TransXNet: Learning Both Global and Local Dynamics With a Dual Dynamic Token Mixer for Visual Recognition

Meng Lou, Shu Zhang, Hong-Yu Zhou, *Member, IEEE*, Sibei Yang, Chuan Wu, *Fellow, IEEE*, and Yizhou Yu, *Fellow, IEEE*

***Abstract*—Recent studies have integrated convolutions into transformers to introduce inductive bias and improve generalization performance. However, the static nature of conventional convolution prevents it from dynamically adapting to input variations, resulting in a representation discrepancy between convolution and self-attention as self-attention calculates attention matrices dynamically. Furthermore, when stacking token mixers that consist of convolution and self-attention to form a deep network, the static nature of convolution hinders the fusion of features previously generated by self-attention into convolution kernels. These two limitations result in a suboptimal representation capacity of the constructed networks. To find a solution, we propose a lightweight dual dynamic token mixer (D-Mixer) to simultaneously learn global and local dynamics, that is, mechanisms that compute weights for aggregating global contexts and local details in an input-dependent manner. D-Mixer works by applying an efficient global attention module and an input-dependent depthwise convolution separately on evenly split feature segments, endowing the network with strong inductive bias and an enlarged effective receptive field. We use D-Mixer as the basic building block to design TransXNet, a novel hybrid CNN–transformer vision backbone network that delivers compelling performance. In the ImageNet-1K image classification task, TransXNet-T surpasses Swin-T by 0.3% in top-1 accuracy while requiring less than half of the computational cost. Furthermore, TransXNet-S and TransXNet-B exhibit excellent model scalability, achieving top-1 accuracy of 83.8% and 84.6%, respectively, with reasonable computational costs. In addition, our proposed network architecture demonstrates strong generalization capabilities in various dense prediction tasks, outperforming other state-of-the-art networks while having lower computational costs. Code is publicly available at https://github.com/LMMMEng/TransXNet.**

***Index Terms*—Visual recognition, Vision Transformer, Dual Dynamic Token Mixer.**

Received 30 October 2023; revised 29 June 2024 and 1 January 2025; accepted 11 March 2025. Date of publication 3 April 2025; date of current version 4 June 2025. This work was supported in part by Hong Kong Research Grants Council through the Collaborative Research Fund under Project HKU C7004-22G. *(Meng Lou and Shu Zhang contributed equally to this work.) (Corresponding author: Yizhou Yu.)*

Meng Lou, Chuan Wu, and Yizhou Yu are with the School of Computing and Data Science, The University of Hong Kong, Hong Kong SAR, China (e-mail: loumeng@connect.hku.hk; cwu@cs.hku.hk; yizhouy@acm.org).

Shu Zhang is with the Artificial Intelligence Laboratory, Deepwise Healthcare, Beijing 100081, China (e-mail: zhangshu@deepwise.com).

Hong-Yu Zhou is with the Department of Biomedical Informatics, Harvard Medical School, Boston, CA 02115 USA, and also with the Department of Computer Science, The University of Hong Kong, Hong Kong SAR, China (e-mail: whuzhouhongyu@gmail.com).

Sibei Yang is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: yangsb@shanghaitech.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2025.3550979

I. Introduction

VISION Transformer (ViT) [1] has shown promising progress in computer vision using multihead self-attention (MHSA) to achieve long-range modeling. However, it does not inherently encode inductive bias as convolutional neural networks (CNNs), resulting in a relatively weak generalization ability [2], [3]. To address this limitation, Swin Transformer [4] introduces shifted window self-attention, which incorporates inductive bias and reduces the computational cost of MHSA. However, Swin Transformer has a limited receptive field due to the local nature of its window-based attention.

To enable vision transformers to possess inductive bias, many previous works [5], [6], [7], [8], [9], [10] have constructed hybrid networks that integrate self-attention and convolution within token mixers. However, the utilization of standard convolutions in these hybrid networks leads to limited performance improvements despite the presence of inductive bias. The reason is twofold. First, unlike self-attention which dynamically calculates attention matrices when given an input, standard convolution kernels are input-independent and unable to adapt to different inputs. This results in a discrepancy in representation capacity between convolution and self-attention. This discrepancy dilutes the modeling capability of self-attention and existing hybrid token mixers. Second, the existing hybrid token mixers face challenges in deeply integrating convolution and self-attention. As a model goes deeper by stacking multiple hybrid token mixers, self-attention is capable of dynamically incorporating features generated by convolution in the preceding blocks while the static nature of convolution prevents it from effectively incorporating and using features previously generated by self-attention. In this work, we aim to design an input-dependent dynamic convolution mechanism that is well-suited for deep integration with self-attention within a hybrid token mixer so as to overcome the aforementioned challenges, resulting in a stronger feature representation capacity of the entire network.

On the other hand, a network should also have a large receptive field along with inductive bias to capture abundant contextual information. To this end, we obtain an interesting insight

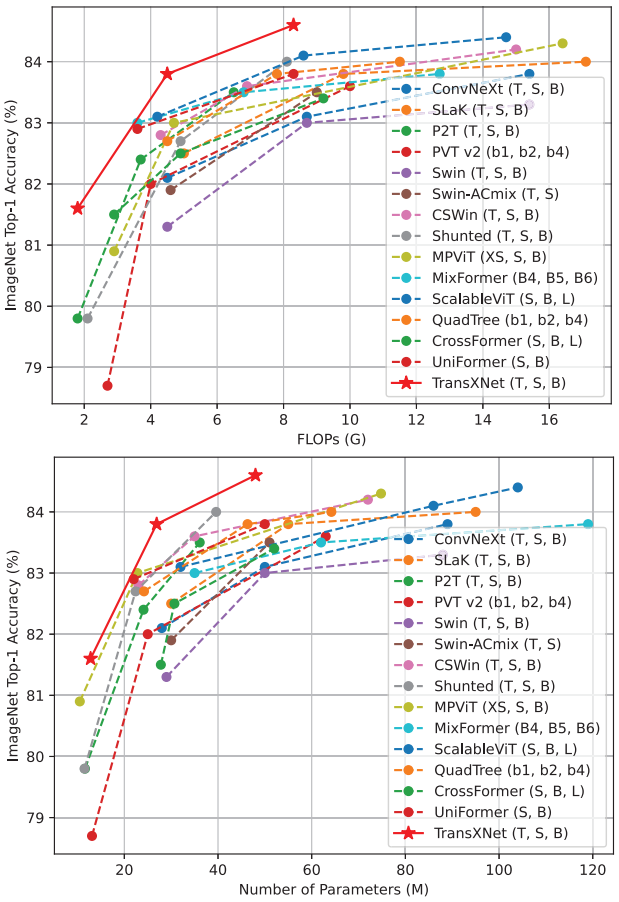

Fig. 1. Comparison of top-1 accuracy on the ImageNet-1K dataset with recent SOTA methods. Our proposed TransXNet model achieves superior performance compared with existing approaches.

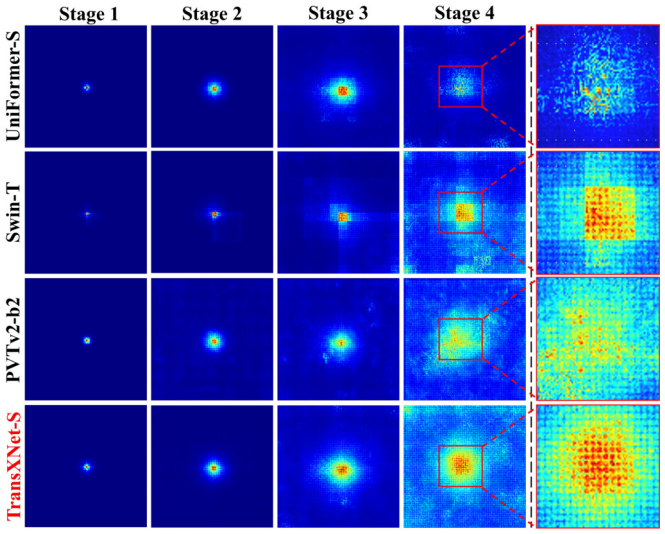

Fig. 2. Visualization of effective receptive fields (ERFs). The results are obtained by averaging over 100 images from ImageNet-1K.

through effective receptive field (ERF) [11] analysis: leveraging global self-attention across all the stages can effectively enlarge a model's ERF. Specifically, we visualize the ERF of three representative networks with similar computational cost, including UniFormer-S [12], Swin-T [4], and PVTv2-b2 [13]. Given a 224 × 224 input image, UniFormer-S and Swin-T exhibit locality at shallow stages and capture global information at the deepest stage, while PVTv2-b2 enjoys global information throughout the entire network. The results in Fig. 2 indicate that while all the three networks use global attention in the deepest layer, the ERF of PVTv2-b2 is clearly larger than that of UniFormer-S and Swin-T. According to this observation, to encourage a large receptive field, an efficient global self-attention mechanism should be encapsulated into all the stages of a network. We also empirically find out that integrating dynamic convolutions with global self-attention can further enlarge the receptive field.

On the basis of the above discussions, we introduce a novel dual dynamic token mixer (D-Mixer) to learn both global and local dynamics, namely, mechanisms that compute weights for aggregating global and local features in an input-dependent way. Specifically, the input features are split into two half segments, which are, respectively, processed by an overlapping spatial reduction attention (OSRA) module and an input-dependent depthwise convolution (IDConv). The resulting two outputs are then concatenated together. Such a simple design can make a network see global contextual information while injecting effective inductive bias. As shown in Fig. 2, our method stands out among its competitors, yielding the largest ERF. In addition, zoom-in views (last column) reveal that our proposed mixer has remarkable local sensitivity in addition to nonlocal attention. We further introduce a multiscale feedforward network (MS-FFN) that explores multiscale information during token aggregation. By hierarchically stacking basic blocks composed of a D-Mixer and an MS-FFN, we construct a versatile backbone network called TransXNet for visual recognition. As illustrated in Fig. 1, our method showcases superior performance when compared with recent state-of-the-art (SOTA) methods in ImageNet-1K [14] image classification. In particular, our TransXNet-T achieves 81.6% top-1 accuracy with only 1.8 GFLOPs and 12.8 M Parameters (Params), outperforming Swin-T while incurring less than half of its computational cost. In addition, our TransXNet-S/B models achieve 83.8%/84.6% top-1 accuracy, surpassing the strong InternImage [15] while incurring less computational cost.

In summary, our main contributions include: First, we propose a novel token mixer called D-Mixer, which aggregates sparse global information and local details in an input-dependent way, giving rise to both large ERF and strong inductive bias. Second, we design a novel and powerful vision backbone called TransXNet using D-Mixer as its token mixer. Finally, we conduct extensive experiments on image classification, object detection, and semantic and instance segmentation tasks. The results show that our method outperforms previous methods while having lower computational cost, achieving SOTA performance.

II. Related Work

A. Convolutional Neural Networks

Throughout the field of computer vision, CNNs have emerged as the standard deep model. Modern CNNs abandon the classical 3 × 3 convolution kernel and gradually adopt a

model design centered on large kernels. For instance, ConvNeXt [16] uses 7×7 depthwise convolution (DWConv) as the network's building block. RepLKNet [17] investigates the potential of large kernels and further extends the convolution kernel to 31×31. SLaK [18] exploits the sparsity of convolution kernel and enlarges the kernel size beyond 51×51. ParC-Net [19] introduces a novel position-aware circular convolution, which achieves a global receptive field while generating location-sensitive features, while ParC-NetV2 [20] further enlarges the receptive field by introducing oversized convolutions and bifurcate gate unit. In addition, some works use gated convolutions to achieve input-dependent modeling, such as FocalNet [21], HorNet [22], VAN [23], MogaNet [24], and Conv2Former [25]. Recently, InternImage [15] proposes a large-scale vision foundation model that surpasses SOTA CNN- and transformer-based models using 3×3 deformable convolutions as the core token mixer.

B. Vision Transformer

Transformer was first proposed in the field of natural language processing [26], and it can effectively perform dense relationships among tokens in a sequence by adopting MHSA. To adapt computer vision tasks, ViT [1] split an image into many image tokens through patch embedding operation, and thus MHSA can be successfully used to model token-wise dependencies. However, vanilla MHSA is computationally expensive for processing high-resolution inputs, while dense prediction tasks such as object detection and segmentation generally require hierarchical feature representations to handle objects with different scales. To this end, many subsequent works adopted efficient attention mechanisms with pyramid architecture designs to achieve dense predictions, such as window attention [4], [27], [28], sparse attention [13], [29], [30], [31], [32], and cross-layer attention [33], [34].

C. CNN-Transformer Hybrid Networks

Since relatively weak generalization is caused by lacking inductive biases in pure transformers [2], [3], CNN-transformer hybrid models have emerged as a promising alternative that can leverage the advantages of both CNNs and transformers in vision tasks. A common design pattern for hybrid models is to use CNNs in the shallow layers and transformers in the deep layers [3], [12], [35], [36]. To further enhance representation capacity, several studies have integrated CNNs and transformers into a single building block [5], [6], [7], [8], [9], [10]. For example, GG-Transformer [6] proposes a dual-branch token mixer, where the glance branch uses a dilated self-attention module to capture global dependencies and the gaze branch leverages a DWConv to extract local features. Similarly, ACmix [7] combines DWConv and window self-attention layers within a token mixer. Moreover, MixFormer [8] introduces a bidirectional interaction module that bridges the convolution and self-attention branches, providing complementary cues. These hybrid CNN-transformer models have demonstrated the ability to effectively merge the strengths of both the paradigms, achieving notable results in various computer vision tasks.

D. Dynamic Weights

Dynamic weight is a powerful factor for the superiority of self-attention, enabling it to extract features dynamically according to the input, in addition to its long-range modeling capability. Similarly, dynamic convolution has been shown to be effective in improving the performance of CNN models [37], [38], [39], [40] by extracting more discriminative local features with input-dependent filters. Among these methods, Han et al. [40] have demonstrated that replacing the shifted window attention modules in Swin Transformer with dynamic depthwise convolutions achieves better results with lower computational cost.

Different from the aforementioned works, our proposed D-Mixer can model both local and global contexts in an input-dependent manner, allowing both convolution and self-attention layers to dynamically calculate convolutional kernels and attention maps, respectively, based on feature clues from preceding layers, thereby achieving both larger receptive fields and stronger inductive biases.

III. Method

A. Overview

As illustrated in Fig. 3, our proposed TransXNet adopts a hierarchical architecture with four stages, which is similar to many previous works [27], [32], [41]. Each stage consists of a patch embedding layer and several sequentially stacked blocks. We implement the first patch embedding layer using a 7×7 convolutional layer (stride = 4) followed by batch normalization (BN) [42], while the patch embedding layers of the remaining stages use 3×3 convolutional layers (stride = 2) with BN. Each block consists of a dynamic position encoding (DPE) [12] layer, a D-Mixer, and an MS-FFN. The basic building block of our TransXNet can be mathematically represented as

$$\begin{aligned} \mathbf{X} &= \text{DPE}(\mathbf{X}_{\text{in}}) \\ \mathbf{Y} &= \text{D-Mixer}(\text{Norm}_1(\mathbf{X})) + \mathbf{X} \\ \mathbf{Z} &= \text{MS-FFN}(\text{Norm}_2(\mathbf{Y})) + \mathbf{Y} \end{aligned} \tag{1}$$

where $\mathbf{X_{in}} \in \mathbb{R}^{C \times H \times W}$ refers to a input feature map, while $\text{DPE}(\cdot)$ is implemented by a residual 7×7 DWConv, i.e., $\text{DPE}(\mathbf{X}) = \text{DWConv}_{7\times7}(\mathbf{X}) + \mathbf{X}$. More details about D-Mixer and MS-FFN are elaborated below.

B. Dual Dynamic Token Mixer

To enhance the generalization ability of the transformer model by incorporating inductive biases, many previous methods have combined convolution and self-attention to build a hybrid model [3], [7], [8], [9], [10], [12], [35], [36]. However, their static convolutions dilute the input dependency of transformers, i.e., although convolutions naturally introduce inductive bias, they have limited ability to improve the model's representation learning capability. In this work, we propose a lightweight token mixer termed D-Mixer, which dynamically leverages global and local information, injecting the potential of large ERF and strong inductive bias without compromising input dependency. The overall workflow of the

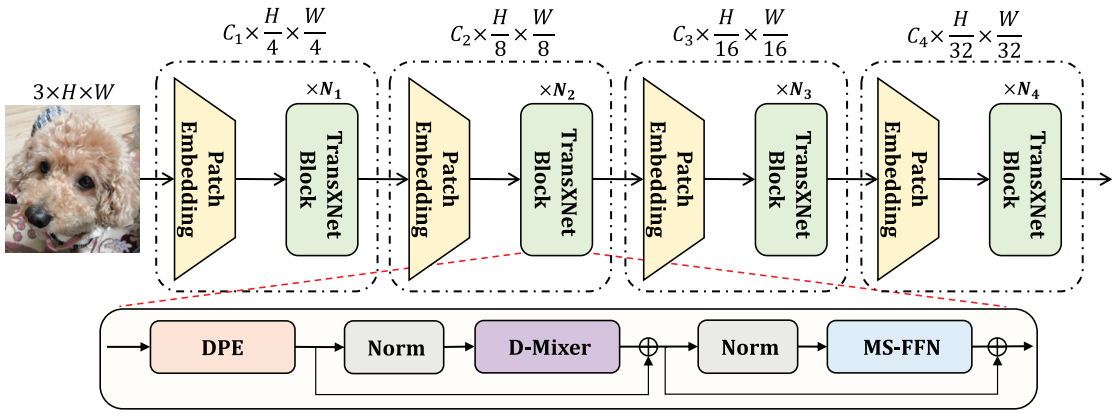

Fig. 3. Overall architecture of the proposed TransXNet.

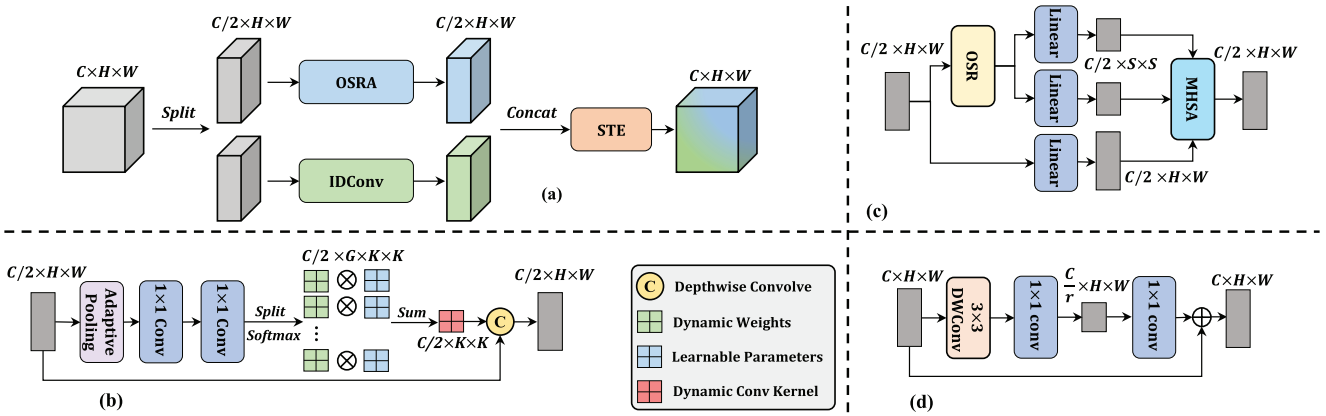

Fig. 4. Workflow of the proposed D-Mixer. (a) D-Mixer. (b) IDConv. (c) OSRA. (d) STE.

proposed D-Mixer is illustrated in Fig. 4(a). Specifically, for a feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, we first divide it uniformly along the channel dimension into two subfeature maps, denoted as $\{\mathbf{X_1}, \mathbf{X_2}\} \in \mathbb{R}^{(C/2)\times H\times W}$. Subsequently, $\mathbf{X_1}$ and $\mathbf{X_2}$ are, respectively, fed to a global self-attention module called OSRA and a dynamic depthwise convolution called IDConv, yielding corresponding feature maps $\{\mathbf{X_1'}, \mathbf{X_2'}\} \in \mathbb{R}^{(C/2)\times H\times W}$, which are then concatenated along the channel dimension to generate output feature map $\mathbf{X'} \in \mathbb{R}^{C\times H\times W}$. Finally, we use a squeezed token enhancer (STE) for efficient local token aggregation. Overall, the proposed D-Mixer is expressed as

$$\begin{aligned} \mathbf{X_1}, \mathbf{X_2} &= \text{Split}(\mathbf{X}) \\ \mathbf{X'} &= \text{Concat}(\text{OSRA}(\mathbf{X}_1), \text{IDConv}(\mathbf{X}_2)) \\ \mathbf{Y} &= \text{STE}(\mathbf{X'}). \end{aligned} \tag{2}$$

From the above equation, we can find out that by stacking D-Mixers, the dynamic feature aggregation weights generated in OSRA and IDConv take into account both global and local information, thus encapsulating powerful representation learning capabilities into the model.

1) Input-Dependent Depthwise Convolution: To inject inductive bias and perform local feature aggregation in a dynamic input-dependent way, we propose a new type of dynamic depthwise convolution, termed IDConv. As shown in Fig. 4(b), taking an input feature map $\mathbf{X} \in \mathbb{R}^{C\times H\times W}$, an adaptive average pooling layer is used to aggregate spatial contexts, compressing spatial dimension to K^2, which is then forwarded into two sequential 1×1 convolutions, yielding attention maps $\mathbf{A'} \in \mathbb{R}^{(G\times C)\times K^2}$, where G denotes the number of attention groups. Then, $\mathbf{A'}$ is reshaped into $\mathbb{R}^{G\times C\times K^2}$ and a softmax function is used over the G dimension, thus generating attention weights $\mathbf{A} \in \mathbb{R}^{G\times C\times K^2}$. Finally, $\mathbf{A}$ is elementwise multiplied with a set of learnable parameters $\mathbf{P} \in \mathbb{R}^{G\times C\times K^2}$, and the output is summed over the G dimension, resulting in IDConv kernels $\mathbf{W} \in \mathbb{R}^{C\times K^2}$, which can be expressed as

$$\begin{aligned} \mathbf{A'} &= \text{Conv}_{1\times 1}^{\frac{C}{r}\to(G\times C)}\left(\text{Conv}_{1\times 1}^{C\to\frac{C}{r}}\left(\text{AdaptivePool}\left(\mathbf{X}\right)\right)\right) \\ \mathbf{A} &= \text{Softmax}\left(\text{Reshape}\left(\mathbf{A'}\right)\right) \\ \mathbf{W} &= \sum_{i=0}^{G} \mathbf{P}_i \mathbf{A}_i. \end{aligned} \tag{3}$$

Since different inputs generate different attention maps $\mathbf{A}$, convolution kernels $\mathbf{W}$ vary with inputs. There are existing dynamic convolution schemes [39], [40]. In comparison to DyConv [39], IDConv generates a spatially varying attention map for every attention group and the spatial dimensions $(K \times K)$ of such attention maps exactly match those of convolution kernels while DyConv only generates a scalar attention weight for each attention group. Hence, our IDConv enables more dynamic local feature encoding. In comparison to the recently proposed D-DWConv [40], IDConv combines dynamic attention maps with static learnable parameters to significantly reduce com-

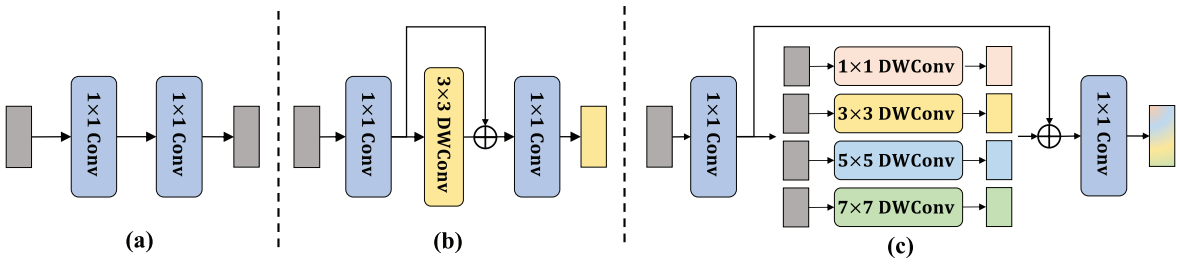

Fig. 5. (a) Vanilla FFN only handles cross-channel information. (b) Inverted Residual FFN further aggregates tokens in a small region. (c) Our MS-FFN performs multiscale token aggregations.

putational overhead. It is noted that D-DWConv applies global average pooling followed by channel squeeze-and-expansion pointwise convolutions on input features, resulting in an output with dimension $(C\times K^2)\times 1\times 1$, which is then reshaped to match the depthwise convolutional kernel. The number of Params incurred in this procedure is $(C^2/r)(K^2+1)$, while our IDConv results in $(C^2/r)(G+1)+GCK^2$ Params. In practice, when the maximum value of G is set to 4, and r and K are set to 4 and 7, respectively, the number of Params of IDConv $(1.25C^2+196C)$ is much smaller than that of D-DWConv $(12.5C^2)$.

2) Overlapping Spatial Reduction Attention: Spatial reduction attention (SRA) [30] has been widely used in previous works [9], [13], [31], [43] to efficiently extract global information by exploiting sparse token-region relationships. However, nonoverlapping spatial reduction (NSR) for reducing the token count breaks spatial structures near patch boundaries and degrades the quality of tokens. To address this issue, we introduce overlapping spatial reduction (OSR) for SRA to better represent spatial structures near patch boundaries using larger and overlapping patches. In practice, the OSR is instantiated as a strided DWConv, where the stride follows the setting of PVT [13], [30] and the kernel size equals the stride plus 3. For instance, in stage 1 of the network, the stride of OSR is 8, and thus OSR is a DWConv with a kernel size of 11 and stride of 8. As depicted in Fig. 4(c), the OSRA can be formulated as

$$\begin{aligned}\mathbf{Y}&=\mathrm{OSR}(\mathbf{X})\\ \mathbf{Q}&=\mathrm{Linear}(\mathbf{X})\\ \mathbf{K},\mathbf{V}&=\mathrm{Split}(\mathrm{Linear}(\mathbf{Y}+\mathrm{LR}(\mathbf{Y})))\\ \mathbf{Z}&=\mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}+\mathbf{B}\right)\mathbf{V}\end{aligned}\tag{4}$$

where $\mathrm{LR}(\cdot)$ denotes a local refinement module that is instantiated by a 3×3 DWConv, $\mathbf{B}$ is a relative position bias matrix that encodes the spatial relationships in attention maps [9], [36], and d is the number of channels in each attention head.

3) Squeezed Token Enhancer: After performing token mixing, most previous methods use a 1×1 convolution to achieve cross-channel communications, which incurs considerable computational overhead. To reduce the computational cost without compromising performance, we propose a lightweight STE, as shown in Fig. 4(d). STE comprises a 3×3 DWConv for enhancing local relationships, channel squeeze-and-expansion 1×1 convolutions for reducing the computational cost, and a residual connection for preserving the representation capacity. The STE can be expressed as follows:

$$\mathrm{STE}(\mathbf{X})=\mathrm{Conv}_{1\times 1}^{\frac{C}{r}\to C}\left(\mathrm{Conv}_{1\times 1}^{C\to\frac{C}{r}}\left(\mathrm{DWConv}_{3\times 3}(\mathbf{X})\right)\right)+\mathbf{X}.\tag{5}$$

According to the above equation, the FLOPs of STE are $HWC(2C/r+9)$. In practice, we set the channel reduction ratio r to 8, but ensure that the number of compressed channels is not less than 16, resulting in FLOPs significantly less than that of a 1×1 convolution, i.e., HWC^2.

C. Multiscale Feedforward Network

Compared with vanilla FFN [1], Inverted Residual FFN [9] achieves local token aggregation by introducing a 3×3 DWConv into the hidden layer. However, due to the larger number of channels in the hidden layer, i.e., typically four times the number of input channels, single-scale token aggregation cannot fully exploit such rich channel representations. To this end, we introduce a simple yet effective MS-FFN. As shown in Fig. 5, instead of using a single 3×3 DWConv, we use four parallel depthwise convolutions with different scales, each of which handles a quarter of the channels. The DWConv kernels with kernel size $=\{3,5,7\}$ can effectively capture multiscale information, while a 1×1 DWConv kernel is in fact a learnable channelwise scaling factor.

D. Architecture Variants

The proposed TransXNet has three different variants: TransXNet-T (Tiny), TransXNet-S (Small), and TransXNet-B (Base). To control the computational cost of different variants, there are two other adjustable hyperparameters in addition to the number of channels and blocks. First, since the computational cost of IDConv is directly related to the number of attention groups, we use a different number of attention groups in IDConv for different variants. In the tiny version, the number of attention groups is fixed at 2 to ensure a reasonable computational cost, while in the deeper small and base models, an increasing number of attention groups is used to improve the flexibility of IDConv, which is similar to the increase in the number of heads of the MHSA module as the model goes deeper. Second, many previous works [13], [30], [31], [32] set the expansion ratio of the FFNs in stages 1 and 2 to 8. However, since feature maps in stages 1 and 2 usually have larger resolutions, this leads to high FLOPs. Hence, we gradually increase the expansion ratio in different architecture variants. Details of different architecture variants are listed in Table I.

TABLE I

DETAILED CONFIGURATIONS OF TRANSXNET VARIANTS, INCLUDING STRIDE OF OSRA (S), NUMBER OF ATTENTION HEADS OF OSRA (H), KERNEL SIZE OF IDCONV (K), NUMBER OF ATTENTION GROUPS IN IDCONV (G), AND EXPANSION RATIO OF MS-FFN (E). FLOPS ARE CALCULATED WITH RESOLUTION 224 × 224

Input Size	Operator	TransXNet-T	TransXNet-S	TransXNet-B
224 × 224	Patch Embed	7×7, 48, stride=4	7×7, 64, stride=4	7×7, 76, stride=4
56 × 56	DPE D − Mixer MS − FFN	$\begin{bmatrix} S=8, H=1 \\ K=7, G=2 \\ E=4 \end{bmatrix} \times 3$	$\begin{bmatrix} S=8, H=1 \\ K=7, G=2 \\ E=6 \end{bmatrix} \times 4$	$\begin{bmatrix} S=8, H=2 \\ K=7, G=2 \\ E=8 \end{bmatrix} \times 4$
	Patch Embed	3×3, 96, stride=2	3×3, 128, stride=2	3×3, 152, stride=2
28 × 28	DPE D − Mixer MS − FFN	$\begin{bmatrix} S=4, H=2 \\ K=7, G=2 \\ E=4 \end{bmatrix} \times 3$	$\begin{bmatrix} S=4, H=2 \\ K=7, G=2 \\ E=6 \end{bmatrix} \times 4$	$\begin{bmatrix} S=4, H=4 \\ K=7, G=2 \\ E=8 \end{bmatrix} \times 4$
	Patch Embed	3×3, 224, stride=2	3×3, 320, stride=2	3×3, 336, stride=2
14 × 14	DPE D − Mixer MS − FFN	$\begin{bmatrix} S=2, H=4 \\ K=7, G=2 \\ E=4 \end{bmatrix} \times 9$	$\begin{bmatrix} S=2, H=5 \\ K=7, G=3 \\ E=4 \end{bmatrix} \times 12$	$\begin{bmatrix} S=2, H=8 \\ K=7, G=4 \\ E=4 \end{bmatrix} \times 21$
	Patch Embed	3×3, 448, stride=2	3×3, 512, stride=2	3×3, 672, stride=2
7 × 7	DPE D − Mixer MS − FFN	$\begin{bmatrix} S=1, H=8 \\ K=7, G=2 \\ E=4 \end{bmatrix} \times 3$	$\begin{bmatrix} S=1, H=8 \\ K=7, G=4 \\ E=4 \end{bmatrix} \times 4$	$\begin{bmatrix} S=1, H=16 \\ K=7, G=4 \\ E=4 \end{bmatrix} \times 4$
	Global Average Pooling			
1 × 1	Fully Connected Layer, 1000			
# FLOPs		1.8 G	4.5 G	8.3 G
# Params		12.8 M	26.9 M	48.0 M

IV. EXPERIMENTS

To assess the efficacy of our TransXNet, we evaluate it on various tasks, including image classification on the ImageNet-1K dataset [14], object detection and instance segmentation on the COCO dataset [44], and semantic segmentation on the ADE20K dataset [45]. In addition, we conduct extensive ablation studies to analyze the impact of different components of our model.

A. Image Classification

1) Setup: Image classification is performed on the ImageNet-1K dataset, following the experimental settings of DeiT [2] for a fair comparison with SOTA methods, i.e., all the models are trained for 300 epochs with the AdamW optimizer [46]. The stochastic depth rate [47] is set to 0.1/0.2/0.4 for tiny, small, and base models, respectively. After pretraining the base model on 224 × 224 inputs, we further fine-tune it on 384 × 384 inputs for 30 epochs to assess its performance when using high input image resolution. Furthermore, to demonstrate the generalizability of our method, we perform additional assessments on the ImageNet-V2 dataset [48] using ImageNet-pretrained weights, adhering to settings outlined in [49]. All the experiments are conducted on eight NVIDIA Tesla V100 GPUs.

2) Results: The proposed method outperforms other competitors in ImageNet-1K image classification with 224 × 224 images, as summarized in Table II. First, TransXNet-T achieves an impressive top-1 accuracy of 81.6% with only 1.8 GFLOPs and 12.8 M Params, surpassing other methods by a large margin. Despite having less than half of the computational cost, TransXNet-T achieves 0.3% higher top-1 accuracy than Swin-T [4]. Second, TransXNet-S achieves a remarkable top-1 accuracy of 83.8%, which is higher than InternImage-T [15] by 0.2% without requiring specialized CUDA implementations. Specifically, as the core operator of InternImage, DCNv3 relies on specialized CUDA implementations for accelerating on GPU, while our method can be more easily generalized to various devices without CUDA support. Moreover, our method outperforms well-known hybrid models, including MixFormer [8] and MaxViT [10], while having a lower computational cost. Notably, our small model performs better than MixFormer-B5 whose number of Params actually exceeds our base model. Note that the performance improvement of TransXNet-S over CMT-S [9] in image classification appears limited because CMT has a more complex classification head to boost performance. In contrast, benefiting from the stronger representation capacity of the backbone network, our method exhibits very clear advantages in downstream tasks including object detection and instance segmentation (see Section IV-B). Finally, TransXNet-B leads other methods by achieving an excellent balance between performance and computational cost, boasting a top-1 accuracy of 84.6%. However, it is worth highlighting that our method exhibits a more pronounced performance advantage on the ImageNet-V2 dataset. Specifically, TransXNet-T, -S, and -B achieve top-1 of 70.7%, 73.8%, and 75.0%, respectively. This demonstrates the superior generalization and transferability of our method compared with its counterparts. The experimental results on 384 × 384 input images are shown in Table III. With only about half of the FLOPs/Params, TransXNet-B significantly outperforms Swin-B and ConvNeXt-B [16], and its performance also surpasses that of CSWin-B [27]. In addition, compared with MaxViT-S, TransXNet-B exhibits a notable performance improvement while saving about 30% of the FLOPs/Params. These results demonstrate the strength of our method in processing higher resolution inputs.

TABLE II

QUANTITATIVE PERFORMANCE COMPARISONS OF IMAGE CLASSIFICATION WITH 224 × 224 INPUTS. #F AND #P DENOTE THE FLOPS AND NUMBER OF PARAMS OF A MODEL, RESPECTIVELY

Method	#F (G)	#P (M)	Top-1 (1K)	Top-1 (V2)
RSB-ResNet-18 [50]	1.8	11.7	70.6	-
RegNetY-1.6G [51]	1.6	11.2	78.0	66.2
PVT-ACmix-T [7]	2.0	13.0	78.0	-
PVTv2-b1 [13]	2.1	13.1	78.7	66.9
Shunted-T [31]	2.1	11.5	79.8	66.9
P2T-T [32]	1.8	11.6	79.8	70.0
QuadTree-B-b1 [52]	2.3	13.6	80.0	67.2
MPViT-XS [53]	2.9	10.5	80.9	70.0
TransXNet-T	1.8	12.8	**81.6**	**70.7**
ConvNeXt-T [16]	4.5	29.0	82.1	71.0
SLaK-T [18]	5.0	30.0	82.5	-
ParCNetV2-T [20]	4.3	25.0	83.5	-
Conv2Former-T [25]	4.4	27.0	83.2	-
InternImage-T [15]	5.0	30.0	83.5	73.0
DeiT-S [2]	4.6	22.0	79.8	68.5
Swin-T [4]	4.5	29.0	81.3	69.7
CSWin-T [27]	4.5	23.0	82.7	72.5
PVTv2-b2 [13]	4.0	25.4	82.0	71.8
Shunted-S [31]	4.9	22.4	82.9	72.4
UniFormer-S [12]	3.6	22.0	82.9	71.9
ScalableViT-S [54]	4.2	32.0	83.1	71.5
MOAT-0 [49]	5.7	27.8	83.3	72.8
BSwin-T [34]	4.5	29.0	82.0	-
Slide-Swin-T [28]	4.6	29.0	82.3	-
Slide-CSWin-T [28]	4.3	23.0	83.2	-
QuadTree-B-b2 [52]	4.5	24.2	82.7	70.4
MPViT-S [53]	4.7	22.8	83.0	72.3
CrossFormer-S [55]	4.9	30.7	82.5	71.5
CvT-13 [5]	4.5	20.0	81.6	70.5
CMT-S [9]	4.0	26.3	83.5	73.4
GG-Transformer-T [6]	4.5	28.0	82.0	-
MaxViT-T [10]	5.6	31.0	83.6	73.1
Swin-ACmix-T [7]	4.6	30.0	81.6	70.7
MixFormer-B4 [8]	3.6	35.0	83.0	-
TransXNet-S	4.5	26.9	**83.8**	**73.8**
ConvNeXt-S [16]	8.7	50.0	83.1	72.4
SLaK-S [18]	9.8	55.0	83.8	-
Conv2Former-S [25]	8.7	50.0	84.1	-
InternImage-S [15]	8.0	50.0	84.2	73.9
DeiT-B [2]	17.5	86.0	81.8	71.2
Swin-B [4]	15.4	88.0	83.5	72.3
CSWin-B [27]	15.0	78.0	84.2	74.1
PVTv2-b4 [13]	10.1	62.6	83.6	73.7
P2T-L [32]	9.8	54.5	83.9	72.7
Shunted-B [31]	8.1	39.6	84.0	73.9
UniFormer-B [12]	8.3	50.0	83.9	73.0
ScalableViT-B [54]	8.6	81.0	84.1	72.9
MOAT-1 [49]	9.1	41.6	84.2	74.2
BSwin-S [34]	8.7	50.0	84.2	-
Slide-Swin-B [28]	15.5	89.0	84.2	-
QuadTree-B-b4 [52]	11.5	64.2	84.1	72.7
MPViT-B [53]	16.4	74.8	84.3	73.7
CrossFormer-L [55]	16.1	92.0	84.0	73.5
GG-Transformer-S [6]	8.7	50.0	83.4	-
Swin-ACmix-S [7]	9.0	51.0	83.5	73.0
MixFormer-B5 [8]	6.8	62.0	83.5	-
TransXNet-B	8.3	48.0	**84.6**	**75.0**

TABLE III

QUANTITATIVE PERFORMANCE COMPARISONS OF IMAGE CLASSIFICATION WITH 384 × 384 INPUTS

Method	#F (G)	#P (M)	Top-1 (1K)	Top-1 (V2)
DeiT-B [2]	49.3	86.0	77.9	73.1
Swin-B [4]	47.1	88.0	84.5	73.2
ConvNeXt-B [16]	45.2	88.6	85.1	75.2
CSWin-B [27]	47.0	78.0	85.4	75.6
MaxViT-S [10]	36.1	69.0	85.2	-
TransXNet-B	24.2	48.0	**85.5**	**76.1**

B. Object Detection and Instance Segmentation

1) Setup: To evaluate our method on object detection and instance segmentation tasks, we conduct experiments on COCO 2017 [44] using the MMDetection[1] codebase. Specifically, for object detection, we use the RetinaNet framework [56], while instance segmentation is performed using the Mask R-CNN framework [57]. For fair comparisons, we initialize all the backbone networks with weights pretrained on ImageNet-1K, while training settings follow the 1× schedule provided by PVT [30].

2) Results: We present results in Table IV. For object detection with RetinaNet, our method attains the best performance in comparison to other competitors. It is noted that previous methods often fail to simultaneously perform well on both small and large objects. However, our method, supported by global and local dynamics and multiscale token aggregation, not only achieves excellent results on small targets but also significantly outperforms previous methods on medium and large targets. For example, the recently proposed Slide-PVTv2-b1 [28], which focuses on local information modeling, achieves comparable AP_S to our tiny model, while our method improves AP_M/AP_L by 1.9%/2.5% while having less computational cost, underscoring its effectiveness in modeling both global and local information. This phenomenon is more prominent in the comparison groups of small and base models, demonstrating the superior performance of our method across different object sizes. Regarding instance segmentation with Mask-RCNN, our method also has a clear advantage over previous methods with a comparable computational cost. It is worth mentioning that even though TransXNet-S shows limited performance improvement over CMT-S [9] in ImageNet-1K classification, it achieves obvious performance improvements in object detection and instance segmentation, which indicates that our backbone has stronger representation capacity and better transferability.

C. Semantic Segmentation

1) Setup: We conduct semantic segmentation on the ADE20K dataset [45] using the MMSegmentation[2] codebase. The commonly used Semantic FPN [60] is used as the segmentation framework. For fair comparisons, all the backbone networks are initialized with ImageNet-1K pretrained weights, and training settings follow PVT [13].

2) Results: Table V demonstrates the performance achieved by our method in comparison to other competitors. Note that since some methods (e.g., CMT [9] and MaxViT [10]) do not report semantic segmentation results in their papers, we do not compare with them. Specifically, our TransXNet-T achieves a remarkable 45.5% mIoU, surpassing the second-best method by 2.1% in mIoU while maintaining a similar

[1]https://github.com/open-mmlab/mmdetection

[2]https://github.com/open-mmlab/mmsegmentation

TABLE IV

PERFORMANCE COMPARISON OF OBJECT DETECTION AND INSTANCE SEGMENTATION ON THE COCO DATASET. FLOPS ARE CALCULATED WITH RESOLUTION 800 × 1280

Backbone	RetinaNet 1× Schedule								Mask R-CNN 1× Schedule							
	#F (G)	#P (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	#F (G)	#P (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-18 [58]	190	21.3	31.8	49.6	33.6	16.3	34.3	43.2	209	31.2	34.0	54.0	36.7	31.2	51.0	32.7
PoolFormer-S12 [41]	188	21.7	36.2	56.2	38.2	20.8	39.1	48.0	207	31.6	37.3	59.0	40.1	34.6	55.8	36.9
PVTv2-b1 [13]	209	23.8	40.2	60.7	42.4	22.8	43.3	54.0	227	33.7	41.8	64.3	45.9	38.8	61.2	41.6
PVT-ACmix-T [7]	232	-	40.5	61.2	42.7	-	-	-	-	-	-	-	-	-	-	-
ViL-T [59]	204	16.6	40.8	61.3	43.6	26.7	44.9	53.6	223	26.9	41.4	63.5	45.0	38.1	60.3	40.8
P2T-T [32]	206	21.1	41.3	62.0	44.1	24.6	44.8	56.0	225	31.3	43.3	65.7	47.3	39.6	62.5	42.3
MixFormer-B3 [8]	-	-	-	-	-	-	-	-	207	35.0	42.8	64.5	46.7	39.3	61.8	42.2
Slide-PVTv2-b1 [28]	204	-	41.5	62.3	44.0	26.0	44.8	54.9	222	33.0	42.6	65.3	46.8	39.7	62.6	42.6
TransXNet-T	187	22.4	**43.1**	**64.1**	**46.0**	**26.2**	**46.7**	**57.4**	205	32.5	**44.5**	**66.5**	**48.6**	**40.6**	**63.7**	**43.8**
Swin-T [4]	248	38.5	41.5	62.1	44.2	25.1	44.9	55.5	264	47.8	42.2	64.6	46.2	39.1	61.6	42.0
CSWin-T [27]	266	35.1	43.8	64.8	46.8	26.0	47.6	59.2	285	45.0	45.3	67.1	49.6	41.2	64.2	44.4
PVTv2-b2 [13]	-	-	-	-	-	-	-	-	279	42.0	46.7	68.6	51.3	42.2	65.6	45.4
PVT-ACmix-S [7]	232	-	40.5	61.2	42.7	-	-	-	-	-	-	-	-	-	-	-
Swin-ACmix-T [7]*	-	-	-	-	-	-	-	-	275	-	47.0	69.0	51.8	-	-	-
P2T-S [32]	260	33.8	44.4	65.3	47.6	27.0	48.3	59.4	279	43.7	45.5	67.7	49.8	41.4	64.6	44.5
MixFormer-B4 [8]	-	-	-	-	-	-	-	-	243	53.0	45.1	67.1	49.2	41.2	64.3	44.1
CrossFormer-S [55]	282	40.8	44.4	65.8	47.4	28.2	48.4	59.4	301	50.2	45.4	68.0	49.7	41.4	64.8	44.6
CMT-S [9]	231	44.3	44.3	65.5	47.5	27.1	48.3	59.1	249	44.5	44.6	66.8	48.9	40.7	63.9	43.4
InternImage-T [15]	-	-	-	-	-	-	-	-	270	49.0	47.2	69.0	52.1	42.5	66.1	45.8
Slide-PVTv2-b2 [28]	255	-	45.0	66.2	48.4	28.8	48.8	59.7	274	43.0	46.0	68.2	50.3	41.9	65.1	45.4
TransXNet-S	242	36.6	**46.4**	**67.7**	**50.0**	**28.9**	**50.3**	**61.1**	261	46.5	**47.7**	**69.9**	**52.3**	**43.1**	**66.9**	**46.4**
Swin-S [4]	336	59.8	44.5	65.7	47.5	27.4	48.0	59.9	354	69.1	44.8	66.6	48.9	40.9	63.4	44.2
CSWin-S [27]	-	-	-	-	-	-	-	-	342	54.0	47.9	70.1	52.6	43.2	67.1	46.2
PVTv2-b3 [13]	354	55.0	45.9	66.8	49.3	28.6	49.8	61.4	372	64.9	47.0	68.1	51.7	42.5	65.7	45.7
P2T-B [32]	344	45.8	46.1	67.5	49.6	30.2	50.6	60.9	363	55.7	47.2	69.3	51.6	42.7	66.1	45.9
CrossFormer-B [55]	389	62.1	46.2	67.8	49.5	30.1	49.9	61.8	408	71.5	47.2	69.9	51.8	42.7	66.6	46.2
InternImage-S [15]	-	-	-	-	-	-	-	-	340	69.0	47.8	69.8	52.8	43.3	67.1	46.7
Slide-PVTv2-b3 [28]	343	-	46.8	67.7	50.3	30.5	51.1	61.6	362	63.0	47.8	69.5	52.6	43.2	66.5	46.6
TransXNet-B	317	58.0	**47.6**	**69.0**	**51.1**	**31.3**	**51.7**	**62.2**	336	67.6	**48.8**	**70.8**	**53.5**	**43.8**	**68.0**	**47.2**

* ACmix uses the 3× schedule to train Mask R-CNN, while our method has better results despite using 1× schedule.

TABLE V

PERFORMANCE COMPARISON OF SEMANTIC SEGMENTATION ON THE ADE20K DATASET. FLOPS ARE CALCULATED WITH RESOLUTION 512 × 2048

Backbone	#F (G)	#P (M)	mIoU
ResNet-18 [58]	129	15.5	32.9
PoolFormer-S12 [41]	124	15.7	37.2
PVTv2-b1 [13]	129	17.8	42.5
PVT-ACmix-T [7]	160	17.0	42.7
P2T-T [32]	121	15.8	43.4
Slide-PVT-T [28]	136	16.0	38.4
TransXNet-T	121	16.6	**45.5**
Swin-T [4]	182	31.9	41.5
PVTv2-b2 [13]	167	29.1	45.2
PVT-ACmix-S [7]	228	29.0	46.4
CSWin-T [27]	202	26.1	48.2
ScalableViT-S [54]	174	30.0	44.9
CrossFormer-S [55]	221	34.3	46.0
UniFormer-S [12]	247	25.0	46.6
P2T-S [32]	162	28.4	46.7
Slide-PVT-S [28]	188	26.0	42.5
TransXNet-S	179	30.6	**48.5**
Swin-S [4]	274	53.2	45.2
PVTv2-b4 [13]	291	66.3	47.9
ScalableViT-B [54]	270	79.0	48.4
CrossFormer-B [55]	270	55.6	47.7
UniFormer-B [12]	471	54.0	48.0
P2T-L [32]	281	58.8	49.4
Slide-PVT-M [28]	278	46.0	44.0
TransXNet-B	256	51.7	**49.9**

computational cost. In addition, TransXNet-S improves the mIoU by 0.3% over CSWin-T [27] but with fewer GFLOPs.

TABLE VI

COMPARISON OF TOKEN MIXERS

Token Mixer	ImageNet-1K			ADE20K		
	#F (G)	#P (M)	Top-1	#F (G)	#P (M)	mIoU
Sep Conv [61]	1.6	11.4	76.9	119.7	14.6	39.9
SRA [30]	2.0	16.2	77.4	126.6	20.0	41.9
Swin [4]	2.0	13.6	78.2	130.5	17.4	40.3
Mixing Block [8]	2.0	13.6	78.9	130.0	17.4	42.3
ACmix Block [7]	2.1	13.9	**79.0**	131.4	17.6	41.5
D-Mixer (Ours)	1.6	11.4	**79.0**	118.0	15.2	**42.7**

Finally, TransXNet-B achieves the highest mIoU of 49.9%, surpassing other competitors but with less computational cost.

D. Ablation Study

1) Setup: To evaluate the impact of each component in TransXNet, we conduct extensive ablation experiments on ImageNet-1K. Due to limited resources, we adjust the number of training epochs to 200 for all the models, while keeping the rest of the experimental settings consistent with Section IV-A. Subsequently, we proceed to fine-tune the ImageNet pretrained model on the ADE20K dataset, applying the identical training configurations as described in Section IV-C.

2) Comparison of Token Mixers: To perform a fair comparison of token mixers, we adjust the tiny model to a similar style as Swin-T [4], i.e., setting the numbers of blocks and channels in the four stages to $[2, 2, 6, 2]$ and $[64, 128, 256, 512]$, respectively, and using nonoverlapping patch embedding and vanilla FFN. The performance of different token mixers is shown in Table VI.

TABLE VII
COMPARISON OF DEPTHWISE CONVOLUTIONS

Local Operator	ImageNet-1K			ADE20K		
	#F (G)	#P (M)	Top-1	#F (G)	#P (M)	mIoU
DWConv [61]	1.8	12.5	80.3	122.0	16.3	44.1
DyConv [39]	1.8	12.3	80.7	121.4	16.5	44.3
Window Attention [43]	1.9	13.3	80.8	124.4	17.0	44.5
D-DWConv [40]	1.8	14.2	**80.9**	121.4	18.0	44.6
IDConv (Ours)	1.8	12.8	**80.9**	121.4	16.6	**45.0**

It can be found that our D-Mixer has clear advantages in terms of performance and computational cost. In ImageNet-1K classification, D-Mixer ties with ACmix block [7] which is also a hybrid module, but our D-Mixer has a significantly lower computational cost. Furthermore, D-Mixer demonstrates a pronounced superiority in semantic segmentation, as demonstrated on the ADE20K dataset.

3) Comparison of Depthwise Convolutions: To evaluate the effectiveness of IDConv, we replace it in the tiny model with a series of alternatives including the standard DWConv, window attention [43], DyConv [39], and D-DWConv [40]. The kernel/window sizes of the above methods are set to 7×7 for fair comparisons. As listed in Table VII, IDConv exceeds DyConv by 0.2% top-1 accuracy and 0.7% mIoU with only a slight increase in Params. Then, window attention performs worse with higher computational cost, possibly due to its nonoverlapping locality. Compared with the recently proposed D-DWConv, IDConv has fewer parameters while achieving comparable top-1 in image classification and superior mIoU in semantic segmentation.

4) Impact of MS-FFN: Based on the tiny model, we investigate the impact of multiscale token aggregation in MS-FFN by conducting comparisons between MS-FFN and vanilla FFN [1], while adjusting the kernel size in the middle layer of MS-FFN. The results presented in Table VIII reveal that MS-FFN surpasses Inverted Residual FFN [9] (i.e., scale = 3) by 0.3% in top-1 accuracy, 0.4% in mIoU, and 0.9% AP^b, respectively. Importantly, this performance boost comes with only a minor increase in computational cost. Our investigation identifies the optimal set of scales for MS-FFN as $\{1, 3, 5, 7\}$, striking a favorable balance between performance and computational efficiency. Although we observe further performance gains by extending the scale set to $\{1, 3, 5, 7, 9\}$, we opt to discard this configuration to avoid the accompanying increase in the number of parameters and FLOPs. However, it can be found that scale = $\{1, 3, 5\}$ only brings a marginal improvement on image classification and semantic segmentation tasks. Specifically, compared with single-scale, scale = $\{1, 3, 5\}$ only improves 0.1% top-1 accuracy on ImageNet-1K and 0.1% mIoU on ADE20K. We believe that the reason for this phenomenon is that the performance of our MS-FFN is closely tied to the input resolution. Basically, the motivation for using MS-FFN is to capture different subregion features since it can generate multiscale representations. In this regard, if we use a relatively small input resolution (e.g., 224×224), then at deeper feature maps of the network (e.g., 14×14 and 7×7), a 3×3 convolution may already cover the sufficient object region. In this case, using a convolution with a larger kernel may not provide additional useful context, thereby resulting in limited performance improvement. However, as the input image resolution increases, the object region contains more pixels, thus a 3×3 convolution can only handle a subregion of the whole object, and convolutions with larger kernels have more potential to extract richer object context. In this regard, multiscale convolutions can provide more effective clues. As depicted in Table VIII, as we progress from image classification to object detection[3], the performance improvement of MS-FFN becomes increasingly notable. More specifically, scale = $\{1, 3, 5\}$ improves over single scale by a notable 0.6% AP^b, which is a more noticeable improvement compared with the other two tasks. It is noteworthy that scale = $\{1, 3\}$ maintains an ignorable performance gap with single scale but with lower computational complexity, while the final design of the MS-FFN (i.e., scale = $\{1, 3, 5, 7\}$) improves AP^b significantly by 1.0% over single scale.

5) Impact of Channel Ratio Between Attention and Convolution: Channel ratio represents the proportion of channels allocated to OSRA in a given feature map. We investigate the impact of channel ratio by setting it to different values in a tiny model. As shown in Table IX, top-1 accuracy and mIoU are greatly improved when the channel ratio is increased from 0.25 to 0.5. However, when the channel ratio becomes greater than 0.5, the improvement in top-1 accuracy and mIoU becomes inconspicuous even though the number of Params increases. Hence, we conclude that a channel ratio of 0.5 has the best tradeoff between performance and model complexity.

6) Other Model Design Choices: We verify the impact of DPE, OSR, and STE on a tiny model by removing or replacing these components. As shown in Table X, DPE brings clear performance improvement, which is consistent with previous works [12], [62]. Regarding the design choice of the self-attention module, OSR demonstrates a slight yet noteworthy improvement of 0.1% in top-1 accuracy and 0.3% in mIoU, all without incurring additional computational costs when compared to nonoverlapping spatial reduction (NSR). Notably, our proposed STE significantly reduces computational cost while maintaining consistent performance on ImageNet-1K and boosting mIoU by 0.3% on ADE20K, highlighting the effectiveness of STE as an efficient design choice for our model.

E. Network Visualization

1) ERF Analysis: To gain further insights into the superiority of our IDConv over the standard DWConv, we visualize the ERF [11] of the deepest stage of all the models considered in Table VII. As shown in Fig. 6(a), DWConv has the smallest ERF in comparison to dynamic operators, including DyConv [38], window attention [43], D-DWConv [40], and IDConv. Furthermore, among the dynamic operators, it is evident that IDConv enables our model to achieve the largest ERF while

[3]Unlike image classification and semantic segmentation, which use fixed input resolutions (i.e., 224×224 and 512×512, respectively), in object detection, the image is resized to the shorter side of 800, while ensuring that the longer side does not exceed 1333. Hence, the resized image generally possesses more fine-grained object information compared with the other two vision tasks.

TABLE VIII
ABLATION ON MS-FFN SCALES

FFN Scale	ImageNet-1K			ADE20K			COCO 2017		
	#F (G)	# P (M)	Top-1	#F (G)	# P (M)	mIoU	#F (G)	# P (M)	AP^b
N/A	1.7	12.5	80.3	119.3	16.2	44.0	185.1	22.1	40.9
$\{1, 3\}$	1.7	12.6	80.6	119.9	16.4	44.5	185.6	22.2	41.5
3 (Single-scale)	1.7	12.7	80.6	120.3	16.4	44.6	186.0	22.3	41.5
$\{1, 3, 5\}$	1.7	12.7	80.7	120.5	16.5	44.7	186.2	22.3	42.1
$\{1, 3, 5, 7\}$	1.8	12.8	80.9	121.4	16.6	**45.0**	187.1	22.4	42.5
$\{1, 3, 5, 7, 9\}$	1.8	13.0	**81.0**	122.5	16.8	44.7	188.2	22.6	**42.7**

TABLE IX
ABLATION ON CHANNEL RATIO BETWEEN ATTENTION AND CONVOLUTION

Attention Ratio	ImageNet-1K			ADE20K		
	#F (G)	#P (M)	Top-1	#F (G)	#P (M)	mIoU
0.25	1.7	12.5	80.6	120.3	16.3	44.1
0.5	1.8	12.8	**80.9**	121.4	16.6	**45.0**
0.75	1.8	13.6	**80.9**	123.2	17.4	44.9

TABLE X
ABLATION STUDY ON DPE, OSR, AND STE

NSR	DPE	OSR	STE	ImageNet-1K			ADE20K		
				#F (G)	#P (M)	Top-1	#F (G)	#P (M)	mIoU
✓				1.8	13.4	80.5	122.2	17.1	44.1
✓	✓			1.9	13.6	80.8	123.4	17.3	44.5
	✓	✓		1.9	13.6	**80.9**	123.5	17.4	44.7
	✓	✓	✓	1.8	12.8	**80.9**	121.4	16.6	**45.0**

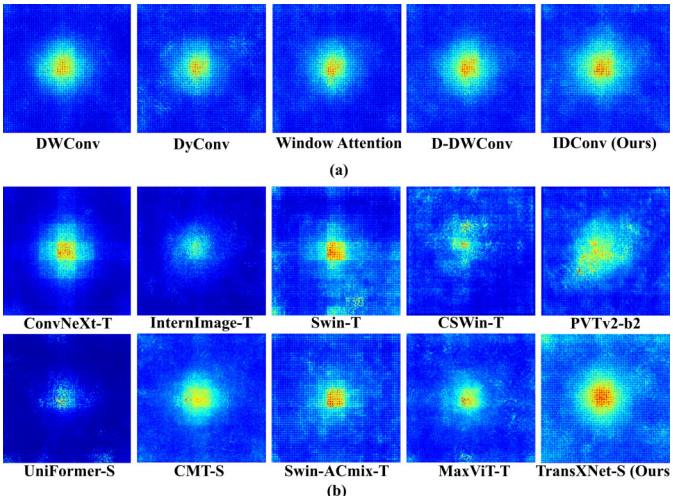

Fig. 6. ERF visualization of (a) models incorporating various local operators and (b) SOTA methods. The results are obtained by averaging over 100 images (resized to 224 × 224) from ImageNet.

preserving a strong locality. These observations substantiate the claim that incorporating suitable dynamic convolutions assists transformers in better capturing global contexts while carrying potent inductive biases, thereby improving their representation capacity.

On the other hand, to demonstrate the powerful representation capacity of our TransXNet, we also compare the ERF of several SOTA methods with similar computational costs. As shown in Fig. 6(b), our TransXNet-S has the largest ERF among these methods while maintaining strong local sensitivity, which is challenging to achieve.

2) Grad-CAM Analysis: To comprehensively assess the quality of the learned visual representations, we use Grad-CAM technique [30] to generate activation maps for visual representations at various stages of TransXNet-S, Swin-T, UniFormer-S, and PVT v2-b2. These activation maps provide insight into the significance of individual pixels in depicting class discrimination for each input image. As depicted in Fig. 7, our approach stands out by revealing more intricate details in early layers and identifying more semantically meaningful regions in deeper layers. This compellingly demonstrates the robust visual representation capabilities of our method compared with other competitors.

3) Visualization of D-Mixer: To demonstrate the hypothesis that convolution operations facilitate the local modeling and self-attention operations drive the global modeling when adopting our D-Mixer, we visualize both local and global activation maps using Grad-CAM and ERF of two branches in D-Mixer. Specifically, the visualization positions are the output of IDConv, the output of OSRA, and the output after the fusion of the two branches using STE, at the last D-Mixer in TransXNet-S. As shown in Fig. 8(a), the local and global branches demonstrate different ERFs. Specifically, the ERF of the local branch exhibits stronger local sensitivity, while the global branch possesses a larger ERF. When the local and global branches are combined, the ERF simultaneously acquires enhanced locality and global responses, thus confirming our hypothesis. It is worth noting that the ERF generated by the local branch also encompasses some long-range dependencies, which can be attributed to the network being stacked to deeper layers, thus the ERF is influenced by the preceding layers that have incorporated both local and global information. Furthermore, we have used Grad-CAM to visualize the activation maps. As shown in Fig. 8(b), the heatmaps generated by the local branch are more focused on detailed information within a local region, whereas the heatmaps produced by the global branch can cover the entire object but may introduce some irrelevant background information. However, when combining the local and global information, the generated heatmaps exhibit a more precise attention on the object regions. This further corroborates our hypothesis.

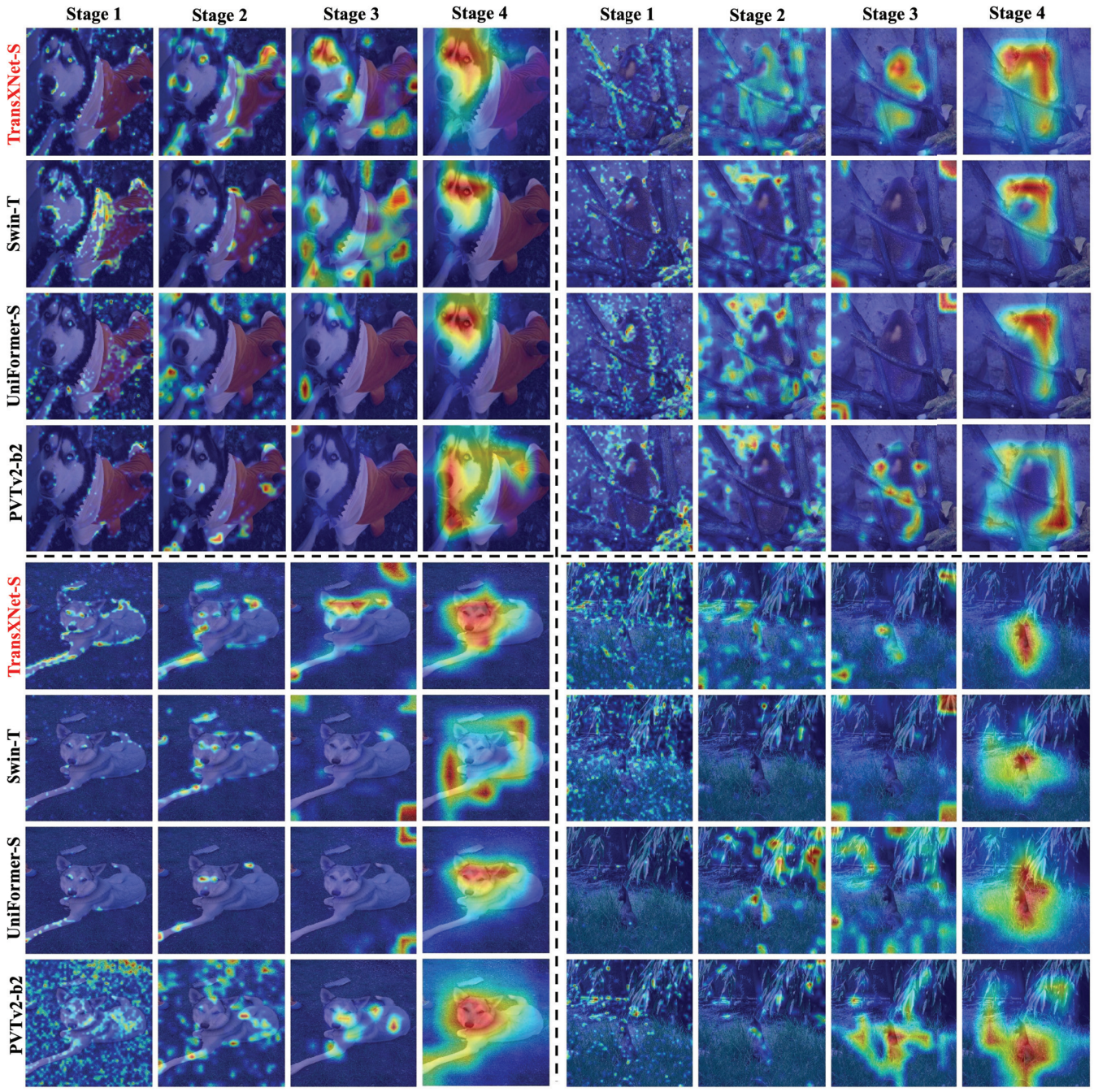

Fig. 7. Grad-CAM visualization of the models trained on ImageNet-1K. The visualized images are randomly selected from the validation set.

F. Computational Efficiency Analysis

In this section, we present a comparison of computational efficiency among different methods. Specifically, using a single RTX 3090 GPU with a batch size of 128, we compare the throughput and GPU memory cost of our method with various classical backbone networks, including Swin [4], ConvNeXt [16], and other related methods that extract both global and local information, such as MPViT [53], QuadTree Transformer [52], and Focal-Transformer [29]. As shown in Table XI, our method achieves a favorable tradeoff between computational efficiency and performance. For example, when TransXNet-T is compared with MPViT-XS, our method achieves nearly comparable speed (857 images/s versus 868 images/s) while consuming less GPU memory (2252 MB versus 2460 MB), and demonstrates a noticeable advantage in top-1 accuracy (81.6% versus 80.9%). Moreover, our small and base models also demonstrate excellent computational efficiency. For instance, TransXNet significantly outperforms Focal-Transformer in terms of performance, speed, and memory usage. Although our method lags behind Swin and ConvNeXt in speed, these methods benefit from the efficiency of local operators for acceleration, while TransXNet includes some operators that may not be as compatible with GPU-based parallel computing, such as multiscale depthwise convolutions in MS-FFN. However, it is noteworthy that TransXNet-S has a notable advantage over Swin-T regarding GPU memory

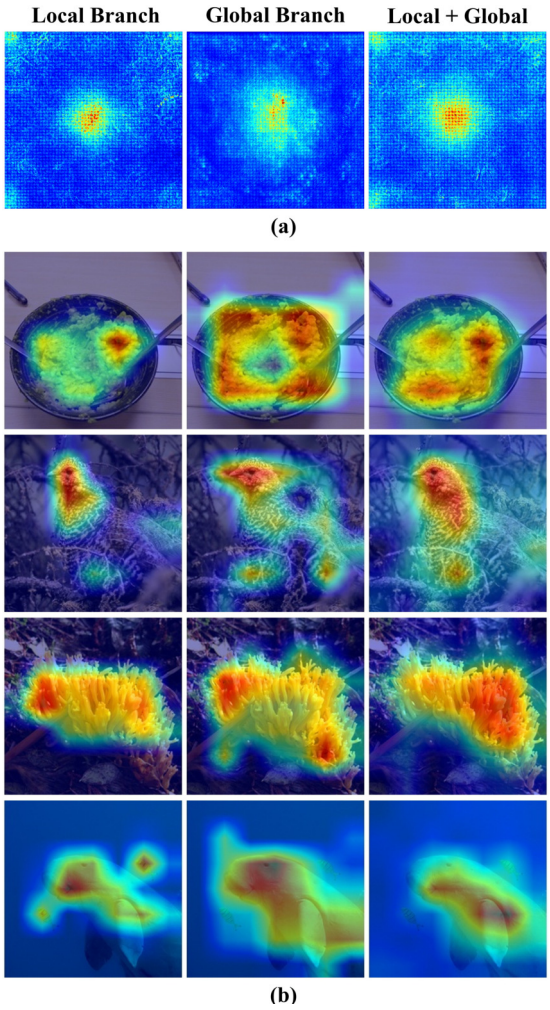

Fig. 8. (a) ERF and (b) Grad-CAM analyses for D-Mixer.

consumption. This advantage may lead to similar speeds between TransXNet-S and Swin-T in practical applications, as our TransXNet has the potential to use a larger batch size with the same memory consumption as Swin.

Furthermore, we compare computational efficiency among different token mixers. It is worth noting that we use a similar architectural design, with the only difference being the token mixer used, namely, Sep Conv [61], SRA [30], Shifted Window [4], Mixing block [8], ACmix block [7], and our D-Mixer. As listed in Table XII, our D-Mixer demonstrates a better tradeoff among performance, GPU memory consumption, and speed compared with other token mixers. This highlights that our D-Mixer is both effective and GPU-friendly.

V. Limitations

Our ablation study suggests that a fixed 1:1 ratio between the numbers of channels allocated to self-attention and dynamic convolutions in all the stages yields a favorable tradeoff. However, we speculate that using different ratios at different stages may further improve performance and reduce computational cost. Regarding the model design, our TransXNet series are manually stacked, and there exist potentially inefficient operators in the building blocks (e.g., multi-kernel depthwise convolutions in MS-FFN). As a result, our model exhibits limited advantages in terms of speed when compared to other models with similar GFLOPs. Nonetheless, these inefficiencies can be mitigated through techniques such as neural architecture search (NAS) [63] and specialized implementation engineering, which we plan to explore in our future work.

TABLE XI

Comparison of Throughput and GPU Memory Cost Among Representative Backbone Networks

Method	# F (G)	# P (M)	# T (imgs/s)	# M (MB)	Top-1
Swin-T [4]	4.4	28.3	858	4392	81.3
Swin-S [4]	8.5	49.6	514	4480	83.0
Swin-B [4]	15.4	88.0	183	8620	83.5
ConvNeXt-T [16]	4.5	28.6	1090	3276	82.1
ConvNeXt-S [16]	8.7	50.2	642	3302	83.1
ConvNeXt-B [16]	15.4	88.6	422	4302	83.8
MPViT-XS [53]	2.9	10.5	868	2460	80.9
MPViT-S [53]	4.7	22.9	513	3078	83.0
MPViT-B [53]	16.1	74.9	256	5502	84.3
QuadTree-B-b1 [52]	2.3	13.6	836	3693	80.0
QuadTree-B-b2 [52]	4.5	24.2	461	3781	82.7
QuadTree-B-b4 [52]	11.5	64.2	215	3921	84.0
Focal-T [29]	4.9	29.1	381	11219	82.2
Focal-S [29]	9.5	51.1	231	11311	83.5
Focal-B [29]	16.0	89.8	168	14889	83.8
TransXNet-T	1.8	12.8	857	2252	81.6
TransXNet-S	4.5	26.8	417	3678	83.8
TransXNet-B	8.2	47.9	250	5998	84.6

TABLE XII

Comparison of Throughput and GPU Memory Cost Among Different Token Mixers

Method	# F (G)	# P (M)	# T (imgs/s)	# M (MB)	Top-1
Sep Conv [61]	1.6	11.4	2164	3615	76.9
SRA [30]	2.0	16.2	1906	4463	77.4
Swin [4]	2.0	13.6	1346	2869	78.2
Mixing Block [8]	2.0	13.6	1240	3469	78.9
ACmix Block [7]	2.1	13.9	1018	4069	79.0
D-Mixer (Ours)	1.6	11.4	1649	3884	79.0

VI. Conclusion

In this work, we propose an efficient D-Mixer, taking advantage of hybrid feature extraction provided by OSRA and IDConv. By stacking D-Mixer-based blocks to a deep network, the kernels in IDConv and attention matrices in OSRA are dynamically generated using both local and global information gathered in previous blocks, empowering the network with a stronger representation capacity by incorporating strong inductive bias and an expanded ERF. Besides, we introduce an MS-FFN to explore multiscale token aggregation in the feedforward network. By alternating D-Mixer and MS-FFN, we construct a novel hybrid CNN-Transformer network termed TransXNet, which has shown SOTA performance on various vision tasks.

References

[1] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, Jan. 2020.

[2] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 10347–10357.

[3] Z. Dai et al., "CoAtNet: Marrying convolution and attention for all data sizes," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 3965–3977.

[4] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[5] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.

[6] Q. Yu, Y. Xia, Y. Bai, Y. Lu, A. Yuille, and W. Shen, "Glance-and-gaze vision transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Jan. 2021, pp. 12992–13003.

[7] X. Pan et al., "On the integration of self-attention and convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 815–825.

[8] Q. Chen et al., "MixFormer: Mixing features across windows and dimensions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5249–5259.

[9] J. Guo et al., "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12175–12185.

[10] Z. Tu et al., "MaxViT: Multi-axis vision transformer," in *Proc. Eur. Conf. Comput. Vis.*Cham, Switzerland: Springer, Jan. 2022, pp. 459–479.

[11] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Dec. 2016, pp. 4898–4906.

[12] K. Li et al., "UniFormer: Unifying convolution and self-attention for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 10, pp. 12581–12600, 2023.

[13] W. Wang et al., "PVT v2: Improved baselines with pyramid vision transformer," *Comput. Vis. Media*, vol. 8, no. 3, pp. 415–424, Sep. 2022.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[15] W. Wang et al., "InternImage: Exploring large-scale vision foundation models with deformable convolutions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 14408–14419.

[16] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11976–11986.

[17] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11963–11975.

[18] S. Liu et al., "More ConvNets in the 2020s: Scaling up kernels beyond 51×51 using sparsity," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2022.

[19] H. Zhang, W. Hu, and X. Wang, "ParC-Net: Position aware circular convolution with merits from ConvNets and transformer," in *Proc. Eur. Conf. Comput. Vis.*Cham, Switzerland: Springer, 2022, pp. 613–630.

[20] R. Xu, H. Zhang, W. Hu, S. Zhang, and X. Wang, "ParCNetV2: Oversized kernel with enhanced attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jan. 2022, pp. 5752–5762.

[21] J. Yang, C. Li, X. Dai, and J. Gao, "Focal modulation networks," in *Proc. Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 35, 2022, pp. 4203–4217.

[22] Y. Rao, W. Zhao, Y. Tang, J. Zhou, S.-N. Lim, and J. Lu, "HorNet: Efficient high-order spatial interactions with recursive gated convolutions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 10353–10366.

[23] M. H. Guo, C. Z. Lu, Z. N. Liu, M. M. Cheng, and S. M. Hu, "Visual attention network," *Comp. Vis. Media*, vol. 9, no. 4, pp. 733–752, Jul. 2023.

[24] S. Li et al., "MogaNet: Multi-order gated aggregation network," in *Proc. Int. Conf. Learn. Represent.*, 2024.

[25] Q. Hou, C.-Z. Lu, M.-M. Cheng, and J. Feng, "Conv2Former: A simple transformer-style ConvNet for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 8274–8283, Dec. 2024.

[26] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[27] X. Dong et al., "CSWin transformer: A general vision transformer backbone with cross-shaped windows," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12124–12134.

[28] X. Pan, T. Ye, Z. Xia, S. Song, and G. Huang, "Slide-transformer: Hierarchical vision transformer with local self-attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 2082–2091.

[29] J. Yang et al., "Focal attention for long-range interactions in vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, May 2021, pp. 30008–30022.

[30] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 568–578.

[31] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, "Shunted self-attention via multi-scale token aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10853–10862.

[32] Y.-H. Wu, Y. Liu, X. Zhan, and M.-M. Cheng, "P2T: Pyramid pooling transformer for scene understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12760–12771, 2023.

[33] H. Zhang, W. Hu, and X. Wang, "Fcaformer: Forward cross attention in hybrid vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6037–6046.

[34] N. Li, Y. Chen, W. Li, Z. Ding, D. Zhao, and S. Nie, "BViT: Broad attention-based vision transformer," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12772–12783, 2024.

[35] T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick, "Early convolutions help transformers see better," in *Proc. NIPS*, vol. 34, Dec. 2021, pp. 30392–30400.

[36] Y. Li et al., "EfficientFormer: Vision transformers at MobileNet speed," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Jan. 2022, pp. 12934–12949.

[37] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019.

[38] J. He, Z. Deng, and Y. Qiao, "Dynamic multi-scale filters for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3562–3572.

[39] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11030–11039.

[40] Q. Han et al., "On the connection between local attention and dynamic depth-wise convolution," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021.

[41] W. Yu et al., "MetaFormer is actually what you need for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10819–10829.

[42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[43] X. Chu et al., "Twins: Revisiting the design of spatial attention in vision transformers," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 9355–9366.

[44] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 740–755.

[45] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 633–641.

[46] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2019.

[47] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.*Cham, Switzerland: Springer, 2016, pp. 646–661.

[48] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, "Do ImageNet classifiers generalize to ImageNet," in *Proc. Int. Conf. Mach. Learn.*, May 2019, pp. 5389–5400.

[49] C. Yang et al., "MOAT: Alternating mobile convolution and attention brings strong vision models," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2022.

[50] R. Wightman, H. Touvron, and H. Jégou, "ResNet strikes back: An improved training procedure in timm," 2021, *arXiv:2110.00476*.

[51] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10428–10436.

[52] S. Tang, J. Zhang, S. Zhu, and P. Tan, "QuadTree attention for vision transformers," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2022.

[53] Y. Lee, J. Kim, J. Willette, and S. J. Hwang, "MPViT: Multi-path vision transformer for dense prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7287–7296.

[54] R. Yang et al., "ScalableViT: Rethinking the context-oriented generalization of vision transformer," in *Proc. Eur. Conf. Comput. Vis.*Cham, Switzerland: Springer, Jan. 2022, pp. 480–496.
[55] W. Wang et al., "CrossFormer: A versatile vision transformer hinging on cross-scale attention," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021.
[56] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[57] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[59] P. Zhang et al., "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2998–3008.
[60] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6399–6408.
[61] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.
[62] X. Chu et al., "Conditional positional encodings for vision transformers," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2021.
[63] P. Ren et al., "A comprehensive survey of neural architecture search: Challenges and solutions," *ACM Comput. Survey*, vol. 54, no. 4, pp. 1–34, 2021.

Meng Lou received the B.E. degree from Lanzhou Jiaotong University, Lanzhou, in 2019, and the M.E. degree from the School of Information Science and Engineering, Lanzhou University, Lanzhou, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Computer Science, The University of Hong Kong, Hong Kong,

From 2022 to 2023, he was a Full-Time Researcher with the AI Lab, Deepwise Healthcare, Beijing, China. His research focuses on deep learning and computer vision.

Shu Zhang received the B.E. degree in automation from Northwestern Polytechnical University, Shenzhen, China, in 2012, and the Ph.D. degree in computer application technology from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2018.

Since April 2018, he has been a Full-Time Senior Researcher with the AI Lab, Deepwise Healthcare, Beijing. His research focuses on pattern recognition and computer vision, with a particular emphasis on medical image analysis and AI for healthcare.

Hong-Yu Zhou (Member, IEEE) received the B.S. degree from Wuhan University, Wuhan, China, in 2015, the M.S. degree from the Department of Computer Science and Technology, Nanjing University, Nanjing, China, in 2018, and the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, Hong Kong, in 2023.

He is currently a Post-Doctoral Fellow with the Harvard Medical School, Boston, MA, USA. His research interests include AI for medicine and AI for healthcare.

Sibei Yang received the B.E. degree from Zhejiang University, Hangzhou, China, in 2016, and the Ph.D. degree from The University of Hong Kong, Hong Kong, in 2020.

She is currently an Assistant Professor and a Principal Investigator with ShanghaiTech University, Shanghai, China. From 2020 to 2021, she worked as a Research Assistant Professor with The Hong Kong Polytechnic University, Hong Kong. Her research interests include computer vision and machine learning.

Chuan Wu (Fellow, IEEE) received the B.E. and M.E. degrees from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2000 and 2002, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, in 2008.

From 2002 to 2004, she worked with the Information Technology Industry, Singapore. Since September 2008, she has been with The University of Hong Kong, Hong Kong, where she is currently a Professor. Her current research interests include distributed machine learning systems and algorithms and intelligent elderly care technologies.

Dr. Wu is an ACM Distinguished Member and served as the Chair for the Interest Group on Multimedia Services and Applications Over Emerging Networks (MEN) of the IEEE Multimedia Communication Technical Committee (MMTC) from 2012 to 2014. She was a co-recipient of the Best Paper Award of HotPOST 2012 and ACM e-Energy 2016. She is an Associate Editor of IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE TRANSACTIONS ON CLOUD COMPUTING.

Yizhou Yu (Fellow, IEEE) received the Ph.D. degree from the University of California at Berkeley, Berkeley, CA, USA, in 2000.

He is a Chair Professor with The University of Hong Kong, Hong Kong. He was a Faculty Member with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, for 12 years. His current research interests include AI foundation models, AI for medicine, AI-based multimedia content generation, and computer vision.

Dr. Yu is an ACM Fellow. He was a recipient of the U.S. National Science Foundation CAREER Award. He has served on the Editorial Board of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS. He has also served on the Program Committee of many leading international conferences, including CVPR, ICCV, and SIGGRAPH.