19.2	3.8	[✓,✓,✓,✓]	[✗,✗,✗,✗]	77.5 (+0.6)
	24.7	4.4	[✓,✓,✓,✓]	[✓,✓,✓,✓]	75.6 *
SeT-16++	24.2	4.1	[✓,✓,✓,✓]	[✗,✗,✓,✓]	81.3
	24.7	4.4	[✓,✓,✓,✓]	[✓,✓,✓,✓]	81.7

regularization techniques we used compared to ViT and DeiT. Unlike ViT that pre-trains itself on other extra-large datasets like ImageNet-21k and JFT. SeT is only trained with the data in the training set of ImageNet. Instead of first training on other datasets, DeiT proposes to train their network only using the data of ImageNet and achieves a competitive result by adopting heavy data augmentations and regularization techniques. As shown in Table 5, when compared with DeiT, SeT-16 can achieve even more competitive results on ImageNet trained with much fewer data augmentations, and techniques for regularization. The last row of Table 5 show that the performance of SeT can be further improved using those bells and whistles like Mixup [49] and Auto-augmentation [6] and SiLU [32]. Concretely, tricks including repeated augmentation [16], random augmentation [7], cutmix [48], random erase [54] and color jittering [25] are used for improving DeiT but are not applied to SeT. We also notice that SeT can be well-tuned by SGD while DeiT or ViT can only be well-optimized using Adam [24]. This demonstrates that SeT is more data-efficient than the token-based transformers, which further implies the importance of retaining the local features in visual modeling.

As shown in Table 4, we also found that when applying the factorization proposed in this paper to DeiT-S, our method can further boost the baseline performance with lower computational cost. For all ablation studies in this paper we propose the training regime in the fourth row of Table 5 if not specified.

TABLE 7
Relative positional encoding is significantly better than the absolute positional encoding.

Positional Encoding	Params	FLOPS	Top-1 Acc (%)
Relative	24.2	4.1	80.5
Absolute	24.2	4.0	78.1

4.4 Effects of local and global separable attention

Table 6 shows the effects of applying local and global attention into SeT. As all images we feed into the network are of size 224×224, the size of the feature map (7×7) at the last stage will be smaller than the query patch size we set (8×8). Therefore, the pixel-wise local attentions here at the last stage are equivalent to the global attention as it can capture all the content of the image. However, in this case, we can still apply patch-wise here so that the weights within the patch-wise attention module can be pre-trained for the down-stream tasks *e.g.* object detection and instance segmentation that need to process images at higher resolutions. We note that all results are trained with 120 epoch using the tricks shown in the second row of Table 5. As shown in 6, when compare the third row with the fourth row, the application of the **patch-wise attention in stage** $3, 4$ **leads to** 0.7% **gain on ImageNet**. However, when applying global attention to the shallow stages like Stage 1 and Stage 2 (last row in Table 6), we observe that the network suffers from an under-fitting problem and cannot be well-tuned using the same training regime. When using the Adam optimizer and trained under the training regime shown in the last row of Table 5, our SeT-16 can be well-tuned and achieve its best performance 81.7%. The comparison between the third and the fifth row also reveals that applying global interactions in the early stages can be the reason why transformer-based method can be hard to be tuned under SGD optimizer. We also observe that SeT can be successfully combined with CNN architectures (the second row in Table 6). If we replace all convolutions within the stage $3, 4$ with the separable attention, the network can have an obvious 1.2% gain compared with the baseline.

4.5 Relative *vs.* Absolute positional encoding

Recent works [2], [31] propose relative positional encoding to incorporate location information into contents. The rel-

TABLE 8
Average Precision (%) of instance segmentation and object detection with Mask RCNN on MS COCO *val-2017* . AP^m_* and AP^b_* denote the average precision for masks and bounding boxes respectively, and AP^*_S, AP^*_M, AP^*_L denote the AP for small, medium and large objects respectively. + denotes better training regime using Adam optimizer and multi-scale training proposed in [43].

			Instance Segmentation			Object Detection		
Backbone	Params (M)	FLOPs (G)	AP^m	$AP^m_S/AP^m_M/AP^m_L$	AP^m_{50}/AP^m_{75}	AP^b	$AP^b_S/AP^b_M/AP^b_L$	AP^b_{50}/AP^b_{75}
ResNet-50	44.2	180	34.7	18.3/37.4/47.2	55.7/37.2	38.2	21.9/40.9/49.5	58.8/41.4
SeT-16	41.3	196	37.2 (+2.5)	21.6/40.3/49.9	60.1/39.0	40.6 (+2.4)	25.6/44.0/52.6	63.4/43.7
PVT-S+	44.1	230	37.8 (+3.1)	20.4/40.3/53.6	60.1/40.3	40.4 (+2.2)	22.9/43.0/55.4	62.9/43.8
Swin-T+	48	267	39.8 (+5.1)	24.4/43.3/54.3	63.3/42.7	43.7 (+5.5)	28.5/47.0/57.3	66.6/47.7
SeT-16+	41.3	196	39.9 (+5.2)	24.4/43.3/54.3	63.3/42.7	44.0 (+5.8)	29.7/47.8/57.2	66.8/48.0

TABLE 9
Experimental results for object detection when embedding into RetinaNet. * SAResNet-50 is trained with a longer training schedule for 150 epochs.

Backbone	Params	FLOPs	AP^b	$AP^b_S/AP^b_M/AP^b_L$	AP^b_{50}/AP^b_{75}
ResNet-50	37.7M	239G	36.5	20.4/40.3/48.1	55.4/39.1
SAResNet-50*	-	-	36.6(+0.1)	18.5/40.6/51.6	54.5/39.2
SeT-16	36.9M	255G	39.2(+2.7)	24.4/42.9/50.6	60.3/41.3

ative positional encoding Table 7 demonstrates the significance of the relative positional encoding r we proposed in the pixel-wise attention. With just a fractional increase in FLOPS, the relative positional encoding performs 2.4% higher than the absolute one.

5 Experiments on MS COCO

Thanks to the pyramid-style design, SeT can be transferred onto other down-stream tasks that require multi-scale features *e.g*. object detection and instance segmentation. We evaluate SeT on the challenging MS COCO dataset, which consists of 115k images for training (*train-2017*) and 5k images (*val-2017*) for validation. We train models on *train-2017* and report the results on *val-2017*. All reported results follow the official definition of Average Precision (AP) metrics by MS COCO, which includes AP_{50} and AP_{75} (averaged over IoU thresholds) and AP_S, AP_M, AP_L (AP at different scales). Annotations of MS COCO include both bounding boxes and polygon masks for object detection and instance segmentation respectively.

We embed SeT into two popular frameworks. The one is RetinaNet, which is an one-stage detector for object detection. The other is Mask-RCNN, which is a two-stage detector that can do object detection and instance segmentation simultaneously. All modules of SeT except for the positional encoding of the patch-wise attention are first pre-trained on ImageNet before conducting experiment on MS COCO. Note that the activation function we used here is ReLU instead of SiLU.

Integrate SeT into RetinaNet. We first integrate SeT into the popular one-stage detector RetinaNet [27] for object detection. We use SeT as the backbone followed by a Feature Pyramid Network (FPN) [26] to refine the multi-scale features of the network. We take 16 images in one batch for 12 epochs ($1\times$) for training. The learning rate is initially warmed up for 500 steps from 0 to 0.02 and then decayed after 8 and 11 epochs by a factor of 0.1.

Table 9 compare set-16 with the other two baselines ResNet-50 and SAResNet-50. Despite the longer training schedule of SAResNet (150 epochs), SeT still outperforms SAResNet by a clear 2.6%. When comparing SeT with ResNet-50, SeT achieves a significant 1.8%. We also notice that SeT is remarkably better in detecting small objects. To be specific, in terms of the AP of small objects (AP^b_S), SeT performs 4.0% better than ResNet-50, and 5.9% better than SAResNet. The improvement compared with these two baselines validate the efficacy of the global attention.

Integrate SeT into Mask-RCNN. We further embed SeT into the two-stage framework Mask-RCNN for object detection and instance segmentation. Similar to what proposed in RetinaNet, we also integrate the features of SeT into FPN for multi-scale interactions. All models reported in Table 8 are trained under a $1\times$ scheme. As shown in Table 8, SeT is 2.5% higher than ResNet-50 for instance segmentation and 2.4% higher for object detection. This means that the multi-scale features extracted from SeT can be generalized onto the down-stream tasks that require precise modeling of the input images. We also train our network under a better training regime for transformers proposed in [43] to further validate the efficacy of our network. As shown in Table 8, under the same training regime (denoted with a + behind), our network SeT-16 can outperform PVT by a significant 2.1% for instance segmentation and 3.6% for object detection. It can also slightly outperform Swin-T as shown in Table 8. This validates the importance of preserving the local features for the input image.

In conclusion, the success on object detection and instance segmentation demonstrates the transferability of SeT to down-stream tasks. We thereby draw the conclusion that SeT can be a promising approach to be used as an alternative to the conventional CNN.

6 Conclusion

In this paper, we present SeT that factorizes the original full spatial self-attention into pixel-wise local attention and patch-wise global attention. Such factorization largely reduces the computational cost while retaining the information of different granularity, which helps generate multi-scale features required by different tasks. Extensive experiments demonstrate that SeT performs better than the state-of-the-art transformer-based approaches on ImageNet, and can be well transferred to other down-stream tasks *e.g*. object detection and instance segmentation.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *TPAMI*, 2017. 2

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. In *ICCV*, 2019. 2, 3, 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2

[4] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, 2019. 2

[5] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020. 2

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019. 6

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 6

[8] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 4, 5

[10] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6824–6835, 2021. 2, 5

[11] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. 2

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 3, 5

[15] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv:1912.12180*, 2019. 2

[16] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: better training with larger batches. In *CVPR*, 2020. 6

[17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017. 2

[18] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *CVPR*, 2018. 2

[19] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *ICCV*, 2019. 1, 2, 5

[20] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018. 2, 5

[21] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017. 2

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3

[23] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *NIPS*, 2016. 2

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014. 6

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1, 2, 6

[26] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 7

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 7

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 2, 5

[29] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 5

[30] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *CVPR*, 2020. 2, 5

[31] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019. 1, 2, 3, 5, 6

[32] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Swish: a self-gated activation function. *arXiv:1710.05941*, 2017. 6

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 4

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 2

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[37] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. *arXiv:2101.11605*, 2021. 2, 3, 4, 5

[38] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NeurIPS*, 2015. 2

[39] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 2

[40] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 2

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv:2012.12877*, 2020. 1, 2, 5

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1, 2, 3, 4

[43] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, pages 568–578, 2021. 2, 5, 7

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 2, 4

[45] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv:2006.03677*, 2020. 2

[46] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 2, 3

[47] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 2

[48] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 6

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 6

[50] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *ICCV*, pages 2998–3008, 2021. 2

[51] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, 2018. 2

[52] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020. 1, 2

[53] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 2

[54] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 6

[55] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2