

DeBiFormer: Vision Transformer with Deformable Agent Bi-level Routing Attention

NguyenHuu BaoLong^{*1}, Chenyu Zhang^{*1}, Yuzhi Shi¹, Tsubasa Hirakawa¹,
Takayoshi Yamashita¹, Tohgoroh Matsui¹, and Hironobu Fujiyoshi¹

Chubu University, Kasugai, Japan

maclong01@gmail.com

{zhang1121, shi, hirakawa}@mprg.cs.chubu.ac.jp

{takayoshi, matsui, fujiyoshi}@isc.chubu.ac.jp

Abstract. Vision Transformers with various attention modules have demonstrated superior performance on vision tasks. While using sparsity-adaptive attention, such as in DAT, has yielded strong results in image classification, the key-value pairs selected by deformable points lack semantic relevance when fine-tuning for semantic segmentation tasks. The query-aware sparsity attention in BiFormer seeks to focus each query on top-k routed regions. However, during attention calculation, the selected key-value pairs are influenced by too many irrelevant queries, reducing attention on the more important ones. To address these issues, we propose the Deformable Bi-level Routing Attention (DBRA) module, which optimizes the selection of key-value pairs using agent queries and enhances the interpretability of queries in attention maps. Based on this, we introduce the Deformable Bi-level Routing Attention Transformer (DeBiFormer), a novel general-purpose vision transformer built with the DBRA module. DeBiFormer has been validated on various computer vision tasks, including image classification, object detection, and semantic segmentation, providing strong evidence of its effectiveness. Code is available at <https://github.com/maclong01/DeBiFormer>

Keywords: Vision Transformer, Self-Attention Mechanism, Image Recognition

1 Introduction

The Vision Transformer has recently demonstrated significant promise in the realm of computer vision [14,27,41]. It can capture long-range dependency in data [27,38], and is almost leading to a convolution-free model more flexible for fitting tons of data [41]. In addition, it enjoys high parallelism, which benefits training and inference for large models [10,38]. The computer vision community has observed a surge in the adoption and development of Vision Transformers [1,13,14,27,41,42].

* Equal contribution.

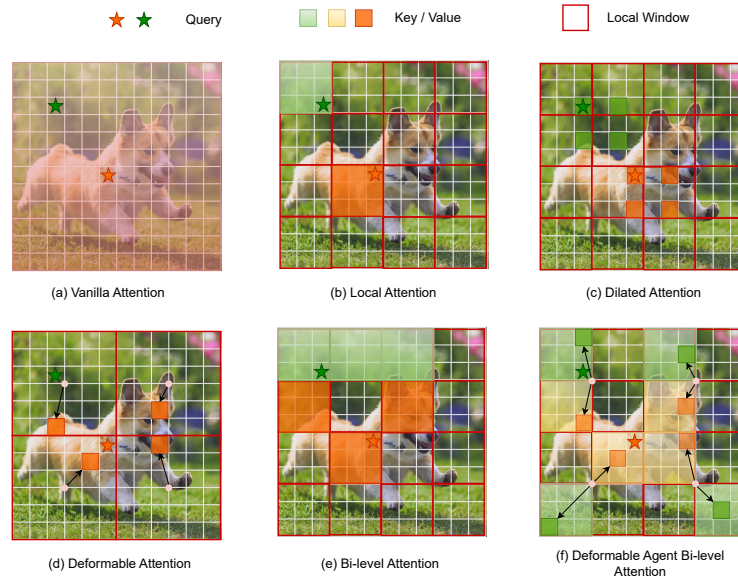


Fig. 1. Vanilla attention and its sparse variants are illustrated in the diagram: (a) vanilla attention functions globally, leading to increased computational complexity and substantial memory footprint. (b)-(c) Multiple strategies strive to reduce complexity by incorporating sparse attention with various handcrafted patterns, such as local window [47] and dilated window [42,37,23]. (d) Deformable attention [44] facilitates image-adaptive sparsity by deforming the regular grid. (e) Bi-level routing attention [53] begins by searching for top-k ($k = 3$ in this case) relevant regions and subsequently attends to the union of these regions. (f) In our approach, we realize bi-level routing attention, where the initial step involves searching for top-k ($k = 1$ in this case) relevant regions. Subsequently, attention is directed to the union of these regions by deforming regular grid attendance via top-k relevant regions.

To improve attention, numerous pieces of research have used thoughtfully crafted, efficient attention patterns in which each query selectively focused by a smaller portion of key-value pairs. As shown in Figure 1, among the various representation approaches, some include local windows [47] and dilated windows [42,37,23]. In addition, some research has taken a different path through sparsity adaptation to data in their methodology, as demonstrated in the works of [5,44]. However, despite the varying strategies for merging or selecting key and value tokens. These tokens are not semantic for queries. With this approach, when applied to other downstream tasks for pretrained ViT [38] and DETR [1]. Queries do not originate from semantic-region key-value pairs. Consequently, compelling all queries to focus on insufficient sets of tokens may not yield the most optimal results. Recently, with the dynamic query-aware sparsity attention mechanism, queries are focused by the most dynamic semantically key-value pairs, which is

referred to as bi-level routing attention [53]. However, in this approach, queries are handled by semantic key-value pairs instead of originating from detailed regions, which may not yield the most optimal results in all cases. In addition, when calculating the attention, these keys and values selected for all queries are influenced by too many less relevant queries, resulting in a decrease of attention for important queries, which has a significant impact when performing segmentations.[12,24].

To make the attention for queries more efficient, we propose the Deformable Bi-level Routing Attention (DBRA), an attention-in-attention architecture for visual recognition. During the process in DBRA, the first problem is how to locate deformable points. We use the observation in [44] that attention has an offset network that takes as input query features and generates corresponding offsets for all reference points. Thus, candidate deformable points are shifted towards important regions with high flexibility and efficiency to capture more informative features. The second problem is how to aggregate information from semantically relevant key-value pairs, and then broadcast the information back to queries. Therefore, we propose an attention-in-attention architecture that shifted toward deformable points as shown above acts as the agent for the queries. As the key-value pairs are selected for deformable points, we use the observation in [53] to select a small portion of the most semantically relevant key-value pairs that a region only needs by focusing on the top-k routed regions. Then, with the semantically relevant key-value pairs selected, we first apply a token-to-token attention with deformable points queries. And then, we apply a second token-to-token attention to broadcast the information back to queries, in which deformable points as key-value pairs are designed to represent the most important points in a portion of semantic regions.

To summarize, our contributions are as follows:

1. We propose Deformable Bi-level Routing Attention (DBRA), an attention-in-attention architecture for visual recognition, where data-dependent attention patterns are obtained flexibly and semantically.
2. By utilizing the DBRA module, we propose a novel backbone, called DeBiFormer, which has a stronger recognition ability based on the visualization results of the attention heat map.
3. Extensive experiments on ImageNet [32], ADE20K [52], and COCO [16] demonstrate that our model consistently outperforms other competitive baselines.

2 Related Work

2.1 Vision Transformers

The Transformer-based backbone incorporates channel-wise MLP [35] blocks to embed per-location features through channel mixing. Additionally, attention [38] blocks are used for cross-location relation modeling and facilitating spatial mixing. Initially devised for natural language processing [38,10], Transformers were

subsequently introduced to the domain of computer vision through works like DETR [1] and ViT [38]. Compared with CNNs, the primary distinction lies in the fact that transformers use attention as a substitute for convolution, thereby facilitating global context modeling. Nevertheless, vanilla attention, which calculates pairwise feature affinity across all spatial locations, imposes a significant computational burden and leads to substantial memory footprints, particularly when dealing with high-resolution inputs. Thus, a key research focus is to devise more efficient attention mechanisms, crucial for mitigating computational demands, especially with high-resolution inputs.

2.2 Attention mechanisms

Numerous studies have aimed to alleviate the computational and memory complexities associated with vanilla attention. Approaches include sparse connection patterns [6], low-rank approximations [39], and recurrent operations [9]. In the context of Vision Transformers, sparse attention has gained popularity, particularly following the remarkable success of the Swin Transformer [27]. Within the Swin Transformer framework, attention is constrained to non-overlapping local windows, and an innovative shift window operation is introduced. This operation facilitates communication between adjacent windows, contributing to its unique approach to handling attention mechanisms. To attain larger or approximate global receptive fields without exceeding computational constraints, recent studies have incorporated diverse manually designed sparse patterns. These include the integration of dilated windows [42,37,23] and cross-shaped windows [13]. Moreover, certain studies endeavor to make sparse patterns adaptable to data, as demonstrated by works like DAT [44], TCFormer [50], and DPT [5]. Despite their efforts to decrease the number of key-value tokens using diverse merging or selection strategies, it is crucial to recognize that these tokens lack semantic specificity. Instead, we reinforce query-aware key-value token selection.

Our work is motivated by an observation: semantically attentive regions for important queries can exhibit significant differences, as illustrated by visualizations from pre-trained models like ViT [38] and DETR [1]. In achieving query-adaptive sparsity through a coarse-to-fine approach, we propose an attention-inattention architecture which utilizes the deformable attention [44] with bi-level routing attention [53]. Diverging from deformable attention [44] and bi-level routing attention [53], our deformable bi-level routing attention aims to reinforce the most semantic and flexible key-value pairs. In contrast, bi-level routing attention only focuses on locating a few highly relevant key-value pairs, while deformable attention prioritizes identifying a few of the most flexible key-value pairs.

3 Our Approach: DeBiFormer

3.1 Preliminaries

Initially, we revisit the attention mechanism used in recent Vision Transformers. Taking a flattened feature map $x \in \mathbb{R}^{N \times C}$ as the input, a multi-head self-

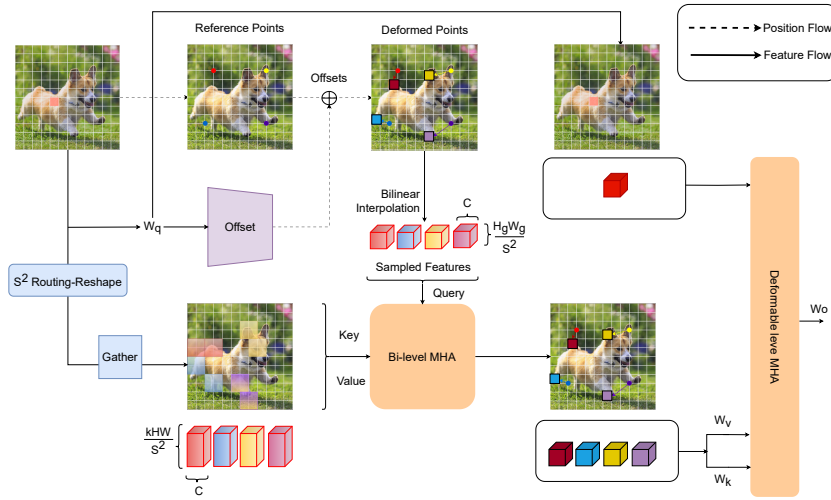


Fig. 2. Detailed architecture of Deformable Bi-level Routing Attention. In the top-left part, the set of reference points is uniformly distributed across the feature map. Offsets for these points are learned from queries through the offset network. Then, in the top-middle part, deformed features are projected from sampled features based on the locations of deformed points. In the bottom-left-middle part, we attend to projected deformed features by utilizing gathered key-value pairs in top-k-related windows.

attention (MHSA) block with M heads is formulated as

$$q = xW_q, k = xW_k, v = xW_v, \quad (1)$$

$$z^{(m)} = \sigma(q^{(m)}k^{(m)\top}/\sqrt{d})v^{(m)}, m = 1, \dots, M, \quad (2)$$

$$z = \text{Concat}(z^{(1)}, \dots, z^{(M)})W_o \quad (3)$$

where $\sigma(\cdot)$ denotes the softmax function, and $d = C/M$ is the dimension of each head. $z^{(m)}$ denotes the embedding output from the m -th attention head, and $q^{(m)}, k^{(m)}, v^{(m)} \in \mathbb{R}^{N \times d}$ denote the query, key, and value embeddings, respectively. $W_q, W_k, W_v, W_o \in \mathbb{R}^{C \times C}$ are the projection matrices. With normalization layers and identity shortcuts, the l -th Transformer block, for which LN means layer normalization, is formulated as

$$z'_l = \text{MHSA}(\text{LN}(z_{l-1})) + z_{l-1}, \quad (4)$$

$$z_l = \text{MLP}(\text{LN}(z'_l)) + z'_l \quad (5)$$

3.2 Deformable bi-level routing attention (DBRA)

The architecture of the proposed Deformable Bi-level Routing Attention (DBRA) is illustrated in Figure 2. We first employ a deformable attention module, which

includes an offset network that generates offsets for reference points based on the query features, creating deformable points. However, these points tend to cluster in important regions, leading to an over-concentration in certain areas.

To address this, we introduce deformable points-aware region partitioning, ensuring that each deformable point interacts with only a small subset of key-value pairs. Yet, solely relying on region partitioning can result in an imbalance between important and less important regions. To tackle this, the DBRA module is designed to distribute attention more effectively. In DBRA, each deformable point acts as an agent query, computing attention with semantic region key-value pairs. This approach ensures that only a few deformable points are assigned to each important region, allowing attention to be spread across all critical areas of the image rather than clustered in one spot.

By employing the DBRA module, attention is reduced in less important regions and increased in more important ones, ensuring a balanced distribution of attention throughout the image.

Deformable attention module and input projection. As illustrated in Fig.2, given the input feature map $x \in \mathbb{R}^{H \times W \times C}$, a uniform grid of points $p \in \mathbb{R}^{H_G \times W_G \times 2}$ is generated by downsampling the input feature map by factor r , $H_G = H/r$, $W_G = W/r$ as a reference. To obtain the offset for each reference point, the features are linearly projected to generate query tokens $q = xW_q$, which are then input into the $\theta_{offset}(\cdot)$ subnetwork to produce the offsets $\Delta p = \theta_{offset}(q)$. Subsequently, the features are sampled at the locations of deformed points as keys and values and further processed by projection matrices:

$$q = xW_q, \Delta p = \theta_{offset}(q), \bar{x} = \varphi(x; p + \Delta p), \quad (6)$$

where \bar{x} represent the deformed key \bar{k} and value \bar{v} embeddings, respectively. Specifically, we set the sampling function $\varphi(\cdot; \cdot)$ to a bilinear interpolation to make it differentiable:

$$\varphi(z; (p_x, p_y)) = \sum_{r_x, r_y} g(p_x, r_x)g(p_y, r_y)z[r_y, r_x, :], \quad (7)$$

where the function $g(a, b) = \max(0, 1 - |a - b|)$ and (r_x, r_y) represent indices for all locations on $z \in \mathbb{R}^{H \times W \times C}$. In a similar setup as deformable attention, where g is nonzero on the four integral points closest to (p_x, p_y) , Equation 7 is simplified to a weighted average across these four locations.

Region partition and region-to-region routing. With the deformable attention feature map input $\bar{x} \in \mathbb{R}^{H_G \times W_G \times C}$ and feature map $x \in \mathbb{R}^{H \times W \times C}$, the process begins by dividing it into regions of size $S \times S$ non-overlapped regions such that each region contains $\frac{H_G W_G}{S^2}$ feature vectors with reshaped \bar{x} as $\bar{x}^r \in \mathbb{R}^{S^2 \times \frac{H_G W_G}{S^2} \times C}$ and x as $x^r \in \mathbb{R}^{S^2 \times \frac{H W}{S^2} \times C}$. Then, we derive the query, key, and value with linear projections:

$$\hat{q} = \bar{x}^r W_q, \hat{k} = x^r W_k, \hat{v} = x^r W_v. \quad (8)$$

Next, we use the region-to-region method, as introduced in BiFormer [53], which is applied to establish the attending relationship by constructing a directed

graph. To initiate the process, region queries and keys $\hat{q}^r, \hat{k}^r \in \mathbb{S}^{S^2 \times C}$ are derived through the application of per-region averaging. Subsequently, the adjacency matrix $A^r \in \mathbb{S}^2 \times \mathbb{S}^2$ for the region-to-region affinity graph is derived through Q^r and K^{r^\top} matrix multiplication:

$$A^r = \hat{q}^r (\hat{k}^r)^\top, \quad (9)$$

where adjacency matrix A^r quantifies the semantic relationship between two regions. The crucial step in this method involves pruning the affinity graph by retaining only the *topk* connections for each region with a routing index matrix $I^r \in \mathbb{N}^{S^2 \times k}$ through the use of the *topk* operator:

$$I^r = \text{topk}(A^r). \quad (10)$$

Bi-level token to deformable-level token attention. Utilizing the region routing matrix I^r , we can then apply token attention. For each deformable query token within region i , its attention spans all key-value pairs located in the *topk* routed regions, that is, those indexed by $I_{i,1}^r, I_{i,2}^r, \dots, I_{i,k}^r$. Hence, we continue the process of gathering the key and value:

$$\hat{k}^g = \text{gather}(\hat{k}, I^r), \hat{v}^g = \text{gather}(\hat{v}, I^r), \quad (11)$$

where $\hat{k}^g, \hat{v}^g \in \mathbb{R}^{S^2 \times \frac{kHW}{S^2} \times C}$ are the gathered key and value. Then, we apply attention on \hat{k}^g, \hat{v}^g as:

$$\hat{O} = \hat{x} + W_{o'}(\text{Attention}(\hat{q}, \hat{k}^g, \hat{v}^g) + \text{LCE}(\hat{v})), \quad (12)$$

$$O = \text{MLP}(\text{LN}(\hat{O})) + \hat{O}, \quad (13)$$

where $W_{o'}$ is a projection weight for output features, and $\text{LCE}(\cdot)$ uses a kernel size 5 depth-wise convolution.

Deformable-level token to token attention. Following that, the deformable features that are semantically attended to via [53] are reshaped O as $O^r \in \mathbb{R}^{H_G \times W_G \times C}$ and parameterized at the locations of keys and values:

$$k = O^r W_k, v = O^r W_v, \quad (14)$$

k and v represent the embeddings of semantically deformed keys and values, respectively. Using existing approaches, we perform self-attention on q, k, v , and relative position offsets R . The output of attention is formulated as follows:

$$z^m = W_o(\sigma(q^m k^{(m)\top} / \sqrt{d} + \phi(\hat{B}; R))v^m). \quad (15)$$

Here, $\phi(\hat{B}; R) \in \mathbb{R}^{HW \times H_G W_G}$ corresponds to the position embedding, following the approach of previous work [27]. Then z^m is projected through W_o to get the final output z as Equation 3.

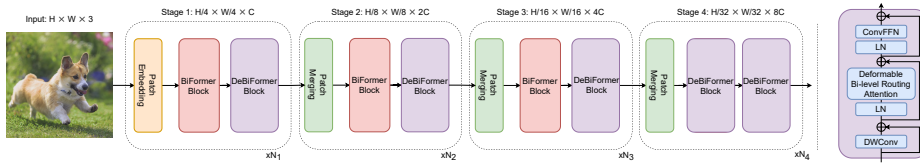


Fig. 3. Overall model architecture of our DeBiFormer. **Left:** Network architecture of DeBiFormer. N_1 to N_4 represent numbers of stacked successive local and Deformable Bi-level Routing Attention blocks. Please consult Table 1 for specific configurations. **Right:** Details on DeBiFormer Block.

| Variant Architectures of DeBiFormer | | | |
|-------------------------------------|---|---|--|
| | DeBi-T | DeBi-S | DeBi-B |
| Stage 1 56×56 | $N_1=2, C=64$ $r=8, M=2, D_r=3$ $G=1, K=9, B_r=3$ | $N_1=4, C=64$ $r=8, M=2, D_r=3$ $G=1, K=9, B_r=3$ | $N_1=4, C=96$ $r=8, M=3, D_r=3$ $G=1, K=9, B_r=3$ |
| Stage 2 28×28 | $N_2=2, C=128$ $r=4, M=4, D_r=3$ $G=2, K=7, B_r=3$ | $N_2=4, C=128$ $r=4, M=4, D_r=3$ $G=2, K=7, B_r=3$ | $N_2=4, C=192$ $r=4, M=6, D_r=3$ $G=2, K=7, B_r=3$ |
| Stage 3 14×14 | $N_3=8, C=256$ $r=2, M=8, D_r=3$ $G=4, K=5, B_r=3$ | $N_3=18, C=256$ $r=2, M=8, D_r=3$ $G=4, K=5, B_r=3$ | $N_3=18, C=384$ $r=2, M=12, D_r=3$ $G=4, K=5, B_r=3$ |
| Stage 4 7×7 | $N_4=2, C=512$ $r=1, M=16, D_r=3$ $G=8, K=3, B_r=3$ | $N_4=6, C=512$ $r=1, M=16, D_r=1$ $G=8, K=3, B_r=2$ | $N_4=4, C=768$ $r=1, M=24, D_r=3$ $G=8, K=3, B_r=3$ |

Table 1. DeBiFormer model architecture specifications. N_i : Number of blocks at stage i . C : Base channels in each block. r : Downsample ratio of deformed points. M : Number of attention heads in DBRMHA. G : Number of offset groups in DBRMHA. D_r : Deformable level MLP expansion ratio. B_r : Bi level MLP expansion ratio. K : Kernel size of offset module.

3.3 Model architectures

Leveraging DBRA as a fundamental building block, we introduce a novel vision transformer called DeBiFormer. As depicted in Figure 3, we adhere to the recent state-of-the-art Vision Transformers [13,27,53,44], using a four-stage pyramid structure. In stage i , we utilize an overlapped patch embedding in the first stage and a patch merging module [25,31] in the second to fourth stages. This is done to decrease the input spatial resolution while increasing the number of channels. Subsequently, N_i consecutive DeBiFormer blocks are used to transform the features. Within each DeBiFormer block, we adhere to recent methodologies [25,37,53] by using a 3×3 depthwise convolution at the outset. This is done to implicitly encode relative position information. Following that, we sequentially use a DBRA module with a 2-ConvFFN module with an expansion ratio e for cross-location relation modeling and per-location embedding, respectively.

| Model | FLOPs (G) | Params (M) | Top-1 (%) | Model | FLOPs (G) | Params (M) | Top-1 (%) |
|---------------------|--------------|---------------|--------------|---------------------|--------------|---------------|--------------|
| ResNet-18 [19] | 1.8 | 11.7 | 69.8 | ConvNeXt-B [28] | 15.4 | 89 | 83.8 |
| PVTv2-b1 [39] | 2.1 | 13.1 | 78.7 | SLaK-S [26] | 9.8 | 55 | 83.8 |
| Shunted-T [31] | 2.1 | 11.5 | 79.8 | Twins-SVT-L [7] | 14.8 | 99 | 83.7 |
| QuadTree-B-b1 [34] | 2.3 | 13.6 | 80.0 | PVTv2-B5 [41] | 11.8 | 82 | 83.8 |
| BiFormer-T [53] | 2.2 | 13.1 | 81.4 | Swin-B [27] | 15.4 | 88 | 83.5 |
| Conv2Former-N [21] | 2.2 | 15.0 | 81.5 | Focal-B [46] | 16.4 | 90 | 84.0 |
| DeBiFormer-T | 2.6 | 21.4 | 81.9 | CSWin-B [13] | 15.0 | 78 | 84.2 |
| PVTv2-B3 [41] | 6.9 | 45 | 81.2 | Shunted-B [31] | 8.1 | 39 | 84.0 |
| Swin-T [27] | 4.5 | 29 | 81.3 | UniFormer-B [25] | 8.3 | 50.0 | 83.9 |
| CSWin-T [13] | 4.5 | 23 | 82.7 | ScalableViT-B [48] | 8.6 | 81 | 84.1 |
| DAT-T [44] | 4.6 | 29 | 82.0 | Slide-Swin-B [30] | 15.5 | 89.0 | 84.2 |
| CrossFormer-S [42] | 5.3 | 31 | 82.5 | DAT-S [44] | 9.0 | 50 | 83.7 |
| RegionViT-S+ [2] | 5.7 | 31 | 83.3 | QuadTree-B-b4 [34] | 11.5 | 64 | 84.1 |
| QuadTree-B-b3 [34] | 7.8 | 46 | 83.7 | CrossFormer-L [42] | 16.1 | 92 | 84.0 |
| MaxViT-T [37] | 5.6 | 31 | 83.6 | RegionViT-B+ [2] | 13.6 | 74 | 83.8 |
| InternImage-T [40] | 5.0 | 30 | 83.5 | InternImage-S [40] | 8.0 | 50 | 84.2 |
| MixFormer-B4 [4] | 3.6 | 35 | 83.0 | MixFormer-B6 [4] | 12.7 | 119 | 83.8 |
| BiFormer-S [53] | 4.5 | 26 | 83.8 | BiFormer-B [53] | 9.8 | 57 | 84.3 |
| UniRepLKNet-T [11] | 4.9 | 31 | 83.2 | UniRepLKNet-S [11] | 9.1 | 56 | 83.9 |
| DeBiFormer-S | 5.4 | 44 | 83.9 | DeBiFormer-B | 11.8 | 77 | 84.4 |

Table 2. Evaluating and comparing different backbones on ImageNet-1K on images with resolution of 224×224 .

DeBiFormer is instantiated in three distinct model sizes, achieved by scaling the network width and depth as outlined in Table 1. Each attention head comprises 32 channels, and we use a bi-level ConvFFN and deformable-level ConvFFN with an MLP expansion ratio of $e = 3$. For the BRA, we use $topk = 1, 4, 16, S^2$, and for the DBRA, we use $topk = 4, 8, 16, S^2$ for the four stages. Moreover, we set the region partition factor S to specific values: $S = 7$ for classification, $S = 8$ for semantic segmentation, and $S = 20$ for object detection tasks.

4 Experiments

We experimentally evaluated the effectiveness of our proposed DeBiFormer on various mainstream computer vision tasks, including image classification (Section 4.1), semantic segmentation (Section 4.2) and object detection and instance segmentation (Section 4.3). In our approach, we commence training from scratch on ImageNet-1K [32] for image classification. Subsequently, we fine-tune the pre-trained backbones on ADE20K [52] for semantic segmentation and on COCO [16] for object detection and instance segmentation. Furthermore, we perform an ablation study to confirm the efficacy of the proposed Deformable Bi-level Routing Attention and top-k choices of DeBiFormer (Section 4.4). Finally, in order to validate that the recognition ability and interpretability of our DeBiFormer, we visualize the attention map (Section 5).

| Backbone | Semantic-FPN mIoU(%) | UperNet mIoU(%) |
|---------------------|-------------------------|--------------------|
| Swin-T [27] | 41.5 | 44.5 |
| DAT-T [44] | 42.6 | 45.5 |
| CSWin-T [13] | 48.2 | 49.3 |
| RegionViT-S+ [2] | 45.3 | - |
| InternImage-T [40] | - | 47.9 |
| CrossFormer-S [42] | 46.0 | 47.6 |
| Uniformer-S [25] | 46.6 | 47.6 |
| Shunted-S [31] | 48.2 | 48.9 |
| BiFormer-S [53] | 48.9 | 49.8 |
| DeBiFormer-S | 49.2 | 50.0 |
| Swin-S [27] | - | 47.6 |
| DAT-S [44] | 46.1 | 48.3 |
| CSWin-S [13] | 49.2 | 50.4 |
| RegionViT-B+ [2] | 47.5 | - |
| InternImage-S [40] | - | 50.1 |
| CrossFormer-B [42] | 47.7 | 49.7 |
| Uniformer-B [25] | 48.0 | 50.0 |
| BiFormer-B [53] | 49.9 | 51.0 |
| DeBiFormer-B | 50.6 | 51.4 |

Table 3. Evaluating DeBiFormer on semantic segmentation with two segmentation heads (Semantic FPN and UperNet) on ADE20K dataset.

4.1 Image classification on ImageNet-1K

Settings. We conducted image classification experiments on the ImageNet-1K [32] dataset, following the experimental settings of DeiT [36] for a fair comparison. Specifically, each model was trained for 300 epochs on 8 V100 GPUs with an input size of 224×224 . We used AdamW as the optimizer with a weight decay of 0.05 and used a cosine decay learning rate schedule with an initial learning rate of 0.001, while the first five epochs were used for linear warm-up. The batch size was set to 1024. To avoid overfitting, we used regularization techniques including RandAugment [8] (rand-m9-mstd0.5-inc1), MixUp [51] (prob = 0.8), CutMix [49] (prob = 1.0), Random Erasing (prob = 0.25), and increasing stochastic depth [22] (prob = 0.1/0.2/0.4 for DeBiFormer-T/S/B, respectively).

Results. We report our results in Table 2 showing the top-1 accuracy with similar computational complexities. Our DeBiFormer outperformed the Swin Transformer [27], PVT [41], DeiT [36], DAT[44], and Biformer [53] in all three scales. Without inserting convolutions in Transformer blocks or using overlapped convolutions in patch embeddings, DeBiFormer achieved gains of 0.5pt, 0.1pt and 0.1pt over BiFormer [53] counterparts.

4.2 Semantic segmentation on ADE20K

Settings. The same as existing works, we used our DeBiFormer on SemanticFPN [43] and UperNet [45]. In both cases, the backbone was initialized with

| Backbone | RetinaNet 1× schedule | | | | | | Mask R-CNN 1× schedule | | | | | |
|--------------------|-----------------------|-----------|-----------|--------|--------|-------------|------------------------|-------------|-------------|---------|-------------|-------------|
| | mAP | AP_{50} | AP_{75} | AP_S | AP_M | AP_L | mAP^b | AP_{50}^b | AP_{75}^b | mAP^m | AP_{50}^m | AP_{75}^m |
| Swin-T [27] | 41.5 | 62.1 | 44.2 | 25.1 | 44.9 | 55.5 | 42.2 | 64.6 | 46.2 | 39.1 | 61.6 | 42.0 |
| DAT-T [44] | 42.8 | 64.4 | 45.2 | 28.0 | 45.8 | 57.8 | 44.4 | 67.6 | 48.5 | 40.4 | 64.2 | 43.1 |
| CSWin-T [13] | - | - | - | - | - | - | 46.7 | 68.6 | 51.3 | 42.2 | 65.6 | 45.4 |
| RegionViT-S+ [2] | 43.9 | 65.5 | 47.3 | 28.5 | 47.3 | 58.0 | 44.2 | 67.3 | 48.2 | 40.9 | 64.1 | 44.0 |
| MixFormer-B4 [4] | - | - | - | - | - | - | 45.1 | 67.1 | 49.2 | 41.2 | 64.3 | 44.4 |
| CrossFormer-S [42] | 44.4 | 55.3 | 38.6 | 19.3 | 40.0 | 48.8 | 45.4 | 68.0 | 49.7 | 41.4 | 64.8 | 44.6 |
| QuadTree-B2 [34] | 46.2 | 67.2 | 49.5 | 29.0 | 50.1 | 61.8 | - | - | - | - | - | - |
| InternImage-T [40] | - | - | - | - | - | - | 47.2 | 69.0 | 52.1 | 42.5 | 66.1 | 45.8 |
| Agent-Swin-T [17] | - | - | - | - | - | - | 44.6 | 67.5 | 48.7 | 40.7 | 64.4 | 43.4 |
| BiFormer-S [53] | 45.9 | 66.9 | 49.4 | 30.2 | 49.6 | 61.7 | 47.8 | 69.8 | 52.3 | 43.2 | 66.8 | 46.5 |
| DeBiFormer-S | 45.6 | 66.6 | 48.9 | 28.7 | 49.3 | 61.6 | 47.5 | 69.7 | 52.1 | 42.5 | 66.2 | 45.7 |
| Swin-S [27] | 44.5 | 65.7 | 47.5 | 27.4 | 48.0 | 59.9 | 44.8 | 66.6 | 48.9 | 40.9 | 63.4 | 44.2 |
| DAT-S [44] | 45.7 | 67.7 | 48.5 | 30.5 | 49.3 | 61.3 | 47.1 | 69.9 | 51.5 | 42.5 | 66.7 | 45.4 |
| CSWin-S [13] | - | - | - | - | - | - | 47.9 | 70.1 | 52.6 | 43.2 | 67.1 | 46.2 |
| RegionViT-B+ [2] | 44.6 | 66.4 | 47.6 | 29.6 | 47.6 | 59.0 | 45.4 | 68.4 | 49.6 | 41.6 | 65.2 | 44.8 |
| CrossFormer-B [42] | 46.2 | 67.8 | 49.5 | 30.1 | 49.9 | 61.8 | 47.2 | 69.9 | 51.8 | 42.7 | 66.6 | 46.2 |
| QuadTree-B3 [34] | 47.3 | 68.2 | 50.6 | 30.4 | 51.3 | 62.9 | - | - | - | - | - | - |
| InternImage-S [40] | - | - | - | - | - | - | 47.8 | 69.8 | 52.8 | 43.3 | 67.1 | 46.7 |
| Agent-Swin-S [17] | - | - | - | - | - | - | 47.2 | 69.6 | 52.3 | 42.7 | 66.6 | 45.8 |
| BiFormer-B [53] | 47.1 | 68.5 | 50.4 | 31.3 | 50.8 | 62.6 | 48.6 | 70.5 | 53.8 | 43.7 | 67.6 | 47.1 |
| DeBiFormer-B | 47.1 | 68.2 | 50.2 | 30.3 | 51.1 | 63.0 | 48.5 | 70.2 | 53.3 | 43.2 | 67.2 | 46.4 |

Table 4. Results on object detection (left group) and instance segmentation (right group) tasks, performed on COCO 2017 dataset.

ImageNet-1K pretrained weights. The optimizer was AdamW [29], and the batch size was 32. For a fair comparison, we followed the same setting as PVT [41] to train the model with 80k steps and Swin Transformer [27] to train the model with 160k steps.

Results. Table 3 shows the results of the two different frameworks. It shows that with the Semantic FPN framework, our DeBiFormer-S/B achieved 49.2/50.6 mIoU, respectively, improving BiFormer by 0.3pt./0.7pt. A similar performance gain for the UperNet framework was also observed. By utilizing the DBRA module, our DeBiFormer could capture the most semantic key-value pairs, which makes the attention selection more reasonable and achieve higher performance on downstream semantic tasks.

4.3 Object detection and instance segmentation

Settings. We used our DeBiFormer as the backbone in the Mask RCNN [18] and RetinaNet [15] frameworks to evaluate the effectiveness of models for object detection and instance segmentation on COCO 2017 [16]. The experiments were conducted with the MMDetection [3] toolbox. Before training on COCO, we initialized the backbone with weights pre-trained on ImageNet-1K and followed the same training strategies as BiFormer [53] to compare our methods fairly. Note that due to device limitations, we set mini batch size as 4 for these experiments, while in BiFormer this value is 16. For details on the specific settings of the

| Sparse Attention | IN1K | ADE20K |
|------------------------------------|-------------|-------------|
| | Top1(%) | mIoU(%) |
| Shifted window [27] | 81.3 | 41.5 |
| Spatially sep [7] | 81.5 | 42.9 |
| Sequential axial [20] | 81.5 | 39.8 |
| Criss-cross [42] | 81.7 | 43.0 |
| Cross-shaped window [13] | 82.2 | 43.4 |
| Deformable [44] | 82.0 | 42.6 |
| Block-grid [37] | 81.8 | 42.8 |
| Bi-level routing [53] | 82.7 | 44.8 |
| Deformable bi-level routing | 82.9 | 48.0 |

Table 5. Ablation study on different attention mechanisms. All models follow the architecture design of the Swin-T model.

experiment, please refer to the supplementary paper.

Results. We list the results in Table 4. For object detection with RetinaNet, we report the mean average precision (mAP) and the average precision (AP) at different IoU thresholds (50%, 75%) for three object sizes (i.e., small, medium, and large (S/M/L)). From the results, we can see that while the overall performance of DeBiFormer was only comparable to some of the most competitive existing methods, the performance on large objects (AP_L) outperformed these methods although we use a limited resources. This may be because the DBRA allocates deformable points more reasonably. These points are not to focus only on small things, but to focus on important things in the image. Therefore the attention is not limited to a small area, which improves the detection accuracy of large objects. For instance segmentation with Mask R-CNN, we report the bounding box and mask the average precision (AP_b and AP_m) at different IoU thresholds (50%, 75%). Note that our DeBiFormer still achieved great performance under the device limitation of mini batch size. We believe that we could achieve better results if the mini batch size could be the same to other methods since it has been proved on semantic segmentation tasks.

4.4 Ablation study

Effectiveness of DBRA. We compared DBRA with several existing sparse attention mechanisms. Following CSWIN [13], we aligned macro architecture designs with Swin-T [27] for fair comparison. Specifically, we used 2, 2, 6, 2 blocks for the four stages and non-overlapped patch embedding, and we set the initial patch embedding dimension to $C = 96$ and MLP expansion ratio to $e = 4$. The results are reported in Table 5. Our Deformable Bi-level Routing Attention had significantly better performance than the existing sparse attention mechanisms, in terms of both image classification and semantic segmentation.

Partition factor S. Similar to BiFormer, we opted to use S as a divisor of the training size to prevent padding. We used an image classification with a resolution of $224 = 7 \times 32$, and we set $S = 7$ to ensure that the size of the feature

| #partition factor | k | Bi-level Routing tokens to attend | CoreML Latency(ms) | Top-1 Acc (%) |
|-------------------|------------|-----------------------------------|--------------------|---------------|
| 7 | 1,4,16,49 | 64, 64, 64, 49 | 291 | 81.81 |
| 7 | 2,8,32,49 | 128, 128, 128, 49 | 459 | 81.74 |
| 7 | 4,8,16,32 | 256, 128, 64, 32 | 276 | 81.47 |
| 7 | 8,16,32,49 | 512, 256, 128, 49 | 498 | 81.60 |
| 7 | 4,8,16,49 | 256, 128, 64, 49 | 382 | 81.90 |

Table 6. Ablation study on top-k selection. The four numbers in the k column represent the top-k values of the four stages and the same to the tokens to attend column. The CoreML Latency is conducted by MacBook Pro M1 using CPU and Neural Engine.

| Stage Configurations | | | | Backbone | | IN-1K |
|----------------------|-----------|-----------|-----------|----------|--------|--------------|
| 1 | 2 | 3 | 4 | FLOPs | #Param | Acc. |
| BB | BB | BB | BB | 2.2 | 13.1 | 81.37 |
| BB | BB | BB | B* | 2.4 | 15.8 | 81.58 |
| BB | BB | BB | ** | 2.6 | 18.5 | 81.63 |
| BB | BB | B* | ** | 2.7 | 21.2 | 81.84 |
| BB | B* | B* | ** | 2.7 | 21.4 | 81.87 |
| B* | B* | B* | ** | 2.6 | 21.4 | 81.90 |

Table 7. Configuration **B***: this stage is constructed with successive bi-level routing attention and DBRMHA blocks, while BB and ** denotes that stage uses the same blocks as bi-level routing attention and DBRMHA, respectively.

maps was divided at each stage. This choice aligns with the strategy used for the Swin Transformer [27], where a window size of 7 was used.

Top-k Choices. We systematically adjusted k to ensure a reasonable number of tokens attended to deformable queries as the region size diminished in later stages. Exploring various combinations of k is a viable option. In Table 6, we present ablation results on IN-1K, following DeBiFormer-STL (“STL” denotes Swin-T Layout). A crucial observation from these experiments is that augmenting the number of tokens paid attention to the deformable queries had a detrimental effect on accuracy and latency, and increasing the number of tokens paid attention in stages 1 and 2 had an effect on accuracy.

Deformable Bi-level Routing Multi-Head Attention (DBRMHA) at different stages. To evaluate the impact of design choices, we systematically replaced bi-level routing attention blocks with DBRMHA blocks across different stages, as shown in Table 7. Initially, all stages used bi-level routing attention, similar to BiFormer-T [53], achieving 81.3% accuracy in image classification. Replacing just one block in the 4th stage with DBRMHA immediately boosted accuracy by +0.21. Replacing all blocks in the 4th stage added another +0.05. Further DBRMHA replacements in the 3rd stage continued to improve performance across tasks. While gains tapered off with earlier stage replacements, we settled on a final version—DeBiFormer—where all stages use Deformable Bi-level Routing Attention for simplicity.

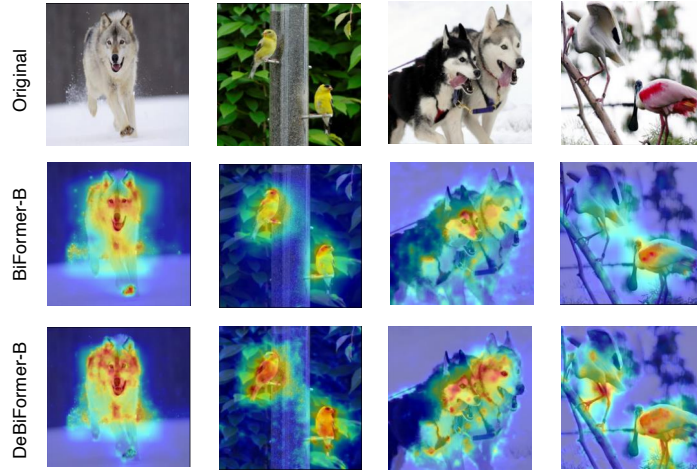


Fig. 4. Grad-CAM Visualization of BiFormer-Base and DeBiFormer-Base. These images are sampled from the validation set of ImageNet-1K.

5 Grad-CAM Visualization

To further illustrate the ability of the proposed DeBiFormer to recognize the attention in important regions, we used Grad-CAM [33] to visualize the areas of greatest concern of BiFormer-Base and DeBiFormer-Base. As shown in Figure 4, by utilizing DBRA module, our DeBiFormer-Base model performed better in target objects locating in which more regions have been focused on. In addition, our model scales down the attention in the unnecessary regions and pays more attention to the necessary regions. Depending on the attention of more necessary regions, our DeBiFormer model focused on semantic areas more continuously and completely, suggesting the stronger recognition ability of our model. Such ability yields better classification, and semantic segmentation performance compared with BiFormer-Base.

6 Conclusion

The paper introduces the Deformable Bi-level Routing Attention Transformer, a novel hierarchical vision transformer designed for both image classification and dense prediction tasks. Through Deformable Bi-level Routing Attention, our model optimizes query-key-value interactions while adaptively selecting semantically relevant regions. This leads to more efficient and meaningful attention. Extensive experiments show our model’s effectiveness compared to strong baselines. We hope this work offers insights into designing flexible and semantically aware attention mechanisms.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Proceedings of the IEEE/CVF European Conference on Computer Vision (ECCV) (2020) [1](#), [2](#), [4](#)
2. Chen, C.F., Panda, R., Fan, Q.: Regionvit: Regional-to-local attention for vision transformers. arXiv preprint arXiv:2106.02689 (2021) [9](#), [10](#), [11](#)
3. 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.: MMDetection: Open mmlab detection toolbox and benchmark. arXiv preprint arXiv:1906.07155 (2019) [11](#)
4. Chen, Q., Wu, Q., Wang, J., Hu, Q., Hu, T., Ding, E., Cheng, J., Wang, J.: Mix-former: Mixing features across windows and dimensions (2022) [9](#), [11](#)
5. Chen, Z., Zhu, Y., Zhao, C., Hu, G., Zeng, W., Wang, J., Tang, M.: Dpt: Deformable patch-based transformer for visual recognition. In: Proceedings of the ACM International Conference on Multimedia (2021) [2](#), [4](#)
6. Child, R., Gray, S., Radford, A., Sutskever, I.: Generating long sequences with sparse transformers. URL <https://openai.com/blog/sparse-transformers> (2019) [4](#)
7. 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. In: NeurIPS 2021 (2021), <https://openreview.net/forum?id=5kT1VBkzSRx> [9](#), [12](#)
8. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.: Randaugment: Practical automated data augmentation with a reduced search space. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 18613–18624. Curran Associates, Inc. (2020) [10](#)
9. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-xl: Attentive language models beyond a fixed-length context (2019) [4](#)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) [1](#), [3](#)
11. Ding, X., Zhang, Y., Ge, Y., Zhao, S., Song, L., Yue, X., Shan, Y.: Unireplknet: A universal perception large-kernel convnet for audio, video, point cloud, time-series and image recognition. arXiv preprint arXiv:2311.15599 (2023) [9](#)
12. Dong, K., Xue, J., Lan, X., Lu, K.: Biunet: Towards more effective unet with bi-level routing attention. In: The British Machine Vision Conference (November 2023) [3](#)
13. 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. In: Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR). pp. 12124–12134 (2022) [1](#), [4](#), [8](#), [9](#), [10](#), [11](#), [12](#)
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houshy, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021) [1](#)
15. Girshick, R., Radosavovic, I., Gkioxari, G., Dollár, P., He, K.: Focal loss for dense object detection. <https://github.com/facebookresearch/detectron> (2018) [11](#)
16. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019) [3](#), [9](#), [11](#)

17. Han, D., Ye, T., Han, Y., Xia, Z., Song, S., Huang, G.: Agent attention: On the integration of softmax and linear attention. arXiv preprint arXiv:2312.08874 (2023) [11](#)
18. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn (2017) [11](#)
19. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015) [9](#)
20. Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T.: Axial attention in multi-dimensional transformers (2019) [12](#)
21. Hou, Q., Lu, C.Z., Cheng, M.M., Feng, J.: Conv2former: A simple transformer-style convnet for visual recognition. arXiv preprint arXiv:2211.11943 (2022) [9](#)
22. Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K.: Deep networks with stochastic depth (2016) [10](#)
23. Jiao, J., Tang, Y.M., Lin, K.Y., Gao, Y., Ma, J., Wang, Y., Zheng, W.S.: Dilateformer: Multi-scale dilated transformer for visual recognition. *IEEE Transactions on Multimedia* (2023) [2, 4](#)
24. Lan, L., Cai, P., Jiang, L., Liu, X., Li, Y., Zhang, Y.: Brau-net++: U-shaped hybrid cnn-transformer network for medical image segmentation. arXiv preprint arXiv:2401.00722 (2024) [3](#)
25. Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatiotemporal representation learning (2022) [8, 9, 10](#)
26. Liu, S., Chen, T., Chen, X., Chen, X., Xiao, Q., Wu, B., Pechenizkiy, M., Mocanu, D., Wang, Z.: More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. arXiv preprint arXiv:2207.03620 (2022) [9](#)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2021) [1, 4, 7, 8, 9, 10, 11, 12, 13](#)
28. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 11976–11986 (June 2022) [9](#)
29. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2017) [11](#)
30. Pan, X., Ye, T., Xia, Z., Song, S., Huang, G.: Slide-transformer: Hierarchical vision transformer with local self-attention. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 2082–2091 (June 2023) [9](#)
31. Ren, S., Zhou, D., He, S., Feng, J., Wang, X.: Shunted self-attention via multi-scale token aggregation (2021) [8, 9, 10](#)
32. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* **115**(3), 211–252 (2015). <https://doi.org/10.1007/s11263-015-0816-y> [3, 9, 10](#)
33. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization (2016). <https://doi.org/10.1007/s11263-019-01228-7> [14](#)
34. Tang, S., Zhang, J., Zhu, S., Tan, P.: Quadtree attention for vision transformers. *ICLR* (2022) [9, 11](#)
35. Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Steiner, A., Keysers, D., Uszkoreit, J., Lucic, M., Dosovitskiy, A.: Mlp-mixer: An all-mlp architecture for vision. arXiv preprint arXiv:2105.01601 (2021) [3](#)

36. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jegou, H.: Training data-efficient image transformers distillation through attention. In: International Conference on Machine Learning. vol. 139, pp. 10347–10357 (July 2021) [10](#)
37. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxvit: Multi-axis vision transformer. ECCV (2022) [2](#), [4](#), [8](#), [9](#), [12](#)
38. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017) [1](#), [2](#), [3](#), [4](#)
39. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity (2020) [4](#), [9](#)
40. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., et al.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. arXiv preprint arXiv:2211.05778 (2022) [9](#), [10](#), [11](#)
41. 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. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 568–578 (2021) [1](#), [9](#), [10](#), [11](#)
42. Wang, W., Yao, L., Chen, L., Lin, B., Cai, D., He, X., Liu, W.: Crossformer: A versatile vision transformer hinging on cross-scale attention. In: Proceedings of the International Conference on Learning Representations (ICLR) (2022), https://openreview.net/forum?id=_PHymLIxuI [1](#), [2](#), [4](#), [9](#), [10](#), [11](#), [12](#)
43. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2. <https://github.com/facebookresearch/detectron2> (2019) [10](#)
44. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4794–4803 (June 2022) [2](#), [3](#), [4](#), [8](#), [9](#), [10](#), [11](#), [12](#)
45. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: European Conference on Computer Vision. Springer (2018) [10](#)
46. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal attention for long-range interactions in vision transformers. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 30008–30022. Curran Associates, Inc. (2021) [9](#)
47. Yang, J., Li, C., Zhang, P., Dai, X., Xiao, B., Yuan, L., Gao, J.: Focal self-attention for local-global interactions in vision transformers (2021) [2](#)
48. Yang, R., Ma, H., Wu, J., Tang, Y., Xiao, X., Zheng, M., Li, X.: Scalablevit: Rethinking the context-oriented generalization of vision transformer (2022) [9](#)
49. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: International Conference on Computer Vision (ICCV) (2019) [10](#)
50. Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11101–11111 (2022) [4](#)
51. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. International Conference on Learning Representations (2018), <https://openreview.net/forum?id=r1Ddp1-Rb> [10](#)
52. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019) [3](#), [9](#)

53. Zhu, L., Wang, X., Ke, Z., Zhang, W., Lau, R.: Biformer: Vision transformer with bi-level routing attention. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) [2](#), [3](#), [4](#), [6](#), [7](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#)