SCASeg: Strip Cross-Attention for Efficient Semantic Segmentation

Guoan Xu, Jiaming Chen, Wenfeng Huang, Wenjing Jia, *Member, IEEE*, Guangwei Gao, *Senior Member, IEEE*, and Guo-Jun Qi, *Fellow, IEEE*

***Abstract*—The Vision Transformer (ViT) has achieved notable success in computer vision, with its variants widely validated across various downstream tasks, including semantic segmentation. However, as general-purpose visual encoders, ViT backbones often do not fully address the specific requirements of task decoders, highlighting opportunities for designing decoders optimized for efficient semantic segmentation. This paper proposes Strip Cross-Attention (SCASeg), an innovative decoder head specifically designed for semantic segmentation. Instead of relying on the conventional skip connections, we utilize lateral connections between encoder and decoder stages, leveraging encoder features as Queries in cross-attention modules. Additionally, we introduce a Cross-Layer Block (CLB) that integrates hierarchical feature maps from various encoder and decoder stages to form a unified representation for Keys and Values. The CLB also incorporates the local perceptual strengths of convolution, enabling SCASeg to capture both global and local context dependencies across multiple layers, thus enhancing feature interaction at different scales and improving overall efficiency. To further optimize computational efficiency, SCASeg compresses the channels of queries and keys into one dimension, creating strip-like patterns that reduce memory usage and increase inference speed compared to traditional vanilla cross-attention. Experiments show that SCASeg's adaptable decoder delivers competitive performance across various setups, outperforming leading segmentation architectures on benchmark datasets, including ADE20K, Cityscapes, COCO-Stuff 164k, and Pascal VOC2012, even under diverse computational constraints.**

***Index Terms*—Vision Transformer, efficient semantic segmentation, decoder head, strip cross-attention, computational efficiency.**

Received 19 February 2025; revised 26 September 2025 and 5 March 2026; accepted 16 April 2026. Date of publication 6 May 2026; date of current version 11 May 2026. This work was supported in part by the Foundation of the State Key Laboratory of Integrated Services Networks of Xidian University under Grant ISN27-4. The associate editor coordinating the review of this article and approving it for publication was Dr. Bo Du. *(Corresponding authors: Wenjing Jia; Guangwei Gao.)*

Guoan Xu is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia, and also with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: xga_njupt@163.com).

Jiaming Chen is with the Department of Computer Science, The University of Manchester, M13 9PL Manchester, U.K. (e-mail: ppjmchen@gmail.com).

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

Guangwei Gao is with the PCA Laboratory and the Key Laboratory 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 State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China (e-mail: gwgao@njust.edu.cn).

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).

Digital Object Identifier 10.1109/TIP.2026.3688157

I. Introduction

SEMANTIC segmentation is a fundamental task in computer vision that involves pixel-level classification [3], [4], [5]. This process entails labeling each pixel in an image to accurately identify object categories, spatial positions, and other critical information, thereby providing a detailed understanding of the scene's composition. Semantic segmentation has widespread applications in various fields, including autonomous driving [6], medical diagnosis [7], and remote sensing [8], among others [9], [10]. A pivotal development in this area was the introduction of the fully convolutional network (FCN) [11], which popularized the encoder-decoder architecture. In this architecture, the encoder extracts high-level semantic features while the decoder integrates these features with spatial details. Despite subsequent advancements [12], traditional CNNs still struggle to capture long-range dependencies effectively.

This limitation has been largely addressed with the emergence of Transformers [13]. Following its groundbreaking success in Natural Language Processing (NLP), the Transformer quickly began to make a significant impact on vision tasks as well. Its self-attention mechanism is highly effective at capturing long-range dependencies within input sequences. Dosovitskiy et al. [14] extended this concept to the visual domain by proposing the Vision Transformer (ViT) backbone, which involved dividing images into small patches, transforming them into one-dimensional sequences, and feeding these sequences into the Transformer encoder to align the input dimensions for visual processing. Since then, many researchers have integrated ViT into the semantic segmentation domain, yielding impressive results. However, most of these approaches have focused primarily on optimizing the efficiency of the Transformer encoder while relying on simple or pre-existing designs for the decoder architecture. For instance, SegFormer [15] emphasizes designing an efficient Transformer encoder while utilizing a straightforward all-MLP decoder. Similarly, SegNeXt [2] develops a lightweight and efficient backbone but adopts a simplistic approach in the decoder stage.

More recently, MetaSeg [16] introduced a new and efficient self-attention module called Channel Reduction Attention (CRA), which simplifies the channel dimensions of the query and key into a single dimension per head within the self-attention process. However, this approach does not effectively facilitate interaction among the various feature representations. FeedFormer [1] uses features directly as queries, rather than relying on class-specific learnable queries. U-MixFormer [18] adaptively incorporates multi-stage features as keys and

values within its specialized mix-attention module. MacFormer [19] introduces a mutual agent cross-attention mechanism to enhance bidirectional feature interaction. Additionally, it proposes detailed enhancements in the frequency domain, achieving notable results. Despite the advancements offered by these attention blocks, they do not adequately consider the importance of local information. As demonstrated by models such as Metaformer [20], CMT [21], SMT [22], and XCiT [23], convolution methods are more effective than Transformers in capturing local features. Therefore, it is crucial to integrate local perception capabilities into the model alongside global attention mechanisms.

Based on the above observations and aiming to achieve a balance between efficiency and performance, we propose a novel Cross-Layer Block (CLB) comprising a *Strip Cross-Attention (SCA)* module and a *Local Perception Module (LPM)*. Specifically, the SCA module captures global long-range context dependencies by employing a low-rank strategy to compress the channel dimensions of the query and key for lightweight computation, while retaining the full value dimension to maintain dense token connectivity. This approach effectively balances global attention modeling with manageable computational complexity. Meanwhile, the LPM leverages the local perception capability of convolution, enhanced with channel attention, to extract and retain fine-grained local details. This design achieves a better trade-off between efficiency and effectiveness, as illustrated in Fig. 1. In summary, the main contributions of our method are as follows:

1) We present an innovative and robust transformer-decoder architecture designed for efficient semantic segmentation. Building on U-Net's strengths in capturing and transmitting hierarchical features, our approach uniquely utilizes lateral connections from the transformer encoder as query features.
2) We introduce a meticulously designed Cross-Layer Block consisting of two key modules: Strip Cross-Attention and the Local Perception Module. These modules work together to capture both local and global contexts effectively.
3) Experiments were conducted using various backbones on benchmark datasets, including ADE20K [1], Cityscapes [24], COCO-Stuff 164k [25], and Pascal VOC2012 [26], resulting in SOTA performance.

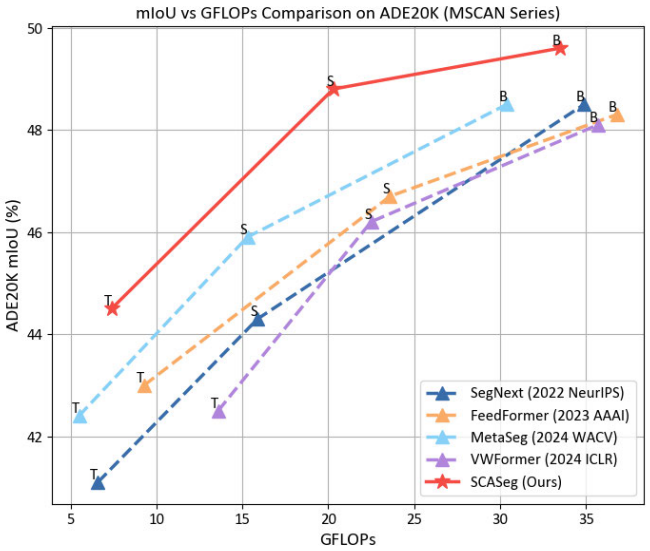

Fig. 1. The mIoU and GFLOPs comparison of SCASeg with SOTA approaches on ADE20K [1] dataset. The results are reported using a single model and single-scale inference based on MSCAN [2] backbones.

II. Related Work

A. Semantic Segmentation

Semantic segmentation can be seen as an evolution of image classification, transitioning from categorizing entire images to assigning labels at the pixel level [27], [28], [29], [30], [31]. During the deep learning era, the Fully Convolutional Network (FCN) [11] marked a significant breakthrough in semantic segmentation by utilizing a fully convolutional architecture for end-to-end pixel-wise classification. Following the development of FCN, subsequent research advanced semantic segmentation from several perspectives: 1) Enlarging the receptive field [32]. DeepLab-v3 [33] introduced dilation rates in the Atrous Spatial Pyramid Pooling (ASPP) module, allowing for a larger and more multi-scale receptive field. 2) Improving contextual information [34]. CPNet [35] enhanced feature learning accuracy by encoding ground truth into a one-hot representation and introducing a context before the encoder, providing more precise guidance for feature learning. 3) Incorporating boundary information [36]. BPKD [37] employed edge detection operators to dilate and erode target objects, effectively extracting their edges, and used knowledge distillation to transfer accurate edge information from a teacher model to a student network. 4) Designing various attention mechanisms [38]. DANet [12] and CCNet [39] extended non-local attention by integrating channel attention concepts to enhance overall model performance. Although these approaches have significantly improved segmentation accuracy, they have also introduced numerous empirical modules, resulting in computationally intensive and complex frameworks.

Another research direction explores prototype-driven designs (*e.g.*, PEM [40]), which constrain cross-attention using learned prototypes and combine them with efficient multi-scale pyramids to reduce computational load. ContrastiveSeg [41] introduced pixel-wise metric learning to leverage semantics across images without additional overhead. Clustering-based approaches (*e.g.*, CLUSTSEG [42]) treated queries as cluster centers, alternating between pixel assignment and center updates to provide a unified framework for segmentation tasks. While these methods mainly innovate at the level of objectives or query formation, our work focuses on an architecture-level contribution through the design of a decoder and token mixer.

B. Encoder Backbone

The Vision Transformer (ViT) [14] was the pioneering work that demonstrated how a pure Transformer-based model could achieve SOTA performance in image classification. ViT treats each image as a sequence of tokens, which are processed through multiple Transformer layers for classification.

Following this, DeiT [43] introduced a more effective training strategy along with a distillation technique for ViT. Recent approaches [21], [22], [23] have incorporated specialized modifications to ViT to further enhance its performance in image classification.

However, since semantic segmentation involves dense prediction tasks distinct from image classification, improvements in classification models do not always translate to equivalent gains in segmentation performance. In this context, SETR [28] was the first to adopt a pure vision transformer-based architecture for semantic segmentation. However, its training costs are quite high due to the large number of tokens generated by high-resolution images, which substantially increases the computational complexity of the self-attention mechanism. Researchers have increasingly focused on improving the computational efficiency of the backbone. PVT [44] employed spatial reduction in the linear projection stage to reduce the dimensions of the key and value matrices. The Swin Transformer [45], on the other hand, partitions images into multiple windows before applying patch sizes, significantly reducing the computational cost. Later, SegFormer [15] proposed a simple yet highly efficient structure that combined a Mix Transformer encoder with an all-MLP decoder. Subsequently, models like CoaT [46], LeViT [47], and Twins [48] further improved the continuity of local features and eliminated fixed-size positional embeddings to enhance Transformer performance in dense prediction tasks. In light of the high computational cost, SegNeXt [2] showed that convolutional attention offers a more efficient and effective means of encoding contextual information compared to the self-attention mechanism used in Transformers, while also providing a comprehensive analysis of the strengths and advantages of various models.

C. Decoder Head

For semantic segmentation, Segmenter [49] leveraged the output embeddings associated with image patches and derived class labels from these embeddings using either a point-wise linear decoder or a mask transformer decoder. MetaSeg [16] introduced a lightweight decoder module called Channel Reduction Attention, which enabled self-attention within each stage's output while reducing computational load. However, a key limitation is the lack of cross-layer interaction, indicating potential areas for improvement. FeedFormer [17] enhanced efficiency by taking high-level encoder features as queries, and the lowest-level encoder features as keys and values. Yet, it processes feature maps independently without progressive propagation across decoder stages, missing opportunities for gradual refinement that can enhance object boundary detection. U-MixFormer [18] addressed this by introducing a mix-attention mechanism that first downsampled features from different levels and concatenated them to form queries. Features from each encoder level were then treated as keys and values, with cross-attention applied progressively across layers. The newly generated feature map was merged back into the original concatenated features to form new queries.

Our approach draws inspiration from this, but U-MixFormer [18] still has a high computational cost due to its reliance on standard cross-attention mechanisms. On the other hand, MacFormer [19] preserved boundary information in the frequency domain and employs Mutual Agent cross-attention. While the use of agents helps control the overall parameter count, the bidirectional cross-attention operations still contribute to considerable computational complexity.

III. Methodology

A. Overall Architecture

As illustrated in Fig. 2 (a), our SCASeg framework is compatible with any pretrained model that features a hierarchical four-stage architecture. In this work, we employ lightweight backbones such as MiT-B0~B5 and MSCAN-T/S/B as encoders, efficiently extracting rich feature representations. Inspired by U-MixFormer [18], we have developed a refined version of the U-Net structure for the decoder. This enhanced design integrates a Cross-Layer Block to facilitate inter-level feature interaction with a low computational burden, significantly improving decoding capability.

B. Hierarchical Encoder and Lightweight Decoder

Given an image I of size $H \times W \times 3$ as input, feature maps $F_i \in \mathbb{R}^{\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i}$ are extracted at each stage of the encoder, where $i \in \{1, 2, 3, 4\}$ indicates the corresponding encoder stage and C_i denotes the number of channels in that stage. These features provide a progression from coarse to fine detail, contributing to the improved performance of semantic segmentation.

The decoder in our SCASeg utilizes the U-Net architecture to better capture global contexts that are insufficiently addressed by the encoder. At each stage of the decoder, refined features D_i are progressively generated through the Cross-Layer Block, where the query features X_q^i correspond to the respective lateral encoder feature maps F_i. The key and value feature X_{kv}^i (denoted as M_i in Fig. 2) are derived from a combination of both encoder and decoder stages. The decoder features are then upsampled using bilinear interpolation to match the height and width of D_1. Finally, the concatenated features are passed through an MLP (Multilayer Perceptron) to generate the segmentation masks, which have dimensions of $\frac{H}{4} \times \frac{W}{4} \times 3$.

The entire decoding process can be summarized by the following formulas:

$$\mathrm{M_i} = \mathrm{Cat}\left[\rho_1\left(\mathrm{F_1}\right), \ldots, \rho_i\left(\mathrm{F_i}\right), \mathrm{D_{i+1}}, \ldots, \mathrm{D_4}\right]_{i=1}^{4}, \quad (1)$$

$$\mathrm{D_i} = \mathrm{CLB}\left(\mathrm{F_i}, \mathrm{M_i}\right), \quad (2)$$

$$\mathrm{D_i^{up}} = \mathrm{Up}\left(\mathrm{D_i}, 2^{i-1}\right), \quad (3)$$

$$\mathrm{O_{mask}} = \mathrm{MLP}(\mathrm{Cat}\left[\mathrm{D_i^{up}}\right]_{i=1}^{4}), \quad (4)$$

where Cat denotes the concatenation operation, ρ represents a downsampling pooling operation, CLB stands for Cross-Layer Block, and Up refers to an upsampling function, which includes the scaling factor. The MLP is implemented using linear functions.

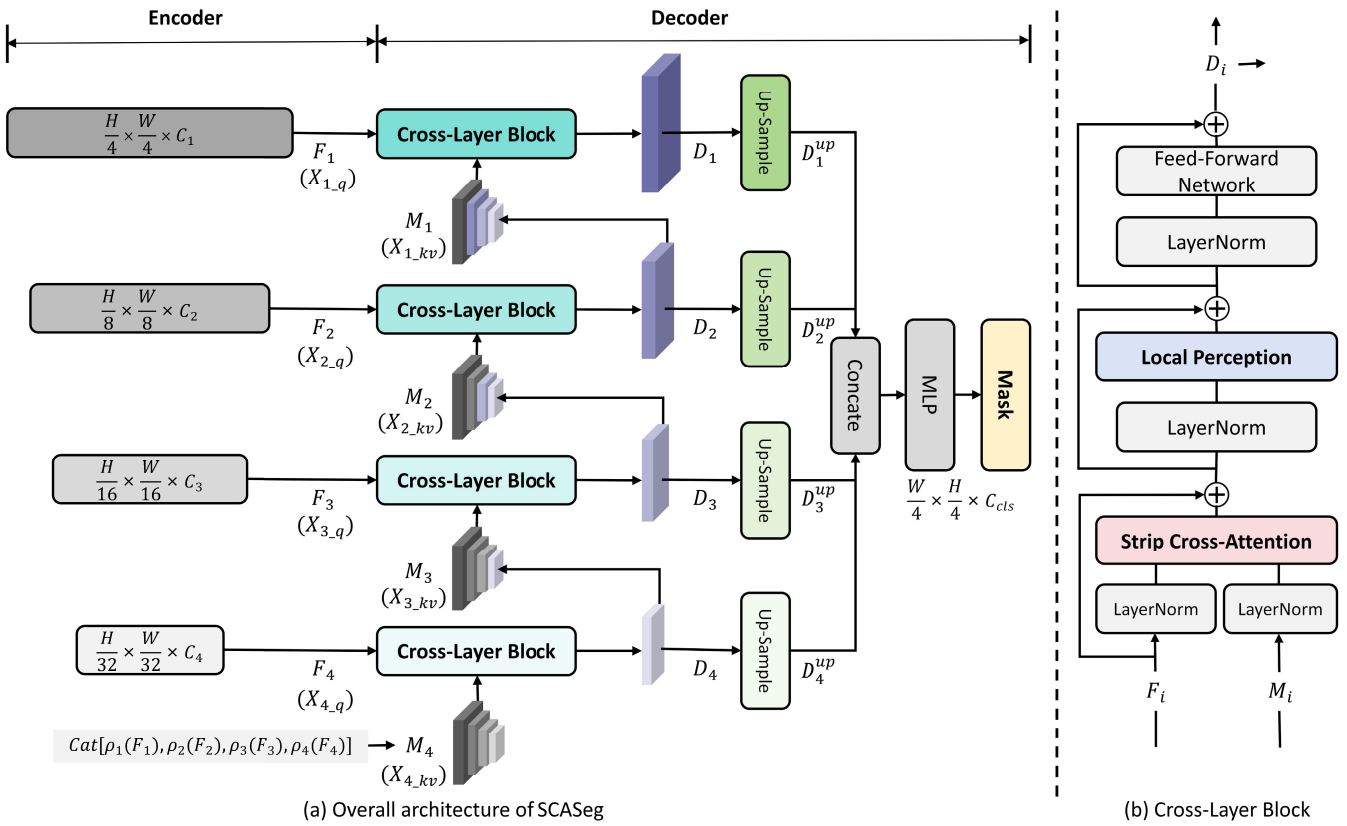

Fig. 2. The Overview of the proposed SCASeg architecture (a) and the detailed structure of the Cross-Layer Block (CLB) (b). SCASeg consists of two main components: an encoder based on hierarchical backbones (e.g., MSCAN [2] and MiT-Bx [15]) and a decoder with a cross-layer interaction design. The key innovation of SCASeg lies in this decoder structure, which promotes multi-scale feature complementarity. The **CLB** further enables efficient global–local feature integration while preserving local structural information.

C. Cross-Layer Block (CLB)

The proposed Cross-Layer Block (CLB) incorporates the MetaFormer [20] block in the decoder to enhance the global context of the feature representations extracted by the encoder, with a primary focus on integrating contextual information across different hierarchical features. As illustrated in Fig. 2 (b), the CLB includes the MetaFormer block, which consists of three residual subblocks, a Local Perception Module (LPM), and a novel Strip Cross-Attention (SCA) module for token mixing. The SCA module effectively captures both local and global contexts of the features while seamlessly integrating information across different hierarchical levels with minimal computational cost. The CLB is applied at each stage and takes two distinct inputs, F_i and M_i.

Thus, the entire process is defined as follows:

$$\mathrm{Z_i^G} = \mathrm{SCA}\left(\mathrm{LN}\left(\mathrm{F_i}\right), \mathrm{LN}\left(\mathrm{M_i}\right)\right) + \mathrm{F_i}, \tag{5}$$

$$\mathrm{Z_i^{GL}} = \mathrm{LPM}\left(\mathrm{LN}\left(\mathrm{Z_i^G}\right)\right) + \mathrm{Z_i^G}, \tag{6}$$

$$\mathrm{D_i} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathrm{Z_i^{GL}}\right)\right) + \mathrm{Z_i^{GL}}, \tag{7}$$

where Z_i^G captures global features, whereas Z_i^{GL} fuses both local and global contexts. Layer Normalization (LN) is employed to standardize these features.

The relationship to other attention blocks is shown in Fig. 3. These works introduce various modifications to the self-attention block, with a focus on enhancing the token-mixer component. MetaSeg [16] addresses the computational complexity of self-attention by employing a channel reduction strategy for Q and K. However, it overlooks the complementary information across multi-scale features. On the other hand, U-MixFormer [18] utilizes a feature mixing strategy but fails to capture local information. MacFormer [19] incorporates bidirectional complementarity but imposes a heavy computational burden. Considering these factors, we aim to propose a more efficient approach that balances both computational efficiency and feature complementarity. These methods often prioritize one aspect over others, resulting in limitations regarding feature interactions, computational cost, or local context preservation. In contrast, our method effectively addresses these challenges by integrating the advantages of cross-layer fusion, local context retention, and computational efficiency.

D. Strip Cross-Attention (SCA)

We introduce the Strip Cross-Attention (SCA) module as an innovative token mixer within the CLB, which is designed to effectively handle both global and local feature extraction while maintaining computational efficiency in cross-attention for semantic segmentation tasks. In transformer blocks, attention modules calculate the scaled dot-product attention for queries (Q), keys (K), and values (V) using the following formula:

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\top}}{\sqrt{\mathrm{d_k}}}\right)\mathbf{V}. \tag{8}$$

Here, $\sqrt{\mathrm{d_k}}$ represents the dimension of the key embeddings.

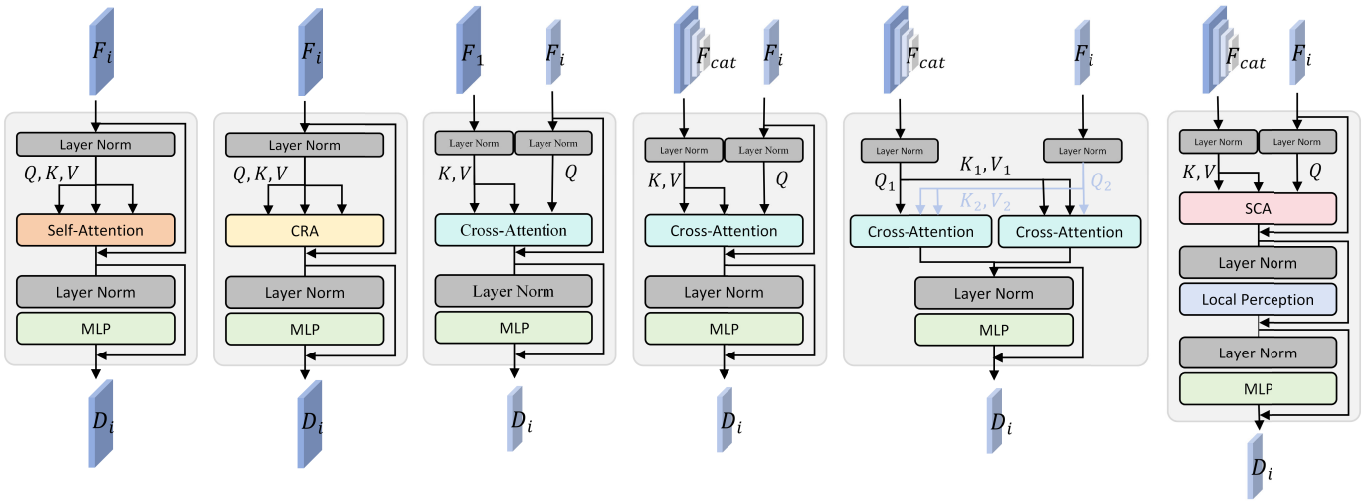

Fig. 3. The Cross-Layer Block (CLB) in our proposed SCASeg in comparison with its counterparts in SOTA approaches: (a) ViT [14], (b) MetaSeg [16], (c) FeedFormer [17], (d) U-MixFormer [18], (e) MacFormer [19]. The main distinction lies in the input features and the token mixer component. Additionally, our CLB takes into account the preservation of local information.

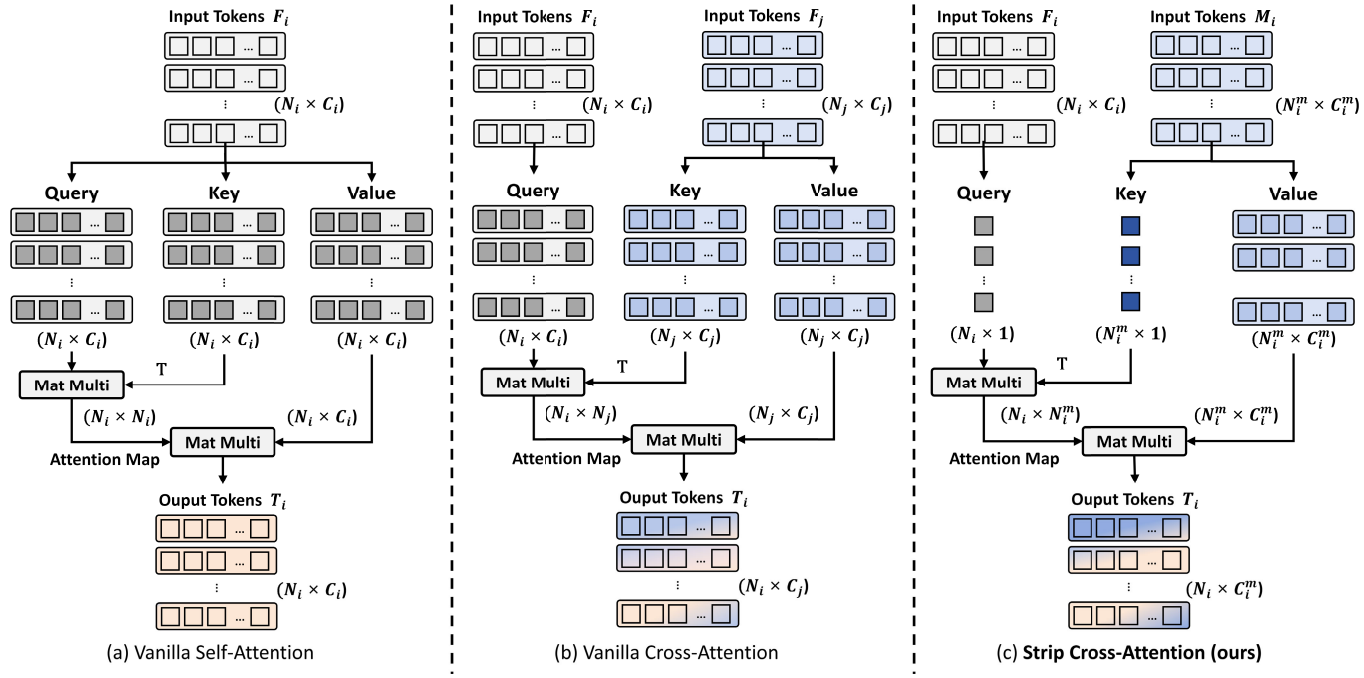

Fig. 4. The proposed Strip Cross-Attention (SCA) in comparison with the vanilla Self-Attention [14] and Cross-Attention [50]. The difference lies in the construction of the attention map: self-attention generates the attention map from the same feature, while cross-attention constructs it from different features. **Strip Cross-Attention (SCA)** leverages the advantages of cross attention while also reducing computational burden. Specifically, strip tokens are obtained by projecting query and key features into a single-channel embedding via linear transformation, resulting in a strip-shaped representation along the channel dimension. Unlike spatial aggregation methods, this operation preserves the spatial token resolution and keeps the token count unchanged, while reducing the dimensionality of attention similarity computation.

In Self-Attention, the features used to generate the queries, keys, and values are identical (denoted as X_{qkv}) and are derived from a common input source F_i, as shown in Fig. 4 (a). In Cross-Attention, two distinct sets of features (X_q and X_{kv}) are processed, each originating from a separate source, F_i and F_j, as depicted in Fig. 4 (b). In Strip Cross-Attention, a fused feature (X_{kv}) is gathered from multiple stages at different scales, denoted as M_i. This design enables the query to identify matches across various stages with varying levels of contextual granularity, thus supporting improved feature refinement. From an efficiency perspective, we strategically designed strip-shaped tokens to implement the attention map. The channel dimensions of the original query and key are embedded into a single dimension, further reducing computational overhead from $\mathcal{O}(H \cdot N^2 \cdot C)$ (vanilla cross-attention) to $\mathcal{O}(H \cdot N^2 \cdot 1)$. This one-dimensional transformation significantly

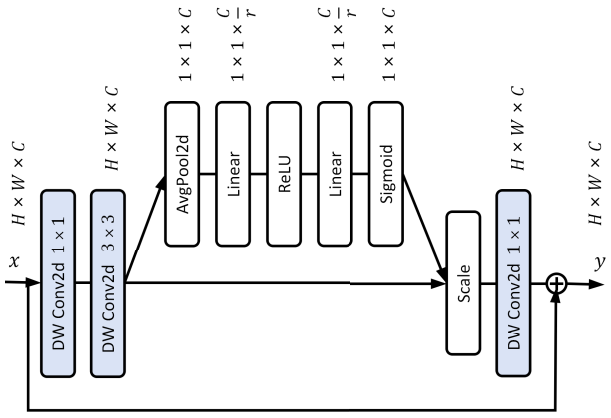

Fig. 5. The architecture of our proposed Local Perception Module (LPM).

reduces computational complexity. The comparison formulas for the computational costs of Self-Attention and Strip Cross-Attention are as follows:

$$\Omega(\mathrm{SA}) = \mathrm{N}^2 \cdot \mathrm{C} + \mathrm{N}^2 \cdot \mathrm{C} \tag{9}$$

$$\Omega(\mathrm{SCA}) = \mathrm{N}^2 \cdot \mathbf{1} + \mathrm{N}^2 \cdot \mathrm{C}, \tag{10}$$

where N represents the total number of tokens.

Motivated by MetaSeg [16], we observed that the channel-compressed feature token query $Q \in \mathbb{R}^{B\times hs\times N\times 1}$ and key $K \in \mathbb{R}^{B\times hs\times N\times 1}$ are effective at capturing global similarities. The SCA operation is expressed as follows:

$$\mathrm{Q_i} = \mathrm{W_i^Q}(\mathrm{F_i}) \in \mathbb{R}^{\mathrm{B}\times \mathrm{hs}\times \mathrm{N_i}\times 1}, \tag{11}$$

$$\mathrm{K_i} = \mathrm{W_i^K}(\mathrm{M_i}) \in \mathbb{R}^{\mathrm{B}\times \mathrm{hs}\times \mathrm{N_i^m}\times 1}, \tag{12}$$

$$\mathrm{V_i} = \mathrm{W_i^V}(\mathrm{M_i}) \in \mathbb{R}^{\mathrm{B}\times \mathrm{hs}\times \mathrm{N_i^m}\times \mathrm{dim_h}}, \tag{13}$$

$$\mathrm{Attn}\left(\mathrm{Q_i}, \mathrm{K_i}\right) = \mathrm{Softmax}\left(\frac{\mathrm{Q_i K_i^T}}{\sqrt{d_k}}\right) \in \mathbb{R}^{\mathrm{B}\times \mathrm{hs}\times \mathrm{N_i}\times \mathrm{N_i^m}}, \tag{14}$$

$$\mathrm{P_i} = \mathrm{Attn}\left(\mathrm{Q_i}, \mathrm{K_i}\right) \mathrm{V_i^T} \in \mathbb{R}^{\mathrm{B}\times \mathrm{hs}\times \mathrm{N_i}\times \mathrm{dim_h}}, \tag{15}$$

$$\mathrm{SCA}\left(\mathrm{F_i}, \mathrm{M_i}\right) = \mathrm{W_i^O}\left(\mathrm{Cat}\left(\mathrm{P_0}, \ldots, \mathrm{P_{hs}}\right)\right) \in \mathbb{R}^{\mathrm{B}\times \mathrm{N_i}\times \mathrm{C_i}}, \tag{16}$$

where W_i^Q, W_i^K and W_i^V are transformation matrices used to map features. B denotes the batch size, hs represents the number of attention heads, N is the number of tokens, and dim_h is the dimension of each head.

E. Local Perception Module (LPM)

Global attention excels at capturing long-range dependencies, but it often overlooks local context. To address the lack of local perception in standard self-attention and cross-attention mechanisms, we introduce a Local Perception Module (LPM) in the CLB, drawing inspiration from backbones such as XCiT [23], SMT [22], and CMT [21]. As shown in Fig. 5, the LPM can be derived using the following equation:

$$x_d = \mathrm{DWConv}_{3\times 3}\left(\sigma\left(\mathrm{DWConv}_{1\times 1}\left(\mathrm{x}\right)\right)\right), \tag{17}$$

$$\omega = \mathrm{Sigmoid}\left(\mathrm{f^{Li}}(\sigma\left(\mathrm{f^{Li}}\left(\mathrm{AvgPool}\left(\mathrm{x_d}\right)\right)\right)\right), \tag{18}$$

$$\mathrm{y} = \mathrm{x} + \mathrm{DWConv}_{1\times 1}\left(\omega \odot \mathrm{x_d}\right), \tag{19}$$

where *DWConv* denotes a depthwise separable convolution, σ represents the ReLU activation function, f^{Li} is the *nn.Linear* operation, and $\odot$ symbolizes matrix multiplication by channel.

IV. Experiments

A. Experimental Settings

1) Datasets: We conducted experiments on four publicly available datasets: ADE20K [1], Cityscapes [24], COCO-Stuff 164K [25], and Pascal VOC2012 [26]. ADE20K [1] contains 150 semantic categories, with 20,210 training images, 2,000 validation images, and 3,352 test images. Cityscapes [24] focuses on urban scenes with 19 categories, including 2,975 training images, 500 validation images, and 1,525 test images. COCO-Stuff164K [25] has 164,000 images annotated with 171 categories, enhancing the COCO dataset with detailed scene parsing. Pascal VOC2012 [26] includes 11,530 images across 20 categories, with pixel-level annotations for classification and segmentation tasks.

2) Implementation Details: Unless otherwise specified, all models are trained using the AdamW optimizer with an initial learning rate of 6e−5 for 160K iterations, following a polynomial learning rate decay schedule. We utilized the *mmsegmentation* [69] codebase (Version 1.2.2) to train our model on eight NVIDIA A100 GPUs. In our experiments, we employed LVT, MiT-B0 ~ 5, and MSCAN as backbone networks, while keeping the encoder unchanged to ensure fair comparisons across different methods. Standard data augmentation strategies were applied, including random horizontal flipping, random scaling with ratios ranging from 0.5 to 2.0, and random cropping to 512×512 for ADE20K [1], COCO-Stuff 164k [25], and Pascal VOC2012 [26], and 1024×1024 for Cityscapes [24]. The batch size was set to 16 for ADE20K, COCO-Stuff 164k, and Pascal VOC2012, and 8 for Cityscapes. All results are reported using single-scale inference, and performance is evaluated using mean Intersection over Union (mIoU).

B. Experimental Results

We categorized SOTA models into lightweight, medium-weight, and heavyweight. The main results on ADE20K [1] and Cityscapes [24] are shown in Tables I, II, and III. For COCO-Stuff 164k [25] and Pascal VOC2012 [26], we conducted fair comparisons using the same backbone networks, with the key results presented in Table IV.

ADE20K [1] & Cityscapes [24]. *Lightweight models:* We presented the performance of the lightweight models in Table I. As indicated in the table, our lightweight model, SCASeg (MiT-B0), achieved an mIoU of 41.6% on ADE20K [1], utilizing only 6.0 million parameters and 5.9 GFLOPs. In comparison to SegFormer [15] (MiT-B0), SCASeg (MiT-B0) achieves a 4.2% improvement in mIoU while reducing computational cost by 29.7%. Although SDPT-Tiny [58] and VWFormer [59] (MiT-B0) have a slight advantage in parameter count, their GFLOPs are nearly the same as ours, and their mIoU is at least 2.2% lower than that of our method. For Cityscapes [24], the performance difference becomes more evident, with our model achieving an mIoU of 79.3% at just 101.7 GFLOPs. This represents a 3.1% improvement in mIoU and an 18.9% reduction in computational cost compared to SegFormer [15] (MiT-B0). Similarly, with LVT and MSCAN-T as backbones, our approach consistently achieves near-SOTA performance.

TABLE I

PERFORMANCE COMPARISONS WITH SOTA LIGHT-WEIGHT MODELS ON ADE20K [1] AND CITYSCAPES [24]. RESULTS ARE TAKEN FROM THE ORIGINAL PAPERS UNDER A UNIFIED MMSEGMENTATION TRAINING PROTOCOL. "–" INDICATES METRICS NOT REPORTED IN THE ORIGINAL PUBLICATIONS

Method	Year	Backbone	Params. (M)↓	ADE20K [1]		Cityscapes [24]	
				GFLOPs↓	mIoU (%)↑	GFLOPs↓	mIoU (%)↑
FCN [11]	2015 CVPR	MobileNet-V2	9.8	39.6	19.7	317.1	61.5
PSPNet [51]	2017 CVPR	MobileNet-V2	13.7	52.9	29.6	423.4	70.2
DeepLabV3+ [32]	2018 ECCV	MobileNet-V2	15.4	69.4	34.0	555.4	75.2
SwiftNetRN [52]	2019 arxiv	ResNet-18	11.8	-	-	104.0	75.5
Semantic FPN [53]	2021 CVPR	ConvMLP-S	12.8	33.8	35.8	-	-
LeMoRe [54]	2025 arxiv	LeMoRe	**1.6**	0.8	33.5	-	-
DataFormer [55]	2025 CVPR	DataFormer	**1.6**	**0.6**	33.8	-	-
ContextFormer [56]	2025 arxiv	TPEM	1.7	**0.6**	35.0	-	-
SeaFormer [57]	2023 ICLR	SeaFormer-S	4.0	1.1	38.1	-	76.1
SCTNet [30]	2024 AAAI	SCTNet-S	4.7	-	37.7	-	72.8
SDPT [58]	2024 TITS	SDPT-Tiny	3.6	5.7	39.4	**63.4**	77.3
SegFormer [15]	2021 NeurIPS	MiT-B0	3.8	8.4	37.4	125.5	76.2
FeedFormer [17]	2023 AAAI	MiT-B0	4.5	7.8	39.2	107.4	77.9
PEM [40]	2024 CVPR	STDC1	17.0	16.0	39.6	92.0	78.3
MetaSeg [16]	2024 WACV	MiT-B0	4.1	3.9	37.9	90.9	76.7
VWFormer [59]	2024 ICLR	MiT-B0	3.7	5.8	38.9	172.0	77.2
EMOv2-2M [60]	2025 TPAMI	MiT-B0	2.6	10.3	40.2	-	-
SCASeg (Ours)	-	MiT-B0	6.0	5.9	**41.6**	101.7	**79.3**
SegFormer [15]	2021 NeurIPS	LVT	**3.9**	10.6	39.3	140.9	77.6
FeedFormer [17]	2023 AAAI	LVT	4.6	10.0	41.0	124.6	78.6
PEM [40]	2024 CVPR	STDC-1	17.0	16.0	39.6	**92.0**	78.3
MetaSeg [16]	2024 WACV	LVT	4.2	6.0	40.8	106.0	78.1
CCASeg [61]	2025 WACV	MiT-B0	6.2	7.2	42.6	115.8	78.7
EDAFormer [62]	2024 ECCV	EDAFormer-T	4.9	**5.6**	42.3	151.7	78.7
VWFormer [59]	2024 ICLR	LVT	5.3	14.3	42.3	194.0	78.9
SCASeg (Ours)	-	LVT	6.3	8.8	**43.8**	122.4	**79.7**
SegNeXt [2]	2022 NeurIPS	MSCAN-T	**4.3**	6.6	41.1	56.0	79.8
FeedFormer [17]	2023 AAAI	MSCAN-T	5.0	9.3	43.0	61.1	80.6
MetaSeg [16]	2024 WACV	MSCAN-T	4.7	**5.5**	42.4	**47.9**	80.1
PEM [40]	2024 CVPR	STDC2	21.0	19.3	**45.0**	118.0	79.0
VWFormer [59]	2024 ICLR	MSCAN-T	5.8	13.6	42.5	131.4	80.3
VRWKV [63]	2025 ICLR	VRWKV-T	8.4	16.6	43.3	-	-
SCASeg (Ours)	-	MSCAN-T	6.5	7.4	44.5	54.8	**81.2**

Medium-weight models: As shown in Table II, our SCASeg demonstrates superior performance compared to other methods when paired with equivalent heavy encoders. SCASeg (MiT-B1) achieved 45.4% mIoU on ADE20K [1] with just 23.4 million parameters and 17.4 GFLOPs. Using MiT-B1 as the backbone, our approach reduces GFLOPs by 15.9% while improving performance by 1.2% compared to FeedFormer [17]. The PEM model [40], which achieves comparable performance (45.5% vs. 45.4%), requires 52% more parameters and 2.7 times the GFLOPs (46.9 vs. 17.4) of our method. Although EfficientMod [5] achieves an mIoU that is 0.6% higher than ours, it incurs an additional computational cost of nearly 11 GFLOPs. On the Cityscapes [24] dataset, our approach achieves a new SOTA accuracy of 80.3%. Similarly, when evaluated using MSCAN-S and MiT-B2 as backbones, SCASeg effectively balances performance and efficiency.

Heavyweight models: Table III presents the experimental comparison using heavy backbones, specifically MSCAN-B and MiT-B3/4/5. Our method also demonstrates solid results. For instance, on the ADE20K [1] dataset, SCASeg (MSCAN-B) achieves 49.6% mIoU with only 33.5 GFLOPs. In comparison, VWFormer [59] shows worse performance while with more computation cost (48.1% mIoU with 35.7 GFLOPs). For the Cityscapes [24] dataset, using MiT-B5 as the backbone, our method achieved an SOTA mIoU of 83.5%, with a relatively small and justifiable cost of 1173.0 GFLOPs. Although the GFLOPs of MetaSeg [16] are 2.6% lower than ours (1143 vs. 1173), its mIoU is 1% lower, indicating a clear performance difference. These experimental results demonstrate that, under the same conditions, our method strikes a better balance between performance and efficiency compared to other approaches, highlighting its distinct advantages and validating its effectiveness.

COCO-Stuff 164k [25] & Pascal VOC2012 [26]: In Table IV, we compared our SCASeg model with previous methods on the COCO-Stuff 164k [25] and Pascal VOC2012 [26] datasets. To ensure a fair assessment, we selected different backbone configurations (MiT-B0/B1/B2, MSCAN-T/S, and SegMAN-T) and applied different methods under the same experimental conditions: 160k iterations with 8 GPUs, each processing a batch size of 8. The inference time per image was tested on a V100 GPU. As observed in Table IV, our method demonstrates outstanding performance compared with other methods.

TABLE II
PERFORMANCE COMPARISONS WITH SOTA MEDIUM-WEIGHT MODELS ON ADE20K [24] AND CITYSCAPES [24]

Method	Year	Backbone	Params. (M)↓	ADE20K [1]		Cityscapes [24]	
				GFLOPs↓	mIoU (%)↑	GFLOPs↓	mIoU (%)↑
CCNet [39]	2019 ICCV	ResNet-101	68.9	278.4	43.7	2224.8	79.5
EncNet [34]	2018 CVPR	ResNet-101	55.1	218.8	44.7	1748.0	76.9
DeepLab-V3+ [32]	2018 ECCV	ResNet-101	52.7	255.1	44.1	2032.3	80.9
Mask2Former [29]	2022 CVPR	ResNet-101	63.0	90.0	47.8	-	-
Auto-DeepLab [27]	2019 CVPR	Auto-DeepLab-L	44.4	-	-	695.0	80.3
OCRNet [3]	2020 ECCV	HRNet-W48	70.5	164.8	45.6	1296.8	81.1
SegFormer [15]	2021 NeurIPS	MiT-B1	13.7	15.9	42.2	243.7	78.5
SegDeformer [4]	2022 ECCV	MiT-B1	14.4	-	44.1	-	-
SeaFormer [57]	2023 ICLR	SeaFormer-B	**8.6**	**1.8**	40.2	-	77.7
SCTNet [30]	2024 AAAI	SCTNet-B	17.4	-	43.0	-	79.8
FeedFormer [17]	2023 AAAI	MiT-B1	17.3	20.7	44.2	256.0	79.0
U-MixFormer [18]	2025 WACV	MiT-B1	24.0	17.8	45.2	246.8	79.9
SFNet	2024 IJCV	ResNet-18	12.3	-	-	-	80.1
MetaSeg [16]	2024 WACV	MiT-B1	16.0	12.4	43.8	**219.0**	78.6
PEM [40]	2024 CVPR	ResNet-50	35.6	46.9	**45.5**	240.0	79.9
VWFormer [59]	2024 ICLR	MiT-B1	13.7	13.2	43.2	289.0	79.0
SCASeg (Ours)	-	MiT-B1	23.4	17.4	45.4	248.8	**80.3**
SegNeXt [2]	2022 NeurIPS	MSCAN-S	**13.9**	15.9	44.3	**124.6**	81.3
FeedFormer [17]	2023 AAAI	MSCAN-S	17.6	23.6	46.7	163.0	81.5
MetaSeg [16]	2024 WACV	MSCAN-S	16.3	**15.3**	45.9	126.0	81.3
VWFormer [59]	2024 ICLR	MSCAN-S	15.5	22.5	46.2	196.0	**81.7**
CPT [64]	2025 TIP	ResNet-101	48.5	100.0	46.8	-	-
EDAFormer [62]	2024 ECCV	EDAFormer-B	29.4	32.0	**49.0**	605.9	81.6
OffSeg [65]	2025 arxiv	OffSeg-L	26.4	17.1	48.5	143.4	81.6
SCASeg (Ours)	-	MSCAN-S	23.7	20.3	48.8	155.6	81.6
SegFormer [15]	2021 NeurIPS	MiT-B2	27.5	62.4	46.5	717.1	81.0
SegDeformer [4]	2022 ECCV	MiT-B2	27.6	-	47.5	-	-
FeedFormer [17]	2023 AAAI	MiT-B2	29.1	42.7	48.0	522.7	81.5
MetaSeg [16]	2024 WACV	MiT-B2	27.8	**25.2**	46.3	**420.0**	81.2
VWFormer [59]	2024 ICLR	MiT-B2	**27.4**	46.6	48.1	469.0	81.7
CPT-M [64]	2025 TIP	ResNet-101	50.4	113.0	47.0	-	-
CPT-M [64]	2025 TIP	D-ResNet-101	50.4	258.0	48.1	-	-
SCASeg (Ours)	-	MiT-B2	35.2	39.6	**48.3**	516.6	**81.9**

C. Visualization Results

Visual Comparison of Feature Maps Before and After Applying CLB: Fig. 6 presents a visual comparison of feature maps in the decoder before and after introducing the CLB. Before applying the CLB, features from different stages (F1–F4) lacked interaction, resulting in no exchange or complementary information between them. This limitation led to errors and inaccuracies in the segmentation results after direct fusion. However, after incorporating the CLB, it is evident that object boundaries are clearly visible at all stages, and the network demonstrates enhanced edge perception and class distinction, ultimately yielding more accurate segmentation outcomes.

Visual Comparison of Segmentation Results with and without Using LPM: The Local Perception Module (LPM) is a critical component of our design. While attention mechanisms typically emphasize global context, they often overlook local perception. By incorporating the LPM, we address this limitation. As illustrated in Fig. 7, there is a noticeable difference in the continuity of local information with and without the LPM. For instance, objects such as the long beak of a red-crowned crane or a snowboard are prone to misprediction due to the influence of surrounding larger objects. The LPM effectively alleviated this issue, enabling more consistent segmentation of small details.

Visual Comparison of Segmentation Results: Figs. 8, 9, 10, and 11show the visual comparison of the segmentation results obtained on the ADE20K [1], Cityscapes [24], Pascal VOC2012 [26], and COCO-Stuff 164k [25] datasets, respectively, using our SCASeg and SOTA methods. The highlighted areas indicate regions where SCASeg outperforms the other methods in segmentation quality. This improvement is evident in two main aspects: first, the prediction accuracy for objects within the same category has increased (*e.g.*, the pole next to the car and the chandelier); second, boundary segmentation accuracy has improved (*e.g.*, billboards). Additionally, small objects, such as traffic lights, are accurately detected and predicted. Compared to SOTA methods, SCASeg achieves better recognition of object details near boundaries. This indicates that our model captures a more relevant visual context by leveraging the capacity of the CLB decoder strategy.

D. Ablation Studies

Effectiveness of Strip Cross-Attention (SCA): In Table V, we demonstrate the effectiveness of incorporating SCA within the decoder. SCA serves as a core component of the CLB,

TABLE III
PERFORMANCE COMPARISONS WITH SOTA HEAVY-WEIGHT MODELS ON ADE20K [1] AND CITYSCAPES [24]

Method	Year	Backbone	Params. (M)↓	ADE20K [1]		Cityscapes [24]	
				GFLOPs↓	mIoU (%)↑	GFLOPs↓	mIoU (%)↑
Seg-B-Mask/16 [49]	2021 ICCV	ViT-Base	106.0	-	48.5	-	-
MaskFormer [66]	2021 NeurIPS	Swin-S	63.0	79.0	49.8	-	-
SETR [28]	2021 CVPR	ViT-Large	318.3	-	50.2	-	82.2
SegNeXt [2]	2022 NeurIPS	MSCAN-B	**27.6**	34.9	48.5	275.7	82.6
FeedFormer [17]	2023 AAAI	MSCAN-B	30.5	36.8	48.3	269.0	82.1
MetaSeg [16]	2024 WACV	MSCAN-B	29.6	**30.4**	48.5	**251.1**	82.7
VWFormer [59]	2024 ICLR	MSCAN-B	28.3	35.7	48.1	302.0	82.3
SCASeg (Ours)	-	MSCAN-B	36.5	33.5	**49.6**	261.0	**83.0**
ContrastiveSeg [41]	2021 ICCV	HRNetV2-W48	-	-	-	-	81.4
SegFormer [15]	2021 NeurIPS	MiT-B3	**47.3**	79.0	49.4	962.9	81.7
FeedFormer [17]	2023 AAAI	MiT-B3	48.3	47.2	49.5	682.0	81.9
MetaSeg [16]	2024 WACV	MiT-B3	47.7	**41.8**	48.7	**645.0**	81.8
VWFormer [59]	2024 ICLR	MiT-B3	**47.3**	63.3	49.6	715.0	82.4
SCASeg (Ours)	-	MiT-B3	55.1	56.3	**50.1**	675.0	**83.0**
CLUSTSEG [42]	2023 ICML	ResNet-50	-	-	50.5	-	-
SegFormer [15]	2021 NeurIPS	MiT-B4	64.1	95.7	50.3	1240.6	81.9
FeedFormer [17]	2023 AAAI	MiT-B4	65.0	63.8	50.7	960.0	82.6
MetaSeg [16]	2024 WACV	MiT-B4	**63.6**	**55.5**	50.5	**923.0**	82.1
VWFormer [59]	2024 ICLR	MiT-B4	64.0	79.9	50.8	993.0	82.7
MacFormer [19]	2024 arxiv	MiT-B4	82.0	76.7	**50.9**	-	-
SCASeg (Ours)	-	MiT-B4	71.8	72.9	**50.9**	953.0	**83.2**
ContrastiveSeg [41]	2021 ICCV	OCR	-	-	-	-	83.2
SegFormer [15]	2021 NeurIPS	MiT-B5	84.7	183.3	51.0	1460.4	82.4
FeedFormer [17]	2023 AAAI	MiT-B5	85.6	79.8	51.2	1180.0	82.7
U-MixFormer [18]	2025 WACV	MiT-B5	93.0	149.5	51.9	1171.0	83.1
ViT-CoMer [31]	2024 CVPR	ViT-CoMer-B	144.7	-	48.8	-	-
MetaSeg [16]	2024 WACV	MiT-B5	85.0	**74.5**	51.4	**1143.0**	82.5
VWFormer [59]	2024 ICLR	MiT-B5	**84.6**	96.1	52.0	1213.0	82.8
MacFormer	2024 arxiv	MiT-B5	103.0	152.4	**52.8**	-	-
SCASeg (Ours)	-	MiT-B5	92.4	88.9	52.7	1173.0	**83.5**

primarily facilitating cross-level feature enhancement and interaction among different hierarchical layers. As shown in this table, incorporating Self-Attention in the Decoder stage leads to a 3.75% improvement in segmentation accuracy. Compared to Self-Attention (SA), Strip Cross-Attention (SCA) achieves a notable reduction in computational overhead, lowering FLOPS by 38.9%, while still delivering a 1.47% improvement in mIoU. Comparisons with Cross-Attention (CA) and Strip Cross-Attention (SCA) show that Strip Cross-Attention not only boosts performance but also reduces parameter count by 0.3M and computation by 0.4G, with a slight advantage in inference speed as well. This demonstrates that SCA not only preserves high segmentation performance but also significantly enhances computational efficiency, making it a highly effective alternative for applications with limited resources.

Effectiveness of the Local Perception Module (LPM): Table V presents the results of combining SCA with LPM, forming the complete CLB structure. With the addition of LPM, parameter count and computational load become comparable to those of Cross-Attention (CA), yet this combination achieves an increase in segmentation performance of over 0.5%. Moreover, inference speed remains nearly identical. When combined with LPM, CA also yields strong results—improving over CA alone by 0.32% and 0.63% on MiT-B0 and MSCAN-T, respectively—demonstrating LPM's effectiveness. However, CA+LPM introduces higher complexity (6.4M parameters, 6.3G FLOPs) than SCA+LPM (6.0M parameters, 5.9G FLOPs), and its accuracy is still 0.28% lower than that of SCA+LPM. In Table VI, we conducted a comparative experiment, using SENet [70] within LPM to enhance the model's sensitivity to local information. When compared with other channel attention mechanisms (CBAM [71], ECANet [72], CooAtt [73]), SENet [70] achieved the best segmentation accuracy, outperforming CooAtt by more than 0.3%. SENet's integration with LPM significantly strengthens the model's local modeling capability, enhancing coherence in segmentation and reducing misclassification for small objects or local regions of the same class.

Effect of Different Channel Dimensions of Key and Value:

To investigate the effect of compressing the channel dimensions of the query and key in our SCA module, we conducted an ablation study on the PASCAL VOC2012 dataset using MSCAN-T as the backbone. Specifically, we varied the output dimensions of the query (Q) and key (K) projections (shared across heads) while keeping the value projection and all other settings unchanged. As shown in Table VII, reducing the query/key dimension to 1 channel achieves a robust mIoU of 77.88%, with the lowest parameter count (6.5M), the lowest

TABLE IV
PERFORMANCE COMPARISONS WITH SOTA MODELS ON COCO-STUFF 164K [25] AND PASCAL VOC2012 [26]

Method	Year	Backbone	Params. (M)↓	FPS (img/s)↑	COCO-Stuff 164k [25] GFLOPs↓	COCO-Stuff 164k [25] mIoU (%)↑	Pascal VOC2012 [26] GFLOPs↓	Pascal VOC2012 [26] mIoU (%)↑
SegFormer [15]	2021 NeurIPS	MiT-B0	3.8	**43.65**	8.4	35.63	8.4	66.49
FeedFormer [17]	2023 AAAI	MiT-B0	4.5	34.80	7.8	39.03	7.8	68.49
U-MixFormer [18]	2025 WACV	MiT-B0	6.1	38.94	6.1	40.24	6.1	71.16
MetaSeg [16]	2024 WACV	MiT-B0	4.1	42.64	**3.9**	38.25	**3.9**	68.72
VWFormer [59]	2024 ICLR	MiT-B0	**3.7**	39.63	5.8	36.28	5.8	70.58
SCASeg (Ours)	-	MiT-B0	6.0	40.25	5.9	**40.56**	5.9	**72.35**
SegNeXt [2]	2022 NeurIPS	MSCAN-T	**4.3**	**33.16**	6.6	38.70	6.6	76.27
FeedFormer [17]	2023 AAAI	MSCAN-T	5.0	25.98	9.3	39.39	9.3	74.80
U-MixFormer [18]	2025 WACV	MSCAN-T	6.7	27.69	7.6	40.04	7.6	77.37
MetaSeg [16]	2024 WACV	MSCAN-T	4.7	31.74	**5.5**	39.70	**5.5**	74.98
VWFormer [59]	2024 ICLR	MSCAN-T	5.8	28.06	13.6	38.85	13.6	76.53
SCASeg (Ours)	-	MSCAN-T	6.5	29.04	7.4	**40.89**	7.4	**77.88**
FeedFormer [17]	2023 AAAI	SegMAN-T	3.1	41.66	7.9	41.54	7.9	75.65
U-MixFormer [18]	2025 WACV	SegMAN-T	4.3	41.93	7.0	41.98	7.0	75.87
MetaSeg [16]	2024 WACV	SegMAN-T	2.9	42.81	**4.5**	41.68	**4.5**	75.70
VWFormer [59]	2024 ICLR	SegMAN-T	**2.8**	42.15	6.3	41.38	6.3	75.26
SegMAN [67]	2025 CVPR	SegMAN-T	6.4	**44.52**	6.2	41.30	6.2	75.10
SCASeg (Ours)	-	SegMAN-T	4.3	43.02	6.9	**42.22**	6.9	**76.02**
SegNeXt [2]	2022 NeurIPS	MSCAN-S	**13.9**	**30.91**	15.9	41.42	15.9	78.62
FeedFormer [17]	2023 AAAI	MSCAN-S	17.6	24.16	23.6	42.61	23.6	77.42
U-MixFormer [18]	2025 WACV	MSCAN-S	24.3	26.55	20.8	42.91	20.8	79.23
MetaSeg [16]	2024 WACV	MSCAN-S	16.3	29.83	**15.3**	42.13	**15.3**	76.73
VWFormer [59]	2024 ICLR	MSCAN-S	15.5	27.37	22.5	41.76	22.5	78.95
SCASeg (Ours)	-	MSCAN-S	23.7	27.71	20.3	**43.65**	20.3	**79.48**
SegFormer [15]	2021 NeurIPS	MiT-B1	13.7	**31.44**	15.9	40.97	15.9	71.13
FeedFormer [17]	2023 AAAI	MiT-B1	17.3	22.29	20.7	42.42	20.7	71.77
U-MixFormer [18]	2025 WACV	MiT-B1	24.0	22.06	17.8	42.71	17.8	74.40
MetaSeg [16]	2024 WACV	MiT-B1	**16.0**	27.12	**12.4**	42.04	**12.4**	73.30
VWFormer [59]	2024 ICLR	MiT-B1	13.7	23.87	13.2	41.54	13.2	73.98
SCASeg (Ours)	-	MiT-B1	23.4	25.37	17.4	**43.19**	17.4	**75.24**
SegFormer [15]	2021 NeurIPS	MiT-B2	27.5	**29.72**	62.4	43.42	62.4	78.74
FeedFormer [17]	2023 AAAI	MiT-B2	29.1	27.56	42.7	44.79	42.7	78.90
U-MixFormer [18]	2025 WACV	MiT-B2	35.8	26.34	40.0	45.47	40.0	79.43
MetaSeg [16]	2024 WACV	MiT-B2	27.8	28.47	**25.2**	44.50	**25.2**	77.76
VWFormer [68]	2024 ICLR	MiT-B2	**27.4**	27.68	46.6	45.18	46.6	79.10
SCASeg (Ours)	-	MiT-B2	35.2	27.00	39.6	**45.85**	39.6	**80.15**

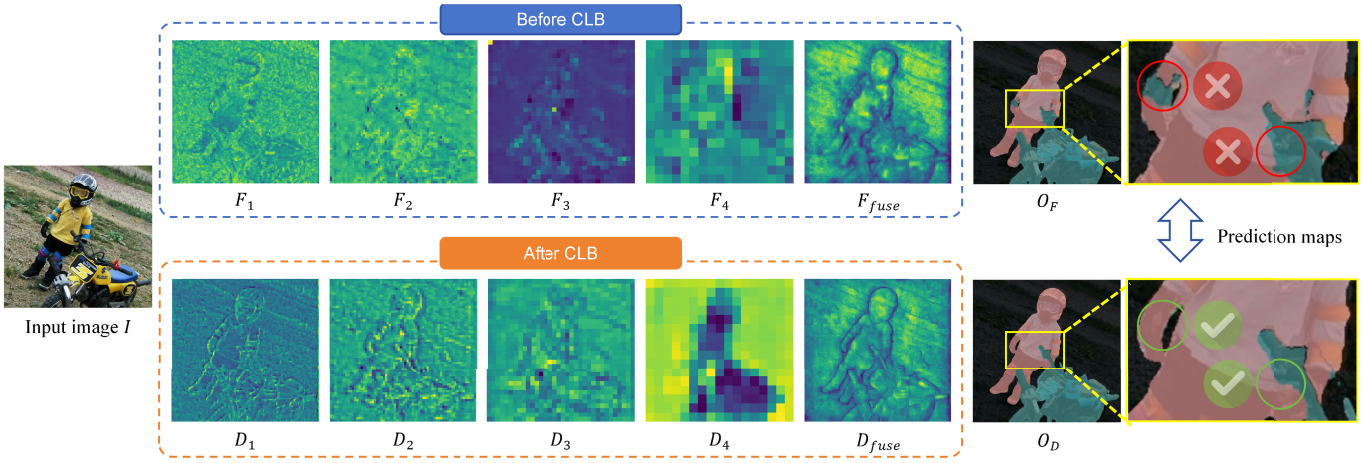

Fig. 6. Visualized feature heatmaps before and after introducing the Cross-Layer Block (CLB).

computational cost (7.4 GFLOPs), and the highest inference speed (29.0 FPS). Increasing the dimension from 1 to 8 slightly improves the mIoU to 77.99%, suggesting that a wider Q/K space may capture more fine-grained attention.

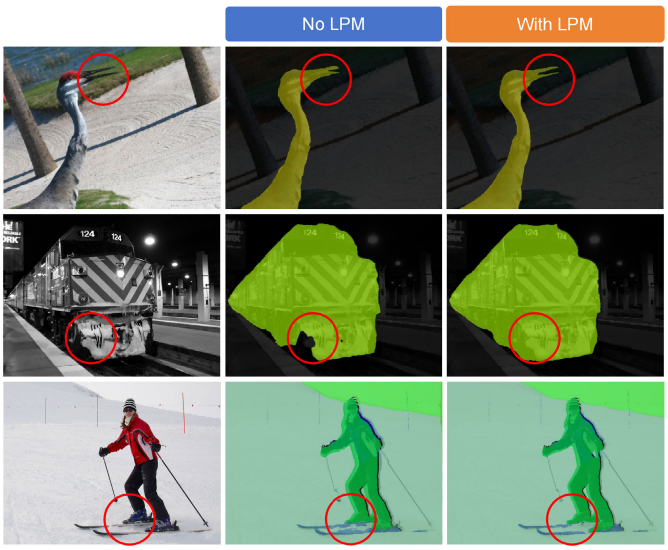

Fig. 7. Visual comparison of segmentation results with and without using our LPM.

TABLE V

ABLATION STUDIES OF SCA ON PASCAL VOC2012 [26]. SA: SELF-ATTENTION, CA: CROSS-ATTENTION, SCA: STRIP CROSS-ATTENTION, LPM: LOCAL PERCEPTION MODULE

Method	Params.↓	GFLOPs↓	mIoU (%)↑	FPS (img/s)↑
MiT-B0	3.8 M	8.4	66.49	44.0
+ SA	4.3 M	9.0	70.06 (+3.57)	45.0
+ CA	6.0 M	5.9	71.79 (+5.30)	42.0
+ CA + LPM	6.4 M	6.3	72.11 (+5.62)	41.0
+ SCA	5.7 M	5.5	71.53 (+5.04)	43.0
+ SCA + LPM	6.0 M	5.9	72.39 (+5.90)	42.0
MSCAN-T	4.3 M	6.6	76.27	33.0
+ SA	4.8 M	10.4	76.92 (+0.65)	30.0
+ CA	6.7 M	7.3	77.38 (+1.11)	29.0
+ CA + LPM	6.9 M	7.6	78.01 (+1.74)	27.0
+ SCA	6.3 M	6.9	76.74 (+0.47)	31.0
+ SCA + LPM	6.5 M	7.4	77.88 (+1.61)	29.0

TABLE VI

ABLATION STUDIES OF THE LOCAL PERCEPTION MODULE (LPM) ON PASCAL VOC2012 [26]

Method	Params. (M)↓	GFLOPs↓	mIoU (%)↑
SCA (MiT-B0)	5.7	5.5	71.53
+ LPM (CBAM)	5.9	5.7	71.29 (-0.24)
+ LPM (ECANet)	5.9	5.7	71.58 (+0.05)
+ LPM (CooAtt)	6.0	5.7	71.87 (+0.34)
+ LPM (SENet)	6.0	5.9	72.39 (+0.86)
SCA (MSCAN-T)	6.3	6.9	76.74
+ LPM (CBAM)	6.5	7.1	76.79 (+0.05)
+ LPM (ECANet)	6.5	7.1	76.49 (-0.25)
+ LPM (CooAtt)	6.5	7.1	77.57 (+0.83)
+ LPM (SENet)	6.5	7.4	77.88 (+1.14)

However, the performance gain plateaus quickly, while model size and complexity rise steadily. Overall, the 1-channel setting offers an optimal balance of efficiency and accuracy, with the attention computation retaining strong semantic expressiveness due to the uncompressed value features. This validates our design of using compressed Q/K with high-dimensional V to achieve compact yet effective attention maps.

TABLE VII

ABLATION STUDIES OF QUERY AND KEY'S CHANNEL DIMENSIONS USING THE MSCAN-T BACKBONE ON PASCAL VOC2012 [26]

Channels	Params. (M)↓	GFLOPs↓	mIoU (%)↑	FPS (img/s)↑
1	6.5	7.4	77.88	29.0
2	6.6	7.5	77.90	29.0
4	6.6	7.5	77.94	28.0
8	6.7	7.6	77.99	28.0

TABLE VIII

ABLATION STUDIES OF CROSS-LAYER BLOCK (CLB) ON PASCAL VOC2012 [26]

Method	Cross Layer 4	3	2	1	Params.↓	FLOPs↓	mIoU (%)↑
SCASeg (MiT-B0)	✔	✗	✗	✗	5.7 M	4.6 G	69.76
	✔	✔	✗	✗	5.7 M	4.9 G	70.68 (+0.92)
	✔	✔	✔	✗	5.7 M	5.2 G	71.16 (+1.40)
	✔	✔	✔	✔	6.0 M	5.9 G	72.35 (+2.59)
SCASeg (MSCAN-T)	✔	✗	✗	✗	6.3 M	6.0 G	75.73
	✔	✔	✗	✗	6.3 M	6.3 G	76.32 (+0.59)
	✔	✔	✔	✗	6.3 M	6.6 G	76.84 (+1.11)
	✔	✔	✔	✔	6.5 M	7.4 G	77.88 (+2.15)

Effectiveness of the Cross-Layer Strategy: The comparative experiments on cross-layer strategies highlight the necessity of information exchange across features at different stages. During the encoding phase, four processing stages generate features that vary in semantic richness and detail when passed to the decoder. Shallow stages, such as Stage 1 and Stage 2, retain more detailed information as they undergo less downsampling and fewer convolutional operations, resulting in clearer edge information. In contrast, deeper stages, such as Stage 3 and Stage 4, capture richer contextual information due to more extensive feature abstraction. This difference is visually evident in Fig. 6, which illustrates the characteristics of the feature maps. The cross-layer strategy facilitates mutual feature enhancement across stages, allowing each stage to compensate for the limitations of the others. As shown in Table VIII, applying the CLB module across all four stages and enabling cross-layer operations significantly boosts segmentation performance (72.35% for MiT-B0, 77.88% for MSCAN-T). This improvement underscores the effectiveness and necessity of cross-layer operations in achieving high segmentation accuracy.

E. Limitations and Future Work

Despite the excellent performance and computational efficiency of our approach, there are some limitations related to parameter size, which may hinder its application in resource-constrained environments. In future work, we plan to explore techniques such as parameter pruning and knowledge distillation to develop a more compact and efficient version of our model, while maintaining its effectiveness.

Our future work will focus on adaptive sampling of both tokens and channels to eliminate redundancy. Rather than

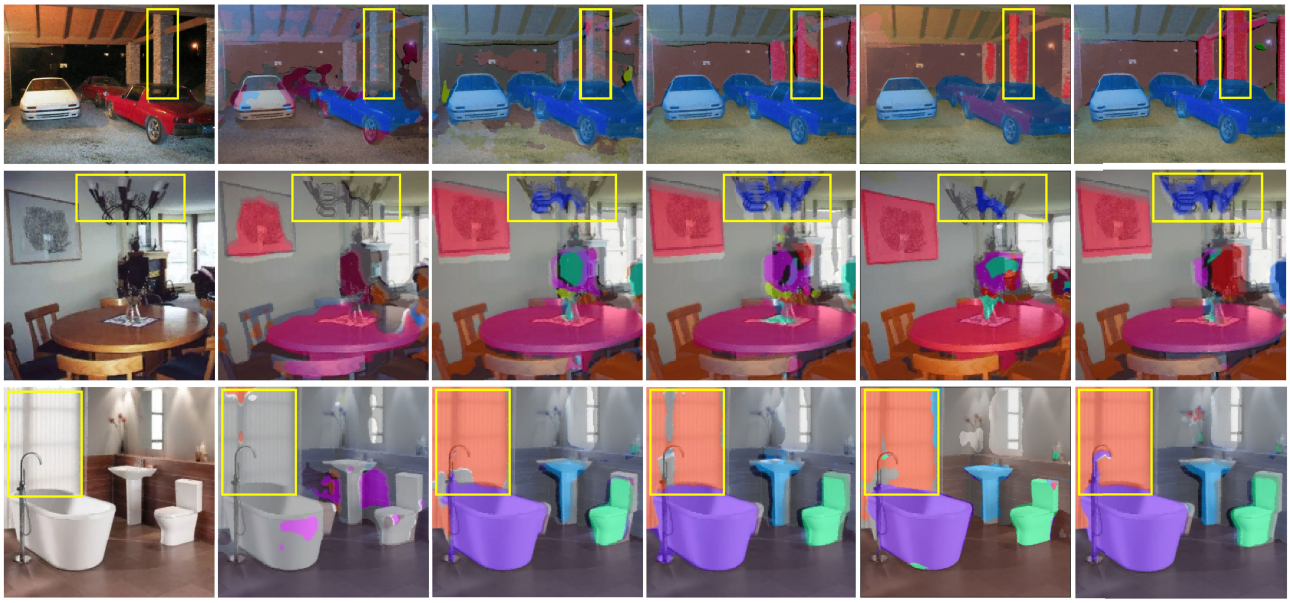

Fig. 8. Visual segmentation results obtained on ADE20K [1].

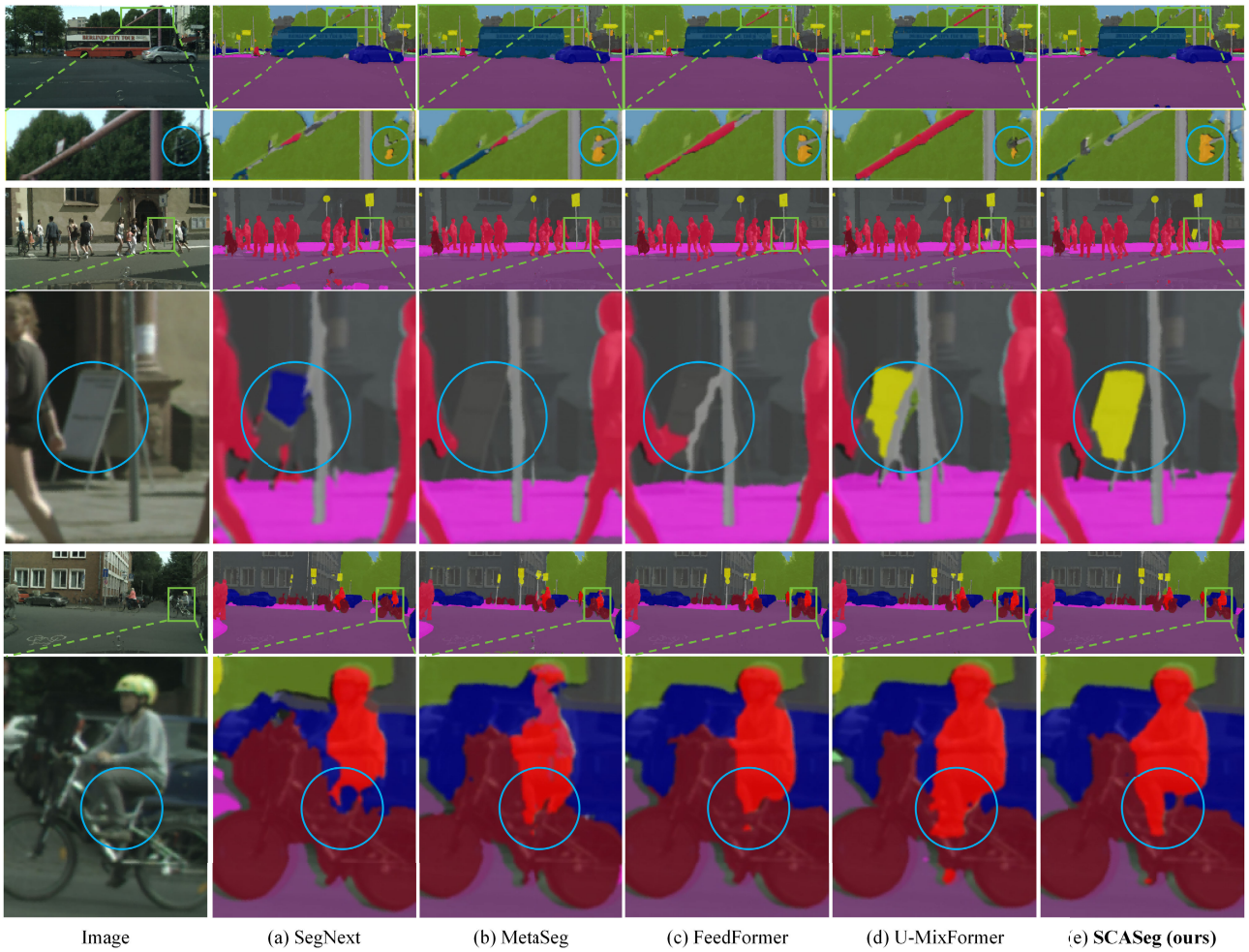

Fig. 9. Visual segmentation results obtained on Cityscapes [24].

compressing channels to a fixed number, we will enable the model to dynamically learn the effective channel count—per layer, per head, and even per sample—using rank-adaptive projections and sparsity-inducing gates (*e.g.*, L_0/L_1 regularization or Gumbel–Softmax masking). For the tokens, we plan to use content-aware selection (differentiable Top-k or

Fig. 10. Visual segmentation results obtained on Pascal VOC2012 [26].

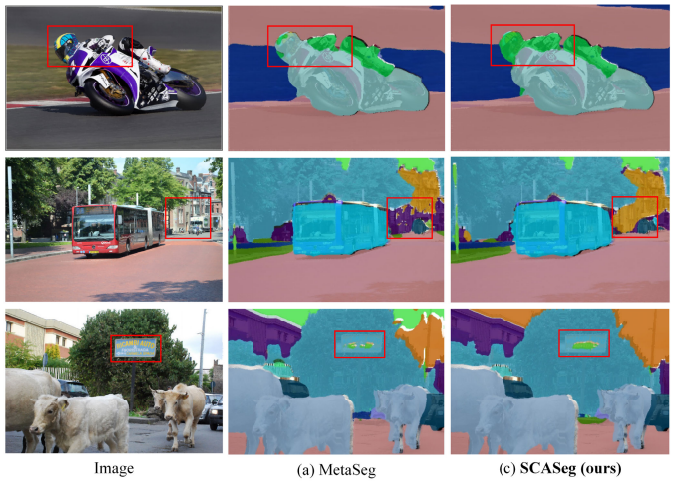

Fig. 11. Visual segmentation results obtained on COCO-Stuff 164K [25].

budget-aware routing) to prioritize computation on salient regions while preserving dense gradients.

V. Conclusion

This paper introduced Strip Cross-Attention (SCASeg), a novel decoder head tailored for semantic segmentation. We developed a Cross-Layer Block (CLB) that integrates hierarchical feature maps from various encoder and decoder stages to create a unified representation for Keys and Values. By incorporating the local perceptual strengths of convolution, the CLB enables SCASeg to effectively capture both global and local context dependencies across multiple layers, enhancing feature interaction at different scales and improving overall efficiency. Experimental results have shown SCASeg's competitive performance across benchmark datasets.

Despite robust empirical results, edge deployment requires further efficiency improvements and reduced computational cost. While cross-layer connections are essential for hierarchical feature aggregation, they introduce significant inference overhead. To address this, we plan to (i) incorporate model-compression techniques, such as structured pruning and knowledge distillation, and (ii) implement self-learning mechanisms that adaptively select tokens and channels on demand, rather than using a fixed number, to develop a more robust and optimized segmentation model in the future.

References

[1] 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.

[2] M.-M. Cheng, M.-H. Guo, Q. Hou, S.-M. Hu, Z. Liu, and C.-Z. Lu, "SegNeXt: Rethinking convolutional attention design for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1140–1156.

[3] Y. Yuan, X. Chen, and J. Wang, "Object-contextual representations for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2020, pp. 173–190.

[4] B. Shi et al., "A transformer-based decoder for semantic segmentation with multilevel context mining," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 624–639.

[5] X. Ma et al., "Efficient modulation for vision networks," 2024, *arXiv:2403.19963*.

[6] G. Xu, J. Li, G. Gao, H. Lu, J. Yang, and D. Yue, "Lightweight real-time semantic segmentation network with efficient transformer and CNN," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15897–15906, Dec. 2023.

[7] R. Azad et al., "Medical image segmentation review: The success of U-Net," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 10076–10095, Dec. 2024.

[8] B. Du et al., "SAMRS: Scaling-up remote sensing segmentation dataset with segment anything model," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 8815–8827.

[9] A. Kirillov et al., "Segment anything," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2023, pp. 4015–4026.

[10] X. Lai et al., "LISA: Reasoning segmentation via large language model," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 9579–9589.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[12] J. Fu et al., "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3146–3154.

[13] A. Vaswani, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 1–11.

[14] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[15] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 12077–12090.

[16] B. Kang, S. Moon, Y. Cho, H. Yu, and S.-J. Kang, "MetaSeg: MetaFormer-based global contexts-aware network for efficient semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 433–442.

[17] J. Shim, H. Yu, K. Kong, and S. Kang, "FeedFormer: Revisiting transformer decoder for efficient semantic segmentation," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2023, pp. 2263–2271.

[18] S.-K. Yeom and J. Von Klitzing, "U-MixFormer: UNet-like transformer with mix-attention for efficient semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Feb. 2025, pp. 1–10.

[19] G. Xu et al., "MacFormer: Semantic segmentation with fine object boundaries," 2024, *arXiv:2408.05699*.

[20] 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. 10809–10819.

[21] J. Guo et al., "CMT: Convolutional neural networks meet vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12165–12175.

[22] W. Lin, Z. Wu, J. Chen, J. Huang, and L. Jin, "Scale-aware modulation meet transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Jun. 2023, pp. 6015–6026.

[23] A. Ali et al., "XCiT: Cross-covariance image transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20014–20027.

[24] M. Cordts et al., "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[25] H. Caesar, J. Uijlings, and V. Ferrari, "COCO-stuff: Thing and stuff classes in context," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1209–1218.

[26] D. Hoiem, S. K. Divvala, and J. H. Hays, "Pascal VOC 2008 challenge," *World Literature Today*, vol. 24, no. 1, pp. 1–4, 2009.

[27] C. Liu et al., "Auto-DeepLab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 82–92.

[28] S. Zheng et al., "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6881–6890.

[29] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1290–1299.

[30] Z. Xu, D. Wu, C. Yu, X. Chu, N. Sang, and C. Gao, "SCTNet: Single-branch CNN with transformer semantic information for real-time segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 6378–6386.

[31] C. Xia, X. Wang, F. Lv, X. Hao, and Y. Shi, "ViT-CoMer: Vision transformer with convolutional multi-scale feature interaction for dense predictions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 5493–5502.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder–decoder with Atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[33] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[34] H. Zhang et al., "Context encoding for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Mar. 2018, pp. 7151–7160.

[35] C. Yu, J. Wang, C. Gao, G. Yu, C. Shen, and N. Sang, "Context prior for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12413–12422.

[36] M. Zhen et al., "Joint semantic segmentation and boundary detection using iterative pyramid contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13663–13672.

[37] L. Liu, Z. Wang, M. H. Phan, B. Zhang, J. Ge, and Y. Liu, "BPKD: Boundary privileged knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2024, pp. 1051–1061.

[38] Z. Zhong et al., "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13062–13071.

[39] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "CCNet: Criss-cross attention for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 603–612.

[40] N. Cavagnero et al., "PEM: Prototype-based efficient MaskFormer for image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 15804–15813.

[41] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. V. Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7303–7313.

[42] J. Liang, T. Zhou, D. Liu, and W. Wang, "CLUSTSEG: Clustering for universal segmentation," 2023, *arXiv:2305.02187*.

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

[44] 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.

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

[46] W. Xu, Y. Xu, T. Chang, and Z. Tu, "Co-scale conv-attentional image transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9961–9970.

[47] B. Graham et al., "LeViT: A vision transformer in ConvNet's clothing for faster inference," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 12239–12249.

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

[49] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7262–7272.

[50] C.-F.-R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 347–356.

[51] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2881–2890.

[52] H.-P. Cheng et al., "SwiftNet: Using graph propagation as meta-knowledge to search highly representative neural architectures," 2019, *arXiv:1906.08305*.

[53] J. Li, A. Hassani, S. Walton, and H. Shi, "ConvMLP: Hierarchical convolutional MLPs for vision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2023, pp. 6307–6316.

[54] M. M. N. Abid, N. Mehta, Z. Wu, and R. Timofte, "LeMoRe: Learn more details for lightweight semantic segmentation," 2025, *arXiv:2505.23093*.

[55] M. M. N. Abid, N. Mehta, Z. Wu, and R. Timofte, "Dataformer: Differential additive transformer for lightweight semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2025, pp. 811–822.

[56] M. M. N. Abid, N. Mehta, Z. Wu, and R. Timofte, "ContextFormer: Redefining efficiency in semantic segmentation," 2025, *arXiv:2501.19255*.

[57] Q. Wan, Z. Huang, J. Lu, Y. Gang, and L. Zhang, "SeaFormer: Squeeze-enhanced axial transformer for mobile semantic segmentation," in *Proc. Int. Conf. Learn. Represent.*, 2023, pp. 5974–5993.

[58] H. Cao et al., "SDPT: Semantic-aware dimension-pooling transformer for image segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 11, pp. 15934–15946, Nov. 2024.

[59] H. Yan, M. Wu, and C. Zhang, "Multi-scale representations by varying window attention for semantic segmentation," in *Proc. Int. Conf. Learn. Represent.*, 2024, pp. 667–684.

[60] J. Zhang et al., "EMOv2: Pushing 5M vision model frontier," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 11, pp. 10560–10576, Nov. 2025.

[61] J. Yoo, D. Ko, and G. Kim, "CCASeg: Decoding multi-scale context with convolutional cross-attention for semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Feb. 2025, pp. 9479–9488.

[62] H. Yu, Y. Cho, B. Kang, S. Moon, K. Kong, and S.-J. Kang, "Embedding-free transformer with inference spatial reduction for efficient semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2024, pp. 92–110.

[63] Y. Duan et al., "Vision-RWKV: Efficient and scalable visual perception with rwkv-like architectures," in *Proc. 13th Int. Conf. Learn. Represent.*, 2025, pp. 61381–61398.

[64] Q. Tang et al., "Rethinking feature reconstruction via category prototype in semantic segmentation," *IEEE Trans. Image Process.*, vol. 34, pp. 1036–1047, 2025.

[65] S.-C. Zhang, Y. Li, Y.-H. Wu, Q. Hou, and M.-M. Cheng, "Revisiting efficient semantic segmentation: Learning offsets for better spatial and class feature alignment," 2025, *arXiv:2508.08811*.

[66] B. Cheng, A. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, p. 17.

[67] Y. Fu, M. Lou, and Y. Yu, "SegMAN: Omni-scale context modeling with state space models and local attention for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2025, pp. 19077–19087.

[68] H. Yan, M. Wu, and C. Zhang, "Multi-scale representations by varing window attention for semantic segmentation," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 667–684.

[69] (2020). *MMSegmentation: OpenMMLab Semantic Segmentation Toolbox and Benchmark*. [Online]. Available: https://github.com/open-mmlab/mmsegmentation

[70] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 7132–7141.

[71] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[72] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539.

[73] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13708–13717.