ScalableViT: Rethinking the Context-oriented Generalization of Vision Transformer

Rui Yang\textsuperscript{1*}, Hailong Ma\textsuperscript{2*}, Jie Wu\textsuperscript{2}, Yansong Tang\textsuperscript{1}, Xuefeng Xiao\textsuperscript{2}, Min Zheng\textsuperscript{2**}, Xiu Li\textsuperscript{1**}

\textsuperscript{1} Tsinghua Shenzhen International Graduate School, Tsinghua University
{r-yang20@mails,tang.yansong@sz,li.xiu@sz}.tsinghua.edu.cn
\textsuperscript{2} ByteDance Inc.
{mahailong.1206,wujie.10,xiaoxuefeng.ailab,zhengmin.666}@bytedance.com

Abstract. The vanilla self-attention mechanism inherently relies on pre-defined and steadfast computational dimensions. Such inflexibility restricts it from possessing context-oriented generalization that can bring more contextual cues and global representations. To mitigate this issue, we propose a Scalable Self-Attention (SSA) mechanism that leverages two scaling factors to release dimensions of query, key, and value matrix while unbinding them with the input. This scalability fetches context-oriented generalization and enhances object sensitivity, which pushes the whole network into a more effective trade-off state between accuracy and cost. Furthermore, we propose an Interactive Window-based Self-Attention (IWSA), which establishes interaction between non-overlapping regions by re-merging independent value tokens and aggregating spatial information from adjacent windows. By stacking the SSA and IWSA alternately, the \textbf{Scalable Vision Transformer} (ScalableViT) achieves state-of-the-art performance on general-purpose vision tasks. For example, ScalableViT-S outperforms Twins-SVT-S by 1.4\% and Swin-T by 1.8\% on ImageNet-1K classification.

Keywords: Vision Transformer, Self-Attention Mechanism, Classification, Detection, Semantic Segmentation,

1 Introduction

Convolutional Neural Networks (CNNs) dominated the computer vision field last few years, which attributes to their capacity in modeling realistic images from local to global perception. Although they have been widely applied in various vision tasks, there are still deficiencies in global visual perception. This global view is essential for downstream tasks, such as object detection and semantic segmentation. Recently, ViT [10] and its follow-ups [35,26,5,40] employed transformer encoders to address the image task and achieved comparable performance against their CNN counterparts because of the global receptive field. However,
the global perception of the transformer entails an unaffordable computation since self-attention is quadratically computed on the whole sequence. To alleviate this overhead, typical Swin [26] employed Window-based Self-Attention (WSA), which partitioned the feature map into many non-overlapped sub-regions and enabled it to process large-scale images with linear complexity. They also proposed a novel Shifted Window-based Self-Attention to compensate for losses of potential long-range dependency. Twins [5] combined the WSA with Global Sub-sampled Attention (GSA) for better performance.

To gain an insight into the window-based self-attention, we visualize the feature maps after the second transformer block. As shown in Fig.1, the feature captured by WSA [5,26] is dispersed, and the activation inclines to partial rather than object-oriented. It may attribute to the fact that the dimension is fixed to \( N \) invariably, which results in limited learning ability and the final performance being highly determined by the difficulty of input data. To alleviate this problem, we develop a novel self-attention mechanism, termed Scalable Self-Attention (SSA), which simultaneously introduces two scaling factors \( (r_n \text{ and } r_c) \) to spatial and channel dimensions. Namely, SSA selectively applies these factors to query, key, and value matrix \( (Q, K, \text{ and } V) \), ensuring the dimension is more elastic and no longer deeply bound to the input. On the one hand, SSA aggregates redundant tokens (similar background) to a more compact one via spatial scalability, reducing unnecessary intermediate multiplication operations. In this way,
the computational complexity is also can be further reduced significantly. In the second row of Fig.1, we can easily observe that the spatial scalability can realize nearly contiguous visual modeling for objects. However, it still losses some contextual cues. Hence, on the other hand, we expand the channel dimension to learn a more graphic representation. As depicted in Fig.1, SSA successes to obtain complete object activations while maintaining context-oriented generalization via channel scalability. For instance, the contextual cues of the cat in the last column are represented in detail. Furthermore, such scaling factors restore the output dimension craftily to $\mathbb{R}^{N \times C}$, which ensures the dimension aligns with the subsequent FFN layers and makes the residual connection feasible.

Moreover, we propose an Interactive Window-based Self-Attention (IWSA) that consists of a regular window-based self-attention and a local interactive module (LIM). IWSA establishes information connections by re-merging independent value tokens and aggregating spatial information from adjacent windows. Thus, it no longer limits the self-attention to local windows, particularly non-overlapping windows. Such characteristic enhances the desired global receptive field and take good advantage of the most significant superiority of the transformer in a single layer. The effectiveness of LIM for window-based self-attention is validated through an ablation study (see Tab.5b). To achieve a more efficient backbone for general-purpose vision tasks, we employ a hierarchical design [14,31] and propose a new vision transformer architecture, termed ScalableViT, who alternately arranges IWSA and SSA blocks in each stage. Because of the context-oriented generalization and the global field achieved in the single block, it is more suitable for visual tasks.

We employ our ScalableViT on several vision tasks, including image-level classification on ImageNet [8], pixel-level object detection and instance segmentation on COCO [25], and semantic segmentation on ADE20K [54]. Extensive experimental results demonstrate that ScalableViT outperforms other state-of-the-art vision transformers with similar or less computational cost. For example, ScalableViT-S achieves $+1.4\%$ improvements against Twins-SVT-S and $+1.8\%$ improvements against Swin-T on ImageNet-1K classification.

2 Related Work

The transformer architecture [37] has become a common template for all natural language processing (NLP) tasks due to its solid global modeling capabilities and convenient parallelization ability. Inspired by this, many researchers tried to equip CNNs with the self-attention to modulate and augment the output of convolutions [2,46]. DETR [3] employed the self-attention mechanism to model the relation between objects for end-to-end detection. Others [20,1,41] combined self-attention with convolutions for full-image contextual information. At the same time, the stand-alone self-attention network [52,28,38] proved that stacking attention layers alone can work for different vision tasks well. Recently, the emergence of ViT [10], DeiT [35], and a series of follow-ups [26,9,48,39,42,47,5] proved the bright prospect of the vision transformer.
2.1 Vision Transformer

ViT [10] applied standard transformer encoders to build a convolution-free image classifier by decomposing the image into a sequence of non-overlapping patches directly. Although it harvested promising results, a gap still existed between data-hungry transformers and top-performing CNNs [34] when only training on the midsize ImageNet-1K [8] from scratch. In order to bridge this gap, DeiT [35] proposed a token-based distillation procedure and a data-efficient training strategy to optimize the Transformer effectively. Later, the follow-ups improved different aspects of the ViT, making them more suitable for vision tasks. T2T-ViT [48] optimized the tokenization by concatenating the neighboring tokens into one token. DynamicViT [30] pruned the tokens of less importance in a dynamic way for a better lightweight module. Cvt [42], CeiT [47] incorporated the convolution designs into the self-attention or the FFN to enhance the locality. CPVT [6] utilized the implicit position representation ability from convolutions (with zero padding) to encode the conditional position information for inputs with the arbitrary size. Then, hierarchical pyramid structures [39,26,9,5] were performed by progressively shrinking the number of tokens and replacing the class token with the average pooling. Thus, the Transformer, supported by multi-level features [23], can handle object detection and image segmentation tasks conveniently. In this paper, we develop a vision transformer, ScalableViT, which achieves a better accuracy and cost trade-off on the visual task.

2.2 Local Self-Attention

The computational complexity of the self-attention mechanism is a barrier that confines it in only downsampled feature maps or small images. So, several previous studies [28,16,38,20,18] had proposed decomposing the global self-attention into much paralleled local self-attention to handle the expensive computation burden. However, this local self-attention limits the model receptive field, which is critical to dense predict tasks. [20,38] proposed generating the sparse attention map on the criss-cross path, realizing global interaction. [18] also captured the information from all the other positions via interlacing elements from different local windows. HaloNet [36] used the overlapped local windows to add the interactions between independent windows. After the pure transformer showed the excellent competitive, several follow-ups [26,5,19,49] applied the self-attention within each non-overlapped local window for linear computational complexity. To compensate for lost information, Swin [26] introduced a novel shifted window strategy, and Twins [5] baked sparse global attention [39] after window-based attention. We design the IWSA, which can aggregate information from a collection of discrete \textit{value} tokens and enable local self-attention to model long-range dependency in a single block.

3 Method

In this section, we elaborately introduce the architecture of ScalableViT and mainly focus on SSA and IWSA mechanisms. SSA simultaneously introduces
different scale factors in the spatial and channel dimensions to maintain context-oriented generalization and reduce computational overhead. IWSA enhances the receptive field of local self-attention by aggregating information from a set of discrete value tokens. Both have linear computational complexity and can learn long-range dependency in a single layer.

3.1 Overall Architecture

The architecture of ScalableViT is illustrated in Fig.2. For an input image with size $H \times W \times 3$, a convolutional patch embedding layer ($7 \times 7$, stride 4) is used to obtain the initial tokens and expand the channel dimension to $C$. Then, a collection of tokens ($H_4 \times W_4$) with the channel dimension $C$ will pass through 4 stages which contain a series of transformer blocks. Between two adjacent stages, another convolutional patch embedding layer ($3 \times 3$, stride 2) is utilized to merge tokens and double the channel dimension. For the $i^{th}$ stage, there are $H_{2i+1} \times W_{2i+1}$ input tokens with $2^{i-1}C$ channels and $L_i$ transformer blocks. As a result, the quantity of tokens will eventually be reduced to $H_{32} \times W_{32}$. This architecture enables us to obtain a hierarchical representation similar to the typical backbones based on CNNs [14,31], allowing ScalableViT to migrate to various vision tasks naturally, such as object detection and segmentation. We design an alternate arrangement of IW-MSA and S-MSA blocks to organize the topology in each stage. In the front of each stage, a position encoding generator (PEG) [6] is inserted between two transformer blocks for position embedding dynamically. The contribution of our ScalableViT lies in two aspects:

1. For global self-attention, we propose SSA to supply context-oriented generalization in the vanilla self-attention block, which significantly reduces computational overhead without sacrificing contextual expressiveness.
2. For local self-attention, we design a LIM to enhance the visual ability of window-based self-attention. Both of them can model long-range dependency in a single layer instead of stacking more self-attention layer.

3.2 Scalable Self-Attention

Self-attention is a critical mechanism in the transformer, and the vanilla self-attention can be calculated as:

\[ Z = A(X)V(X) = \text{Softmax}(\frac{Q(X)K(X)^T}{\sqrt{d_k}})V(X) \]  

(1)

where \( A(X) \) is the attention matrix of the input \( X \), \( Q(X), K(X), V(X) \in \mathbb{R}^{N \times C} \) are the \textit{query}, \textit{key}, and \textit{value} matrices, \( d_k \) is the \textit{query/key} channel dimension, \( N \) is the number of tokens, and \( C \) is the channel dimension of these tokens. The original self-attention mechanism obtains a global receptive field by establishing associations between all input tokens, which is a vital advantage of the Transformer over CNNs. However, it has quadratic computational overhead with \( N \), leading to inefficiency in the intermediate multiplication operations.

Generally, there is much homologous information in natural images, but vanilla self-attention still calculates their similarity. However, not all information is necessary to calculate self-attention in the vision transformer. For example, similar background tokens should be aggregated as a representative token to attend to other foreground tokens. Namely, the dimension of \( Q(X), K(X) \), and \( V(X) \) should not be bounded to the input. More importantly, the fixed dimension results in limited learning ability. Thus, we develop the Scalable Self-Attention (SSA), where two scaling factors (\( r_n \) and \( r_c \)) are introduced to spatial and channel dimensions, respectively, so that a more efficient intermediate calculation than the vanilla one. As illustrated in Fig. 3, the spatial dimension \( N \)
and channel dimension $C$ are selectively scaled to $N \times r_n$ and $C \times r_c$, respectively, by three transformation functions $f_q(\cdot)$, $f_k(\cdot)$, and $f_v(\cdot)$. These scaling factors can also restore the output dimension to align with input afresh, making the subsequent FFN layers and residual connections feasible. As a result, the intermediate dimension is more elastic and no longer deeply bound to the input $X$. The model can reap context-oriented generalization while dwindling computational overhead significantly. SSA can be naturally written as:

$$Z' = A'(X)V'(X) = \text{Softmax}\left(\frac{Q'(X)K'(X)^T}{\sqrt{d_k}}\right)V'(X)$$

$$Q'(X) = f_q(X), \quad K'(X) = f_k(X), \quad V'(X) = f_v(V)$$

where $Q'(X) \in \mathbb{R}^{N \times Cr_c}$, $K'(X) \in \mathbb{R}^{Nr_n \times Cr_c}$, $V'(X) \in \mathbb{R}^{Nr_n \times C}$ are the scaled query, key and value matrices of the input $X \in \Omega^{H \times W \times C}$, $A'(X) \in \mathbb{R}^{N \times Nr_n}$ is the scaled attention matrix, and $Z'$ is the weighted sum of $V'(X)$. The transformation $f_q(\cdot)$ scales the channel dimension of query from $C$ to $C r_c$. $f_k(\cdot)$ is the scaling function for key, which scales the spatial dimension from $N$ to $Nr_n$ while scaling the channel dimension from $C$ to $C r_c$. $f_v(\cdot)$ is the scaling function for value, which scales the spatial dimension from $N$ to $Nr_n$. Hence, some unnecessary intermediate multiplication is decreased significantly, and the computation complexity of the proposed SSA is equal to $O(NN_{r_n}C + NN_{r_n}C r_c)$ that is linear with the input size ($N = H \times W$). In practice, three transformations are operated by convolution and fused with the linear projection getting query, key, and value for utility and briefness. The efficient SSA does not change the size of $Z$ and can be expanded to the scalable multi-head self-attention (S-MSA) easily.

More importantly, the introduced spatial and channel scalability can bring context-oriented generalization (see Fig.1). If only introduced spatial scalability ($r_c \equiv 1$), there would realize nearly contiguous visual modeling for objects but a lack of critical graphic representation. When further introducing channel scalability, SSA can maintain contextual cues and obtain complete object activations successfully, which essential in visual tasks. The values of these two scaling factors vary with model configurations and different network stages. As the network gradually deepens, the number of tokens shrinks, and the degree of redundancy is also dropped. Thus, $r_n$ is largen with the stage depth. Similarly, the channel dimension does not always mismatch with spatial dimension in the self-attention operation. Thus, we set $r_c \geq 1$ in ScalableViT-S and ScalableViT-B. Because of a too-large channel dimension, we set $r_c \leq 1$ in ScalableViT-L. Details about two scale factors are displayed in Table 1.

### 3.3 Interactive Window-based Self-Attention

Besides efficient self-attention, several earlier research has developed local self-attention to avoid quadratic computational complexity with the number of tokens. For example, WSA divides an image $(H \times W \times C)$ into multiple partial windows which contains $M \times M$ tokens. Then, the self-attention would be calculated in each isolated window to produce a set of discrete outputs $\{Z_n\}_{n=1}^{H/M \times W/M}$,
where $Z_n$ can be calculated as:

$$Z_n = A_n(X_n)V_n(X_n) = Softmax\left(\frac{Q_n(X_n)K_n(X_n)^T}{\sqrt{d_k}}\right)V_n(X_n) \quad (4)$$

in which $X_n \in \Omega_n^{M \times M \times C}$ is the partial window field, $Q_n(X_n), K_n(X_n), V_n(X_n) \in \mathbb{R}^{M^2 \times C}$ are the query, key, and value matrices of the discrete window $X_n$, respectively. $d_k$ is equal to the channel dimension of discrete query/key. Finally, a collection of discrete $\{Z_n\}_{n=1}^{H \times W}$ is merged back to $Z \in \mathbb{R}^{N \times C}$. Thus, the computational complexity of attention would be reduced from $O(2H^2W^2C)$ to $O(2M^2HW'C)$ for this image. WSA can be suitable for various vision tasks that require high-resolution input due to its linear complexity.

However, such computationally efficient WSA only yields an activation map with integrated shapes but isolated features (see Fig.1), which ascribed to the missed global receptive field in a single layer. It is far from the initial aims of self-attention. To alleviate this problem, we propose the Interactive Window-based Self-Attention (IWSA) that incorporates a local interactive module (LIM) into the WSA, as illustrated in Fig.4. After getting a collection of discrete values $\{V_n(X_n)\}_{n=1}^{H \times W}$, the LIM reshapes them into $\mathbb{R}^{M \times M \times C}$ and merges them into a shape-integrated value map $V \in \mathbb{R}^{H \times W \times C}$. Then, a function $F(x)$ is employed to establish marriages and connections between adjacent $V_n(X_n)$s. As a result, the output $Y = F(V)$ is an integrated feature map with the global receptive field. Finally, this feature map is added with $Z$ as the final output $Z'$ that owns global information. Without loss of generality, IWSA is calculated as:

$$Z' = Z + F(V) \quad (5)$$

where $Z \in \mathbb{R}^{N \times C}$ is merged by a set of $\{Z_n\}_{n=1}^{H \times W}$. In order to implement friendly, a deep-wise convolution with zero padding is employed to take the place of function $F(x)$. If the kernel size of this deep-wise convolution is $k \times k$ (set to 3 by default), the computational cost from LIM is negligible in practice. Additionally, [21] demonstrated that convolution with zero padding could implicitly encode position information through experiments. Thus, IWSA allows self-attention to benefit from the translation invariance, which enables itself to perform better on the limited dataset. Furthermore, our IWSA can be easily expanded to interactive window-based multi-head self-attention (IW-MSA) format easily if calculated in different heads.

CoaT [45] also introduced a deep-wise convolution into self-attention. However, they only considered the convolution as a positional encoding method and inserted it deeply into the calculation. If this convolution is expanded into the WSA, it would be limited in the discrete $V_n(X_n)$. Differently, we regard our LIM as a matchmaker. It is applied on the spliced value map $V$ and parallels with self-attention. By making the sufficient ablation study in Section 4.4, we demonstrate that LIM is capable of delivering stable improvements.
Table 1: Detailed configurations of ScalableViT series. $r_c$ and $r_n$ are the scale factors for the channel dimension and the spatial dimension, respectively. #Blocks and #Head refers to the number of blocks ([$L_1, L_2, L_3, L_4$]) and heads in four stages, respectively. #Channel refers to the channel dimension of the first stage.

| Models     | #Channels | #Blocks   | #Heads   | $r_c$              | $r_n$              |
|------------|-----------|-----------|----------|--------------------|--------------------|
| ScalableViT-S | 64        | [2,2,20,2] | [2,4,8,16] | [1.25,1.25,1.25,1.0] | [$\frac{1}{64}$, $\frac{1}{16}$, $\frac{1}{4}$, 1] |
| ScalableViT-B | 96        | [2,2,14,6] | [3,6,12,24] | [2.0,1.25,1.25,1.0] | [$\frac{1}{64}$, $\frac{1}{16}$, $\frac{1}{4}$, 1] |
| ScalableViT-L | 128       | [2,6,12,4] | [4,8,16,32] | [0.25,0.5,1.0,1.0] | [$\frac{1}{64}$, $\frac{1}{16}$, $\frac{1}{4}$, 1] |

### 3.4 Position Encoding

Besides the position information introduced by LIM, we utilize the positional encoding generator (PEG) [6], composed of a convolution layer with a fixed weights, to acquire the implicit positional information. As illustrated in Fig.2, it is plugged between two consecutive transformer blocks, with only one in the front of each stage. After the PEG, input tokens are sent to subsequent blocks where position bias could enable the transformer to realize the input permutation.

### 3.5 Architecture Variants

In order to fairly compare with other models under similar computation complexity, we set three models: ScalableViT-S, ScalableViT-B, and ScalableViT-L. The detailed configurations are provided in Table 1, where $r_c$ and $r_n$ denote the expand or reduce factors for the channel and spatial dimensions, respectively, as described in Section 3.2. Due to the varying representational capability, we set different $r_c$ for three models. Additionally, the number of blocks, channels, and heads vary with the computational cost.

### 4 Experiments

In the following, we compare the proposed model with other state-of-the-art works on ImageNet-1K [8], COCO [25], and ADE20K [54]. Then, we conduct ablation studies on the upgraded parts to verify their effectiveness.

#### 4.1 Image Classification on ImageNet-1K

**Settings.** Image classification experiments are conducted on the ImageNet-1K [8] dataset. For a fair comparison, the settings mainly follow DeiT [35]. During training, we apply data augmentations and regularizations, including random cropping, random horizontal flipping [32], mixup [51], CutMix [50], random erasing [53], label-smoothing [33], and stochastic depth [17]. We employ the AdamW optimizer [27] to train models for 300 epochs from scratch. The learning rate is set to 0.001 initially and varies with the cosine scheduler. The global
Table 2: Comparison with different state-of-the-art backbones on ImageNet-1K classification. Except for EfficientNet, other models are trained and evaluated on $224 \times 224$ input size. Top-1 refers to top-1 accuracy (%).

| Method            | #Param. | FLOPs | Top-1 |
|-------------------|---------|-------|-------|
| **ConvNet**       |         |       |       |
| RegNetY-4G [29]   | 21M     | 4.0G  | 80.0  |
| RegNetY-8G [29]   | 39M     | 8.0G  | 81.7  |
| RegNetY-16G [29]  | 84M     | 16.0G | 82.9  |
| EfficientNet-B4 [34] | 19M   | 4.2G  | 82.9  |
| EfficientNet-B5 [34] | 30M   | 9.9G  | 83.6  |
| EfficientNet-B6 [34] | 43M   | 19.0G | 84.0  |
| **Transformer**   |         |       |       |
| DeiT-Small/16 [35] | 22M   | 4.6G  | 79.9  |
| T2T-ViT-14 [48]   | 22M     | 5.2G  | 81.3  |
| TNT-S [12]        | 24M     | 5.2G  | 81.3  |
| CoaT-Lite(S) [45] | 20M     | 4.0G  | 81.9  |
| PVT-Small [39]    | 25M     | 3.8G  | 79.8  |
| Swin-T [26]       | 29M     | 4.5G  | 81.3  |
| CvT-13 [42]       | 20M     | 4.5G  | 81.6  |
| Twins-SVT-S [5]   | 24M     | 2.9G  | 81.7  |
| CrossFormer-S [40] | 31M   | 4.9G  | 82.5  |
| **ScalableViT-S(ours)** | 32M | 4.2G  | **83.1** |

| Method            | #Param. | FLOPs | Top-1 |
|-------------------|---------|-------|-------|
| **Transformer**   |         |       |       |
| T2T-ViT-19 [48]   | 39M     | 8.9G  | 81.9  |
| CoaT(S) [45]      | 22M     | 12.6G | 82.1  |
| CoaT-Lite(M) [45] | 45M     | 9.8G  | 83.6  |
| PVT-Medium [39]   | 44M     | 6.7G  | 81.2  |
| Swin-S [26]       | 50M     | 8.7G  | 83.0  |
| CvT-21 [42]       | 32M     | 7.1G  | 82.5  |
| Twins-SVT-B [5]   | 56M     | 8.6G  | 83.2  |
| CrossFormer-B [40] | 52M   | 9.2G  | 83.4  |
| **ScalableViT-B(ours)** | 81M | 8.6G  | **84.1** |

The batchsize is set to 1024 on 8 V100 GPUs. During testing on the validation set, the shorter side of an input image is first resized to 256, and a center crop of $224 \times 224$ is used to evaluate the classification accuracy. More details are supplied in the Appendix B.

**Result.** Classification results on ImageNet-1K are reported in Table 2, where all models are divided into small (around 4G), base (around 9G), and large (around 15G) levels according to computation complexity (FLOPs). ScalableViT-S outperforms comparable models (1.4% better than Twins-SVT-S, and 1.8% better than Swin-T). Moreover, it can even approach or exceed other base models. For the base level, ScalableViT-B surpasses Twins-SVT-B by 0.9% and SWin-S by 1.1% with similar FLOPs. ScalableViT-L also achieves a prominent accuracy-cost trade-off. Additionally, our ScalableViT outperforms the EfficientNet by 0.2%, 0.5%, and 0.4% under three magnitude receptively.

### 4.2 Object Detection on COCO

**Settings.** Object detection experiments are conducted on the COCO 2017 [25] dataset. We verify the model effectiveness on RetinaNet [24] and Mask R-CNN [13] detection frameworks using the MMDetection [4]. Before training, we initialize the backbone with the weight pre-trained on ImageNet-1K dataset, FPN with Xavier [11] scheme, and other new layers with Normal scheme ($std = 0.01$). All models utilize the same settings as [5]: AdamW [27] optimizer, $1 \times$ (12 epochs), and $3 \times$ (36 epochs) schedule with a global batchsize of 16 on 8 GPUs.
Table 3: Results on COCO object detection using the RetinaNet framework. 1× refers to 12 epochs, and 3× refers to 36 epochs. MS means multi-scale training. AP\textsubscript{b} and AP\textsubscript{m} denotes box mAP and mask mAP, respectively. FLOPs are measured at resolution 800 × 1280.

| Backbone                  | #Param. | FLOPs | RetinaNet 1× | RetinaNet 3× + MS |
|--------------------------|---------|-------|--------------|------------------|
|                          | (M)     | (G)   | AP\textsubscript{b} | AP\textsubscript{m} |
|                          |         |       | AP\textsubscript{b} \textsubscript{50} | AP\textsubscript{b} \textsubscript{75} | AP\textsubscript{m} \textsubscript{50} | AP\textsubscript{m} \textsubscript{75} |
| ResNet50[14]             | 38      | 239   | 36.3 55.3 8.4 | 19.3 40.0 48.8 | 39.0 58.4 41.8 | 22.4 42.8 51.6 |
| PVT-Small[39]            | 34      | 226   | 40.4 61.3 43.0 | 25.0 42.9 55.7 | 42.2 62.7 45.0 | 26.2 54.2 57.2 |
| Swin-T[26]               | 39      | 245   | 41.5 62.1 44.2 | 25.1 44.9 55.5 | 43.9 64.8 47.1 | 28.4 47.1 57.8 |
| Twins-SVT-S[5]           | 34      | 210   | 43.0 64.2 46.3 | 28.0 46.4 57.5 | 45.6 67.1 48.6 | 29.8 49.3 60.0 |
| CrossFormer-S[40]        | 41      | 272   | 44.4 65.6 46.3 | 32.8 48.4 59.4 | — — — — — —   | — — — — — —   |
| **ScalableViT-S(ours)**  | 36      | 238   | 45.2 66.5 48.4 | 29.2 49.1 60.3 | 47.8 69.2 51.2 | 31.4 51.5 63.4 |
| ResNet101[14]            | 58      | 315   | 38.5 57.8 41.2 | 21.4 42.6 51.1 | 40.9 60.1 44.0 | 23.7 45.0 53.8 |
| PVT-Medium[39]           | 54      | 283   | 41.7 61.6 43.6 | 26.0 47.2 55.7 | 43.1 61.6 45.0 | 28.7 46.3 58.9 |
| Swin-S[26]               | 60      | 335   | 44.5 65.7 47.5 | 27.4 48.3 58.3 | 46.3 67.8 49.8 | 31.4 50.1 60.9 |
| Twins-SVT-B[5]           | 67      | 326   | 46.5 66.7 48.1 | 28.5 48.9 60.6 | 46.9 68.2 50.1 | 31.6 53.9 57.8 |
| CrossFormer-B[40]        | 62      | 389   | 46.2 67.8 49.5 | 30.1 49.9 61.8 | 47.6 69.2 50.1 | 32.1 52.6 61.8 |
| **ScalableViT-B(ours)**  | 85      | 330   | 45.8 67.6 49.2 | 29.9 49.5 61.0 | 48.0 69.3 51.4 | 32.8 51.6 62.4 |

For the 1× schedule, the short side of training images is resized to 800 pixels, and the long side is never more than 1333 pixels. The learning rate is declined at the 8th and 11th epoch with a decay rate of 0.1. For the 3× schedule, we adopt the multi-scale training, which randomly resizes the short side of the input within the range of [480, 800] while keeping the longer side at most 1333. The learning rate is declined at the 27th and 33rd with a decay rate of 0.1. More details are provided in the Appendix B.

**Result.** We present results of RetinaNet and Mask R-CNN frameworks in Table 3, where AP\textsubscript{b} and AP\textsubscript{m} refer to box mAP and mask mAP, respectively. For object detection with RetinaNet, ScalableViT perform a notable advantage against its CNN and Transformer counterparts. To be more specific, with 1× scheduler, our ScalableViT brings 7.3-8.9 AP\textsubscript{b} against ResNet at comparable settings. Compared with the popular Swin and Twins transformer, our ScalableViT performs 3.5-3.7 AP\textsubscript{b} and 0.5-2.2 AP\textsubscript{b} improvements, respectively. With
3× scheduler, our ScalableViT still achieves competitive performance. For object detection and instance segmentation with Mask R-CNN, our ScalableViT-S outperforms ResNet50 by 7.8 AP$^b$ and 7.3 AP$^m$ with 1× schedule. ScalableViT-S achieves 3.6 AP$^b$ and 2.6 AP$^m$ improvements than Swin-T transformer. With 3× scheduler, ScalableViT-S brings 7.7 AP$^b$ and 6.5 AP$^m$ against ResNet50. Similarly, it also surpasses Swin-T and Twins-SVT-S transformers. Under base level, there is also a similar improvement, demonstrating its stronger context-oriented generalization. Additionally, Fig.5 depicts some qualitative object detection and instance segmentation results from ScalableViT-S-based RetinaNet and Mask R-CNN, which demonstrate that contextual representation from the backbone enables the model to detect objects better.

4.3 Semantic Segmentation on ADE20K

**Settings.** Semantic segmentation experiments are conducted on the challenging ADE20K [54] dataset. We use the typical Semantic FPN [22] and the UperNet [43] as segmentation frameworks to evaluate our models. Following the common practice, we use the MMSegmentation [7] to implement all related experiments, and the settings follow [5,26,39]. For the Semantic FPN, we train 80K iterations with a batch size 16 on 4 GPUs. For the UperNet, we train 160K iterations with a batch size 16 on 8 GPUs. During training, we first resize the short side of input images to 512 pixels, and the long side is never more than 2048 pixels, then randomly crop to 512×512. During testing, we resize input images the same as the training phase but without cropping. We also use the test time augmentation for UperNet, including multi-scale test ([0.5, 0.75, 1.0, 1.25, 1.5, 1.75]× resolution) and flip, for better results. More details are in the Appendix B.

**Result.** Table 4 reports the segmentation results using the Semantic FPN and UperNet frameworks. For the Semantic FPN, our ScalableViT outperforms the Swin transformer by +3.4 mIoU, +3.2 mIoU, and +3.4 mIoU, respectively, under three comparable FLOPs levels. Compared with CrossFormer-S [40], ScalableViT-S performs a modest mIoU but has a fewer computation. When equipped with the UperNet, ScalableViT achieves +4 mIoU, +1.9 mIoU, and +1.6 mIoU improvements under different model sizes. The same competitive results are achieved when test time augmentation is adopted. In addition, ScalableViT-S outperforms CrossFormer-S by +0.9 mIoU. Although ScalableViT achieves a modest performance under the base and large size compared with CrossFormer, there is still significant competitiveness. Moreover, Fig.5(c) shows some qualitative results from ScalableViT-S-based Semantic FPN on the validation part. These results indicate that the ScalableViT backbone can obtain high-quality semantic segmentation results under contextual-oriented generalization.

4.4 Ablation Study

**Analysis for Self-Attention mechanisms.** Our ScalableViT contains two important designs: SSA and IWSA. We ablate their benefits in Table 5a. Firstly,
Table 4: Results on ADE20K segmentation using the Semantic FPN and UperNet framework. FLOPs are measured at resolution $512 \times 2048$. MS refers to the test time augmentation, including flip and multi-scale test.

| Backbone          | Semantic FPN 80k | UperNet 160k |
|-------------------|-----------------|--------------|
|                   | #Param. FLOPs mIoU(%) | #Param. FLOPs mIoU/MS mIoU(%) |
| ResNet50[14]      | 29M 183G 36.7   | — — —/—       |
| PVT-Small[39]     | 28M 161G 39.8   | — — —/—       |
| Swin-T[26]        | 32M 182G 41.5   | 60M 945G 44.5/45.8 |
| Twins-SVT-S[5]    | 28M 144G 43.2   | 54M 901G 46.2/47.1 |
| CrossFormer-S[40] | 34M 221G 46.0   | 62M 980G 47.6/48.4 |
| **ScalableViT-S(ours)** | 30M 174G 44.9 | 57M 931G 48.5/49.4 |
| ResNet101[14]     | 48M 260G 38.8   | 86M 1092G —/44.9 |
| PVT-Medium[39]    | 48M 219G 41.6   | — — —/—       |
| Swin-S[26]        | 53M 274G 45.2   | 81M 1038G 47.6/49.5 |
| Twins-SVT-B[5]    | 60M 261G 45.3   | 89M 1020G 47.7/48.9 |
| CrossFormer-B[40] | 56M 331G 47.7   | 84M 1090G 49.7/50.6 |
| **ScalableViT-B(ours)** | 79M 270G 48.4 | 107M 1029G 49.5/50.4 |
| ResNeXt101-64×4d[44] | 86M — 40.2   | — — —/—       |
| PVT-Large[39]     | 65M 283G 42.1   | — — —/—       |
| Swin-B[26]        | 91M 422G 46.0   | 121M 1188G 48.1/49.7 |
| Twins-SVT-L[5]    | 104M 404G 46.7  | 133M 1164G 48.8/50.2 |
| CrossFormer-L[40] | 95M 497G 48.7   | 126M 1258G 50.4/51.4 |
| **ScalableViT-L(ours)** | 105M 402G 49.4 | 135M 1162G 49.8/50.7 |

All attention modules in ScalableViT-S are replaced with the regular window-based self-attention (WSA), which equals to the $7 \times 7$ deep-wise convolution with dynamic weights. Although WSA achieves 82.4% top-1 accuracy, the dispersed feature (see Fig.1) hinders it from better performance on the downstream visual task. Then, we substituted all attention modules with our IWSA and SSA, respectively. Both of them outperform WSA 0.4% top-1 accuracy. More importantly, they bring +4.6 mIoU and +5.5 mIoU improvements on the ADE20K because of the ability modeling long-range dependency. When only introducing spatial scalability ($r_c \equiv 1$), SSA only achieve 82.6% top-1 accuracy and 43.7 mIoU. Thus, the context-oriented generalization from the cooperation between spatial and channel scalability plays a critical role in visual tasks. Additionally, we examine the topology of our architecture by rearranging the IWSA and SSA. The result demonstrates that prioritizing IWSA followed by SSA produces the best results. We also compare IWSA with Shifted Window-based Self-Attention (SWSA) in ScalableViT, where our IWSA is more appropriate than SWSA.

**Effectiveness of Local Interactive Module.** We examine the effectiveness of the LIM in Table 5b. The ScalableViT-S without position encoding generator (PEG), locally enhanced module (LEM), or LIM is regarded as a baseline model which achieves 82.7% top-1 accuracy on ImageNet-1K. Then, three modules are inserted and yield +0.2%, +0.1%, and +0.3% gain than baseline, respectively.
Table 5: Ablation study for different self-attention mechanisms and local interactive module. Top-1 refers to top-1 accuracy (%) on ImageNet-1K. Semantic segmentation results are yielded from Semantic FPN. The experiment settings are the same as ScalableViT-S.

(a) Analysis for self-attention mechanisms

| Method          | #Param. | FLOPs | Top-1 mIoU(%) |
|-----------------|---------|-------|---------------|
| WSA             | 30M     | 4.3G  | 82.4 38.9     |
| IWSA            | 30M     | 4.3G  | 82.9 43.5     |
| SSA             | 34M     | 4.1G  | 82.9 44.4     |
| SSA ($r_c\equiv 1$) | 32M   | 3.9G  | 82.6 43.7     |
| IWSA & SSA      | 32M     | 4.2G  | 83.1 44.9     |
| SWSA & SSA      | 32M     | 4.2G  | 82.9   —       |
| SSA & IWSA      | 32M     | 4.2G  | 83.0   —       |

(b) Analysis for local interactive module and positional encoding generator.

| PEG LEM LIM | #Param. | FLOPs | Top-1 |
|-------------|---------|-------|-------|
| ✓           | 32M     | 4.2G  | 82.7  |
| ✓           | 32M     | 4.2G  | 82.9  |
| ✓ ✓         | 32M     | 4.2G  | 83.0  |

Fig. 5: Qualitative results based on ScalableViT-S. (a), (b) and (c) are yielded by RetinaNet [24], Mask R-CNN [13], and Semantic FPN [22], respectively.

It fully demonstrates that reasonable convolution can help the model perform better. Due to the window connection, LIM outperforms LEM by +0.2% top-1 accuracy, proving the significance of information interaction from our module. Additionally, we combine PEG with LEM or LIM, whose results are better than only using a single module. Note that the combination of PEG and LIM outperforms the PEG and LEM under the same FLOPs, confirming the usefulness and effectiveness of our LIM again.

5 Conclusion

In this paper, we have presented a vision transformer backbone named ScalableViT, composed of two highly effective self-attention mechanisms (SSA and IWSA). SSA employs two cooperated scaling factors in spatial and channel di-
dimensions for context-oriented generalization, which maintains more contextual cues and learns graphic representations. IWSA develops a local interactive module to establish information connections between independent windows. Both of them owns the capability to model long-range dependency in a single layer. The proposed ScalableViT alternately stakes these two self-attention modules. It pushes the whole framework into a more effective trade-off state and achieves state-of-the-art performance on various vision tasks.

A Additional Analyses for IW SA

As shown in Fig. 6, IW SA is composed of a window-based self-attention (WSA) and a local interactive module (LIM). WSA splits the global self-attention into many limited windows and yields a collection of discrete value matrices. LIM build connections between these value matrices through a fusion function $F$. In practice, this function is replaced with a $3 \times 3$ deep-wise convolution. Additionally, WSA can be viewed as a $7 \times 7$ deep-wise convolution with an adaptive weight. Thus, $F$ brings information exchange through a kind of interleaving effect (illustrated by yellow squares in Fig. 6). This parallel stagger makes IW SA realize a global receptive field in a single layer.

In Table 6, we compare the LIM and the LEM on the ADE20K [54] using Semantic FPN [22] framework. All settings are recorded in the Section B. ScalableViT-S with the LIM achieves +3.8 mIoU than the LEM under the same overhead because IW SA can model the long-range dependency in single layer. This result also proves that the global receptive field plays a more critical role in the downstream vision task. Moreover, the LIM can be expanded to other window-based self-attention with different window division styles.

Table 6: LIM vs. LEM on ADE20K using Semantic FPN. #Param. refers to total parameters of Semantic FPN based on ScalableViT-S backbone. FLOPs are measured at resolution $512 \times 2048$.

| Model                  | #Param. | FLOPs | Top-1 mIoU(%) |
|------------------------|---------|-------|---------------|
| ScalableViT-S w. LEM   | 30M     | 174G  | 83.0          | 41.1           |
| ScalableViT-S w. LIM   | 30M     | 174G  | **83.1**      | **44.9**       |

B More Implementary Details

Classification. The classification settings mainly follow DeiT [35]. All variants are trained under a resolution of $224 \times 224$. During training from scratch, we employ the AdamW optimizer [27] with a weight decay of 0.05 and a momentum of 0.9 to train models for 300 epochs. The learning rate is set to 0.001 initially.
Fig. 6: Interactive Window-based Self-Attention (IWSA). Besides the proposed LIM, other parts compose the WSA. The LIM extracts a set of discrete value matrices $\{V_1, V_2, V_3, V_4\}$ from WSA and merges them via a fusion function $\mathcal{F}$. The output $Y$ is added to $Z$ for an output $Z'$ with information interaction.

and varies with the cosine scheduler, where a 5-epochs linear warm-up is used to stabilize training. The global batchsize is set to 1024 on 8 V100 GPUs. Moreover, we apply data augmentations and regularizations, including random cropping, random horizontal flipping [32], mixup [51], CutMix [50], random erasing [53], label-smoothing [33], stochastic depth [17], and repeated augmentation [15]. For stochastic depth augmentation, we set the drop rate to 0.2, 0.5, and 0.5 for ScalableViT-S, ScalableViT-B, and ScalableViT-L, respectively. During testing on the validation set, the shorter side of an input image is first resized to 256, and a center crop of $224 \times 224$ is used to evaluate the classification accuracy.

Object Detection. We adopt RetinaNet [24] and Mask R-CNN [13] detection frameworks on COCO [25] that contains 118K training images and 5K validation images. Before training, we initialize the backbone with the weight pre-trained on ImageNet-1K, FPN with Xavier [11] scheme, and other new layers with Normal scheme ($std = 0.01$). All models utilize AdamW [27] optimizer, 500-iteration warm-up, $1 \times$ (12 epochs), and $3 \times$ (36 epochs) schedule with a global batchsize of 16 on 8 GPUs. Settings of initial learning rate and weight decay are shown in Table 7. For $1 \times$ schedule, the short side of training images is resized to 800 pixels, and the long side is never more than 1333 pixels. The learning rate is declined at the 8th and 11th epoch with a decay rate of 0.1. For the $3 \times$ schedule, we adopt the multi-scale training, which randomly resizes the short side of the input images within the range of $[480, 800]$ while keeping the longer side at most 1333. The learning rate is declined at the 27th and 33rd with a decay rate of 0.1. When testing, the image size is set as the same as the $1 \times$ schedule.

Semantic Segmentation. Semantic segmentation experiments are conducted on the challenging ADE20K [54], with 20K images for training and 2K images for validation. We use the typical Semantic FPN [22] and UperNet [43] as segmen-
Table 7: Settings of the initial learning rate and weight decay.

| Model                 | #lr scheduler | learning rate | weight decay |
|-----------------------|---------------|---------------|--------------|
| **Object Detection**  |               |               |              |
| RetinaNet(1×)         | Multi-step    | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| RetinaNet(3×)         | Multi-step    | $1 \times 10^{-4}$ | $5 \times 10^{-2}$ |
| Mask R-CNN(1×)        | Multi-step    | $2 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| Mask R-CNN(3×)        | Multi-step    | $1 \times 10^{-4}$ | $5 \times 10^{-2}$ |
| **Semantic Segmentation** |             |               |              |
| Semantic FPN          | Polynomial    | $1 \times 10^{-4}$ | $1 \times 10^{-4}$ |
| UperNet               | Polynomial    | $6 \times 10^{-5}$ | $1 \times 10^{-2}$ |

Migration frameworks to evaluate our models. Following the common practice, we use the MMSegmentation [7] to implement all related experiments. We employ the AdamW [27] to optimize two models. The initial learning rate and weight decay are shown in Table 7. For the Semantic FPN, we train 80K iterations with a batch size 16 on 4 GPUs. The polynomial policy schedules the learning rate with a power of 0.9. For the UperNet, we train 160K iterations with a batch size 16 on 8 GPUs. The polynomial policy schedules the learning rate with a power of 1.0. During training, we first resize the short side of input images to 512 pixels, and the long side is never more than 2048 pixels, then randomly crop to $512 \times 512$. During testing, we resize input images the same as the training phase but without cropping. We also use the test time augmentation for UperNet, including multi-scale test ([0.5, 0.75, 1.0, 1.25, 1.5, 1.75] \times \text{resolution}) and flip, for better results.
References

1. Bello, I., Zoph, B., Le, Q., Vaswani, A., Shlens, J.: Attention augmented convolutional networks. In: ICCV. pp. 3285–3294. IEEE (2019)
2. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H.: Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In: ICCV. pp. 1971–1980 (2019)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. Lecture Notes in Computer Science, vol. 12346, pp. 213–229 (2020)
4. Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D.: MM Detection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019)
5. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. arXiv preprint arXiv:2104.13840 (2021)
6. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. arXiv preprint arXiv:2102.10882 (2021)
7. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation (2020)
8. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
9. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. arXiv preprint arXiv:2107.00652 (2021)
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
11. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Teh, Y.W., Titterington, D.M. (eds.) Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. vol. 9, pp. 249–256 (2010)
12. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. arXiv preprint arXiv:2103.00112 (2021)
13. He, K., Gkioxari, G., Dollár, P., Girshick, R.B.: Mask R-CNN. IEEE TPAMI 42(2), 386–397 (2020)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778. IEEE Computer Society (2016)
15. Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., Soudry, D.: Augment your batch: Improving generalization through instance repetition. In: CVPR. pp. 8126–8135 (2020)
16. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition. In: ICCV. pp. 3463–3472 (2019)
17. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.Q.: Deep networks with stochastic depth. In: ECCV. vol. 9908, pp. 646–661 (2016)
18. Huang, L., Yuan, Y., Guo, J., Zhang, C., Chen, X., Wang, J.: Interlaced sparse self-attention for semantic segmentation. arXiv preprint arXiv:1907.12273 (2019)
19. Huang, Z., Ben, Y., Luo, G., Cheng, P., Yu, G., Fu, B.: Shuffle transformer: Re-thinking spatial shuffle for vision transformer. arXiv preprint arXiv:2106.03650 (2021)

20. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: ICCV. pp. 603–612 (2019)

21. Islam, M.A., Jia, S., Bruce, N.D.B.: How much position information do convolutional neural networks encode? In: ICLR (2020)

22. Kirillov, A., Girshick, R.B., He, K., Dollár, P.: Panoptic feature pyramid networks. In: CVPR. pp. 6399–6408 (2019)

23. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR. pp. 936–944 (2017)

24. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. IEEE TPAMI 42(2), 318–327 (2020)

25. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: ECCV. Lecture Notes in Computer Science, vol. 8693, pp. 740–755 (2014)

26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)

27. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)

28. Parmar, N., Ramachandran, P., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. In: NeurIPS. pp. 68–80 (2019)

29. Radosavovic, I., Kosaraju, R.P., Girshick, R.B., He, K., Dollár, P.: Designing network design spaces. In: CVPR. pp. 10425–10433 (2020)

30. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. arXiv preprint arXiv:2106.02034 (2021)

31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) ICLR (2015)

32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015)

33. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. pp. 2818–2826 (2016)

34. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning, ICML. vol. 97, pp. 6105–6114 (2019)

35. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: Proceedings of the 38th International Conference on Machine Learning, ICML. vol. 139, pp. 10347–10357 (2021)

36. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B.A., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones. In: CVPR. pp. 12894–12904 (2021)

37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)

38. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A.L., Chen, L.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In: ECCV. Lecture Notes in Computer Science, vol. 12349, pp. 108–126 (2020)
39. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. arXiv preprint arXiv:2102.12122 (2021)
40. Wang, W., Yao, L., Chen, L., Lin, B., Cai, D., He, X., Liu, W.: Crossformer: A versatile vision transformer hinging on cross-scale attention. arXiv preprint arXiv:2108.00154 (2021)
41. Wang, X., Girshick, R.B., Gupta, A., He, K.: Non-local neural networks. In: CVPR. pp. 7794–7803 (2018)
42. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L.: Cvt: Introducing convolutions to vision transformers. arXiv preprint arXiv:2103.15808 (2021)
43. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV. Lecture Notes in Computer Science, vol. 11209, pp. 432–448 (2018)
44. Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR. pp. 5987–5995 (2017)
45. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-scale conv-attentional image transformers. arXiv preprint arXiv:2104.06399 (2021)
46. Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., Hu, H.: Disentangled non-local neural networks. In: ECCV. vol. 12360, pp. 191–207 (2020)
47. Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W.: Incorporating convolution designs into visual transformers. arXiv preprint arXiv:2103.11816 (2021)
48. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. arXiv preprint arXiv:2101.11986 (2021)
49. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution transformer for dense prediction. arXiv preprint arXiv:2110.09408 (2021)
50. Yun, S., Han, D., Chun, S., Oh, S.J., Yoo, Y., Choe, J.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV. pp. 6022–6031 (2019)
51. Zhang, H., Cissé, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
52. Zhao, H., Jia, J., Koltun, V.: Exploring self-attention for image recognition. In: CVPR. pp. 10073–10082 (2020)
53. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI. pp. 13001–13008 (2020)
54. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ADE20K dataset. Int. J. Comput. Vis. 127(3), 302–321 (2019)