Neural Networks 180 (2024) 106653

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

DiagSWin: A multi-scale vision transformer with diagonal-shaped windows for object detection and segmentation

Ke Li, Di Wang [*], Gang Liu, Wenxuan Zhu, Haodi Zhong, Quan Wang

Key Laboratory of Smart Human–Computer Interaction and Wearable Technology of Shaanxi Province, Xidian University, Xi'an, 710071, China

ARTICLE INFO

Keywords:
Vision transformer
Multi-scale
Diagonal-shaped windows
Object detection and semantic segmentation

ABSTRACT

Recently, Vision Transformer and its variants have demonstrated remarkable performance on various computer vision tasks, thanks to its competence in capturing global visual dependencies through self-attention. However, global self-attention suffers from high computational cost due to quadratic computational overhead, especially for the high-resolution vision tasks (e.g., object detection and semantic segmentation). Many recent works have attempted to reduce the cost by applying fine-grained local attention, but these approaches cripple the long-range modeling power of the original self-attention mechanism. Furthermore, these approaches usually have similar receptive fields within each layer, thus limiting the ability of each self-attention layer to capture multi-scale features, resulting in performance degradation when handling images with objects of different scales. To address these issues, we develop the Diagonal-shaped Window (DiagSWin) attention mechanism for modeling attentions in diagonal regions at hybrid scales per attention layer. The key idea of DiagSWin attention is to inject multi-scale receptive field sizes into tokens: before computing the self-attention matrix, each token attends its closest surrounding tokens at fine granularity and the tokens far away at coarse granularity. This mechanism is able to effectively capture multi-scale context information while reducing computational complexity. With DiagSwin attention, we present a new variant of Vision Transformer models, called DiagSWin Transformers, and demonstrate their superiority in extensive experiments across various tasks. Specifically, the DiagSwin Transformer with a large size achieves 84.4% Top-1 accuracy and outperforms the SOTA CSWin Transformer on ImageNet with 40% fewer model size and computation cost. When employed as backbones, DiagSWin Transformers achieve significant improvements over the current SOTA modules. In addition, our DiagSWin-Base model yields 51.1 box mAP and 45.8 mask mAP on COCO for object detection and segmentation, and 52.3 mIoU on the ADE20K for semantic segmentation.

1. Introduction

The emergence of Vision Transformer (ViT) has showcased the impressive competitiveness and immense potential of Transformer-based architectures in the realm of computer vision (Dosovitskiy et al., 2020). Building upon its success, there has been a growing endeavor to apply ViT to various computer vision tasks, including image classification (Dong et al., 2022; Fan et al., 2021; Li, Xie, Zhang, & Shi, 2023; Vaswani et al., 2021; Wen, Liu, Deng, Liu, Fei, Yan, & Xu, 2024; Yang et al., 2021), object detection (Carion, Massa, Synnaeve, Usunier, Kirillov, & Zagoruyko, 2020; Dai, Cai, Lin, & Chen, 2021; Wu, Liu, Wen, Xu, Yang, & Li, 2023; Wu, Wen, Xu, Yang, Li, & Zhang, 2024; Zheng et al., 2020), segmentation (Lin, Yan, et al., 2021; Liu et al., 2023; Wang, Xu, et al., 2021; Zhang et al., 2023) and tracking (Cai et al., 2023).

The multi-head self-attention (MSA) mechanism is the key to the success of Transformers in computer vision. By leveraging the MSA mechanism, ViT models global dependencies and captures long-range contextual information, thereby facilitating their understanding of intricate relationships and dependencies within images. However, when dealing with fine-grained vision tasks such as high-resolution object detection or pixel-level semantic segmentation, Transformers are computationally inefficient due to the quadratic computational cost with respect to the number of tokens in feature maps. To improve the efficiency, one typical way is to limit the attention region of each token from full-attention to window-based local self-attention (Fan et al., 2021; Vaswani et al., 2021). To bridge connections between windows, researchers further proposed shift operations to exchange information with neighboring windows (Liu et al., 2021). However, the receptive field is enlarged quite slowly and it requires stacking a great number of blocks to achieve global self-attention. A sufficiently large receptive field is crucial to the performance especially for the downstream

[*] Corresponding author.
E-mail address: wangdi@xidian.edu.cn (D. Wang).

https://doi.org/10.1016/j.neunet.2024.106653
Received 21 November 2023; Received in revised form 3 August 2024; Accepted 19 August 2024
Available online 22 August 2024

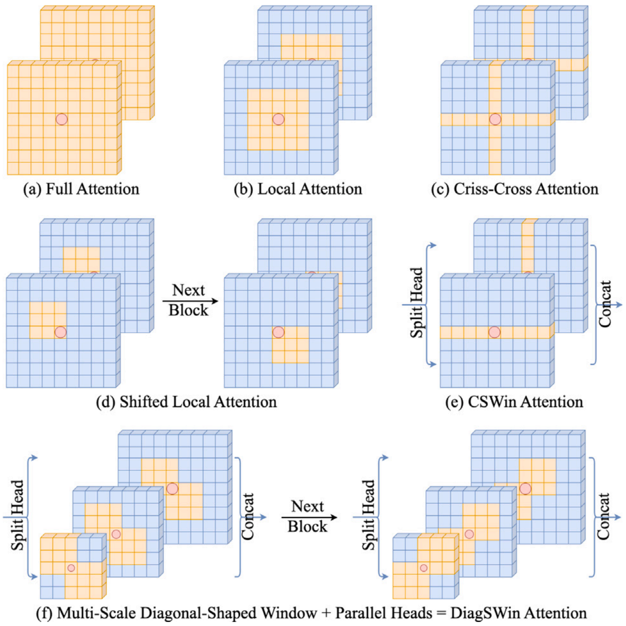

(a) Full Attention
(b) Local Attention
(c) Criss-Cross Attention
Next Block
(d) Shifted Local Attention
Split Head
Concat
(e) CSWin Attention
Split Head
Concat
Next Block
Split Head
Concat
(f) Multi-Scale Diagonal-Shaped Window + Parallel Heads = DiagSWin Attention

Fig. 1. Comparison of different attention mechanisms. (a) Full Attention, (b) Local Attention, (c) Criss-Cross Attention, (d) Shifted Local Attention, (e) CSWin Attention and (f) our DiagSWin Attention. Our DiagSWin has two characteristics. First, DiagSWin allows self-attention heads within the same layer to capture short- and long-range interactions. Second, DiagSWin can effectively expand the attention region by adjusting the direction of the diagonal window in two consecutive self-attention layers.

tasks (e.g., object detection and segmentation). Furthermore, other approaches aim to compute self-attention at high-resolution features by downsampling the key and value feature maps to lower computation cost (Wang, Xie, et al., 2021; Wang et al., 2022). However, these approaches tend to merge too many tokens within a single self-attention layer This leads to a blend of tokens from both small objects and background noise, ultimately leading to a degradation in the model's performance for high-resolution vision tasks.

In addition, previous Transformer models largely neglect the multi-scale nature of scene objects within one attention layer, rendering performance degradation when confronted with visual scenes containing objects of various sizes. This is because the receptive field of each token remains consistent throughout the entire attention layer, making it inefficient to capture features at different scales within the same attention layer.

To address this limitation, we present a novel multi-scale **Diag**onal-**S**haped **Win**dow (DiagSWin) attention mechanism, which allows self-attention heads within the same layer to capture fine- and coarse-grained dependencies for high-resolution images. As shown in Fig. 1(f), a query token in the feature map attends to its different surroundings in different heads. Furthermore, we alternately compute self-attention in the left and right diagonal-shaped windows in successive blocks to expand the receptive field.

We show a comparison with vanilla self-attention. Previous self-attention mechanisms typically apply the same attention operation across multi-heads (Fig. 1(a), (b), (c), (d)), and perform different attention operations sequentially (Fig. 1(d)). However, recent self-attention mechanisms have split the multi-heads into parallel groups and apply different self-attention operations onto different groups (Fig. 1(e)) We will show through ablation analysis that DiagSWin attention effectively models visual dependencies among all regions covering the entire feature map, while requiring fewer tokens in computation than standard self-attention mechanisms.

Based on the DiagSWin attention, we follow the hierarchical design and propose a new vision Transformer architecture named "DiagSWin Transformer" for general-purpose vision tasks. As depicted in Fig. 2, this architecture provides stronger modeling capabilities with lower computation costs and fewer parameters. Under the similar FLOPs (Floating Point Operations) and model size, DiagSWin Transformer variants significantly outperform previous state-of-the-art (SOTA) vision Transformers. On ImageNet (Deng et al., 2009), our DiagSWin Transformer outperforms the SOTA CSWin Dong et al. (2022) with only **2/3** parameters (78M→46M) and computational cost (15.0G→9.0G). When scaling down to small sizes, DiagSWin Transformer achieves performance similar to that of SWin-Base (Liu et al., 2021), yet with only **35%** parameters. For object detection, instance segmentation, and semantic segmentation, DiagSWin Transformer consistently outperforms Focal Transformer (Yang et al., 2021) on COCO (Lin et al., 2014) and ADE20K (Zhou et al., 2017) with a similar model size. Besides, our base variant DiagSWin-B achieves **51.1** box mAP and **45.8** mask mAP on the COCO detection task, and **52.3** mIOU on the ADE20K semantic segmentation task, surpassing previous Swin Transformer counterpart by +**2.6**, +**2.4**, and +**2.6**, respectively. Under a smaller FLOPs setting, Our tiny variant DiagSWin-T even shows larger performance gains, i.e., +**3.2** box mAP, +**2.1** mask mAP on COCO detection, and +**4.8** on ADE20K segmentation.

In summary, our contribution are listed as follows.

1. We introduce Diagonal-shaped Window (DiagSWin) attention mechanism, which combines multi-scale feature extractions within a single self-attention layer via multi-scale token aggregation.
2. Based on DiagSWin attention, we build DiagSWin Transformer module, which can easily capture multi-scale objects in high-resolution images.
3. We evaluate the performance of the proposed DiagSWin Transformer across a range of visual tasks, including classification, object detection, and segmentation. Experimental results consistently show that the proposed DiagSWin Transformer outperforms previous Vision Transformers under similar model size.

The rest of this article is organized as follows. Related works of our method are introduced in Section 2. The proposed DiagSWin Transformer is elaborated in Section 3. Extensive experiments and a detailed analysis are presented in Section 4. Conclusions are drawn in Section 5.

2. Related work

2.1. Efficient self-attention

In natural language processing, self-attention mechanisms have been shown to efficiently process long documents by capturing contextual information. In image processing, this approach can also help us better understand the semantic information in images. Since the image resolution is often very high in vision tasks, designing efficient self-attention mechanisms is very crucial. Early vision Transformers (Dosovitskiy et al., 2020; Fan et al., 2021) used the original full self-attention, whose computation complexity is quadratic to image size. To address this issue, some vision Transformers (Liu et al., 2021; Yang et al., 2021) have implemented short-range modeling through local attention mechanisms to reduce computation complexity and memory costs. Another approach is criss-cross attention (Huang et al., 2019), which calculates specific long-range attention in stripe windows along horizontal and vertical axes. However the performance of criss-cross attention is inefficient in practice due to its overlapped window design and ineffective due to its restricted window size. Focal attention (Yang et al., 2021) takes the lead to provide mechanism that incorporates local and global attention in a single Transformer layer. But it tends to merge too many tokens within one self-attention layer, thereby reducing the efficiency of the model in capturing small objects. In this article, we propose a multi-scale diagonal-shaped window (DiagSWin) attention mechanism that allows the self-attention heads within the

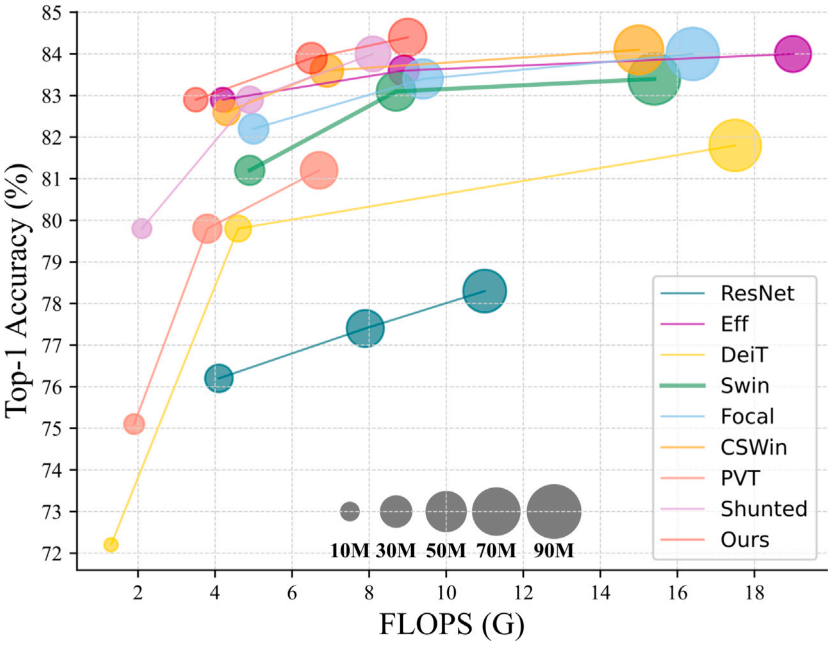


Fig. 2. Our DiagSWin Transformer and its variants achieve comparable or superior performance on ImageNet, while utilizing fewer parameters and FLOPs. The size of the circle is proportional to the model size.

same layer to capture fine- and coarse-grained dependencies for high-resolution images. DiagSWin attention employs a more rational window design approach and can perform secondary attention in overlapping horizontal and vertical regions by alternately using left and right diagonal windows. Fig. 1 shows a visual comparison of the proposed method with other related attention methods.

2.2. *Vision transformers*

Deep Convolutional Neural Networks (DCNNs) have been the predominant force in the field of computer vision, delivering remarkable achievements. Inspired by the success of Transformers (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, & Polosukhin, 2017) in natural language processing (NLP), the seminal work ViT (Dosovitskiy et al., 2020) has shown that non-convolutional architectures relying solely on transformer layers can also achieve competitive results in visual tasks. Numerous follow-up studies have been conducted to enhance the model's classification performance through more complex data augmentation or knowledge distillation techniques (Ren, Gao, et al., 2022; Touvron et al., 2021). Since the computational complexity of self-attention is quadratic with respect to the number of tokens, it becomes prohibitively expensive for handling high-resolution tasks. Therefore, these Vision Transformer (ViT) models commonly employ non-overlapping and larger-sized patches (tokens) when partitioning the image. However, this kind of partitioning is overly coarse and results in the loss of valuable fine-grained information. Recent works based on windowed attention achieves a better accuracy/FLOPs trade-off by performing fine-grained local attention. The Swin Transformer (Shifted Windows Transformer) (Liu et al., 2021), a representative example of local attention, divides feature maps into non-overlapping square regions and performs self-attention locally. However, to model global dependencies via self-attention, these local attention needs to shift the windows over the image or stack a lot of layers to obtain a global receptive field. Other works adopt the strategy of token mergings. PVT (Pyramid Vision Transformer) (Wang, Xie, et al., 2021) designs a spatial reduction attention to merge tokens of key and query. Focal Transformer (Yang et al., 2021) demonstrates attention to objects across different scales by aggregating multi-grained tokens around the query. DilateFormer (Multi-Scale Dilated Transformer) (Jiao et al., 2023) utilizes dilated convolution in the shallow layer of the transformer to achieve a larger receptive field. However, these models usually downsample feature maps and work with low-resolution features, which limits their effectiveness in high-resolution vision tasks. Therefore, we introduce the DiagSWin self-attention to address the limitations mentioned above. It can preserve both coarse- and fine-grained details while simultaneously maintaining a global dependency modeling over the image tokens.

2.3. *Multi-scale visual methods*

The exploration of multi-scale feature representations has consistently been a focus research in computer vision. In image processing (Bracewell & Bracewell, 1986; Torrence & Compo, 1998), various transformation analysis methods are used to decompose images into multiple sub-bands of different scales for processing. Image pyramids (Adelson, Anderson, Bergen, Burt, & Ogden, 1984; Meer, 1989) obtain different scales of images by scaling up or down the original image. In the era of CNNs, different sizes of convolutional kernels have been used to extract features at different scales since the introduction of AlexNet (Krizhevsky, Sutskever, & Hinton, 2017). Pooling methods (Tolias, Sicre, & Jégou, 2015) reduce the size of feature maps while preserving more detailed information. FPN (Feature Pyramid Networks) (Lin, Dollár, et al., 2017) extracts and fuses multi-scale information by scaling and reconstructing images. Currently, there have been studies on integrating multi-scale representations into transformer-based architectures. For example, CrossViT (Chen, Fan, & Panda, 2021) proposes a dual-branch structure, where each branch takes image patches of different sizes as input to produce stronger image features. MViT (Multiscale vision transformer) (Fan et al., 2021) constructs several channels-resolution scale stages, the stages hierarchically expands channel capacity while reducing the spatial resolution. MPViT (Multi-path vision transformer) (Lee, Kim, Willette, & Hwang, 2022) consists of multi-scale patch embedding and multi-path transformer blocks. EAPT (Efficient Attention Pyramid Transformer) (Lin, Sun, et al., 2021) and ViTAE (Xu, Zhang, Zhang, & Tao, 2021) design additional convolution branches outside or inside the self-attention blocks to integrate multiscale information. However, the methods mentioned above are unsuitable for use as a general-purpose backbone

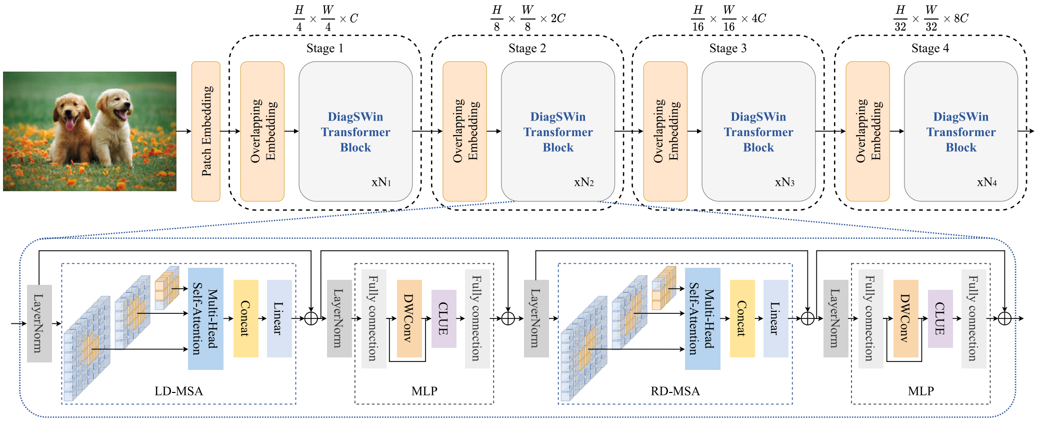

$\frac{H}{4} \times \frac{W}{4} \times C$
$\frac{H}{8} \times \frac{W}{8} \times 2C$
$\frac{H}{16} \times \frac{W}{16} \times 4C$
$\frac{H}{32} \times \frac{W}{32} \times 8C$
Stage 1
Stage 2
Stage 3
Stage 4
Patch Embedding
Overlapping Embedding
DiagSWin Transformer Block
xN1
xN2
xN3
xN4
LayerNorm
Multi-Head Self-Attention
Concat
Linear
LD-MSA
LayerNorm
Fully connection
DWConv
CLUE
Fully connection
MLP
LayerNorm
Multi-Head Self-Attention
Concat
Linear
RD-MSA
LayerNorm
Fully connection
DWConv
CLUE
Fully connection
MLP

Fig. 3. The architecture of a DiagSWin Transformer; The top part shows DiagSWin Transformer, consisting of Overlapping Embedding and DiagSWin Transformer Block. The bottom part shows DiagSWin Transformer Block (notation presented with Eq. (6)). LD-MSA and RD-MSA are multi-head self attention modules with left and right diagonal-shaped windowing configurations, respectively.

network on high-resolution dense prediction tasks, due to its low-resolution feature maps and the quadratic increase in complexity with image size. To the best of our knowledge, the proposed method samples the feature maps at multiple scales in a single transformer layer (horizontal dimension) to form a feature pyramid, and across the entire network architecture (vertical dimension) to form a cascaded feature pyramid, which is a unique contribution of our work.

3. The proposed method

3.1. Overall architecture

An overview of the DiagSWin Transformer architecture is presented in Fig. 3(a). To adapt to high-resolution or pixel-dense prediction tasks, we follow the design approach of Liu et al. (2021), Wang, Xie, et al. (2021), an image $I \in \mathcal{R}^{H \times W \times 3}$ is first embedded by a 7×7 convolution layer with stride 4, followed by a normalization layer to get $\frac{h}{4} \times \frac{w}{4}$ patch tokens, and the dimension of each token is C. Aiming to build a hierarchical feature pyramid (vertical dimension), the backbone network consists of 4 stages. We use a overlapping embedding layer (Ren, Zhou, He, Feng, & Wang, 2022) between two adjacent stages to reduce the spatial size of the feature map by half and double the channel dimension. Therefore, the constructed feature maps have $\frac{h}{2^{i+1}} \times \frac{w}{2^{i+1}}$ tokens for the ith stage which has the same feature map resolution as those typical convolutional networks, e.g., ResNet (He, Zhang, Ren, & Sun, 2016) and EfficientNet (Tan & Le, 2019). As a result, the proposed architecture can be conveniently substituted for the backbone networks in existing methods for various visual tasks. For image classification tasks, we first normalize the feature map output from the last stage and then use a linear classifier for prediction. For object detection or semantic segmentation tasks, the DiagSWin Transformer is used as the backbone network to extract multi-scale features. We add a normalization layer to the features at each stage and then feed them into the object detection head or image reconstruction decoder. Although self-attention mechanisms balance short-range and long-range modeling, the standard self-attention mechanism still suffers from high computational cost in high-resolution tasks. In the next section, we describe in detail how we use the proposed DiagSWin attention mechanism to alleviate this problem.

DiagSWin Transformer Block. DiagSWin Transformer is built by replacing the standard MSA module in a Transformer block with a module based on diagonal-shaped windows (described in Section 3.2). As shown in Fig. 3 (bottom), DiagSWin Transformer block consists of alternating left and right diagonal MSA modules, followed by a 2-layer MLP with a depth-wise convolution (DWConv) and a GELU nonlinearity in between. A LayerNorm (LN) layers are applied before each MSA module and each MLP, and a residual connection is applied after each module.

3.2. Multi-scale diagonal-shaped window self-attention

We proposed DiagSWin attention to make the Transformer layers applicable to high-resolution visual tasks. Unlike the approach of simply designing self-attention windows of different sizes, we focus on the fine-grain tokens only locally, but the summarized ones globally. Therefore, DiagSWin attention can cover the same number of image regions with less cost compared to standard self-attention. It is worthwhile to note that with DiagSWin attention mechanism, the self-attention in multi-granularity diagonal-shaped windows is calculated in parallel. We split the multi-heads into parallel groups and apply different self-attention operations onto different groups. This parallel strategy effectively reduces extra computation cost introduced by merging too many tokens within one self-attention layer. In addition, we alternate between computing left and right diagonal-shaped window attention within a DiagSWin block. The non-overlapping regions in the left and right diagonal-shaped windows attend to global contextual information, while the overlapping region enhances local attention to neighboring tokens. In theory, DiagSWin attention can achieve global interaction with less computational cost. In practice, extracting the surrounding tokens for each query position can lead to significant time overhead, as it necessitates redundant extraction of each token for all queries around it. Therefore, in our DiagSWin Transformer, we resort to performing DiagSWin attention at the window level. We provide a detailed elaboration of the DiagSWin attention.

3.2.1. DiagSWin attention

For a given input $X \in \mathcal{R}^{H \times W \times C}$ with spatial size $H \times W$ and channel dimension C, we first partition it into uniformly sized image patches (7 in our settings). Then, we use the image patch as the minimum unit for attention calculation and perform multi-scale fine-grained and coarse-grained attention. Fig. 4 shows the proposed DiagSWin Attention. To illustrate more clearly, we introduce three terms.

- **Downsampling Rate** r_i denotes the downsampling rate that summarized tokens for different heads indexed by $i \in \{1, \dots head\}$.
- **Aggregation size** s_i is the size of the window on which the summarized tokens are aggregated in the ith head.

Now, we will elaborate on how DiagSWin attention works in the following 3 steps: multi-scale downsampling, DiagSWin aggregating and attention calculation.

Multi-scale Downsampling. Before computing attention, the feature map $X \in \mathcal{R}^{H \times W \times C}$ is downsampled to different sizes for different heads indexed by i:

$$\tilde{X}_i = \mathcal{P}(X; r_i) \tag{1}$$

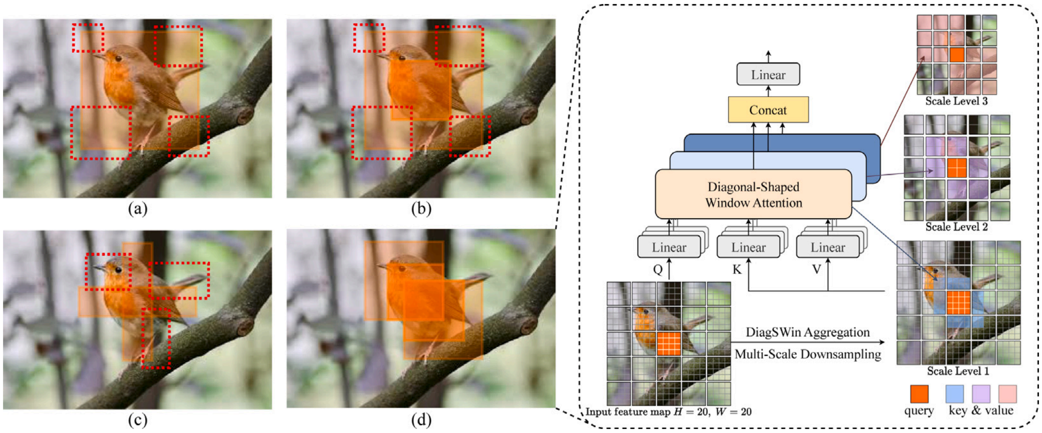


Fig. 4. Left: An illustrative description of different attention mechanisms. Right: An illustration of DiagSWin attention at patch level. Each of the square cells represents a visual token. Taking the left diagonal-shaped window as an example, the input feature map size is 20 × 20. We first partition it into 5 × 5 windows of size 4 × 4. Take the 4 × 4 orange window in the middle as the query set, we extract its surrounding tokens at three granularity levels as its keys and values. For the first level, we aggregate the left diagonal-shaped window 56 tokens closest to the orange window at the finest grain. At the second level, we expand the attention region and nd downsample with $r = 4$ to form summarized tokens, which results in 28 summarized tokens. At the third level, we attend a larger region covering the whole feature map and downsample the sub-windows by $r = 8$, which results in 17 summarized tokens. Finally, these three levels of tokens are distributed into different heads to compute keys and values for 16 queries in the orange window in parallel.

Here $\mathcal{P}(\cdot; r_i)$ is the multi-scale pooling layer in the ith head with the downsampling rate of r_i. In practice, we take a convolution layer with kernel size and stride of r_i to implement the downsampling. The downsampled feature maps $\{x_i\}_1^{head}$ provide rich information at both fine- and coarse-grain in a single layer across the attention heads. Since we set $r = 1$ for the first focal level which has the same granularity as the input feature map, there is no need to perform pooling operator. In addition, the operator $\mathcal{P}(\cdot; \Theta)$ can also select different methods, such as max pooling, mean pooling, 2D convolution, dilated convolution, or fully connected layers. In , we conducted comparative experiments on these different pooling methods.

DiagSWin Aggregating. For images, the importance of the relationship between pixels often decreases inversely with distance. The closer the distance, the higher the correlation. We attend to 8 surrounding regions (top, left, bottom, right, top-left, bottom-left, top-right, bottom-right) of the same patch. From the distance perspective, pixels in the horizontal and vertical directions are closer to the center point than those in the diagonal direction. Therefore, we expand the patch to its upper-left and lower-right (or lower-left and upper-right) corners, based on which we form a left (or right) diagonal-shaped window W:

$$W_i = AGG(\tilde{X}_i, s_i) \tag{2}$$

where $AGG(\cdot)$ is the diagonal-shaped window token aggregation layer in the ith head. For a given query $q \in \mathcal{R}^{m\times m}$, the DiagSWin attended region is in the range $\left[-s_i - m, m + s_i\right]$. This aggregation size is a crucial parameter of the diagonal-shaped window as it enables us to achieve strong modeling capability while lowering computation cost. Specifically, the diagonal-shaped window coverage area is related to the depth of the network: small coverage area for shallow layers and large coverage area for deep layers. A larger diagonal-shaped window size encourages a stronger connection between long-range elements and achieves better network capacity with a small increase in computation cost. We provide a mathematical analysis of how the diagonal-shaped window size affects the modeling capability and computation cost.

Attention Computing. As mentioned earlier, we use patches as units of computation, which means that tokens within a patch share the same set of surroundings. For the feature map $\tilde{X}_i$, we use three linear mapping layers f^q, f^k, and f^v to calculate the queries Q, keys K, and values V. The calculation of query is only performed on the feature map where $r_i = 1$. For the jth patch in ith head:

$$\begin{aligned} Q^j &= f^q(\tilde{X}^1, \dots \tilde{X}^n), \\ K_i^j &= f_i^k(W_i^1, \dots W_i^n), \\ V_i^j &= f_i^v(W_i^1, \dots W_i^n), \end{aligned} \tag{3}$$

Table 1
Comparison of computational complexity of different attention methods.

Model	Complexity	Description
ViT	$\mathcal{O}\left(\frac{H^2W^2}{p^4}C\right)$	p is the patch size.
Swin	$\mathcal{O}(CHWp^2)$	p is the window size.
Focal	$\mathcal{O}\left(CHW(L + \sum_l (s_r^l)^2)\right)$	s_r^l represents focal region size at level $l \in \{1, \dots, L\}$.
DiagSWin	$\mathcal{O}\left(CHW \sum_i \left(\frac{1}{r_i^2} + W_i^2\right)\right)$	r_i and W_i are downsampling rate and diagonal-shaped window size at scale i, respectively.

where $n \in [\frac{H}{p} \times \frac{W}{p}]$, and $Q^j, K_i^j, V_i^j, \in \mathcal{R}^{p\times p\times C}$. Then, we follow (Liu et al., 2021) to include a relative position bias and compute the DiagSWin attention for Q^j in the ith head by,

$$Attention(Q^j, K_i^j, V_i^j) = Softmax(\frac{Q^j K_i^{j\,T}}{\sqrt{d}} + B)V_i^j \tag{4}$$

where B represents the learnable relative positional offsets (Liu et al., 2021) added within the diagonal-shaped window.

3.2.2. *Complexity analysis*

For a feature map $x \in \mathcal{R}^{H\times W\times C}$, we first downsample it to obtain $\tilde{X}_i \in \mathcal{R}^{\frac{H}{r_i}\times\frac{W}{r_i}\times C}$, with a complexity of $\mathcal{O}(\sum_i \frac{(H\times W)}{r_i^2}C)$. Regarding attention computation, the cost of computing a $p \times p$ patch is $\mathcal{O}(p^2 \sum_i W_i^2 C)$, and the cost of computing the entire feature map is $\mathcal{O}(\sum_i W_i^2 HWC)$. In summary, the computational cost of our DiagSWin attention is $\mathcal{O}(CHW \sum_i (\frac{1}{r_i^2} + W_i^2))$. Table 1 shows the complexity comparison of our DiagSWin and other attention methods. It can be observed that ViT has the highest complexity, especially for large-size input feature maps. In contrast, the proposed DiagSWin demonstrates the lowest complexity due to its primarily linear complexity and smaller constant factors. The proposed attention method significantly reduces the complexity as the resolution increases.

3.3. *Detail feed-forward layer*

In the traditional feed-forward layer, the fully connected layer operates on a point-wise basis, and it cannot learn cross-token information. As shown in Fig. 3, we complement the local details in the feed forward layer by adding a depth-wise convolution (DWConv) layer between

Table 2
Model configurations for DiagSWin Transformers. Ω indicate that DiagSWin transformer block uses the full attention mechanism.

	Layer Name	Output size	DiagSWin-Tiny	DiagSWin-Small	DiagSWin-Base
Stage 1	Overlapping Embedding	56 × 56	$p_1 = 4; C_1 = 64$	$p_1 = 4; C_1 = 64$	$p_1 = 4; C_1 = 64$
	Transformer Block	56 × 56	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 8, s_i = 2 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 2,\ N_1 = 1$	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 8, s_i = 2 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 2,\ N_1 = 2$	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 8, s_i = 2 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 2,\ N_1 = 2$
Stage 2	Overlapping Embedding	28 × 28	$p_2 = 2; C_2 = 128$	$p_2 = 2; C_2 = 128$	$p_2 = 2; C_2 = 128$
	Transformer Block	28 × 28	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 4, s_i = 2 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 4,\ N_2 = 2$	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 4, s_i = 2 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 4,\ N_2 = 2$	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 4, s_i = 2 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 2,\ N_2 = 2$
Stage 3	Overlapping Embedding	14 × 14	$p_3 = 2; C_3 = 256$	$p_3 = 2; C_3 = 256$	$p_3 = 2; C_3 = 256$
	Transformer Block	14 × 14	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 2, s_i = 4 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 8,\ N_3 = 3$	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 2, s_i = 4 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 8,\ N_3 = 8$	$\begin{bmatrix} r_i = 1, s_i = 3 & i < \frac{head}{2} \\ r_i = 2, s_i = 4 & i \geq \frac{head}{2} \end{bmatrix}$ $head = 2,\ N_3 = 12$
Stage 4	Overlapping Embedding	7 × 7	$p_4 = 2; C_4 = 512$	$p_4 = 2; C_4 = 512$	$p_4 = 2; C_4 = 512$
	Transformer Block	7 × 7	$r = 1, s = \Omega$ $head = 16,\ N_4 = 1$	$r = 1, s = \Omega$ $head = 16,\ N_4 = 1$	$r = 1, s = \Omega$ $head = 16,\ N_4 = 2$

the first fully connected layer and the GELU nonlinearity in the feed forward layer:

$$\begin{aligned} x' &= fc(x), \\ x'' &= fc(\sigma(x' + DWConv(x'))), \end{aligned} \tag{5}$$

By utilizing the above method, DiagSWin attention sequentially focuses on the left diagonal and right diagonal regions in a DiagSWin transformer blocks. Consecutive DiagSWin Transformer blocks are computed as,

$$\begin{aligned} \hat{z}^l &= LD\text{-}MSA(LN(z^{l-1})) + z^{l-1}, \\ z^l &= DFF(LN(\hat{z}^l)) + \hat{z}^l, \\ \hat{z}^{l+1} &= RD\text{-}MSA(LN(z^l)) + z^l, \\ z^{l+1} &= DFF(LN(\hat{z}^{l+1})) + \hat{z}^{l+1}, \end{aligned} \tag{6}$$

where $\hat{z}^l$ and z^l denote the output features of the L(/R)W-MSA module and the Detail Feed-Forward Layer for block l, respectively. The overall calculate process of the proposed DiagSWin Transformer is shown in Algorithm 1.

Algorithm 1 DiagSWin Transformer

```
Require: Input image $I \in \mathbb{R}^{H \times W \times 3}$, stage layers $L$, block depths $D$
Ensure: Final feature of the DiagSWin Transformer
Perform patch embedding to extract visual feature $X \in \mathbb{R}^{H \times W \times C}$
for stage ← 1 to $L$ do
    Perform overlapping embedding to reduce spatial size by half and double the channel dimension
    for block ← 1 to $D$ do
        for direction ∈ {left, right} do
            Perform multi-scale downsampling to obtain multi-scale visual features $\tilde{X}_i$ by Eq. (1)
            Aggregate tokens to construct diagonal-shaped window $W_i$ by Eq. (2)
            Generate queries $Q$, keys $K$, and values $V$ by Eq. (3)
            Compute self-attention to obtain feature $x$ by Eq. (4)
            Input $x$ to detail feed-forward layer to obtain output of block $x''$ by Eq. (5)
        end for
    end for
end for
return Final feature of the DiagSWin Transformer
```

3.4. Model settings

Following Wang, Xie, et al. (2021), Wu et al. (2021), we consider three configurations for DiagSWin Transformers: DiagSWin-Tiny, DiagSWin-Small and DiagSWin-Base, as shown in Table 2. Our models take image $I \in \mathcal{R}^{224 \times 224 \times 3}$ as input, and the window partition size is set to 7 to make our models comparable to Swin Transformers (Liu et al., 2021). As show in Table 2, *head* and the N_i indicate the number of heads in one block and the number of blocks in one stage. The variants only come from the number of layers in different stage. Specifically, the number of head in each block is set to $\{2, 4, 8, 16\}$. For fine-grained attention ($i < \frac{head}{2}$), the downsampling rates are all set to 1 with the aggregation sizes all set to 3. For the coarse-grained attention ($i \geq \frac{head}{2}$), the downsampling rates for the four stages are $\{8, 4, 2, 1\}$, and the aggregation sizes are $\{2, 2, 4, 6\}$, respectively. For the overlapping embedding layer, except for the first stage where the spatial reduction ratio p is 4, the remaining three stages are all 2.

4. Experiments

To assess the performance of our DiagSwin Transformer, we employ it across a range of benchmark tasks, including ImageNet-1K classification (Deng et al., 2009), COCO object detection and instance segmentation (Lin et al., 2014), and ADE20K semantic segmentation (Zhou et al., 2017). Furthermore, we conduct comprehensive ablation studies to systematically evaluate the impact of individual model components on overall performance.

4.1. ImageNet-1K classification

For a fair comparison, we follow the training strategies in Touvron et al. (2021) and compare different methods on ImageNet-1K (Deng et al., 2009). Specifically, all our models are trained for 300 epochs with the input size of 224×224 and the batch size of 1024. We set the initial learning rate to 10^{-3} with 20 epochs of linear warm-up starting from 10^{-5}. For optimization, we utilize the AdamW (Loshchilov & Hutter, 2017) with a cosine learning rate scheduler and the weight decay is set to 0.05. The stochastic depth drop rates are set to 0.2, 0.2 and 0.3 for DiagSWin-tiny, DiagSWin-small and DiagSWin-base models, respectively. We incorporate most of the augmentation and regularization techniques from Touvron et al. (2021), with the exception of repeated augmentation (Hoffer, Ben-Nun, Hubara, Giladi, Hoefler, Soudry, 2020) and exponential moving average (Polyak & Juditsky, 1992).

In Table 3, we summarize the results for DiagSWin Transformers and other backbones on image classification task. It shows that DiagSWin Transformers consistently outperform other architectures (including Transformer- and ConvNet-based) with fewer model sizes (#Params.), computation cost (FLOPs) and inference speed (Throughput), using regular ImageNet-1K training.

Compared with SOTA CNNs, our DiagSWin Transformer generally achieves a 6.1%–6.7% points higher accuracy compared to a ResNet

Table 3
ImageNet-1K classification. These models are trained with 224 × 224 resolution.

Method	Image size	#Param.	FLOPs	Throughput	Top-1
Eff-B4 (Tan & Le, 2019)	380^2	19M	4.2G	349/s	82.9
Eff-B5 (Tan & Le, 2019)	456^2	30M	9.9G	169/s	83.6
Eff-B6 (Tan & Le, 2019)	528^2	43M	19.0G	96/s	84.0
ResNet-50 (He et al., 2016)	380^2	25M	4.1G	317/s	76.2
ResNet-101 (He et al., 2016)	456^2	45M	7.9G	146/s	77.4
ResNet-152 (He et al., 2016)	528^2	60M	11.0G	73/s	78.3
DeiT-S (Touvron et al., 2021)	224^2	22M	4.6G	940/s	79.8
DeiT-B (Touvron et al., 2021)	224^2	87M	17.5G	292/s	81.8
CrossViT-15 (Chen et al., 2021)	224^2	27M	5.8G	640/s	81.5
CrossViT-18 (Chen et al., 2021)	320^2	43M	9.0G	430/s	82.5
CvT-13 (Wu et al., 2021)	224^2	20M	4.5G	–	81.6
CvT-21 (Wu et al., 2021)	224^2	32M	7.1G	–	82.5
CvT-21 (Wu et al., 2021)	384^2	32M	24.9G	–	83.3
MViT-B-16 (Fan et al., 2021)	224^2	37M	4.2G	583/s	82.5
MViT-B-24 (Fan et al., 2021)	224^2	54M	9.9G	219/s	83.1
Swin-T (Liu et al., 2021)	224^2	28M	4.9G	755/s	81.2
Swin-S (Liu et al., 2021)	224^2	49M	8.7G	437/s	83.1
Swin-B (Liu et al., 2021)	224^2	87M	15.4G	278/s	83.4
Focal-T (Yang et al., 2021)	224^2	29M	4.9G	319/s	82.2
Focal-S (Yang et al., 2021)	224^2	51M	9.4G	192/s	83.6
Focal-B (Yang et al., 2021)	224^2	90M	16.4G	138/s	84.0
CSWin-T (Dong et al., 2022)	224^2	23M	4.3G	701/s	82.6
CSWin-S (Dong et al., 2022)	224^2	35M	6.9G	437/s	83.6
CSWin-B (Dong et al., 2022)	224^2	78M	15.0G	250/s	84.2
PVT-S (Wang, Xie, et al., 2021)	224^2	25M	3.8G	820/s	79.8
PVT-M (Wang, Xie, et al., 2021)	224^2	44M	6.7G	526/s	81.2
PVT-L (Wang, Xie, et al., 2021)	224^2	61M	9.8G	367/s	81.7
Shunted-T (Ren, Zhou, et al., 2022)	224^2	12M	2.1G	883/s	79.8
Shunted-S (Ren, Zhou, et al., 2022)	224^2	22M	4.9G	708/s	82.9
Shunted-B (Ren, Zhou, et al., 2022)	224^2	40M	8.1G	483/s	84.0
PVTv2-B1 (Wang et al., 2022)	224^2	13M	2.1G	936/s	78.7
PVTv2-B2 (Wang et al., 2022)	224^2	25M	4.0G	785/s	82.0
PVTv2-B4 (Wang et al., 2022)	224^2	63M	10.1G	329/s	83.6
MPViT-XS (Lee et al., 2022)	224^2	11M	2.9G	758/s	80.9
MPViT-S (Lee et al., 2022)	224^2	23M	4.7G	396/s	83.0
MPViT-B (Lee et al., 2022)	224^2	75M	16.4G	208/s	84.3
DiagSWin-T(Ours)	224^2	19M	3.4G	728/s	**82.9**
DiagSWin-S(Ours)	224^2	31M	6.5G	523/s	**83.9**
DiagSWin-B(Ours)	224^2	46M	9.0G	413/s	**84.4**

model with fewer parameter size. In addition, we find that our DiagSWin Transformer is the only Transformer-based architecture that achieves comparable or even superior results than Eff-Net under the small and base settings, while maintaining lower computation complexity. It is also worth noting that neural architecture search is used in Eff-Net but not in our DiagSWin Transformer design.

To be exact, DiagSWin-T achieves higher performance than the Transformer baseline DeiT-B, but only requires one-fifth parameters (87 M → 19 M) and computation cost (17.5 G → 3.4 G FLOPs). Compared to Shunted-S, which adopts the same multi-scale idea and achieves the same accuracy, DiagSWin-Tiny only needs fewer parameters (22 M → 19 M) and computational cost (4.9 G → 3.4 G FLOPs). In contrast to the latest SOTA models such as Focal and CSWin, our DiagSwin Transformer consistently delivers superior performance. Notably, our compact model outperforms the Focal-T, by a margin of 0.3%, while reducing the model size by 39%. When model size grows large, our base model achieve SOTA performance with only half of parameters and computation cost comparing with Focal-B, surpassing CSWin-B, PVTv2-B4 and MPViT-B by 0.2%, 0.8% and 0.1% respectively. As the model scales up, our base model manages to achieve SOTA performance with just half of the parameters and computational cost compared to Focal-B. It surpasses CSWin-B, PVTv2-B4, and MPViT-B by 0.2%, 0.8% and 0.1% respectively.

4.2. Semantic segmentation

We further investigate the capability of DiagSWin Transformer for pixel-dense tasks, using the widely-used ADE20K (Zhou et al., 2017) dataset for semantic segmentation, which covers a broad range of 150 semantic categories. It has 25 K images in total, with 20 K for training,

Table 4
ADE20K semantic segmentation results using Semantic FPN (Kirillov, Girshick, He, & Dollár, 2019).

Backbone	#Param.	FLOPs	mIoU
ResNet-50 (He et al., 2016)	29M	183G	36.7
ResNet-101 (He et al., 2016)	48M	260G	38.8
ResNeXt101-64 × 4d (Xie, Girshick, Dollár, Tu, & He, 2017)	86M	–	40.2
Swin-T (Liu et al., 2021)	32M	182G	41.5
Swin-S (Liu et al., 2021)	53M	274G	45.2
Swin-B (Liu et al., 2021)	91M	422G	46.0
CSWin-T (Dong et al., 2022)	26M	202G	48.2
CSWin-S (Dong et al., 2022)	36M	271G	49.2
CSWin-B (Dong et al., 2022)	81M	464G	**49.9**
PVT-T (Wang, Xie, et al., 2021)	17M	–	35.7
PVT-S (Wang, Xie, et al., 2021)	28M	116G	39.8
PVT-M (Wang, Xie, et al., 2021)	48M	219G	41.6
PVTv2-B1 (Wang et al., 2022)	18M	–	42.5
PVTv2-B2 (Wang et al., 2022)	29M	–	45.2
PVTv2-B4 (Wang et al., 2022)	66M	–	47.9
Shunted-S (Ren, Zhou, et al., 2022)	26M	183G	48.2
Dilate-S (Jiao et al., 2023)	28M	178G	47.1
Dilate-B (Jiao et al., 2023)	51M	288G	48.8
DiagSWin-T	23M	172G	48.4
DiagSWin-S	34M	234G	49.1
DiagSWin-B	50M	288G	**49.9**

Table 5
ADE20K semantic segmentation results using UpperNet (Xiao, Liu, Zhou, Jiang, & Sun, 2018). We omit models pretrained on larger-datasets (e.g., ImageNet-21K). Single- and multi-scale evaluations are reported in the last two columns.

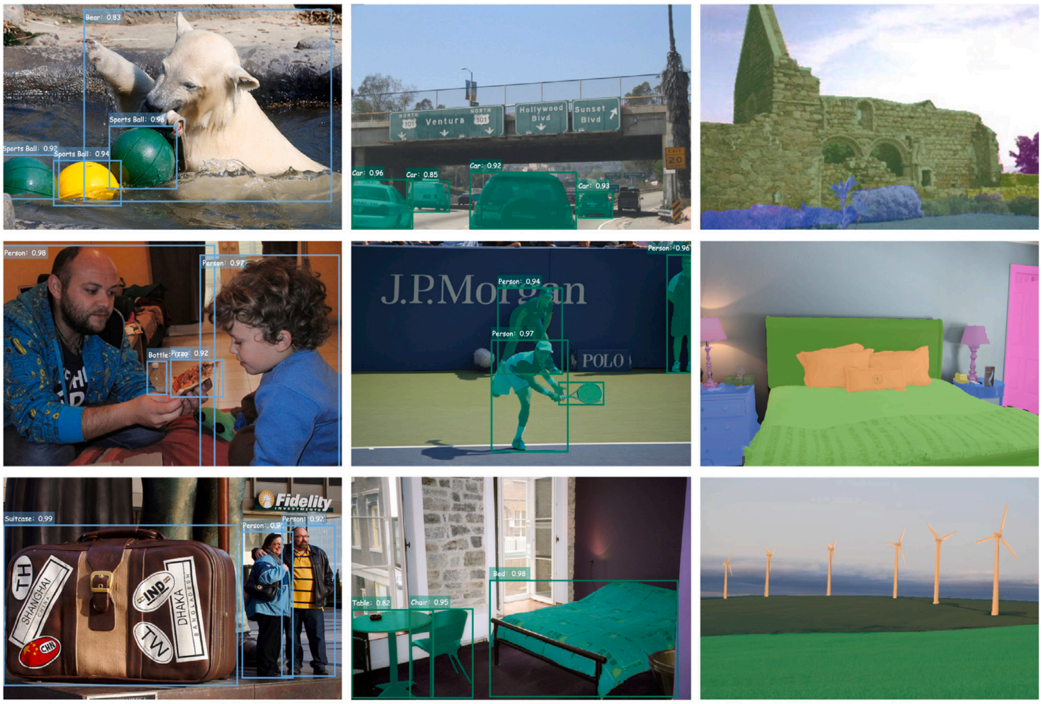
Backbone	#Param.	FLOPs	mIoU	+MS
ResNet-101 (He et al., 2016)	86M	1029G	–	44.9
Swin-T (Liu et al., 2021)	60M	945G	44.5	45.8
Swin-S (Liu et al., 2021)	81M	1038G	47.6	49.5
Swin-B (Liu et al., 2021)	121M	1188G	48.1	49.7
Focal-T (Yang et al., 2021)	62M	998G	45.8	47.0
Focal-S (Yang et al., 2021)	85M	1130G	48.0	50.0
Focal-B (Yang et al., 2021)	126M	1354G	49.0	50.5
CSWin-T (Dong et al., 2022)	60M	959G	49.3	50.7
CSWin-S (Dong et al., 2022)	65M	1027G	50.4	51.5
CSWin-B (Dong et al., 2022)	109M	1222G	51.1	52.2
MPViT-S (Lee et al., 2022)	52M	943G	48.3	–
MPViT-B (Lee et al., 2022)	105M	1186G	50.3	–
Shunted-S (Ren, Zhou, et al., 2022)	52M	940G	48.9	49.9
Dilate-S (Jiao et al., 2023)	54M	935G	47.1	47.6
Dilate-B (Jiao et al., 2023)	79M	1046G	50.8	51.1
DiagSWin-T	49M	884G	49.3	50.6
DiagSWin-S	61M	976G	50.3	51.4
DiagSWin-B	75M	1078G	51.3	**52.3**

2 K for validation, and another 3 K for testing. We report the mIOU results with and without multi-scale testing while employing UperNet and Semantic FPN as the primary frameworks and take different architectures as backbones. We adhere to the default configurations of Focal Transformer and mmsegmentation (Contributors, 2020). In the case of UperNet (Xiao et al., 2018), we opt for the AdamW optimizer with a weight decay of 0.01, running for 160K iterations. The learning rate initiates at 6×10^{-5}, featuring a 1500-iteration warm-up at the beginning of training and linear decay. Our augmentations incorporates random flipping, scaling, and photo-metric distortion, with input sizes set is 512 × 512 during training. We conduct single-scale and multi-scale (MS) testing. For Semantic FPN (Kirillov et al., 2019), we employ AdamW optimizer with a weight decay of 0.0001, coupled with a learning rate of 0.0001 for 80K iterations.

As shown in Table 4, when the framework is Semantic FPN, our DiagSWin Transformer is 5.9%, 3.9% and 2.0% higher than PVTv2. Table 5 shows the results of different methods with the UpperNet framework in terms of mIoU and Multi-scale tested mIoU (MS mIoU). Specifically, DiagSWin-T, DiagSWin-S, and DiagSWin-B outperform the corresponding versions of Swin by 4.8% (60 M → 49 M), 2.7% (81 M → 61 M), 3.2% (121 M → 75 M) mIOU. Compared to CNNs (DiagSWin-B and ResNet-101), mIoU increases by 7.4%, demonstrating significant performance gains and the potential of Vision Transformers. In Fig. 5, we also visualize some qualitative semantic segmentation results on ADE20K. The results of segmentation the shows the superiority of our DiagSWin Transformer.

4.3. COCO object detection

Next, we evaluate our models on object detection with the COCO dataset (Lin et al., 2014), which includes 118 K training images and 5 K validation images. We adopt DiagSWin as the backbone in RetinaNet (Lin, Goyal, et al., 2017) and Mask R-CNN (He et al., 2016) frameworks to assess the effectiveness of our method. Prior to fine-tuning, we pretrain our models on the ImageNet-1K dataset for 300 epochs and follow similar training strategies as Swin Transformer (Liu et al., 2021) to ensure a fair comparison. During training, we employ AdamW (Loshchilov & Hutter, 2017) for optimization with an initial



Object Detection on COCO　Instance Segmentation on COCO　Semantic Segmentation on ADE20K

Fig. 5. Qualitative results of object detection and instance segmentation on COCO val2017, and semantic segmentation on ADE20K. The results (from left to right) are generated by DiagSWin-B-based RetinaNet, Mask R-CNN, and Semantic FPN, respectively.

learning rate of 10^{-4} and weight decay of 0.05. Additionally, we introduce stochastic depth with drop rates of 0.2, 0.3, and 0.5 to regularize the training of our Tiny, Small, and Base models, respectively.

We report DiagSWin in 1× with 12 epochs and 3× with 36 epochs training schedules. We compare DiagSWin Transformer with various backbones: previous CNN backbones ResNet (He et al., 2016), ResNeXt(X) (Xie et al., 2017), and Transformer backbones PVT (Wang, Xie, et al., 2021), ViL (Dosovitskiy et al., 2020), Swins (Liu et al., 2021), Focal (Yang et al., 2021), CSWin Dong et al. (2022), PVT (Wang, Xie, et al., 2021), PVTv2(Wang et al., 2022), MPViT (Lee et al., 2022), Shunted (Ren, Zhou, et al., 2022) and Dilate (Jiao et al., 2023). DiagSWin Transformers exhibit significant improvements across all settings and metrics

In Table 6, we take RetinaNet and Mask-RCNN for object detection in 1× schedule and report the box mAP (AP^b) and mask mAP (AP^m) of different CNN and Transformer backbones. Specifically, DiagSWin outperforms CNN-based models, with an improvement of 8.6% / 8.9% over ResNet-50/101 and 7.2% / 6.8% over ResNeXt101-32/64x4d. Compared to other methods that use transformer-based structures, DiagSWin Transformers achieves higher scores than the current best method CSWin Transformer with fewer parameters and computational complexity. Different from the other multi-scale Transformer models, DiagSWin Transformers can capture short- and long-range interactions for each token at different heads, and thus capture richer visual contexts at each layer for high-resolution tasks. We present some qualitative object detection and instance segmentation results on COCO val2017 in Fig. 5, which also shows good performance of our models.

To have more comprehensive comparisons, we train all models using the 3× schedule and show the detailed numbers for RetinaNet and Mask R-CNN in Table 7. Even with 3× schedules, DiagSWin-Tiny still achieves an 2.3% gain over Swin-Tiny but only needs fewer parameters (39/48 M → 29/39 M) and computational cost (245/264 G → 224/249 G FLOPs). We also achieve similar performance gain on small and base configuration. Moreover, when model size grows large, our base model achieves the best AP^b of 51.1%, surpassing CSWin-Base (51.1% vs. 50.8%), while our parameter number is 30% fewer.

4.4. Visualization results

Fig. 6 shows a visualization of the comparative results between the proposed DiagSwin and other SOTA methods. The first column represents the original image, blue region and boxes denote predictions by Focal-Base, pink region and boxes indicate predictions by CSWin-Base, and orange region and boxes show predictions by our DiagSWin-Base. It is evident that the predictions made by Focal and CSWin exhibit some missing details, particularly for small objects in the image and objects that are occluded. These models often lose visual attention in such cases. In contrast, DiagSWin shows significant improvement, which can be attributed to our better multi-scale window modeling of visual queries.

4.5. Ablation study

In this section, we conducted a series of ablation studies to scrutinize the model's capabilities from various angles. We report the results on image classification and object detection based on DiagSWin-Tiny.

Attention mechanism comparison. We compare DiagSWin attention with existing self-attention mechanisms. For a fair comparison, we use the Swin-Tiny as backbone and only change the self-attention mechanism. In detail, we use [2-2-6-2] blocks for the four stages, non-overlapped token embedding, and relative position bias. The results are reported in Table 8. Obviously, our DiagSWin attention mechanism performs better than existing self-attention mechanisms across all the tasks.

Different pooling methods. The ablation in Table 9 looks at the pooling operator (max/average/conv/fc). First, we observe that

Table 6
COCO detection and instance segmentation with RetinaNet (Lin, Goyal, Girshick, He, & Dollár, 2017) and Mask R-CNN (He et al., 2016). Models are trained for 1× schedule with multi-scale training inputs (MS). All backbones are pretrained on ImageNet-1K. The numbers before and after "/" at column 2 and 3 are the Parameter size and complexity for RetinaNet and Mask R-CNN, respectively.

Backbone	Params	FLOPs	RetinaNet 1× schedule + MS						Mask R-CNN 1× schedule + MS					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 (He et al., 2016)	38/44	239/260	36.3	55.3	38.6	19.3	40.0	48.8	38.0	58.6	41.4	34.4	55.1	36.7
ViL-Small (Dosovitskiy et al., 2020)	36/45	252/174	41.6	–	–	–	–	–	41.8	–	–	38.5	–	–
Swin-Tiny (Liu et al., 2021)	39/48	245/264	42.0	63.0	44.7	26.6	45.8	55.7	42.2	64.6	46.2	39.1	61.6	42.0
Focal-Tiny (Yang et al., 2021)	39/49	265/291	43.7	65.2	46.7	28.6	47.4	56.9	44.8	67.7	49.2	41.0	64.7	44.2
CSWin-Tiny (Dong et al., 2022)	−/42	−/279	–	–	–	–	–	–	46.7	68.6	51.3	42.2	65.6	45.4
PVT-Tiny (Wang, Xie, et al., 2021)	23/33	−/−	36.7	56.9	38.9	22.6	38.8	50.0	36.7	59.2	39.3	35.1	56.7	37.3
PVTv2-B1 (Wang et al., 2022)	24/34	−/−	41.2	61.9	43.9	25.4	44.5	54.3	41.8	64.3	45.9	38.8	61.2	41.6
MPViT-XS (Lee et al., 2022)	20/30	211/231	43.8	65.0	47.1	28.1	47.6	56.5	44.2	66.7	48.4	40.4	63.4	43.4
DiagSWin-Tiny	29/39	224/249	**45.2**	**65.3**	**48.9**	**30.3**	**49.4**	**58.6**	**46.8**	**68.8**	**51.7**	**42.4**	**65.9**	**46.0**
ResNet101 (He et al., 2016)	57/63	315/336	38.5	57.8	41.2	21.4	42.6	51.1	40.4	61.1	44.2	36.4	57.7	38.8
ResNeXt101-32 × 4d (Xie et al., 2017)	56/63	319/340	39.9	59.6	42.7	22.3	44.3	52.5	41.9	62.5	45.9	37.5	59.4	40.2
ViL-Medium (Dosovitskiy et al., 2020)	51/60	339/261	42.9	–	–	–	–	–	43.4	–	–	39.7	–	–
Swin-Small (Liu et al., 2021)	60/69	335/354	45.0	66.2	48.3	27.9	48.8	59.5	46.5	68.7	51.3	42.1	65.8	45.2
Focal-Small (Yang et al., 2021)	62/71	367/401	45.6	67.0	49.8	**31.7**	**50.4**	60.8	47.4	69.8	51.9	42.8	66.6	46.1
CSWin-Small (Dong et al., 2022)	−/54	−/342	–	–	–	–	–	–	47.9	70.1	52.6	43.2	**67.1**	46.2
PVT-Small (Wang, Xie, et al., 2021)	34/44	−/279	40.4	61.3	43.0	25.0	42.9	55.7	40.4	62.9	43.8	37.8	60.1	40.3
PVTv2-B2 (Wang et al., 2022)	35/45	−/−	44.6	65.6	47.6	27.4	48.8	58.6	45.3	**67.1**	49.6	41.2	64.2	44.4
MPViT-S (Lee et al., 2022)	32/43	248/268	45.7	57.3	48.8	28.7	49.7	59.2	46.4	68.6	51.2	42.4	65.6	45.7
Shunted-Small (Ren, Zhou, et al., 2022)	32/42	−/−	45.4	65.9	49.2	28.7	49.3	60.0	47.1	68.8	52.1	42.5	65.8	45.7
Dilate-Small (Jiao et al., 2023)	−/44	−/262	–	–	–	–	–	–	45.8	68.2	50.1	41.7	65.3	44.7
DiagSWin-Small	42/52	304/328	**47.1**	**67.2**	**50.4**	**31.7**	**50.4**	**61.8**	**48.3**	**70.2**	**53.2**	**43.3**	**67.1**	**46.4**
ResNeXt101-64 × 4d (Xie et al., 2017)	96/102	473/493	41.0	60.9	44.0	23.9	45.2	54.0	42.8	63.8	47.3	38.4	60.6	41.3
PVT-Large (Wang, Xie, et al., 2021)	71/81	345/364	43.4	63.6	46.1	26.1	46.0	59.5	44.5	66.0	48.3	40.7	64.4	43.7
ViL-Base (Dosovitskiy et al., 2020)	67/76	443/365	44.3	–	–	–	–	–	45.1	–	–	41.0	–	–
Swin-Base (Liu et al., 2021)	98/107	477/496	45.0	66.4	48.3	28.4	49.1	60.6	46.9	69.2	51.6	42.3	66.0	45.5
Focal-Base (Yang et al., 2021)	101/110	514/533	46.3	68.0	49.8	31.7	50.4	60.8	47.8	70.2	52.5	43.2	67.3	46.5
CSWin-Base (Dong et al., 2022)	−/97	−/526	–	–	–	–	–	–	48.7	**70.4**	53.9	43.9	67.8	**47.3**
PVT-Medium (Wang, Xie, et al., 2021)	54/64	−/−	41.9	63.1	44.3	25.0	44.9	57.6	42.0	64.4	45.6	39.0	61.6	42.1
PVTv2-B4 (Wang et al., 2022)	72/82	−/−	46.1	66.9	49.2	28.4	50.0	**62.2**	47.5	68.7	52.0	42.7	66.1	46.1
MPViT-B (Lee et al., 2022)	85/95	482/503	47.0	68.4	50.8	29.4	51.3	61.5	48.2	70.0	52.9	43.5	67.1	46.8
Shunted-Base (Ren, Zhou, et al., 2022)	−/59	−/−	–	–	–	–	–	–	48.0	69.8	53.3	43.2	66.9	46.8
Dilate-Base (Jiao et al., 2023)	−/67	−/370	–	–	–	–	–	–	47.6	70.2	**55.2**	43.4	67.2	46.8
DiagSWin-Base	57/67	346/367	**47.8**	**68.5**	**51.3**	**32.2**	**51.6**	62.1	**49.2**	70.3	54.0	**44.2**	**68.1**	**47.3**

max/average pooling achieves Top-1 accuracies of 82.4%/82.2% with the smallest number of parameters. Using conv achieves the highest Top-1 accuracy. Dilated convolution enlarges the receptive field but does not improve the results under the same training conditions. It is worth noting that regardless of the pooling operator used in DiagSWin, it consistently outperforms Swin in terms of accuracy. This reflects that the structure of DiagSWin is superior to Swin, rather than the choice of downsampling method.

Detail Feed-Forward. We introduced a detail feed-forward (DFF) layer in DiagSWin, composed of a residual structure formed by depth-wise separable convolution. We compared the DFF layer with the traditional feed-forward (FF) layer in ViT and the convolutional feed-forward (CFF) layer in PVTv2. As shown in Table 10, the DFF consistently brings performance gains, indicating that introducing local information in the feed-forward layer significantly improves its learning capability.

Effect of varying the aggregation size. We have demonstrated the superiority of the multi-scale structure of DiagSWin. Thus, a related question is whether increasing the window size would be helpful as it leads to a larger receptive field. In Fig. 7, we study the trade-off between window expand size and accuracy. We change the s_i of $r_i = 1$ in DiagSWin Transformer for parameter analysis. The results show that as the expand size increases, the computational cost (FLOPS) increases, and the Top-1 classification accuracy initially improves significantly but slows down when the window is sufficiently large. This gain is attributed to the excellent ability of our DiagSWin attention to model multi-scale relationships among visual cues. Our default setting of $s_i = [3, 3, 3, 7]$ achieves a good balance between accuracy and FLOPs.

5. Conclusion

In this paper, we have presented a new Vision Transformer architecture named DiagSWin Transformer. The core design of DiagSWin Transformer is the DiagSWin attention mechanism, which can attend to multi-scale objects on various-scale feature maps within one self-attention layer, enabling the vision Transformer to simultaneously obtain multi-scale attention at both fine-grained and coarse-grained levels. Meanwhile, DiagSWin attention alternately calculates self-attention

Table 7
COCO detection and instance segmentation with RetinaNet (Lin, Goyal, et al., 2017) and Mask R-CNN (He et al., 2016). Models are trained for 3× schedule with multi-scale training inputs (MS). All backbones are pretrained on ImageNet-1K. The numbers before and after "/" at column 2 and 3 are the Parameter size and complexity for RetinaNet and Mask R-CNN, respectively.

Backbone	Params	FLOPs	RetinaNet 3× schedule + MS						Mask R-CNN 3× schedule + MS					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet50 (He et al., 2016)	38/44	239/260	39.0	58.4	41.8	22.4	42.8	51.6	41.0	61.7	44.9	37.1	58.4	40.1
ViL-Small (Dosovitskiy et al., 2020)	36/45	252/174	42.9	63.8	45.6	27.8	46.4	56.3	43.4	64.9	47.0	39.6	62.1	42.4
Swin-Tiny (Liu et al., 2021)	39/48	245/264	45.0	65.9	48.4	29.7	48.9	58.1	46.0	68.1	50.3	41.6	65.1	44.9
Focal-Tiny (Yang et al., 2021)	39/49	265/291	45.5	66.3	48.8	31.2	49.2	58.7	47.2	69.4	51.9	42.7	66.5	45.9
CSWin-Tiny (Dong et al., 2022)	–/42	–/279	–	–	–	–	–	–	49.0	**70.7**	53.7	43.6	67.9	46.6
PVT-Tiny (Wang, Xie, et al., 2021)	23/33	–/–	39.4	59.8	42.0	25.5	42.0	52.1	39.8	62.2	43.0	37.4	59.3	39.9
MPViT-XS (Lee et al., 2022)	20/30	211/231	46.1	67.4	49.3	31.4	50.2	58.4	46.6	68.5	51.1	42.3	65.8	45.8
DiagSWin-Tiny	29/39	224/249	**47.3**	**67.7**	**50.3**	**32.7**	**51.5**	**60.1**	**49.2**	**70.7**	**54.1**	**43.7**	**68.0**	**47.2**
ResNet101 (He et al., 2016)	57/63	315/336	40.9	60.1	44.0	23.7	45.0	53.8	42.8	63.2	47.1	38.5	60.1	41.3
ResNeXt101-32 × 4d (Xie et al., 2017)	56/63	319/340	41.4	61.0	44.3	23.9	45.5	53.7	44.0	64.4	48.0	39.2	61.4	41.9
ViL-Medium (Dosovitskiy et al., 2020)	51/60	339/261	43.7	64.6	46.4	27.9	47.1	56.9	44.6	66.3	48.5	40.7	63.8	43.7
Swin-Small (Liu et al., 2021)	60/69	335/354	46.4	67.0	50.1	31.0	50.1	60.3	48.5	70.2	53.5	43.3	67.3	46.6
Focal-Small (Yang et al., 2021)	62/71	367/401	47.3	67.8	51.0	31.6	50.9	61.1	48.8	70.5	53.6	43.8	67.7	47.2
CSWin-Small (Dong et al., 2022)	–/54	–/342	–	–	–	–	–	–	50.0	71.3	54.7	44.5	68.4	47.7
PVT-Small (Wang, Xie, et al., 2021)	34/44	–/279	42.2	62.7	45.0	26.2	45.2	57.2	43.0	65.3	46.9	39.9	62.5	42.8
MPViT-S (Lee et al., 2022)	32/43	248/268	47.6	68.7	51.3	32.1	51.9	61.2	48.4	70.5	52.6	43.9	67.6	47.5
Shunted-Small (Ren, Zhou, et al., 2022)	32/42	–/–	46.4	66.7	50.4	31.0	51.0	60.8	49.1	70.6	53.8	43.9	67.8	47.5
Dilate-Base (Jiao et al., 2023)	–/67	–/370	–	–	–	–	–	–	49.0	70.9	53.8	43.7	67.7	46.9
DiagSWin-Small	42/52	304/328	**48.6**	**69.6**	**51.8**	**33.1**	**51.9**	**61.7**	**50.1**	**71.6**	**54.9**	**45.1**	**68.5**	**47.8**
ResNeXt101-64 × 4d (Xie et al., 2017)	96/102	473/493	41.8	61.5	44.4	25.2	45.4	54.6	44.4	64.9	48.8	39.7	61.9	42.6
PVT-Large (Wang, Xie, et al., 2021)	71/81	345/364	43.4	63.6	46.1	26.1	46.0	59.5	44.5	66.0	48.3	40.7	64.4	43.7
ViL-Base (Dosovitskiy et al., 2020)	67/76	443/365	44.7	65.5	47.6	29.9	48.0	58.1	45.7	67.2	49.9	41.3	64.4	44.5
Swin-Base (Liu et al., 2021)	98/107	477/496	45.8	66.4	49.1	29.9	49.4	60.3	48.5	69.8	53.2	43.4	66.8	46.9
Focal-Base (Yang et al., 2021)	101/110	514/533	46.9	67.8	50.3	31.9	50.3	61.5	49.0	70.1	53.6	43.7	67.6	47.0
CSWin-Base (Dong et al., 2022)	–/97	–/526	–	–	–	–	–	–	50.8	72.1	55.8	44.9	69.1	48.3
PVT-Medium (Wang, Xie, et al., 2021)	54/64	–/–	43.2	63.8	46.1	27.3	46.3	58.9	44.2	66.0	48.2	40.5	63.1	43.5
MPViT-B (Lee et al., 2022)	85/95	482/503	48.3	69.5	51.9	32.3	52.2	62.3	49.5	70.9	54.0	43.5	68.3	48.3
Shunted-Base (Ren, Zhou, et al., 2022)	–/59	–/–	–	–	–	–	–	–	50.1	70.9	54.1	45.2	68.0	48.0
Dilate-Base (Jiao et al., 2023)	–/67	–/370	–	–	–	–	–	–	49.9	71.9	55.1	44.5	68.9	47.7
DiagSWin-Base	57/67	346/367	**49.1**	**70.3**	**53.4**	**34.6**	**53.0**	**62.8**	**51.1**	**72.3**	**56.2**	**45.8**	**69.4**	**48.5**

Table 8
Comparison of different self-attention mechanisms.

Backbone	ImageNet-1K	COCO		ADE20K
	Top-1(%)	AP^b	AP^m	mIoU(%)
Sliding window	81.4	39.8	38.7	41.5
Sequential Axial	81.5	40.4	37.6	39.8
Criss-Cross	81.6	42.4	39.5	43.2
Shifted window	81.3	42.2	39.1	41.6
Cross-shaped window	82.2	43.4	40.2	43.4
Shunted window	81.9	45.7	41.9	47.2
Diagonal-shaped(Ours)	**83.4**	**47.5**	**42.8**	**48.5**

Table 9
Parameter size, FLOPs and Top-1 accuracy on ImageNet-1K of different pooling operators.

Pooling operator $\mathcal{P}(\cdot;r)$	#Param.	FLOPs(%)	Top-1(%)
max pooling layer	18.5M	3.2G	82.4
average pooling layer	18.5M	3.2G	82.2
convolutional layer	18.7M	3.4G	**82.9**
fully connected layer	18.5M	3.2G	82.6
dilated convolution layer	20.1M	3.4G	81.9

Table 10
Parameter size, FLOPs and Top-1 accuracy on ImageNet-1K of different feed-forward layers.

Layers	#Param.	FLOPs	Top-1(%)
Feed-Forward	18.5M	3.4G	82.5
Convolutional Feed-Forward	18.7M	3.4G	82.7
Detail Feed-Forward	18.7M	3.4G	**82.9**

within left and right diagonal-shaped windows, reducing computational cost while expanding visual attention. A comprehensive empirical study demonstrates that our DiagSWin Transformers surpass SOTA Vision Transformers in various vision tasks, including image classification, object detection, and segmentation. We are looking forward to applying it to more vision tasks.

Although DiagSWin has demonstrated superior performance compared to previous methods, manually designed sparse windows can still introduce additional biased information. We will further investigate how to generate adaptive window shape and size, thereby enhancing the generalization and scalability of the proposed method for future research.

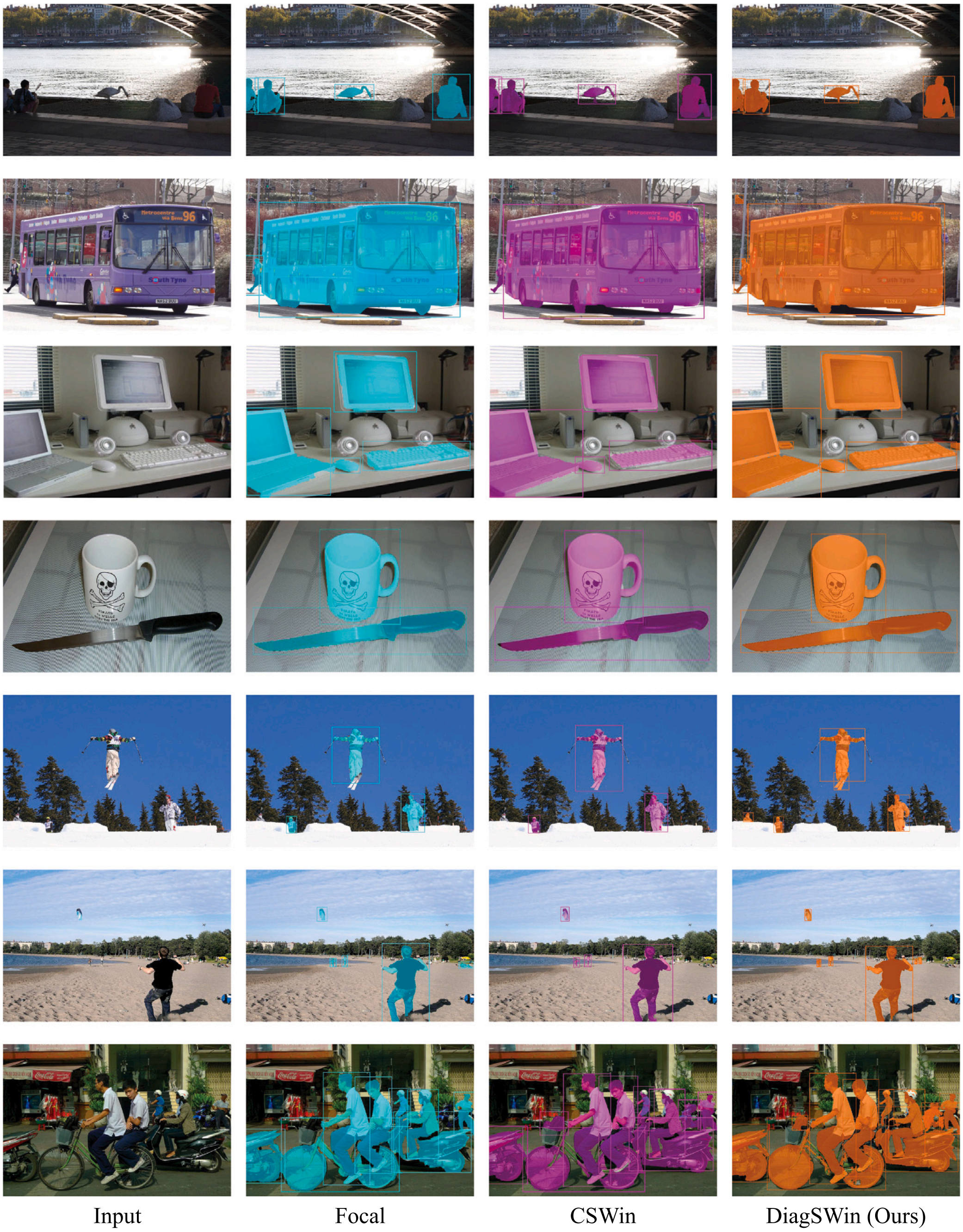


Fig. 6. Qualitative comparison of object detection and instance segmentation on COCO val2017. In each column from left to right: original images; result generated by Focal-Base, CSWin-Base and the proposed DiagSwin-Base.

CRediT authorship contribution statement

Ke Li: Investigation, Methodology, Writing – original draft. **Di Wang:** Conceptualization, Funding acquisition, Supervision, Writing – review & editing. **Gang Liu:** Writing – review & editing. **Wenxuan Zhu:** Software, Validation, Visualization. **Haodi Zhong:** Writing – review & editing. **Quan Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2022ZD0117103, in part by the National Natural Science Foundation of China under Grants 62072354, 62172222, 62402368 and 62072355, in part by the Fundamental Research Funds for the Central Universities under Grant QTZX23084, in part by the Natural Science Basic Research Program of Shaanxi under Grants 2024JC-YBQN-0732, 2024JC-YBQN-0340, 2024JC-YBQN-0715 and 2024GX-YBXM-039, in part by the Project of Collaborative Innovation Center of

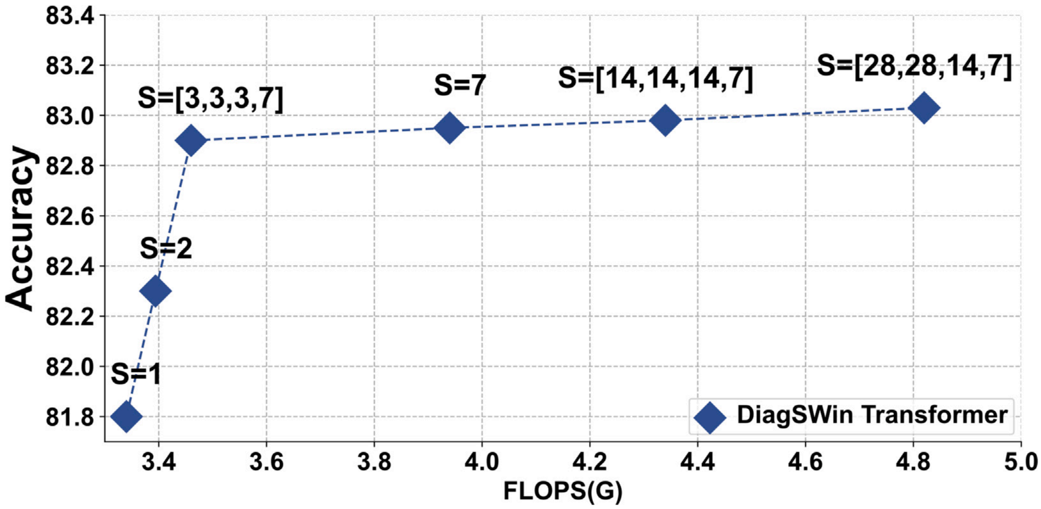


Fig. 7. Ablation on dynamic DiagSWin aggregation size.

Shaanxi Provincial Department of Education under grant 23JY038, in part by the Shaanxi Qinchuangyuan "Scientists+Engineers" team under Grant 2023KXJ-040, in part by the Research and Innovation Funds of Xidian University under Grant YJSJ24015, and in part by China Scholarship Council.

References

Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., & Ogden, J. M. (1984). Pyramid methods in image processing. *RCA Engineer*, *29*(6), 33–41.

Bracewell, R. N., & Bracewell, R. N. (1986). *The Fourier transform and its applications* (p. 31999). New York: McGraw-Hill.

Cai, H., Lan, L., Zhang, J., Zhang, X., Zhan, Y., & Luo, Z. (2023). Iouformer: Pseudo-iou prediction with transformer for visual tracking. *Neural Networks*.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. In *Computer vision–ECCV 2020: 16th European conference, Glasgow, UK, August 23–28, 2020, proceedings, part I 16* (pp. 213–229). Springer.

Chen, C.-F. R., Fan, Q., & Panda, R. (2021). Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 357–366).

Contributors, M. (2020). Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark.

Dai, Z., Cai, B., Lin, Y., & Chen, J. (2021). Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1601–1610).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). IEEE.

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., et al. (2022). Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12124–12134).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., et al. (2021). Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6824–6835).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hoffer, E., Ben-Nun, T., Hubara, I., Giladi, N., Hoefler, T., & Soudry, D. (2020). Augment your batch: Improving generalization through instance repetition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8129–8138).

Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., & Liu, W. (2019). Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 603–612).

Jiao, J., Tang, Y.-M., Lin, K.-Y., Gao, Y., Ma, J., Wang, Y., et al. (2023). Dilateformer: Multi-scale dilated transformer for visual recognition. *IEEE Transactions on Multimedia*.

Kirillov, A., Girshick, R., He, K., & Dollár, P. (2019). Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 6399–6408).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Lee, Y., Kim, J., Willette, J., & Hwang, S. J. (2022). Mpvit: Multi-path vision transformer for dense prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7287–7296).

Li, Q., Xie, X., Zhang, J., & Shi, G. (2023). Few-shot human–object interaction video recognition with transformers. *Neural Networks*, *163*, 1–9.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117–2125).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980–2988).

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., et al. (2014). Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, Zurich, Switzerland, September 6-12, 2014, proceedings, part V 13* (pp. 740–755). Springer.

Lin, X., Sun, S., Huang, W., Sheng, B., Li, P., & Feng, D. D. (2021). Eapt: efficient attention pyramid transformer for image processing. *IEEE Transactions on Multimedia*.

Lin, L., Yan, P., Xu, X., Yang, S., Zeng, K., & Li, G. (2021). Structured attention network for referring image segmentation. *IEEE Transactions on Multimedia*, *24*, 1922–1932.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Liu, J., Tan, H., Hu, Y., Sun, Y., Wang, H., & Yin, B. (2023). Global and local interactive perception network for referring image segmentation. *IEEE Transactions on Neural Networks and Learning Systems*.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Meer, P. (1989). Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, *45*(3), 269–294.

Polyak, B. T., & Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, *30*(4), 838–855.

Ren, S., Gao, Z., Hua, T., Xue, Z., Tian, Y., He, S., et al. (2022). Co-advise: Cross inductive bias distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16773–16782).

Ren, S., Zhou, D., He, S., Feng, J., & Wang, X. (2022). Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10853–10862).

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning* (pp. 6105–6114). PMLR.

Tolias, G., Sicre, R., & Jégou, H. (2015). Particular object retrieval with integral max-pooling of cnn activations. arXiv preprint arXiv:1511.05879.

Torrence, C., & Compo, G. P. (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, *79*(1), 61–78.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347–10357). PMLR.

Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., & Shlens, J. (2021). Scaling local self-attention for parameter efficient visual backbones. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 12894–12904).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., et al. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 568–578).

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., et al. (2022). Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, *8*(3), 415–424.

Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., et al. (2021). End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8741–8750).

Wen, Jie, Liu, Chengliang, Deng, Shijie, Liu, Yicheng, Fei, Lunke, Yan, Ke, et al. (2024). Deep double incomplete multi-view multi-label learning with incomplete labels and missing views. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(8), 11396–11408.

Wu, Zhihao, Liu, Chengliang, Wen, Jie, Xu, Yong, Yang, Jian, & Li, Xuelong (2023). Selecting high-quality proposals for weakly supervised object detection with bottom-up aggregated attention and phase-aware loss. *IEEE Transactions on Image Processing*, *32*, 682–693.

Wu, Zhihao, Wen, Jie, Xu, Yong, Yang, Jian, Li, Xuelong, & Zhang, David (2024). Enhanced spatial feature learning for weakly supervised object detection. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(1), 961–972.

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., et al. (2021). Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 22–31).

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., & Sun, J. (2018). Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision* (pp. 418–434).

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1492–1500).

Xu, Y., Zhang, Q., Zhang, J., & Tao, D. (2021). Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems*, *34*, 28522–28535.

Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., et al. (2021). Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, *34*, 30008–30022.

Zhang, N., Yu, L., Zhang, D., Wu, W., Tian, S., Kang, X., et al. (2023). Ct-net: Asymmetric compound branch transformer for medical image segmentation. *Neural Networks*.

Zheng, M., Gao, P., Zhang, R., Li, K., Wang, X., Li, H., et al. (2020). End-to-end object detection with adaptive clustering transformer. arXiv preprint arXiv:2011.09315.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., & Torralba, A. (2017). Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 633–641).