S2AFormer: Strip Self-Attention for Efficient Vision Transformer

Guoan Xu, Wenfeng Huang, Wenjing Jia, *Member, IEEE*, Jiamao Li, *Member, IEEE*, Guangwei Gao, *Senior Member, IEEE*, Guo-Jun Qi, *Fellow, IEEE,*

***Abstract*—The Vision Transformer (ViT) has achieved remarkable success in computer vision due to its powerful token mixer, which effectively captures global dependencies among all tokens. However, the quadratic complexity of standard self-attention with respect to the number of tokens severely hampers its computational efficiency in practical deployment. Although recent hybrid approaches have sought to combine the strengths of convolutions and self-attention to improve the performance–efficiency trade-off, the costly pairwise token interactions and heavy matrix operations in conventional self-attention remain a critical bottleneck. To overcome this limitation, we introduce S2AFormer, an efficient Vision Transformer architecture built around a novel Strip Self-Attention (SSA) mechanism. Our design incorporates lightweight yet effective Hybrid Perception Blocks (HPBs) that seamlessly fuse the local inductive biases of CNNs with the global modeling capability of Transformer-style attention. The core innovation of SSA lies in simultaneously reducing the spatial resolution of the key (K) and value (V) tensors while compressing the channel dimension of the query (Q) and key (K) tensors. This joint spatial-and-channel compression dramatically lowers computational cost without sacrificing representational power, achieving an excellent balance between accuracy and efficiency. We extensively evaluate S2AFormer on a wide range of vision tasks, including image classification (ImageNet-1K), semantic segmentation (ADE20K), and object detection/instance segmentation (COCO). Experimental results consistently show that S2AFormer delivers substantial accuracy improvements together with superior inference speed and throughput across both GPU and non-GPU platforms, establishing it as a highly competitive solution in the landscape of efficient Vision Transformers.**

***Index Terms*—Transformer, strip self-attention, local perception, global context**

This work was supported in part by the foundation of Key Laboratory of Artificial Intelligence of Ministry of Education under Grant AI202404. *(Corresponding authors: Wenjing Jia, Guangwei Gao.)*

Guoan Xu, Wenfeng Huang, and Wenjing Jia are with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: xga_njupt@163.com, huangwenfeng@outlook.com, and Wenjing.Jia@uts.edu.au).

Jiamao Li is with the Bionic Vision System Laboratory, State Key Laboratory of Transducer Technology, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China (e-mail:jmli@mail.sim.ac.cn).

Guangwei Gao is with the PCA Lab, Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai 200240, China (e-mail: csggao@gmail.com).

Guo-Jun Qi is with the Research Center for Industries of the Future and the School of Engineering, Westlake University, Hangzhou 310024, China, and also with OPPO Research, Seattle, WA 98101 USA (e-mail: guojunq@gmail.com).

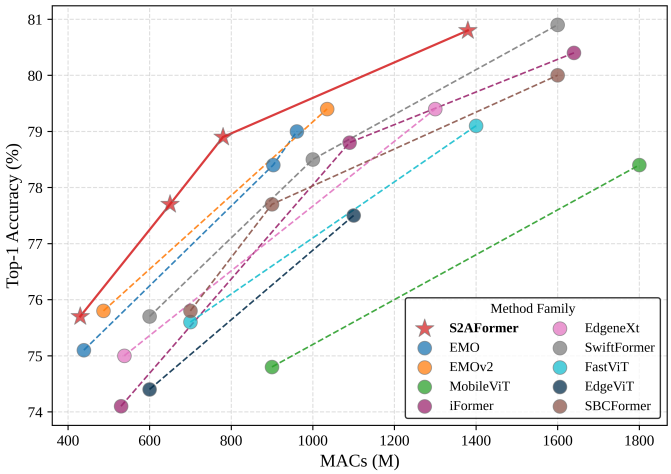


Fig. 1: Comparison of our proposed S2AFormer model with SOTA methods, including EMOv2 [8], SwiftFormer [9], and iFormer [10]. (Top-1 accuracy v.s. MACs on ImageNet-1k [11]). In theory, the optimal performance is in the upper-left of the plot, which means higher top-1 accuracy and fewer MACs.

I. Introduction

Vision Transformers (ViTs) [1], [2] have emerged as a notable competitor to CNNs, demonstrating a superior ability to capture long-range interactions between image patches. While Convolutional Neural Networks (CNNs) limit interactions to local regions through shared kernels, ViTs divide the input image into patches and use self-attention (SA) to update token features, enabling global interactions. Consequently, Transformer-based architectures have gained increasing attention in computer vision, achieving remarkable results across various tasks such as image classification [3], [4], object detection [1], [5], and semantic segmentation [6], [7].

However, the quadratic complexity in the token mixer makes self-attention inefficient in terms of parameters and computational load. For images of size $h \times w$, the complexity $O(h^2w^2)$ limits the applicability of ViTs in high-resolution tasks (such as detection and segmentation), unsuitable for real-time applications and edge devices due to their significant computational overhead.

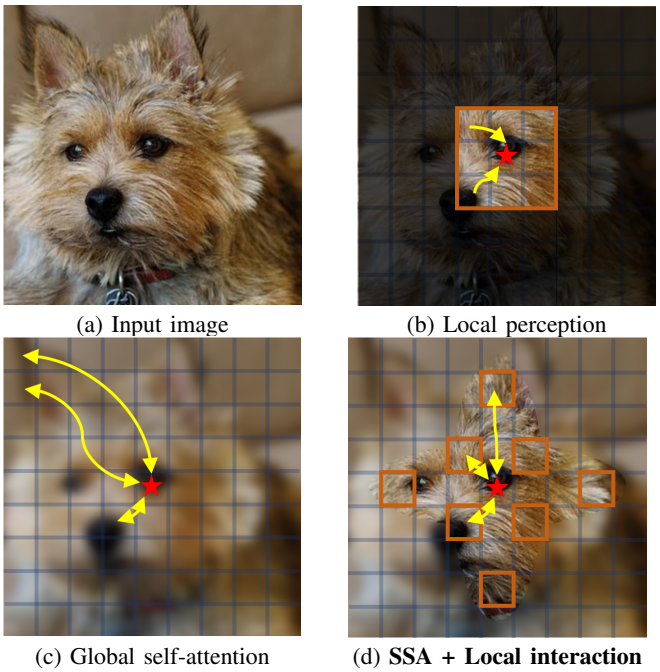
To address computational overhead, various optimized self-attention strategies have been introduced. Methods such as [3], [12] have been developed to limit the self-attention operation to local windows, rather than to the entire feature map. However, this approach compromises global information interaction between patches. Another approach involves token

pruning or merging to selectively compute only the most informative tokens and eliminate uninformative tokens. For instance, Dynamic-ViT [13] reduces the number of tokens by eliminating redundant tokens, while EViT [14] merges redundant tokens into a single token. DVT [15] adopts a flexible and dynamic patch-splitting strategy (*e.g.*, 4×4, 7×7, etc.) tailored to the complexity of each image, moving away from the standard 14×14 approach. MG-ViT [16] emphasizes the need for different regions of an image to receive attention at varying levels of granularity, offering a more nuanced and effective representation. While reducing tokens decreases computation, it presents challenges when dealing with complex, object-rich images, as it becomes increasingly difficult to determine which tokens should be retained.

Some approaches [17]–[19] aim to streamline the global sequence length by aggregating tokens across the entire key-value feature space, implementing a coarse-grained global attention mechanism. For instance, the Pyramid Vision Transformer (PVT) [17] uses a large kernel with significant strides to achieve uniform token aggregation, generating a consistent coarse representation across the feature map. Efficient-ViT [20] introduces grouped attention heads to capture multi-scale features while distributing the spatial dimensions of each token, thereby effectively balancing the computational load. Similarly, SG-Former [19] combines the windowing mechanism of the Swin Transformer [3] with the spatial dimensionality reduction strategy of PVT [17], allowing for the extraction of information at different scales. Notably, it further optimizes efficiency by reducing the sequence length of Q, significantly decreasing computational complexity. As pointed out in MetaSeg [21], beyond spatial redundancy, significant redundancy also exists in the channel dimension. By leveraging this channel redundancy, the computational complexity can be further reduced. While ViTs excel at modeling long-range dependencies, they tend to struggle with local feature representation compared to CNNs, a limitation that is often overlooked in the aforementioned architectures.

A comparative analysis of receptive field properties and cross-region information exchange mechanisms of different architectures is illustrated in Fig. 2. Convolutional networks excel in local sensitivity due to their inductive bias toward spatial locality (via small kernels), enabling fine-grained detail extraction but struggling to model long-range contextual relationships. Conversely, vanilla Transformers establish global receptive fields through all-to-all token interactions, achieving holistic scene understanding at the cost of quadratic computational complexity and suboptimal local structural awareness.

In this paper, we propose an efficient Transformer architecture, named *S2AFormer*. Our goal is to address the primary challenge of preserving both local sensitivity and global receptive fields while reducing heavy computational load. To achieve this, we first design simple yet effective Hybrid Perception Blocks (HPBs) in the backbone network. Specifically, we propose a novel Strip Self-Attention (SSA) module that first applies spatial convolutions to the Key and $Value$ features, thereby extracting more distilled feature representations. Subsequently, the $Query$ and Key undergo channel-wise compression, significantly improving memory



(a) Input image (b) Local perception (c) Global self-attention (d) **SSA + Local interaction**

Fig. 2: Visualization of perception fields in different strategies. Red stars represent the current positions, and the black areas represent the regions that the current position cannot perceive. Convolutional networks cannot model long-range contexts. Global self-attention establishes global receptive fields at the cost of quadratic computational complexity. SSA integrated with local interaction prioritizes semantically salient regions globally while suppressing redundant spatial correlations.

efficiency and reducing computational complexity from $O(n^2 \cdot C)$ to $O((\frac{n}{r})^2 \cdot h)$, where r is the reduction rate and h is the number of attention heads.

Meanwhile, rotations and translations are common data augmentations in vision tasks, and a well-designed model should ideally produce consistent predictions under such transformations, demonstrating a desirable degree of geometric invariance. However, Transformers relying on absolute positional encodings often introduce patch-specific biases that anchor predictions to fixed spatial coordinates, thereby undermining this invariance. Moreover, purely global attention mixing tends to neglect fine-grained local neighborhood structures and precise boundary details within patches. To address these limitations, we incorporate a dedicated Local Interaction Module (LIM) into HPB. This module effectively complements global receptive fields with localized processing, significantly enhancing boundary fidelity and robustness to rotations and translations. The LIM is implemented using only lightweight depthwise separable and grouped convolutions, ensuring minimal additional computational overhead. Furthermore, we integrate SENet [22] to dynamically reweight features, enabling the model to focus on semantically salient information within the local receptive field. A comprehensive series of rigorous experiments confirms that our S2AFormer architecture either rivals or surpasses the most advanced state-of-the-art backbones (as shown in Fig. 1).

The key contributions of this paper are summarized as:

TABLE I: Key properties observed from successful backbones that benefit model design.

Methods \ Properties	Global dependency	Local perception	Low computation	High performance	Fast inference
DeiT [4]	✓	✗	✗	✓	✗
PoolFormer [23]	✓	✓	✗	✓	✗
RepViT [24]	✗	✓	✓	✗	✓
SHViT [25]	✓	✓	✓	✗	✓
S2AFormer (ours)	✓	✓	✓	✓	✓

1) We propose a novel token-mixer mechanism, termed SSA, which enables token interaction in a lightweight manner through both spatial and channel-wise compression, effectively reducing computational cost and achieving faster inference.
2) We introduce HPB, a simple yet effective design that enhances the global self-attention with local perceptual capabilities. By incorporating the LIM module, HPB improves the model's ability to capture fine-grained visual details, producing more comprehensive and expressive feature representations.
3) We pretrain our backbone on ImageNet-1K [11] and evaluate its generalization capability on multiple downstream tasks, including object detection, instance segmentation, and semantic segmentation, all of which demonstrate strong performance, providing a solid pretrained model for future research.

II. Related Work

A. Efficient ViTs

Originally developed for long-sequence learning in NLP tasks [26], Transformers were later adapted for image classification [2] and for object detection [1], both achieving performance on par with CNNs due to advanced training techniques and larger datasets. DeiT [4] further improved the training process by incorporating distillation, eliminating the need for extensive pretraining [27]. Since then, various modifications of vision Transformers (ViT) and hybrid architectures have emerged, introducing image-specific inductive biases to ViTs, enhancing performance across a range of vision tasks [17], [28]. While ViTs have outperformed CNNs in several vision tasks due to their token mixer's strong global context capabilities, the complexity of pairwise token interactions and the heavy computational demands of matrix operations present challenges for deploying ViTs in resource-constrained environments and real-time applications.

As a result, researchers have focused on developing various optimization methods to make ViTs more lightweight and better suited for mobile devices. These methods include token pruning or token merging [13], [29], [30], the introduction of novel architectures or modules [9], [31], [32], re-evaluating self-attention and sparse-attention mechanisms [33]–[35], and applying search algorithms commonly used in CNNs to identify more compact and faster ViT models [36]. Notable examples include TPS [37], which reduces the number of tokens by truncating based on a threshold. This method shortens the sequence length processed by self-attention and has proved its effectiveness in scenarios where the target objects in the image are relatively simple and singular. EdgeViT [38] improved efficiency by employing a global sparse attention module that focuses on a few selected tokens. In contrast, [17] achieves a better efficiency-accuracy balance by downsampling the key and value vectors. EdgeNeXt [32] introduced a transposed self-attention mechanism that computes attention maps along the channel dimension rather than the spatial dimension. This approach, combined with token mixing, results in linear complexity relative to the number of tokens. Reformer [39] used locality-sensitive hashing to group tokens, replacing dot-product attention and reducing complexity from $O(n^2)$ to $O(nlogn)$. LinFormer [40] employed a low-rank matrix factorization technique to approximate the self-attention matrix, reducing complexity from $O(n^2)$ to $O(n)$. Likewise, RAVLT [33] introduced a rank-augmented linear attention framework that effectively balances the performance of traditional softmax attention with the efficiency of linear attention methods. SwiftFormer [9] introduces a more efficient additive attention mechanism, replacing quadratic matrix multiplications with linear element-wise operations.

Although existing methods can alleviate the quadratic computational complexity of self-attention, they often do so at the cost of reduced accuracy. Moreover, approaches that rely exclusively on global self-attention remain insufficient for capturing fine-grained local features, owing to the intrinsic limitations of self-attention in modeling detailed local patterns.

B. Hybrid Variants

Recent research has introduced a variety of hybrid architectures that integrate CNNs and ViTs within a single framework. By leveraging ViTs' self-attention for capturing long-range dependencies and CNNs' local kernels to preserve fine-grained details, these models have achieved enhanced performance across various vision tasks. MobileViT [41] employed a hybrid architecture by combining lightweight MobileNet blocks with multi-head self-attention (MHSA) blocks. The MobileFormer [42] architecture merged MobileNetv3 [43] with ViT [2] to achieve state-of-the-art performance within a 6M-parameter budget, making it well-suited for resource-constrained environments. While these methods achieve fast inference speeds and demonstrate strong suitability for edge deployment, their compromised accuracy presents a critical limitation for performance-sensitive applications.

The CMT [44] architecture introduced a convolutional stem, placing a convolutional layer before each Transformer block, resulting in an alternating sequence of convolutional and

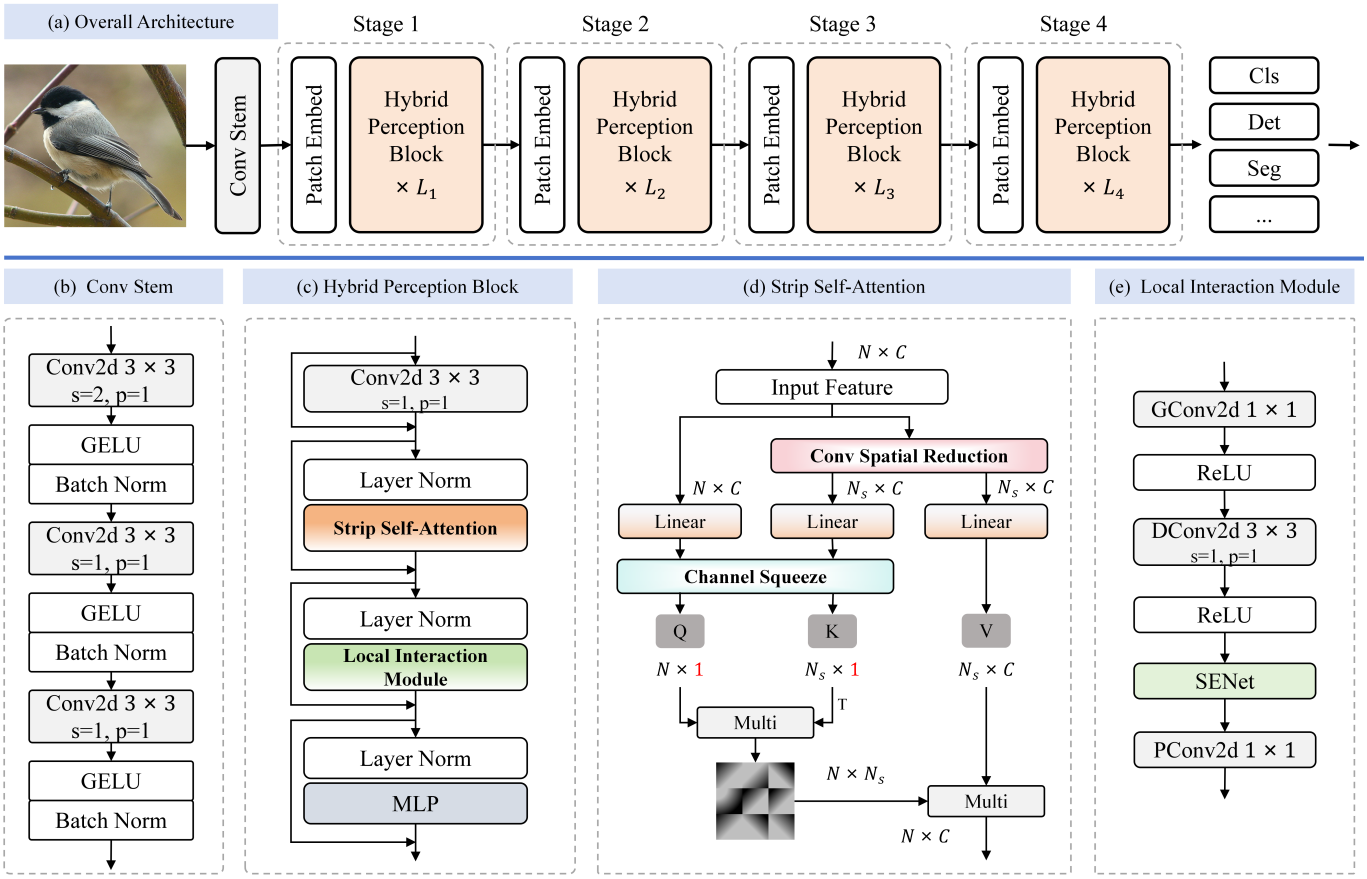


Fig. 3: Overview of our proposed S2AFormer. Similar to [3], [17], we employ a hierarchical architecture with four stages, each containing L_i Hybrid Perception Blocks. Table II provides the detailed network configurations of S2AFormer variants.

Transformer layers. CvT [45] replaced ViTs' linear embeddings with convolutional token embeddings and uses a convolutional transformer layer to fully leverage these embeddings, ultimately improving performance. However, their improved accuracy comes at the cost of reduced inference speed.

Since 2024, to avoid the computational burden and inference speed bottleneck caused by self-attention, efficient methods have been exploring the use of convolution-based techniques as alternatives in the token mixer component. For example, RepViT [24] built upon MobileNet-V3 [43] by streamlining the block structure and removing the residual connection used during training to achieve faster inference speeds. RepNeXt [46] utilized multi-scale feature representations while combining both serial and parallel structural reparameterization (SRP) methods to expand the network's depth and width, resulting in a unique convolutional attention mechanism. CAS-ViT [47] adopted a serial structure with CBAM [48], where spatial and channel-wise weights for the query (Q) and key (K) are calculated. These weights are then summed and used to perform matrix calculations with value (V), resulting in an additive attention mechanism. This is a promising approach, given that the computational complexity of convolution is $O(n)$. However, purely CNN-based architectures that rely on large-kernel convolutions still fall short in performance compared to self-attention mechanisms, despite offering notable improvements in speed and efficiency.

TABLE II: Network configurations of S2AFormer variants.

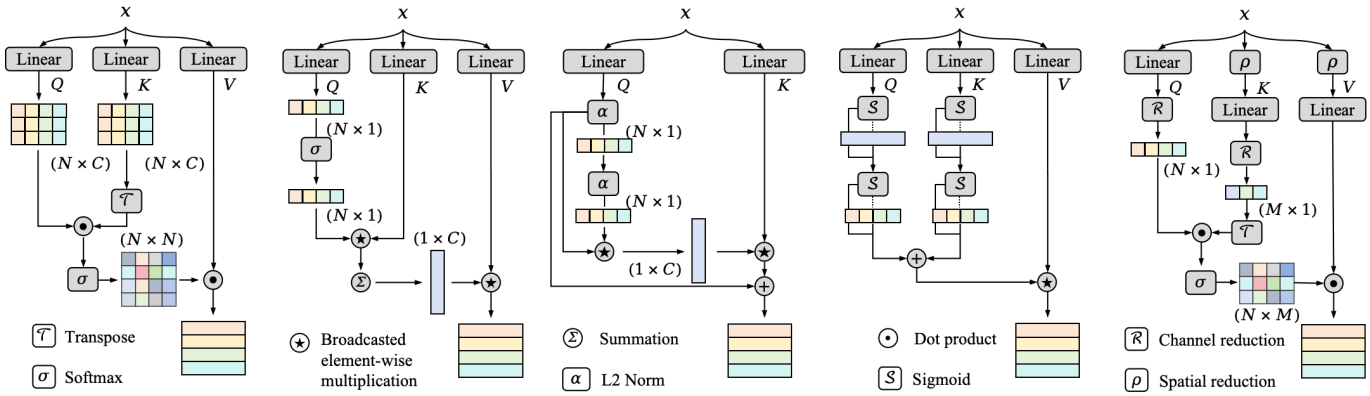
Model	Blocks L	Channels C	#Para. (M)	GMACs
mini	[2,2,2,2]	[32, 64, 128, 256]	5.02	0.43
T	[2,2,6,2]	[48, 64, 128, 256]	5.80	0.66
XS	[2,2,10,2]	[48, 64, 128, 256]	6.54	0.79
S	[2,4,24,4]	[48, 64, 128, 256]	10.69	1.38
M	[2,4,20,2]	[96, 128, 256, 512]	24.87	4.12
L	[4,4,20,4]	[96, 192, 384, 768]	76.58	12.53

These observations collectively underscore the essential qualities of an efficient backbone: the ability to capture both local and global features, achieve high performance, and maintain fast inference speed. Making the trade-off between efficiency and precision is a continuing focus of research. As presented in Table I, the identified properties from successful backbone designs offer important guidance for constructing models that balance accuracy, efficiency, and scalability.

III. Methodology

A. Overall Architecture

Building on these insights, we propose S2AFormer, an efficient Transformer-based backbone featuring a hybrid architecture that effectively preserves both local and global receptive fields while significantly reducing computational overhead. The overall structure of S2AFormer, as illustrated in Fig. 3,



(a) Self-attention (b) Separable Self-attention (c) Swift Self-attention (d) Conv Additive Self-attention (e) **Strip Self-attention**

Fig. 4: Comparison of different self-attention mechanisms: (a) Vanilla self-attention in ViTs [2], which computes global attention using standard dot-product operations. (b) Separable self-attention in MobileViT-v2 [49], which applies element-wise operations on query (Q) and key (K) to form a context vector. (c) Swift self-attention in SwiftFormer [9], where Q is weighted and pooled into global queries, broadcast, and multiplied element-wise with K to generate global context. (d) Convolutional additive self-attention in CAS-ViT [47], which replaces the global dot-product with a cascaded design applying spatial attention followed by channel attention. (e) Our proposed strip self-attention jointly compresses spatial and channel dimensions to effectively eliminate redundant information, achieving a lightweight design while preserving dense global dependencies.

adopts the four-stage framework commonly used in traditional CNNs [50] and hierarchical ViTs [3], enabling the generation of multi-scale feature maps suitable for dense prediction tasks such as object detection and semantic segmentation.

Our approach begins with a Stem Block consisting of three 3×3 convolutional layers, with the first layer using a stride of 2 [51]. This is followed by four stages, each responsible for generating feature maps at different scales. These stages share a consistent architecture, featuring a patch embedding and multiple L_i Hybrid Perception Blocks (HPBs). The HPBs, equipped with two core modules, SSA and LIM, effectively capture both local and global dependencies without the heavy computation overhead.

The resulting feature maps, F_1, F_2, F_3, and F_4, have strides of 4, 8, 16, and 32 pixels relative to the input image. By leveraging this feature pyramid, our approach integrates seamlessly with various downstream tasks, including image classification, object detection, instance segmentation, and semantic segmentation.

B. Hybrid Perception Block (HPB)

The Hybrid Perception Block (HPB) framework, shown in Fig. 3 (c), consists of two main components: the Strip Self-Attention (SSA) module in Fig. 3 (d) and the Local Interaction Module (LIM) in Fig. 3 (e). The overall operation is defined as follows:

$$f_{Conv} = DWConv\left(x\right) + x, \tag{1}$$

where $x \in \mathbb{R}^{H \times W \times C}$, $H \times W$ represents the resolution of the input at the current stage, while C denotes the feature dimension, and $DWConv$ means the depth-wise convolution. The SSA module can be formulated as:

$$f_{ssa} = SSA\left(LN\left(f_{Conv}\right)\right) + f_{Conv}, \tag{2}$$

Algorithm 1 Pseudocode of the Strip Self-Attention Module

procedure SSA(x, H, W)
  $Q \leftarrow$ Channel Squeeze(Linear Projection(x))
  if sr_ratio > 1 **then**
    $\widetilde{x} \leftarrow$ Spatial Reduction(x)
    $K \leftarrow$ Channel Squeeze(Linear Projection($\widetilde{x}$))
    $V \leftarrow$ Linear Projection($\widetilde{x}$)
  else
    $K \leftarrow$ Channel Squeeze(Linear Projection(x))
    $V \leftarrow$ Linear Projection(x)
  end if
  $attn \leftarrow$ Softmax($(Q \cdot K^T)$/Scale)
  $y \leftarrow$ Dropout(Linear Projection($attn \cdot V$))
  return y
end procedure

where SSA denotes Strip Self-Attention operation, and LN represents the Layer Norm.

To compensate for the limited local perceptual capacity of self-attention, we integrate a specialized Local Interaction Module, LIM, into the HPB, seamlessly combining global and local receptive fields to improve overall performance. The LIM can be formulated as

$$f_{lim} = LIM\left(LN\left(f_{ssa}\right)\right) + f_{ssa}. \tag{3}$$

Finally, the MLP is conducted, as:

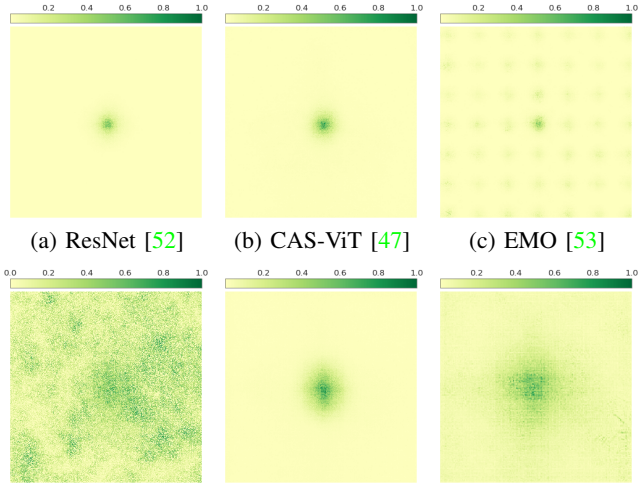
$$f_{mlp} = MLP\left(LN\left(f_{lim}\right)\right) + f_{lim}. \tag{4}$$

Next, we elaborate on the details of these components in the following sections.

1) Strip Self-Attention (SSA): In the standard self-attention mechanism [2], the input $x \in \mathbb{R}^{N \times C}$, where $N = HW$, is linearly projected into the query $Q \in \mathbb{R}^{N \times d_q}$, the key $K \in$

TABLE III: Complexity comparison between Multi-Head Self-Attention (MHSA) and Strip Self-Attention (SSA).

Stage	Operation	Computation Complexity	
		MHSA	SSA
Linear Projection	$Q = xW_Q \in \mathbb{R}^{N\times d_q}$ $K = xW_K \in \mathbb{R}^{N\times d_k}$ $V = xW_V \in \mathbb{R}^{N\times d_v}$	$\mathcal{O}(Nd(d_q + d_k + d_v))$	$\mathcal{O}(Nd(h + \frac{h}{k^2} + \frac{d_v}{k^2}))$
Attention Scores	$QK^T \in \mathbb{R}^{N\times N}$	$\mathcal{O}(N^2 d_q)$	$\mathcal{O}(\frac{N^2}{k^2} \cdot h)$
Weighted Sum	$Attn \cdot V \in \mathbb{R}^{N\times d_v}$	$\mathcal{O}(N^2 d_v)$	$\mathcal{O}(\frac{N^2}{k^2} d_v)$
Total Complexity	$h \ll d_q = d_k = d_v = d$	$\mathcal{O}(3Nd^2 + 2dN^2)$	$\mathcal{O}(\frac{1}{k^2}Nd^2 + \frac{d+h}{k^2}N^2 + (1+\frac{1}{k^2})Ndh)$ $\ll (1+\frac{2}{k^2})Nd^2 + \frac{2}{k^2}dN^2$ $< 3Nd^2 + 2dN^2$



(a) ResNet [52] (b) CAS-ViT [47] (c) EMO [53]

(d) Vanilla ViT [2] (e) SwiftFormer [9] (f) **Ours**

Fig. 5: Visualization of effective respective fields (ERFs) across different models. Convolution-based models (a), (b), and (c) exhibit highly localized receptive fields, while Vanilla ViT (d) distributes attention broadly across all spatial positions. Model (e) demonstrates more limited receptive fields compared to ours. In contrast, our method (f) achieves a balanced pattern—effectively capturing key local regions and progressively expanding outward in a strip-like manner.

$\mathbb{R}^{N\times d_k}$, and the value $V \in \mathbb{R}^{N\times d_v}$. Subsequently, the self-attention mechanism is executed as follows:

$$Attn\,(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \tag{5}$$

To reduce computational complexity, a Convolution Spatial Reduction (CSR) [17] is employed, which uses a $k \times k$ depth-wise convolution with a stride of k to downsample the spatial dimensions of x before the linear projection operation. This results in $\widetilde{x} \in \mathbb{R}^{N_s\times C}$, where $N_s = \frac{N}{k^2}$. In our work, we propose a Strip Self-Attention (SSA) module as an innovative token mixing mechanism in the HPB to balance global feature extraction and the computational efficiency of self-attention.

As shown in Fig. 3 (d), the channel dimensions of the query Q and key K are compressed into a single dimension to further minimize computational overhead. Experimental results demonstrate that a strip-shaped configuration for $Q \in \mathbb{R}^{N\times 1}$ and $K \in \mathbb{R}^{N_s\times 1}$ can effectively capture global similarities. The pseudocode for the entire SSA operation is presented in Algorithm 1, followed by a detailed complexity analysis.

Fig. 4 compares the proposed Strip Self-Attention with several notable self-attention variants from prior works. (a) shows the vanilla self-attention used in ViTs [2], which computes global attention via standard dot-product operations. (b) depicts the separable self-attention in MobileViT-v2 [49], which forms a context vector via element-wise interactions between query (Q) and key (K), then multiplies it with value (V) to produce the output. (c) presents the swift self-attention in SwiftFormer [9], an additive variant that relies solely on Q and K: the query matrix is reweighted with learnable parameters, pooled into global queries, broadcast, and multiplied element-wise with K to obtain a global context. (d) illustrates the convolutional additive self-attention in CAS-ViT [47], which replaces the core token mixer with a sequence of convolutional spatial attention followed by channel attention, inspired by CBAM [48]. Our Strip Self-Attention differs in two key respects. Firstly, the mechanisms in (b) and (c) operate at the attention-map level using element-wise products rather than full matrix attention, limiting their ability to model true global context. Secondly, CAS-ViT's [47] purely convolutional path struggles to capture dense token correlations. In contrast, our method jointly reduces redundancy in spatial and channel dimensions, minimizing computational load while preserving global awareness and effectively modeling dense token interactions.

To facilitate clearer understanding, the effective receptive fields (ERFs) of various architectures are distinctly visualized in Fig. 5. Convolution-based methods display highly localized receptive fields, which restrict their ability to capture global context. In contrast, the Vanilla ViT [2] covers all spatial positions, allowing for comprehensive global information extraction, but at the cost of significantly increased computational complexity. Our method achieves a balance between these two extremes. As illustrated, the receptive field in our approach initially focuses on the most critical regions and then progressively expands in a strip-like manner along the

TABLE IV: Comparison of image classification performance on the ImageNet-1K dataset [11]. *Mixer* refers to the token mixer operation, *Res.* indicates the input resolution of training, and *#Para.* corresponds to the total number of parameters.

Model	Mixer	Res.	#Para.(M)↓	GMACs↓	Top-1(%)↑
MobileFormer-52M [42]	Conv+Attn	224^2	3.50	0.052	68.7
MobileViT-XXS [41]	Conv+Attn	256^2	1.30	0.364	69.0
MobileViT-v2-0.5 [49]	Attn	256^2	1.40	0.500	70.2
EfficientViT-M2 [54]	Conv+Attn	224^2	4.20	0.201	70.8
EdgeNeXt-XXS [32]	Conv+Attn	256^2	1.30	0.261	71.2
EMO-1M [53]	Conv	224^2	1.30	0.260	71.5
FasterNet-T0 [55]	Conv+Attn	224^2	3.90	0.340	71.9
MobileNet-v3 [56]	Conv	256^2	1.40	0.481	72.3
EMOv2-1M [8]	Conv	224^2	1.40	0.285	72.3
SHViT-S1 [25]	Conv+Attn	224^2	6.30	0.240	72.8
Vim-Ti [57]	Conv+SSM	224^2	7.00	1.500	73.1
MobileMamba-T2 [58]	Conv+SSM	192^2	8.80	0.255	73.6
iFormer-T [10]	Conv+Attn	224^2	2.90	0.530	74.1
LSNet-T [59]	Conv+Attn	224^2	11.40	0.300	74.9
VRWKV-T [60]	Conv+Attn	224^2	6.20	1.200	75.1
S2AFormer-mini (Ours)	**Conv+Attn**	224^2	**5.02**	**0.430**	**75.1**
S2AFormer-mini (Ours)	**Conv+Attn**	256^2	**5.02**	**0.560**	**75.7**
EdgeViT-XXS [38]	Conv+Attn	224^2	4.10	0.600	74.4
DeiT-T [4]	Attn	224^2	5.90	1.200	74.5
MobileViT-XS [41]	Conv+Attn	224^2	2.30	0.706	74.8
SHViT-S2 [25]	Conv+Attn	224^2	11.40	0.370	75.2
InceptionNeXt-A [61]	Conv	224^2	4.20	0.510	75.3
FastViT-T8 [62]	Conv+Attn	256^2	3.60	0.700	75.6
EMOv2-2M	Conv	224^2	2.30	0.487	75.8
MobileMamba-T4 [58]	Conv+SSM	192^2	14.20	0.413	76.1
EfficientVMamba-T [63]	Conv+SSM	224^2	6.00	0.800	76.5
EfficientViT-M5 [54]	Conv+Attn	224^2	12.40	0.522	77.1
PoolFormer-S12 [23]	Pool	224^2	12.00	1.900	77.2
SHViT-S3 [25]	Conv+Attn	224^2	14.20	0.600	77.4
LSNet-S [59]	Conv+Attn	224^2	16.10	0.500	77.8
FAT-B0 [64]	Conv+Attn	224^2	4.50	0.700	77.6
S2AFormer-T (Ours)	**Conv+Attn**	224^2	**5.80**	**0.655**	**77.7**
S2AFormer-T (Ours)	**Conv+Attn**	256^2	**5.80**	**0.855**	**78.3**
MobileViG-S [65]	Conv+GNN	224^2	7.20	1.000	78.2
EdgeViT-XS [38]	Conv+Attn	224^2	6.70	1.100	77.5
MobileFormer-294M [42]	Conv+Attn	224^2	11.40	0.294	77.9
SwiftFormer-S [9]	Conv+Attn	224^2	6.10	1.000	78.5
MobileViT-v2-1.0 [49]	Attn	256^2	4.90	1.800	78.1
EdgeNeXt-S [32]	Conv+Attn	224^2	5.60	0.965	78.8
MobileMamba-S6 [58]	Conv+SSM	224^2	15.00	0.652	78.0
RepViT-M0.9 [24]	Conv	224^2	5.10	0.800	78.7
FastViT-T12 [62]	Conv+Attn	256^2	6.80	1.400	79.1
EfficientFormer-L1 [31]	Conv+Attn	224^2	12.30	1.300	79.2
EfficientMod-xs [35]	Conv	224^2	6.60	0.800	78.3
EfficientVMamba-S [63]	Conv+SSM	224^2	11.00	1.300	78.7
MobileViT-S [41]	Conv+Attn	224^2	5.60	2.010	78.4
StarNet-S4 [66]	Conv	224^2	7.50	1.075	78.4
S2AFormer-XS (Ours)	**Conv+Attn**	224^2	**6.54**	**0.786**	**78.9**
S2AFormer-XS (Ours)	**Conv+Attn**	256^2	**6.54**	**1.030**	**79.3**

Model	Mixer	Res.	#Para.(M)↓	GMACs↓	Top-1(%)↑
EdgeNeXt-S [32]	Conv+Attn	256^2	5.60	1.300	79.4
SHViT-S4 [25]	Conv+Attn	256^2	16.50	0.990	79.4
MobileMamba-B1 [58]	Conv+SSM	256^2	17.10	1.080	79.9
PoolFormer-S24 [23]	Pool	224^2	21.00	3.500	80.3
FAT-B1 [64]	Conv+Attn	224^2	7.80	1.200	80.1
VRWKV-S [60]	Conv+Attn	224^2	23.80	4.600	80.1
Vim-S [57]	Conv+SSM	224^2	26.00	5.100	80.3
LSNet-B [59]	Conv+Attn	224^2	23.20	1.300	80.3
MobileViG-M [65]	Conv+GNN	224^2	14.00	1.500	80.6
SwiftFormer-L1 [9]	Conv+Attn	224^2	12.10	1.620	80.9
EfficientViT-M5 [54]	Conv+Attn	512^2	12.40	2.670	80.8
FastViT-SA12 [62]	Conv+Attn	256^2	10.90	1.900	80.6
RepViT-M1.1 [24]	Conv	224^2	8.20	1.300	80.7
EdgeViT-S [38]	Conv+Attn	224^2	11.10	1.900	81.0
EfficientMod-s [35]	Conv	224^2	12.90	1.400	81.0
S2AFormer-S (Ours)	**Conv+Attn**	224^2	**10.69**	**1.380**	**80.8**
S2AFormer-S (Ours)	**Conv+Attn**	256^2	**10.69**	**1.800**	**81.3**
RMT-T [67]	Conv	224^2	14.00	2.500	82.4
TransNeXt-Micro [68]	Conv+Attn	224^2	12.80	2.700	82.5
EfficientFormer-L3 [31]	Conv+Attn	224^2	31.30	3.900	82.4
MobileViG-Ti [65]	Conv+GNN	224^2	26.70	2.800	82.6
RepViT-M1.5 [24]	Conv	224^2	14.00	2.300	82.3
CAS-ViT-M [47]	Conv	224^2	12.42	1.887	81.4
MobileMamba-B2 [58]	Conv+SSM	384^2	17.10	2.427	81.6
VRWKV-B [60]	Conv+Attn	224^2	93.70	18.200	82.0
SHViT-S4 [25]	Conv+Attn	512^2	16.50	3.970	82.0
MobileMamba-B4 [58]	Conv+SSM	512^2	17.10	4.313	82.5
MambaOut-Tiny [69]	Conv	224^2	27.00	4.500	82.7
EfficientVMamba-B [63]	Conv+SSM	224^2	33.00	4.000	81.8
InceptionNeXt-T [61]	Conv	224^2	28.00	4.200	82.3
PeLK-T [70]	Conv	224^2	29.00	5.600	82.6
S2AFormer-M (Ours)	**Conv+Attn**	224^2	**24.87**	**4.120**	**82.3**
S2AFormer-M (Ours)	**Conv+Attn**	256^2	**24.87**	**5.380**	**82.7**
ConvNeXt-B [71]	Conv	224^2	89.00	15.400	83.8
Swin-B [3]	Attn	224^2	88.00	15.400	83.5
EfficientFormer-L7 [31]	Conv+Attn	224^2	82.10	10.200	83.3
PlainMamba-L3 [72]	Conv+SSM	224^2	50.00	14.400	82.3
Focal-Base [73]	Attn	224^2	89.80	16.000	83.8
GroupMixFormer-B [74]	Attn	224^2	45.80	17.600	84.7
CAFormer-M36 [75]	Attn	224^2	56.00	13.200	85.2
VRWKV-L [60]	Conv+Attn	384^2	334.90	189.500	86.0
RAVLT-L [33]	Attn	224^2	95.00	16.000	85.8
XCiT-M24/16 [34]	Attn	224^2	84.00	16.200	84.3
InceptionNeXt-B [61]	Conv	224^2	87.00	14.900	84.0
PoolFormer-M48 [23]	Conv	224^2	73.00	11.600	82.5
Conv2Former-B [76]	Conv+Attn	224^2	90.00	15.900	84.4
TransNeXt-Base [68]	Conv+Attn	224^2	89.70	18.400	84.8
S2AFormer-L (Ours)	**Conv+Attn**	224^2	**76.58**	**12.530**	**85.6**
S2AFormer-L (Ours)	**Conv+Attn**	256^2	**76.58**	**16.370**	**86.0**

four cardinal directions, effectively emulating human visual perception mechanisms.

2) Local Interaction Module (LIM): Vision Transformers (ViTs) face two main limitations compared to CNNs. First, absolute positional encoding disrupts translation invariance by assigning unique encodings to each patch. This issue can be mitigated by using relative or no positional encoding. Second, ViTs often overlook local relationships and structural information within patches. To overcome this, we propose a Local Interaction Module (LIM) to effectively capture local features, which is formulated as

$$f_{ds} = DWConv\left(\sigma\left(PWConv\left(f\right)\right)\right), \tag{6}$$

where $DWConv$ refers to depth-wise convolution, while $PWConv$ represents point-wise convolution, together forming the structure of depth-wise separable convolution. The symbol σ typically denotes the activation function $ReLU$. The final output of LIM can be expressed as:

$$f_{loc} = PWConv\left(\sigma\left(SE\left(f_{ds}\right)\right)\right), \tag{7}$$

where SE denotes the channel attention module [22].

Due to its depth-wise architectural design, the proposed LIM has an almost negligible computational footprint. It has minimal parameter overhead and minimal impacts on inference throughput and memory efficiency.

C. Complexity Analysis

To demonstrate the efficiency of our proposed SSA, we compared its computational overhead with standard self-attention [2]. Table III presents a detailed computational complexity comparison of the two attention mechanisms.

Specifically, given an input feature $x \in \mathbb{R}^{N\times d}$, the total computational cost of Multi-Head Self-Attention (MHSA), measured in FLOPs, is given by

$$\mathcal{O}(\text{MHSA}) = Nd(d_q + d_k + d_v) + N^2(d_q + d_v). \tag{8}$$

When assuming $d_q = d_k = d_v = d$, this simplifies to

$$\mathcal{O}(\text{MHSA}) = 3Nd^2 + 2N^2d. \tag{9}$$

In the case of our SSA, a spatial reduction is introduced, with $N_s = \frac{N}{k^2}$, where k represents the kernel size of the spatial reduction convolution.

TABLE V: Comparison of semantic segmentation performance on the ADE20K dataset [77].

Model	Semantic FPN [78]		
	#Para.(M)↓	GFLOPs↓	mIoU(%)↑
ResNet-18 [50]	-	-	32.9
EMO-1M [53]	5	23	34.2
PVT-T [17]	17	33	35.7
ResNet-50 [50]	29	46	36.7
S2AFormer-mini (Ours)	**6**	**23**	**36.7**
PoolFormer-S12 [23]	16	31	37.2
PVTv2-B0 [18]	8	25	37.2
EMO-2M [53]	6	24	37.3
CASViT-XS [47]	7	24	37.1
S2AFormer-T (Ours)	**7**	**25**	**38.0**
FastViT-SA12 [62]	14	29	38.0
ResNet-101 [50]	48	65	38.8
EfficientFormer-L1 [31]	16	28	38.9
ResNeXt-101-32x4d [79]	47	65	39.7
S2AFormer-XS (Ours)	**8**	**26**	**39.2**
LSNet-T [59]	-	-	40.1
ResNeXt-101-64x4d [79]	86	104	40.2
EMO-5M [53]	9	26	40.4
PVT-S [17]	28	45	39.8
RepViT-M1.1 [24]	-	-	40.6
PoolFormer-S24 [23]	25	39	40.3
S2AFormer-S (Ours)	**12**	**28**	**40.8**
SwiftFormer-L1 [9]	16	30	41.4
EfficientViM-M4 [80]	-	-	41.3
PoolFormer-S36 [23]	35	48	42.0
EfficientFormerV2-S2 [51]	16	28	42.4
iFormer-M [10]	-	-	42.4
PVTv2-B1 [18]	18	34	42.5
MobileMamba-B4 [81]	20	-	42.5
PoolFormer-M48 [23]	77	82	42.7
InceptionNeXt-T [61]	28	44	43.1
FastViT-SA36 [62]	34	44	42.9
LSNet-B [59]	-	-	43.0
EfficientMod-S [35]	33	-	43.5
S2AFormer-M (Ours)	**26**	**43**	**43.7**

Thus, the total computational cost for SSA is expressed as

$$\mathcal{O}(\text{SSA}) = \frac{1}{k^2}Nd^2 + \frac{h+d}{k^2}N^2 + \left(1+\frac{1}{k^2}\right)Ndh. \quad (10)$$

For $h \ll d$, the dominant terms simplify to

$$\mathcal{O}(\text{SSA}) \ll \frac{1}{k^2}Nd^2 + \frac{2d}{k^2}N^2 + \left(1+\frac{1}{k^2}\right)Nd^2. \quad (11)$$

Let $\mathcal{O}(\text{A}) = \frac{1}{k^2}Nd^2 + \frac{2d}{k^2}N^2 + \left(1+\frac{1}{k^2}\right)Nd^2$, the complexity can be further simplified as

$$\begin{aligned}\mathcal{O}(\text{A}) &= \left(1+\frac{2}{k^2}\right)Nd^2 + \frac{2d}{k^2}N^2 \\ &< 3Nd^2 + 2dN^2.\end{aligned} \quad (12)$$

Eq. 12 shows that the computational cost of SSA is significantly lower than that of MHSA, especially when the reduction factor k is large, satisfying

$$\mathcal{O}(\text{SSA}) \ll \mathcal{O}(\text{A}) < \mathcal{O}(\text{MHSA}). \quad (13)$$

IV. Experiments

In this section, we evaluate the proposed S2AFormer architecture through extensive experiments on large-scale image classification datasets and further assess its adaptability across various downstream vision tasks, including semantic segmentation, object detection, and instance segmentation. We first benchmark S2AFormer against existing state-of-the-art approaches, and then conduct ablation studies to identify the key components contributing to its effectiveness.

A. Image Classification

We train the proposed S2AFormer models from scratch on ImageNet-1K [11] without using pre-trained weights or additional data. Following the strategy outlined in EdgeNeXt [32], we use a 224×224 input resolution and train for 300 epochs on four NVIDIA H100 GPUs (96 GB memory). The implementation is done in PyTorch 2.5.1 with Timm 0.4.9, using AdamW [84] as the optimizer and a batch size of 1024. The initial learning rate is set to 6×10^{-3} and follows a cosine decay schedule with a 20-epoch warmup. Data augmentation includes label smoothing of 0.1, random resize cropping, horizontal flipping, RandAugment, and a multi-scale sampler. Additionally, we apply EMA [85] with a momentum factor of 0.9995.

The results on the ImageNet-1K [11] dataset, shown in Table IV, highlight the significant improvements achieved by S2AFormer in image classification. Compared to state-of-the-art approaches, S2AFormer enhances classification accuracy while effectively balancing model complexity and computational efficiency. S2AFormer-mini has 1.28M fewer parameters than that of SHViT-S1 [25] but outperforms it by 2.3% in accuracy. S2AFormer-T achieves 77.7% Top-1 accuracy, significantly surpassing models like SHViT-S3 [25], which has more than twice as many parameters (14.2M) and achieves only 77.4% accuracy with nearly identical MACs. Similarly, EfficientVMamba-T [63], the top-performing model in the Mamba series, falls short by 1.2% in accuracy compared to S2AFormer-T, while requiring 0.15G more MACs. S2AFormer-XS outperforms StarNet-S4 [66], which has 7.5M parameters, 1.075 GMACs, and 78.4% accuracy, while being more efficient in terms of parameters and MACs. S2AFormer-S has a slight disadvantage of 0.2% lower Top-1 accuracy compared to EdgeViT-S [38] (81.0% vs 80.8%), but it achieves this with much fewer parameters and a reduction of 0.5 GMACs, making it more lightweight. In comparison to InceptionNeXt-T [61], which has 28M parameters, 4.2 GMACs, and 82.3% accuracy, S2AFormer-M is 3.13M smaller in terms of parameter count while delivering the same Top-1 accuracy, further emphasizing its efficiency and reduced computational burden.

B. Semantic Segmentation

We evaluate the models' performance in semantic segmentation using the ADE20K dataset [77], a challenging benchmark for scene parsing. The dataset comprises 20,000 training images and 2,000 validation images, covering 150 detailed semantic categories. The proposed model is tested with Semantic FPN [78] as the backbone, with normalization layers frozen and pre-trained weights from ImageNet-1K [11] classification. Following standard practices [78], [86], the network is trained for 40,000 iterations with a batch size of 32, using the AdamW [84] optimizer. The initial learning rate is set to 2×10^{-4} and follows a polynomial decay schedule

TABLE VI: Comparison of object detection and instance segmentation performance (%) on the COCO val2017 dataset [82]. FLOPs are tested on images of size 800×1280. Our results, highlighted in bold, demonstrate superior performance with comparative computational overhead.

Model	#Para.(M)↓	GFLOPs↓	RetinaNet 1×						Mask R-CNN 1×					
			AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
MobileNetV2 [83]	-/-	-/-	28.3	46.7	29.3	14.8	30.7	38.1	29.6	48.3	31.5	27.2	45.2	28.6
MobileNetV3 [43]	-/-	-/-	29.9	49.3	30.8	14.9	33.3	41.1	29.2	48.6	30.3	27.1	45.5	28.2
ResNet-18 [50]	21/31	-/-	31.8	49.6	33.6	16.3	34.3	43.2	34.0	54.0	36.7	31.2	51.0	32.7
EfficientViT-M4 [54]	9/-	-/-	32.7	52.2	34.1	17.6	35.3	46.0	32.8	54.4	34.5	31.0	51.2	32.2
S2AFormer-mini (Ours)	**12/22**	**159/177**	**33.4**	**53.2**	**34.9**	**19.9**	**36.3**	**44.5**	**33.4**	**55.4**	**35.2**	**31.7**	**52.5**	**33.3**
LSNet-T [59]	-/-	-/-	34.2	54.6	35.2	17.8	37.1	48.5	35.0	57.0	37.3	32.7	53.8	34.3
PoolFormer-S12 [23]	22/32	-/-	36.2	56.2	38.2	20.8	39.1	48.0	37.3	59.0	40.1	34.6	55.8	36.9
CAS-ViT-XS [47]	12/23	162/181	36.5	56.3	38.9	21.8	39.9	48.4	37.4	59.1	40.4	34.9	56.2	37.0
ResNet-50 [50]	38/44	-/260	36.3	55.3	38.6	19.3	40.0	48.8	38.0	58.6	41.4	34.4	55.1	36.7
FastViT-SA12 [62]	-/-	-/-	-	-	-	-	-	-	38.9	60.5	42.2	35.9	57.6	38.1
SHViT-S3 [25]	-/-	-/-	36.1	56.6	38.0	19.9	39.1	50.8	36.9	59.4	39.6	34.4	56.3	36.1
S2AFormer-T (Ours)	**12/23**	**164/182**	**36.7**	**57.0**	**39.1**	**21.1**	**39.7**	**48.6**	**37.6**	**59.8**	**40.6**	**35.4**	**57.2**	**37.6**
PVT-T [17]	23/33	221/240	36.7	56.9	38.9	22.6	38.8	50.0	36.7	59.2	39.3	35.1	56.7	37.3
LSNet-S [59]	-/-	-/-	36.7	57.2	38.6	20.0	39.7	51.8	37.4	59.9	39.8	34.8	56.8	36.6
MF-508M [42]	8/-	168/-	38.0	58.3	40.3	22.9	41.2	49.7	-	-	-	-	-	-
EfficientFormer-L1 [31]	-/32	-/196	-	-	-	-	-	-	37.9	60.3	41.0	35.4	57.3	37.3
PVTv2-B0 [18]	13/24	-/-	37.2	57.2	39.5	23.1	40.4	49.7	38.2	60.5	40.7	36.2	57.8	38.6
S2AFormer-XS (Ours)	**13/24**	**166/185**	**37.9**	**58.6**	**40.3**	**22.9**	**41.5**	**49.8**	**38.4**	**60.2**	**41.5**	**35.8**	**57.3**	**38.1**
SHViT-S4 [25]	-/-	-/-	38.8	59.8	41.1	22.0	42.4	52.7	39.0	61.2	41.9	35.9	57.9	37.9
EMO-5M [53]	-/-	-/-	38.9	59.8	41.0	23.8	42.2	51.7	39.3	61.7	42.4	36.4	58.4	38.7
ResNet-101 [50]	57/63	315/336	38.5	57.8	41.2	21.4	42.6	51.1	40.4	61.1	44.2	36.4	57.7	38.8
EfficientViM-M4 [80]	-/-	-/-	38.8	59.6	41.1	22.1	42.4	52.8	39.3	60.2	42.5	35.8	57.1	37.4
PoolFormer-S24 [23]	31/41	-/233	38.9	59.7	41.3	23.3	42.1	51.8	40.1	62.2	43.4	37.0	59.1	39.6
PoolFormer-S36 [23]	41/51	-/272	39.5	60.5	41.8	22.5	42.9	52.4	41.0	63.1	44.8	37.7	60.1	40.0
RepViT-M1.1 [24]	-/-	-/-	-	-	-	-	-	-	39.8	61.9	43.5	37.2	58.8	40.1
S2AFormer-S (Ours)	**17/28**	**178/197**	**40.0**	**60.9**	**42.7**	**24.4**	**43.6**	**52.9**	**41.0**	**62.5**	**45.0**	**37.6**	**59.7**	**40.3**
PVT-S [17]	34/44	286/305	38.7	59.3	40.8	21.2	41.6	54.4	40.4	62.9	43.8	37.8	60.1	40.3
EfficientFormer-L3 [31]	-/51	-/250	-	-	-	-	-	-	41.4	63.9	44.7	38.1	61.0	40.4
SwiftFormer-L1 [9]	-/31	-/202	-	-	-	-	-	-	41.2	63.2	44.8	38.1	60.2	40.7
Swin-T [3]	38/48	248/267	41.5	62.1	44.2	25.1	44.9	55.5	42.2	64.6	46.2	39.1	61.6	42.0
EfficientFormer-L7 [31]	-/101	-/378	-	-	-	-	-	-	42.6	65.1	46.1	39.0	62.2	41.7
SwiftFormer-L3 [9]	-/48	-/252	-	-	-	-	-	-	42.7	64.4	46.7	39.1	61.7	41.8
S2AFormer-M (Ours)	**32/42**	**234/253**	**41.7**	**62.4**	**44.5**	**25.8**	**44.6**	**55.4**	**42.6**	**64.5**	**46.9**	**39.3**	**62.0**	**41.7**

with a power of 0.9. During training, input images are resized and cropped to 512×512. The implementation is based on the MMSegmentation [87] framework.

The comparative results shown in Table V demonstrate that all S2AFormer variants effectively balance computational efficiency and segmentation accuracy. S2AFormer-mini, with just 6M parameters and 23 GFLOPs, achieves a competitive mIoU of 36.7%, surpassing ResNet-18 [50] and PVT-T [17] by 3.8% and 1.0%, respectively. With 7M parameters and 25 GFLOPs, S2AFormer-T improves the mIoU to 38.0%, outperforming PoolFormer-S12 [23] (16M parameters, 31 GFLOPs, 37.2% mIoU) and CASViT-XS [47] (7M parameters, 24 GFLOPs, 37.1% mIoU) by 0.8% and 0.9%, respectively, while maintaining a smaller or comparable model size. S2AFormer-XS achieves 39.2% mIoU, surpassing EfficientFormer-L1 [31] by 0.3% using fewer parameters, demonstrating its high efficiency. S2AFormer-S offers better performance than PoolFormer-S24 [23] (40.8% vs 40.3%) with fewer parameters and lower GFLOPs. Similarly, S2AFormer-M outperforms InceptionNeXt-T [61] (28M parameters, 44 GFLOPs, 43.1% mIoU) by 0.6%, while being 7.14% smaller in parameter count.

C. Object Detection and Instance Segmentation

We assess the performance of S2AFormer on object detection and instance segmentation tasks using the COCO2017 dataset [82], which contains 118,000 training images and 5,000 validation images, along with bounding box and mask annotations across 80 categories. We assess the performance of our model with two widely used object detectors: RetinaNet [88] and Mask R-CNN [89]. To initialize the backbone, we use weights pre-trained on ImageNet. All models are trained with a batch size of 16 on four H100 GPUs, utilizing the AdamW optimizer [84] with an initial learning rate of 1×10^{-4}. Following standard practices [88]–[90], we adopt a $1\times$ (12 epochs) training schedule. The models are implemented using the MMDetection [90] framework.

We conducted a detailed evaluation on the COCO val2017 dataset [82], comparing backbone models based on Average Precision (AP) across various scales (small, medium, large) and tasks (detection and segmentation), along with their efficiency in terms of parameters (Para.) and FLOPs. As shown in Table VI, S2AFormer-mini achieves an impressive 33.4% AP on RetinaNet [88], surpassing models like ResNet-18 [50] and EfficientViT-M4 [54] while maintaining fewer parameters and lower GFLOPs. S2AFormer-T achieves 36.7%

Fig. 6: Qualitative results of semantic segmentation on ADE20K val dataset [77]. Upper: input validation images, Lower: results generated by S2AFormer-S-based Semantic FPN [78].

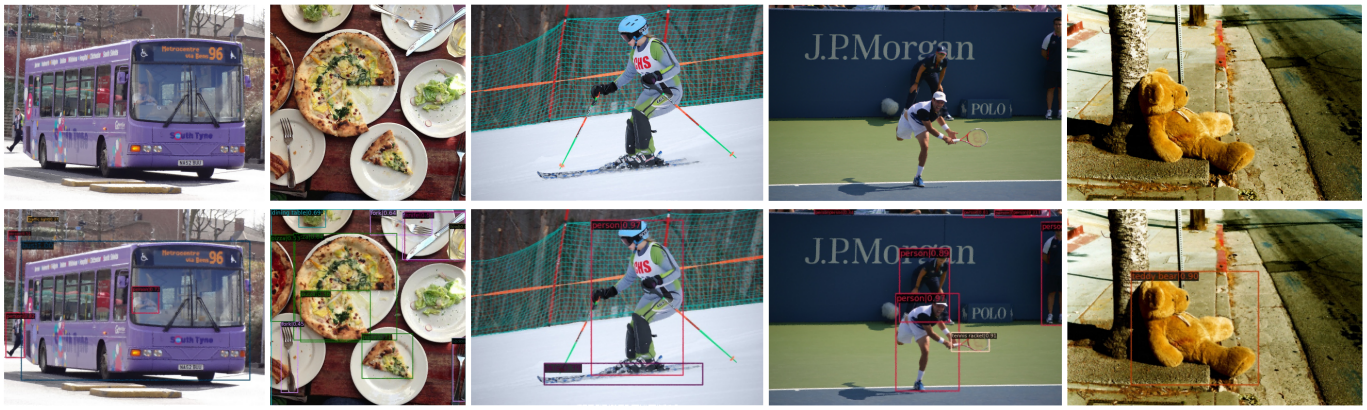


Fig. 7: Qualitative results of object detection on the COCO val2017 dataset [82]. Top: input validation images. Bottom: results generated by S2AFormer-S-based RetinaNet [88].

AP for RetinaNet [88] and 37.6% AP^b for Mask R-CNN [89], surpassing PoolFormer-S12 [23], which has 36.2% AP and 37.3% in AP^b, while maintaining a reduced computational burden. Under similar parameter conditions, S2AFormer-XS (with 13/24 parameters and 166/185 GFLOPs) achieves 37.9% AP for RetinaNet and 38.4% AP^b for Mask R-CNN, outperforming PVTv2-B0 [18] (with 13/24 parameters and 37.2% AP for RetinaNet and 38.2% AP^b for Mask R-CNN) by a noticeable margin. Similarly, S2AFormer-M achieves excellent performance across benchmarks.

D. Qualitative Results

The qualitative results in Fig. 6 highlight the efficacy of our method in generating visually coherent and fine-grained segmentation maps on the ADE20K validation set [77]. Complementing these findings, Figs. 7 and 8 showcase task-specific visualizations from our S2AFormer-S model, addressing instance segmentation and object detection challenges, respectively. These outcomes underscore the model's robustness in accurately localizing and delineating objects across diverse and cluttered scenes, even under varying scales and occlusions.

E. Inference Speed

We implemented S2AFormer on three different platforms and evaluated its efficiency through throughput analysis. Note that, to ensure fairness and reproducibility, we report GPU, CPU, and ONNX throughput using a single codebase and the same checkpoint, rather than referencing results from prior works. All experiments were conducted on an NVIDIA H100 NVL GPU and an Intel Xeon Gold 6426Y CPU, using PyTorch 2.5.1 (for GPU/CPU) and ONNX Runtime (for ONNX).

As shown in Table VII, for GPU-based inference, S2AFormer demonstrates near-optimal performance with comparable ImageNet-1K accuracy. For example, S2AFormer-mini achieves a throughput of 6330 images/s on the GPU, significantly outperforming other models. It exceeds the throughput of EMO-2M [53], which achieves 3359 images/s, making S2AFormer-mini nearly twice as fast in GPU throughput. This highlights its strong image processing capabilities compared to similar models like EfficientViT-M3 [54] (4355 images/s) and PVT-T [17] (5033 images/s).

In CPU and ONNX-based inference, S2AFormer also shows highly competitive performance. For example, S2AFormer-T surpasses EdgeViT-XS [38] in throughput on both CPU (41.75 images/s) and ONNX (112.89 images/s) platforms. This highlights the model's robustness in non-GPU environments, where inference speed can become a bottleneck. This is particularly important for real-world deployment scenarios, where model efficiency across various hardware platforms, including CPUs and optimized ONNX models, plays a crucial

Fig. 8: Qualitative results of instance segmentation on the COCO val2017 dataset [82]. Top: input validation images. Bottom: results generated by S2AFormer-S-based Mask R-CNN [89].

TABLE VII: Throughput comparison across GPU, CPU, and ONNX platforms. GPU: NVIDIA H100 NVL, CPU: Intel(R) Xeon(R) Gold 6426Y.

Model	#P.↓	Top-1↑	Throughput (images/s) GPU↑	CPU↑	ONNX↑
EMO-1M [53]	1.3	71.5	4674	45.57	118.72
EfficientViT-M3 [54]	6.9	73.4	4355	35.34	169.35
MobileMamba-T2 [81]	8.8	73.6	6279	63.75	178.38
EfficientViT-M4 [54]	8.8	74.3	4289	34.34	159.51
EMOv2-1M [8]	1.4	72.3	4886	45.70	112.98
EMO-2M [53]	2.3	75.1	3359	41.20	83.79
PVT-T [17]	13.2	75.1	5033	61.01	117.63
S2AFormer-mini	**5.0**	**75.1**	**6330**	**58.21**	**118.26**
FastViT-T8 [62]	3.6	75.6	7615	50.76	129.27
EfficientFormerV2-S0 [51]	3.5	75.7	520	44.57	136.32
MobileMamba-T4 [81]	14.2	76.1	5124	52.33	146.37
EdgeViT-XXS [38]	4.1	74.4	3614	50.82	134.79
PoolFormer-S12 [75]	12.0	77.2	2148	32.67	102.16
EfficientViT-M5 [54]	12.4	77.1	3843	28.48	126.28
EdgeViT-XS [38]	6.7	77.5	2872	35.14	106.10
S2AFormer-T	**5.8**	**77.7**	**4768**	**41.75**	**112.89**
EMO-5M [53]	5.1	78.4	2370	38.18	62.52
PVTv2-B1 [18]	14.0	78.7	3930	79.51	37.96
EMO-6M [53]	6.1	79.0	2102	35.90	57.00
StarNet-S4 [66]	7.5	78.4	2570	30.83	64.26
EfficientFormerV2S1 [51]	6.1	79.0	472	37.89	104.35
EfficientFormer-L1 [31]	12.3	79.2	3572	66.93	88.72
iFormer-S [10]	6.5	78.8	3764	32.89	66.30
S2AFormer-XS	**6.5**	**78.9**	**4185**	**34.09**	**71.63**
FastViT-T12 [62]	6.8	79.1	5277	42.06	88.31
FastViT-S12 [62]	8.8	79.8	4952	39.41	68.09
FastViT-SA12 [62]	10.9	80.6	4812	38.48	50.12
LSNet-B [59]	23.2	80.3	2890	27.20	52.88
PVT-S [17]	24.5	79.8	2736	43.50	53.01
EdgeViT-S [38]	11.1	81.0	1951	23.15	58.32
EfficientViT-M5 [54]	12.4	80.8	2396	29.05	54.62
S2AFormer-S	**10.7**	**80.8**	**2644**	**25.78**	**55.84**
PVT-M [17]	44.2	81.2	1717	25.96	36.89
EfficientFormerV2-S2 [51]	12.6	81.6	256	25.08	46.94
PVT-L [17]	61.4	81.7	587	18.84	27.77
EfficientVMamba-B [63]	33.0	81.8	2432	26.75	32.54
PVTv2-B2 [18]	25.4	82.0	2218	40.20	40.83
EfficientFormer-L3 [31]	31.3	82.4	1550	35.96	25.31
VRWKV-B [60]	93.7	82.0	1520	20.49	24.68
S2AFormer-M	**24.9**	**82.3**	**1652**	**15.50**	**27.15**

role in practical applications.

TABLE VIII: Performance comparisons with and without LIM on COCO val2017 dataset [82].

Model	LIM	Top-1↑	RetinaNet 1× #Para.(M)↓	GFLOPs↓	AP(%)↑
S2AFormer-mini	✗	74.89%	11.55	159.06	32.6
	✓	75.06%	11.65	159.20	33.4
S2AFormer-T	✗	77.63%	12.29	163.50	35.3
	✓	77.73%	12.44	163.71	36.7

TABLE IX: Performance comparisons between convolution and pooling operations for spatial reduction purposes using Top-1 accuracy and inference throughput.

Model	#Para.(M)↓	GMACs↓	Top-1↑	TP (imgs/s) GPU↑	CPU↑
S2AFormer-mini	5.02	0.43	75.1%	6330	58.21
Conv → Pooling	5.01	0.57	75.0%	3649	54.10
S2AFormer-T	5.80	0.66	77.7%	4768	41.75
Conv → Pooling	5.79	1.08	77.3%	1695	34.20

F. Ablation Studies

Our comparative experiments across three vision benchmarks demonstrate the robustness and efficiency of the proposed strip self-attention in S2AFormer. In addition to the core HPBs, we introduced a specialized LIM to compensate for the limited local perceptual capacity of self-attention. In this section, we conduct ablation studies to validate the effectiveness of both LIM and the convolution-based spatial reduction operation.

1) Effectiveness of LIM: As shown in Table VIII, for S2AFormer-Mini, the parameter count increases slightly from 11.55M to 11.65M, while the GFLOPs remain nearly unchanged, with values of 159.06 and 159.20, respectively. Initially, the model achieves a Top-1 accuracy of 74.89%. However, after incorporating LIM, there is a noticeable improvement in performance, with the Top-1 accuracy rising to 75.06%. A similar trend is observed for S2AFormer-T.

Without LIM, the model achieves a Top-1 accuracy of 77.63%, but with LIM, the accuracy increases slightly to 77.73%. The parameter count for this model increases modestly from 12.29M to 12.44M, and GFLOPs show a minor increase from 163.50 to 163.71. Despite these small increases in computational cost, the model's Average Precision (AP) improves significantly, with LIM boosting the AP from 35.3 to 36.7. This demonstrates that the improvements in accuracy come with minimal additional computational overhead, particularly in terms of detection performance.

2) Effectiveness of Conv Spatial Reduction: As shown in Table IX, our comparison between Convolution and Pooling for reducing spatial dimensions reveals that Convolution slightly outperforms Pooling in both the S2AFormer-mini and S2AFormer-T models. Specifically, S2AFormer-mini achieves a Top-1 accuracy of 75.1% with Convolution, which is slightly higher than the 75.0% obtained with Pooling. Similarly, S2AFormer-T reaches a Top-1 accuracy of 77.7% using Convolution, compared to 77.3% with Pooling.

In terms of efficiency, Convolution also demonstrates superiority, requiring fewer MACs: 0.43 GMACs versus 0.57 GMACs for S2AFormer-mini and 0.66 GMACs versus 1.08 GMACs for S2AFormer-T. Regarding inference speed, Convolution outpaces Pooling by a significant margin. On the GPU, S2AFormer-mini with Convolution processes 6330 images/s, compared to 3649 images/s with Pooling. Similarly, S2AFormer-T processes 4768 images/s with Convolution, versus 1695 images/s with Pooling. On the CPU, Convolution also leads, with 58.21 images/s for S2AFormer-mini and 41.75 images/s for S2AFormer-T, while Pooling reaches 54.10 images/s and 34.20 images/s, respectively.

In conclusion, convolution consistently delivers better accuracy, lower computational complexity, and faster inference speeds on both GPU and CPU, making it the more efficient option for spatial dimension reduction in our model.

G. Limitations and Future Works

While our proposed S2AFormer strikes an effective balance between efficiency and accuracy, there is still potential to refine the spatial–channel compression strategy. In future work, we aim to explore learnable token compression mechanisms that dynamically adapt to the content distribution in both spatial and channel dimensions. A promising approach is to integrate mask-guided autoregressive compression, inspired by MAE [91], to selectively retain informative tokens for global reasoning while discarding redundant ones. This adaptive compression could further reduce computational costs without sacrificing accuracy, enhancing the scalability and robustness of our framework for more complex dense prediction tasks.

V. Conclusion

In this work, we developed a new family of hybrid architecture that efficiently combines CNNs and Vision Transformers, called S2AFormer. Our approach reduces redundancies in both the spatial and channel dimensions, contributing to a more efficient architectural design. Extensive experiments across three downstream benchmarks have validated the effectiveness and superiority of S2AFormer over some state-of-the-art methods. The results highlight S2AFormer as a promising solution for achieving both high performance and efficiency in computer vision tasks.

References

[1] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proceedings of the European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[3] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[4] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proceedings of the International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[5] M. Maaz, H. Rasheed, S. Khan, F. S. Khan, R. M. Anwer, and M.-H. Yang, "Class-agnostic object detection with multi-modal transformer," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 512–531.

[6] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1290–1299.

[7] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[8] J. Zhang, T. Hu, H. He, Z. Xue, Y. Wang, C. Wang, Y. Liu, X. Li, and D. Tao, "Emov2: Pushing 5 m vision model frontier," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 11, pp. 10 560–10 576, 2025.

[9] A. Shaker, M. Maaz, H. Rasheed, S. Khan, M.-H. Yang, and F. S. Khan, "Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 425–17 436.

[10] C. Zheng, "Iformer: Integrating convnet and transformer for mobile application," *arXiv preprint arXiv:2501.15369*, 2025.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, pp. 248–255.

[12] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo, "Cswin transformer: A general vision transformer backbone with cross-shaped windows," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 124–12 134.

[13] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," *Advances in Neural Information Processing Systems*, vol. 34, pp. 13 937–13 949, 2021.

[14] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," *arXiv preprint arXiv:2202.07800*, 2022.

[15] Y. Wang, R. Huang, S. Song, Z. Huang, and G. Huang, "Not all images are worth 16x16 words: Dynamic vision transformers with adaptive sequence length," *arXiv preprint arXiv:2105.15075*, vol. 2, no. 3, p. 8, 2021.

[16] Y. Zhang, Y. Liu, D. Miao, Q. Zhang, Y. Shi, and L. Hu, "Mg-vit: a multi-granularity method for compact and efficient vision transformers," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[17] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 568–578.

[18] W. Wang, E. Xie, and X. Li, "Pvtv2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 1–10, 2022.

[19] S. Ren, X. Yang, S. Liu, and X. Wang, "Sg-former: Self-guided transformer with evolving token reallocation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6003–6014.

[20] Y. Xie and Y. Liao, "Efficient-vit: A light-weight classification model based on cnn and vit," in *Proceedings of the International Conference on Image and Graphics Processing*, 2023, pp. 64–70.

[21] B. Kang, S. Moon, Y. Cho, H. Yu, and S.-J. Kang, "Metaseg: Metaformer-based global contexts-aware network for efficient semantic segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 434–443.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[23] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan, "Metaformer is actually what you need for vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 819–10 829.

[24] A. Wang, H. Chen, Z. Lin, J. Han, and G. Ding, "Repvit: Revisiting mobile cnn from vit perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 909–15 920.

[25] S. Yun and Y. Ro, "Shvit: Single-head vision transformer with memory efficient macro design," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5756–5767.

[26] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[27] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z.-H. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 558–567.

[28] G. Xu, W. Jia, T. Wu, L. Chen, and G. Gao, "Haformer: Unleashing the power of hierarchy-aware features for lightweight semantic segmentation," *IEEE Transactions on Image Processing*, vol. 33, pp. 4204–4214, 2024.

[29] M. Kim, S. Gao, Y.-C. Hsu, Y. Shen, and H. Jin, "Token fusion: Bridging the gap between token pruning and token merging," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 1383–1392.

[30] Q. Tang, B. Zhang, J. Liu, F. Liu, and Y. Liu, "Dynamic token pruning in plain vision transformers for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 777–786.

[31] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 934–12 949, 2022.

[32] M. Maaz, A. Shaker, H. Cholakkal, S. Khan, S. W. Zamir, R. M. Anwer, and F. Shahbaz Khan, "Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 3–20.

[33] Q. Fan, H. Huang, and R. He, "Breaking the low-rank dilemma of linear attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 25 271–25 280.

[34] A. Ali, H. Touvron, M. Caron, P. Bojanowski, M. Douze, A. Joulin, I. Laptev, N. Neverova, G. Synnaeve, J. Verbeek *et al.*, "Xcit: Cross-covariance image transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 014–20 027, 2021.

[35] X. Ma, X. Dai, J. Yang, B. Xiao, Y. Chen, Y. Fu, and L. Yuan, "Efficient modulation for vision networks," *arXiv preprint arXiv:2403.19963*, 2024.

[36] Q. Zhou, K. Sheng, X. Zheng, K. Li, X. Sun, Y. Tian, J. Chen, and R. Ji, "Training-free transformer architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 894–10 903.

[37] S. Wei, T. Ye, S. Zhang, Y. Tang, and J. Liang, "Joint token pruning and squeezing towards more aggressive compression of vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2092–2101.

[38] Z. Chen, F. Zhong, Q. Luo, X. Zhang, and Y. Zheng, "Edgevit: Efficient visual modeling for edge computing," in *Proceedings of the International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2022, pp. 393–405.

[39] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.

[40] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[41] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[42] Y. Chen, X. Dai, D. Chen, M. Liu, X. Dong, L. Yuan, and Z. Liu, "Mobile-former: Bridging mobilenet and transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5270–5279.

[43] B. Koonce, *MobileNetV3*. Berkeley, CA: Apress, 2021, pp. 125–144. [Online]. Available: https://doi.org/10.1007/978-1-4842-6168-2_11

[44] J. Guo, K. Han, H. Wu, Y. Tang, X. Chen, Y. Wang, and C. Xu, "Cmt: Convolutional neural networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 12 175–12 185.

[45] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 22–31.

[46] M. Zhao, Y. Luo, and Y. Ouyang, "Repnext: A fast multi-scale cnn using structural reparameterization," *arXiv preprint arXiv:2406.16004*, 2024.

[47] T. Zhang, L. Li, Y. Zhou, W. Liu, C. Qian, and X. Ji, "Cas-vit: Convolutional additive self-attention vision transformers for efficient mobile applications," *arXiv preprint arXiv:2408.03703*, 2024.

[48] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[49] S. Mehta and M. Rastegari, "Separable self-attention for mobile vision transformers," *arXiv preprint arXiv:2206.02680*, 2022.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[51] Y. Li, J. Hu, Y. Wen, G. Evangelidis, K. Salahi, Y. Wang, S. Tulyakov, and J. Ren, "Rethinking vision transformers for mobilenet size and speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 16 889–16 900.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, 2012.

[53] J. Zhang, X. Li, J. Li, L. Liu, Z. Xue, B. Zhang, Z. Jiang, T. Huang, Y. Wang, and C. Wang, "Rethinking mobile block for efficient attention-based models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2023, pp. 1389–1400.

[54] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan, "Efficientvit: Memory efficient vision transformer with cascaded group attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 420–14 430.

[55] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 021–12 031.

[56] A. G. Howard, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[57] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, "Vision mamba: Efficient visual representation learning with bidirectional state space model," *arXiv preprint arXiv:2401.09417*, 2024.

[58] H. He, J. Zhang, Y. Cai, H. Chen, X. Hu, Z. Gan, Y. Wang, C. Wang, Y. Wu, and L. Xie, "Mobilemamba: Lightweight multi-receptive visual mamba network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 4497–4507.

[59] A. Wang, H. Chen, Z. Lin, J. Han, and G. Ding, "Lsnet: See large, focus small," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 9718–9729.

[60] Y. Duan, W. Wang, Z. Chen, X. Zhu, L. Lu, T. Lu, Y. Qiao, H. Li, J. Dai, and W. Wang, "Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures," *arXiv preprint arXiv:2403.02308*, 2024.

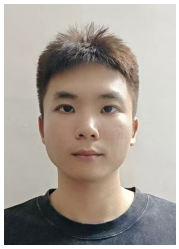
[61] W. Yu, P. Zhou, S. Yan, and X. Wang, "Inceptionnext: When inception meets convnext," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5672–5683.

[62] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, "Fastvit: A fast hybrid vision transformer using structural reparameterization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5785–5795.

[63] X. Pei, T. Huang, and C. Xu, "Efficientvmamba: Atrous selective scan for light weight visual mamba," *arXiv preprint arXiv:2403.09977*, 2024.

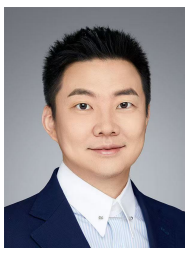
[64] Q. Fan, H. Huang, X. Zhou, and R. He, "Lightweight vision transformer with bidirectional interaction," *Advances in Neural Information Processing Systems*, vol. 36, pp. 15 234–15 251, 2023.
[65] M. Munir, W. Avery, and R. Marculescu, "Mobilevig: Graph-based sparse attention for mobile vision applications," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2211–2219.
[66] X. Ma, X. Dai, Y. Bai, Y. Wang, and Y. Fu, "Rewrite the stars," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5694–5703.
[67] Q. Fan, H. Huang, M. Chen, H. Liu, and R. He, "Rmt: Retentive networks meet vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 5641–5651.
[68] D. Shi, "Transnext: Robust foveal visual perception for vision transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 773–17 783.
[69] W. Yu and X. Wang, "Mambaout: Do we really need mamba for vision?" *arXiv preprint arXiv:2405.07992*, 2024.
[70] H. Chen, X. Chu, Y. Ren, X. Zhao, and K. Huang, "Pelk: Parameter-efficient large kernel convnets with peripheral convolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 5557–5567.
[71] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 976–11 986.
[72] C. Yang, Z. Chen, M. Espinosa, L. Ericsson, Z. Wang, J. Liu, and E. J. Crowley, "Plainmamba: Improving non-hierarchical mamba in visual recognition," *arXiv preprint arXiv:2403.17695*, 2024.
[73] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao, "Focal attention for long-range interactions in vision transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 30 008–30 022, 2021.
[74] C. Ge, X. Ding, Z. Tong, L. Yuan, J. Wang, Y. Song, and P. Luo, "Advancing vision transformers with group-mix attention," *arXiv preprint arXiv:2311.15157*, 2023.
[75] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, and X. Wang, "Metaformer baselines for vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 2, pp. 896–912, 2024.
[76] Q. Hou, C.-Z. Lu, M.-M. Cheng, and J. Feng, "Conv2former: A simple transformer-style convnet for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 8274–8283, 2024.
[77] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 633–641.
[78] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6399–6408.
[79] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.
[80] S. Lee, J. Choi, and H. J. Kim, "Efficientvim: Efficient vision mamba with hidden state mixer based state space duality," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 14 923–14 933.
[81] H. He, J. Zhang, Y. Cai, H. Chen, X. Hu, Z. Gan, Y. Wang, C. Wang, Y. Wu, and L. Xie, "Mobilemamba: Lightweight multi-receptive visual mamba network," *arXiv preprint arXiv:2411.15941*, 2024.
[82] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision*. Springer, 2014, pp. 740–755.
[83] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.
[84] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[85] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.
[86] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
[87] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," https://github.com/open-mmlab/mmsegmentation, 2020.
[88] T.-Y. Ross and G. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2980–2988.
[89] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
[90] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.
[91] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.

Guoan Xu received the M.S. degree from the College of Automation, Nanjing University of Posts and Telecommunications. He is currently a Ph.D. candidate at the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS). His main research interests include image segmentation, depth estimation, and multi-modal image processing.



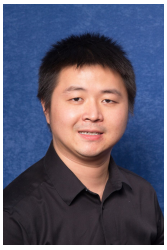
Wenfeng Huang is currently a Ph.D. candidate at the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Australia. His research interests include computer vision, medical image analysis, 3D reconstruction, and AI-generated content.

Wenjing Jia received her Ph.D. degree in Computing Sciences from the University of Technology Sydney in 2007. She is currently an Associate Professor at the Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney (UTS). Her research falls in the fields of image processing and analysis, computer vision, and pattern recognition.



Jiamao Li (Member, IEEE) received the Ph.D. degree in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 2013. He is currently a Professor and the director of the Biovision System Laboratory with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai, China. He has published more than 90 papers in journals and conferences. His main research interests include embodied intelligence, machine vision, edge computing chips, and microsystems for robotics.

Guangwei Gao (Senior Member, IEEE) received the Ph.D. degree in pattern recognition and intelligent systems from the Nanjing University of Science and Technology, Nanjing, in 2014. He is currently a Professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include pattern recognition and image understanding. He has published more than 70 scientific papers in IEEE-TPAMI/TIP/TCSVT/TITS/TMM, CVPR, AAAI, IJCAI, NeurIPS, etc. Personal website: *https://guangweigao.github.io*.



Guo-Jun Qi (Fellow, IEEE) has been a faculty member with the Department of Computer Science, University of Central Florida, since August 2014. Since August 2018, he has been the Chief Scientist, leading and overseeing the International Research and Development Team for multiple artificial intelligence services on the Huawei Cloud. He is currently a Professor and the Chief Scientist who oversees the Artificial Intelligence Research Center, Westlake University, and the OPPO U.S. Research Center. His research interests include machine learning and knowledge discovery from multi-modal data to build smart and reliable information and decision-making systems.