2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

HSAN: A Side Adapter Network with Hybrid Compression and Local Enhancement Attention

Yankui Fu[1], Fang Yang[1[0000-0001-5948-3965]] and Qingxuan Shi[1]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071002, Hebei, China
yangfang@hbu.edu.cn

Abstract. Significant progress has been made in open-vocabulary semantic segmentation tasks, particularly in recognizing and segmenting unseen categories by leveraging Contrastive Language-Image Pre-training (CLIP). Among existing methods, the Side Adapter Network (SAN) stands out as an effective approach, achieving strong performance. However, we identify that SAN does not perform well in capturing fine-grained local features in complex scenes and high-resolution images. Additionally, it suffers from high computational costs and struggles to effectively fuse the features generated by its internal modules with those extracted by CLIP, resulting in segmentation accuracy. To address these issues, we propose HSAN, which introduces the Hybrid Compression and Local Enhancement Attention (HCLEA) mechanism to reduce dimensionality for lower computational complexity while using additional convolutional neural networks to preserve and enhance local features. Furthermore, we design an Adaptive Feature Fusion Block (AFFB) that dynamically adjusts fusion weights based on input features, achieving better global-local feature fusion and fully leveraging CLIP's generalization ability. Extensive benchmarks show that our method improves both accuracy and inference speed over SAN and other baselines.

Keywords: Open-Vocabulary Semantic Segmentation, Attention Mechanism, Feature Fusion, CLIP.

1 Introduction

Semantic segmentation assigns category labels to each pixel, requiring object recognition and precise boundaries. Traditional models [1-5] work well on known categories but struggle with unseen ones due to fixed label sets. This limitation hinders their flexibility and robustness in complex environments. To address this challenge, open-vocabulary semantic segmentation uses vision-language models for cross-modal alignment, enabling segmentation of unseen categories and improving generalization.

Open-vocabulary semantic segmentation typically relies on large-scale pre-trained vision-language models [6-7], especially CLIP [6]. Nevertheless, CLIP is originally built for image-level alignment and lacks the detailed feature granularity required for pixel-level semantic understanding. One solution is to frame the segmentation task as a region recognition task, as demonstrated by [8-9], which helps in unseen category

segmentation but suffers from limited performance in complex scenes and high computational cost.

To overcome these limitations, improved model architectures have been proposed. Among them, the Side Adapter Network (SAN) [10] introduces a lightweight auxiliary network to improve feature representation and handle unseen categories. Nevertheless, SAN integrates mask generation and classification effectively, it still faces challenges in achieving a balance between efficiency and accuracy. The self-attention mechanism used in SAN is computationally intensive for high-resolution images, and its element-wise fusion struggles to capture fine-grained local details in complex scenes.

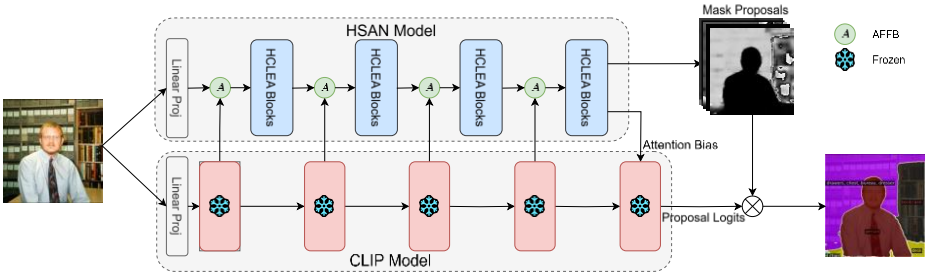

Fig. 1. Overall architecture of the proposed framework. The CLIP model is kept frozen during training. The HSAN module generates attention biases and mask proposals, guiding deeper CLIP layers in making full-image predictions. During inference, the final segmentation is obtained by combining the proposal masks with the corresponding logits.

To improve upon the existing SAN framework, we introduce an upgraded architecture, as illustrated in Fig. 1. Specifically, we incorporate the Hybrid Compression and Local Enhancement Attention (HCLEA) mechanism into the design to reduce computation cost while improving fine-grained feature representation. This modification enhances model inference efficiency without compromising segmentation quality. In addition, we develop the Adaptive Feature Fusion Block (AFFB), which adaptively recalibrates fusion weights based on input features, further boosting model performance without increasing latency.

The proposed improvements lead to significant gains in both segmentation accuracy and computational efficiency. Experimental evaluations reveal that the HCLEA-AFFB enhanced model consistently surpasses several established baselines across diverse benchmark datasets, particularly excelling in scenarios demanding a balance between speed and fine-detail recognition.

The contributions of this paper are as follows: (1) We present an improved local information enhancement architecture by integrating SAN with the HCLEA mechanism, targeting performance gains for open-vocabulary segmentation. (2) AFFB is employed to better align CLIP-derived visual representations with internal features, yielding improved segmentation quality. (3) Extensive comparisons across five public benchmarks confirm that our approach achieves highly competitive performance in terms of accuracy, generalization, and runtime efficiency.

2 Related Work

2.1 Open-Vocabulary Semantic Segmentation and Side Adapter Network

Open-vocabulary semantic segmentation is a computer vision task that requires a model to perform pixel-level classification and segmentation for unseen categories. Early approaches [11-12] primarily focused on mapping visual inputs and label names into a shared embedding space. More recently, vision-language models such as CLIP [6] have been leveraged to address this challenge by aligning cross-modal representations. Some methods [8, 13] attempt to fine-tune vision-language models, but this requires large amounts of additional data and may weaken the model's open-vocabulary capabilities. Other strategies [9, 14, 15] adopting a two-stage method suffer from the separation between the mask generator and classifier, leading to feature misalignment and affecting classification performance. Furthermore, processing each mask proposal individually increases computational load, resulting in lower inference efficiency.

SAN [10] proposes an end-to-end architecture that uses side adapter modules to combine the backbone network and the CLIP model, enabling efficient classification of mask proposals. SAN integrates the mask generation and classification processes seamlessly, avoiding the incompatibility issues caused by the separation of the mask generator and CLIP model in two-stage methods. However, the self-attention mechanism used in SAN is computationally expensive when handling large-scale images, and its feature fusion method (element-wise addition) has certain limitations in complex scenes, failing to adequately capture fine-grained local information.

To address these drawbacks, we proposes the Hybrid Compression and Local Enhancement Attention (HCLEA) mechanism and an Adaptive Feature Fusion Block (AFFB), which together reduce overhead and improve the preservation of fine-grained local features.

2.2 Vision Transformers

In recent developments, the Vision Transformer (ViT) [16] has gained significant attention in computer vision, achieving notable success across a variety of tasks. This has, in turn, stimulated the creation and refinement of numerous ViT derivatives [17-19]. Convolutional Neural Networks (CNNs) are known for their strong capabilities in extracting localized visual features, while attention mechanisms possess the unique advantage of capturing global contextual information. Combining the two allows for leveraging both CNN's local perception capabilities and self-attention's global relational modeling strengths. This combination can take several forms, including using CNNs in the model backbone [20], integrating convolution operations into the MHSA module [21-23], or embedding MHSA into ResNet-like architectures [24].

Recently, researchers have been exploring effective alternatives to the multi-head self-attention (MHSA) module, as MHSA often becomes the computational bottleneck in ViT architectures. A common optimization strategy is to reduce the spatial dimensions of the MHSA module to lower computational complexity. During MHSA computation, the input is mapped to query, key, and value. Some recent studies [25-27]

have found that downsampling the query, key, and value can significantly improve computational efficiency while maintaining accuracy. Our work adopts a similar approach to achieve a more efficient model design.

3 Method

3.1 Overall Architecture

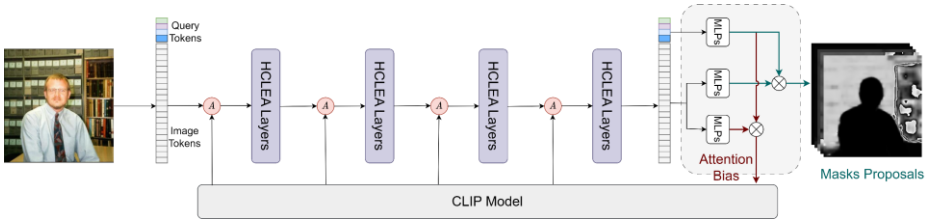

Fig. 2. HSAN Network Architecture. The input image is tokenized and combined with learnable query tokens. These tokens are processed through multiple HCLEA layers, where shallow CLIP features are fused at different stages. The final tokens are fed into MLPs to produce attention biases, which guide the CLIP model and generate mask proposals.

To address the challenges of fine-grained feature extraction and open-vocabulary segmentation, we introduce HSAN, a lightweight end-to-end network designed to work with a frozen CLIP model. HSAN enables effective segmentation by producing two key outputs: mask proposals for locating object regions, and attention bias for recognizing mask categories. This decoupled design allows category recognition to leverage feature regions beyond the mask areas, enhancing segmentation performance in complex scenes.

Fig. 2 shows that the input image is projected into visual tokens and fused with shallow features from the CLIP model. These tokens are fused with trainable queries, which are then refined via a stack of HCLEA layers to better capture local semantics. This refinement enhances the model's ability to focus on spatially detailed regions and improves representation quality. MLPs are applied afterward to generate both spatial mask hypotheses and adaptive attention biases. These outputs serve as intermediate guidance for the downstream attention injection module.

The attention bias is injected into the self-attention layers of the CLIP model, specifically targeting SLS tokens, which are initialized from the CLS tokens and attend only to visual inputs. During attention computation, the bias modifies the interaction as follows:

$$X^{l}_{[SLS]} = softmax\left(Q^{l}_{[SLS]}K^{l}_{visual} + B\right)V^{l}_{[SLS]} \tag{1}$$

This allows the SLS tokens to dynamically align with the semantic regions, ultimately producing accurate segmentation results.

3.2 Hybrid Compression and Local Enhancement Attention

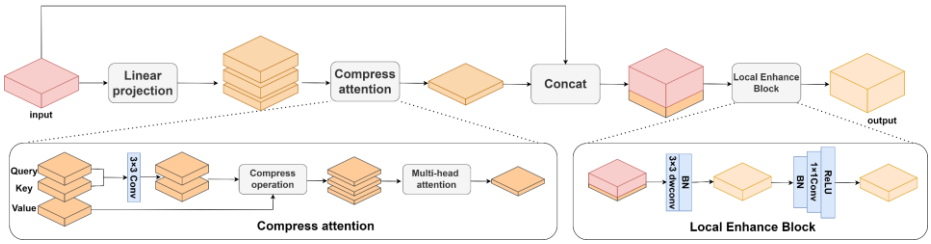

Fig. 3. The proposed Hybrid Compression and Local Enhancement Attention module. The top illustrates the overall workflow, while the bottom left and right show the compression attention and local enhancement components, respectively.

We propose Hybrid Compression and Local Enhancement Attention to improve efficiency and local feature modeling in transformer-based segmentation. The structure of HCLEA is illustrated in Fig. 3, and it comprises two main stages: a compression attention module that reduces computational cost and a local enhancement module that recovers fine-grained details.

In the compression attention module of HCLEA, the input sequence $x \in \mathbb{R}^{L\times N}$ where L is the sequence length and N is the embedding dimension, first undergoes linear projection to obtain the initial Q, K, and V representations. To boost the model's sensitivity to local features, both the Q and K are further processed by a convolutional layer, producing locally enhanced representations. These features are then compressed to a lower-dimensional space using a convolutional layer, resulting in the compressed query, key, and value, denoted as $Q_{lc}, K_{lc}, V_c \in \mathbb{R}^{L\times Nc}$. This process is formally defined as:

$$Q_{lC} = f_c(Conv_{local}(Q)),\ K_{lC} = f_c(Conv_{local}(K)),\ V_c = f_c(V) \tag{2}$$

Here, $Conv_{local}(\cdot)$ denotes a convolution applied for local enhancement, and $f_c(\cdot)$ denotes a linear transformation that reduces the feature dimension from N to N_c. Based on these compressed representations, scaled dot-product attention is applied to compute the final output of the compression module:

$$x_{ca} = softmax\left(\frac{Q_{lc}k_k^{\mathrm{T}}}{\sqrt{N_c}}\right)V_c \tag{3}$$

This design reduces the attention complexity from $O(L^2 \times N)$ to $O(L^2 \times N_c)$, while preserving global context and enhancing local feature sensitivity through convolutional transformation.

After obtaining the output $x_{ca} \in \mathbb{R}^{L\times Nc}$ from the compression attention module, we enhance local details through a local enhancement operation. Specifically, we concatenate the original input x with x_{ca} along the channel dimension to form a fused feature $h \in \mathbb{R}^{L\times(N+Nc)}$. Then h are first passed through a 3×3 convolution to strengthen local feature extraction, followed by a BN layer and a ReLU activation function. To adjust

the channel dimension and generate the final output, a 1×1 convolution is applied. The output of the local enhancement module is denoted as $x_{hclea} \in \mathbb{R}^{L \times N_c}$.

By combining the compression attention stage with the local enhancement stage, the overall time complexity of HCLEA is $O\left(L^2 \times N_C + L \times (N + N_C)\right)$. Compared to the traditional self-attention mechanism, this time complexity significantly reduces the computational demand, while the convolution operations improve the model's capacity for extracting localized features. This enables HCLEA to efficiently handle complex visual tasks while retaining global information.

3.3 Adaptive Feature Fusion Block

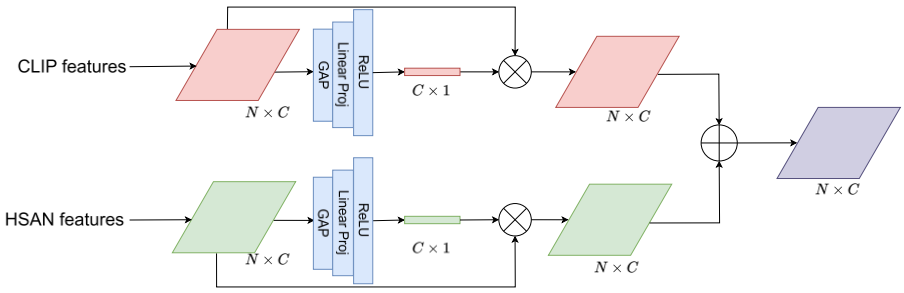

Fig. 4. Diagram of the AFFB module. The upper and lower paths correspond to the input CLIP features and HSAN features, respectively.

To enable more flexible and context-aware feature fusion, we introduce the Adaptive Feature Fusion Block (AFFB), which dynamically adjusts fusion weights based on the global characteristics of input features. By assigning adaptive importance to different channels, AFFB helps the network distinguish between informative and redundant features under varying visual conditions.

As illustrated in Fig. 4, given two input feature maps $F_1, F_2 \in \mathbb{R}^{H \times W \times C}$, global average pooling is first applied to each to obtain compact global descriptors. These descriptors are processed by a shared fully connected layer with ReLU activation to generate channel-wise attention weights $\alpha, \beta \in \mathbb{R}^{1 \times 1 \times C}$. The weights are then broadcasted and used to modulate the input features, which are subsequently fused and refined. The overall computation can be formulated as:

$$F_{fused} = \sigma\left(Conv(\alpha \cdot F_1 + \beta \cdot F_2)\right) \tag{4}$$

Here, $\cdot$ denotes element-wise multiplication, $Conv(\cdot)$ is a 1×1 convolution layer, and $\sigma(\cdot)$ is a non-linear activation function. This design allows the network to adaptively adjust fusion weights based on global context, enhancing feature representation and robustness across diverse inputs. By learning content-aware weighting for each feature map, AFFB enables the model to emphasize more informative features while suppressing redundant or irrelevant ones. Compared to static fusion methods, AFFB introduces

greater flexibility with minimal computational overhead, making it suitable for real-time applications.

4 Experiments

4.1 Datasets and Evaluation Metrics

We conduct open-vocabulary semantic segmentation experiments on six benchmark datasets: COCO Stuff [28], ADE20K-150 [29], ADE20K-847 [29], Pascal Context-59 [30], Pascal Context-459 [30], and Pascal VOC [31]. Among these, the COCO Stuff dataset is used for model training, while the other datasets are used for testing.

COCO Stuff. COCO Stuff contains 164,000 images annotated with 171 categories, split into 118,000 training, 5,000 validation, and 41,000 test images. We utilize the entire training set in our experiments.

ADE20K-150 (ADE-150). ADE20K-150 offers dense annotations for 150 categories, including scenes, objects, and parts. It contains 20,000 training and 2,000 validation images, and is widely used in semantic segmentation tasks.

ADE20K-847 (ADE-847). Sharing the same images as ADE20K-150, ADE20K-847 expands the label set to 847 categories. It focuses on testing generalization to unseen concepts, making it suitable for open-vocabulary evaluation.

Pascal VOC (VOC). Pascal VOC provides 20 segmentation classes and serves as a benchmark for classification, detection, and segmentation. It contains 1,464 training and 1,449 validation samples.

Pascal Context-59 (PC-59). PC-59 extends Pascal VOC with 59 categories and fine-grained pixel-level annotations. The dataset consists of 5,000 training and 5,000 validation images.

Pascal Context-459 (PC-459). PC-459 builds upon PC-59 by increasing the number of annotated categories to 459, using the same image set. It is commonly used in open-vocabulary segmentation benchmarks.

Evaluation Metrics. We adopt the mean Intersection over Union (mIoU) metric to evaluate segmentation performance. As a standard and category-balanced measure, mIoU computes average IoU across all classes, making it suitable for performance comparison and robustness analysis.

4.2 Implementation Details

The default HSAN includes 8 HCLEA layers, each configured with 240 output channels. Within the multi-head attention unit, 6 attention heads are used with a block size of 16, and 100 learnable query tokens are incorporated. For visual encoding, the ViT-B/16 CLIP model pre-trained on image-text pairs is adopted, using 224×224 resolution inputs. The first 9 layers are responsible for feature integration, while the final 3 focus on recognizing masks. The input image resolution is standardized to 640×640.

Model training is conducted on the COCO Stuff dataset. During training, we apply data augmentation techniques [5, 9, 15] , including random scaling (short edge from 320 to 1024) and cropping to 640×640. Optimization is performed with the AdamW algorithm, starting with a base learning rate of 1e-4 and a weight decay parameter also set to 1e-4. The batch size is maintained at 8, and the model is trained over 60,000 iterations. A polynomial decay schedule is used to adjust the learning rate, with a decay power of 0.9.

For inference, all experiments are run on a single NVIDIA GeForce RTX 4090 GPU using CUDA 11.3 and PyTorch 1.10.0 on Ubuntu 20.04. The system setup includes 60 GB of RAM and 24 GB of VRAM. Training is conducted in an Anaconda virtual environment to ensure consistent dependencies.

Table 1. Performance of Different Methods on Various Datasets.

Method	VL-Model	Training Dataset	ADE-847	PC-459	ADE-150	PC-59	VOC
Group VIT[32]	rand.init.	CC12M+YFCC	-	-	-	22.4	52.3
LSeg+[33]	ALIGN RN101	COCO	2.5	5.2	13.0	36.0	59.0
OpenSeg[33]	ALIGN RN101	COCO	4.0	6.5	15.3	36.9	60.0
LSeg+[33]	ALIGN EN-B7	COCO	3.8	7.8	18.0	46.5	-
OpenSeg[33]	ALIGN EN-B7	COCO	6.5	9.0	21.1	42.1	-
OpenSeg[33]	ALIGN EN-B7	COCO+Loc. Narr.	8.8	12.2	28.6	48.2	72.2
SimSeg[10]	CLIP ViT-B/16	COCO	7.0	8.7	20.5	47.7	88.4
OvSeg[9]	CLIP ViT-B/16	COCO	7.1	11.0	24.8	53.3	92.6
SAN[10]	CLIP ViT-B/16	COCO	9.5	12.0	26.6	52.2	93.5
HSAN(ours)	CLIP ViT-B/16	COCO	**10.3**	**12.9**	**27.7**	**54.3**	**94.7**

4.3 System Comparison

We performed a systematic comparison between our method and other open-vocabulary semantic segmentation methods, and the results are presented in Table 1. In the table, VL-Model indicates the vision-language model used, Training Dataset refers to the dataset used during model training, and rand. init. refers to random initialization. Additionally, RN101 represents ResNet-101 [35], while EN-B7[36] denotes EfficientNet-B7 [36]. All methods were trained using the COCO dataset, with images of resolution 640×640 applied to all models, and each model processing only one image per inference. Under these settings, our method achieves superior accuracy compared to other methods, with an mIoU improvement of approximately +1.2 mIoU.

Table 2. Comparison of inference speeds (FPS) across methods.

Method	FPS
SimSeg[10]	0.8
OvSeg[10]	0.8
MaskClip[10]	4.1
SAN[10]	15.2
HSAN(ours)	20.7

To further validate the efficiency of our method, we conducted an additional experiment to compare the inference speed (FPS) of various models. To highlight the speed advantage of our method, we use Titan XP for the experiments in Table 2. Our method achieved significantly higher FPS compared to other open-vocabulary semantic segmentation methods, demonstrating its superior efficiency while maintaining state-of-the-art accuracy.

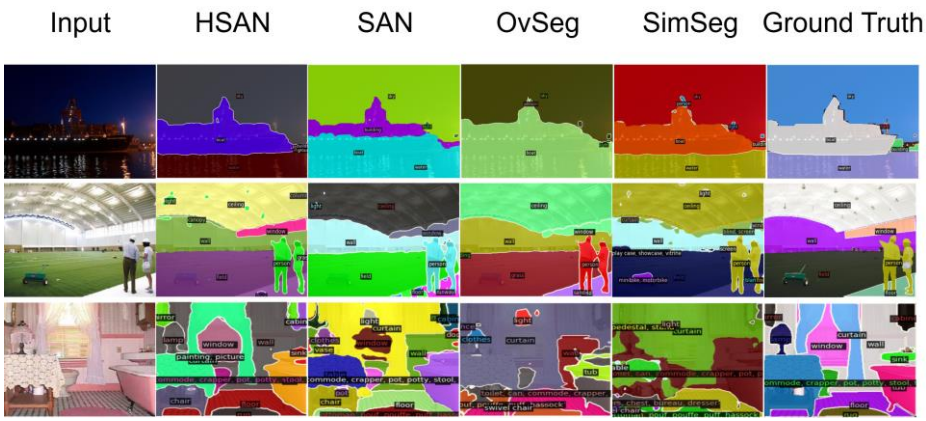

Fig. 5. Visual comparison of open-vocabulary segmentation results from different methods.

For a more intuitive demonstration of the performance, we visualized the model's predictions and displayed them in Fig. 5. In the experiments, we selected several scenes with fine-grained local features for testing, with images taken from the ADE150 and ADE847 datasets.

4.4 Ablation Study

Unless otherwise specified, the default configuration uses the ViT-B/16 CLIP model, with a feature dimension of 240, 6 heads in the multi-head attention, and an 8-layer side adapter network.

HCLEA Module. The reason HSAN improves both model accuracy and inference speed is that it replaces plain self-attention with our proposed HCLEA. To evaluate

HCLEA in practical applications and to demonstrate its advantages compared to other mainstream attention mechanisms, we designed a set of comparative experiments. The primary motivation of this experiment is to compare the performance of HCLEA with other attention mechanisms in semantic segmentation tasks using a unified model architecture and training environment. This aims to verify HCLEA's actual contributions in handling irregular-length sequences, reducing computational complexity, and improving model accuracy.

To ensure the fairness of the experiments, we only replaced the attention mechanism in the HSAN model architecture. Each experiment used the same dataset, training strategy, and hyperparameter settings, ensuring that the final results reflect the differences in the performance of the attention mechanisms themselves.

Table 3. Efficiency and performance comparison of attention mechanisms on PC-59.

Method	mIoU	FPS	Param.(M)
Self Attention[36]	52.2	56.5	8.4
Deformable Attention[37]	51.1	77.8	7.9
SwiftFormer[17]	51.3	92.5	8.6
HCLEA(ours)	52.7	69.4	8.6

Table 3 presents the performance of various attention mechanisms in the segmentation task, evaluated within the same model framework (HSAN) by solely replacing the attention mechanism used for computation. The HCLEA mechanism stands out in terms of mIoU, achieving 52.7, outperforming other methods. This result demonstrates that HCLEA can capture global information while retaining fine-grained local features, making it perform better in handling complex scenes, especially in detail recognition. Although other methods show advantages in inference speed, they fall short in balancing global and local features, resulting in slightly lower segmentation accuracy compared to HCLEA. Its ability to retain structural integrity while enhancing detailed regions contributes to better overall performance. While some alternative methods demonstrated faster inference, they struggled to maintain consistency across both local and global features, leading to marginally lower accuracy compared to HCLEA.

Table 4 shows the impact of different compression levels on the performance of the HCLEA module. Compression Level refers to how many dimensions remain after compression compared to the original dimensions. The uncompressed HCLEA module achieved a segmentation accuracy (mIoU) of 26.85, with a parameter count of 12.3M. When the compression level increased to 1/2, the model's accuracy slightly dropped to 26.75, but the parameter count was significantly reduced to 8.6M, improving computational efficiency. As the compression level further increased to 1/4 and 1/8, the model's accuracy dropped to 26.69 and 26.62, respectively, while the parameter count reduced to 8.1M and 7.7M. These results indicate that moderate compression reduces model complexity with only a slight loss in performance. In contrast, higher compression levels continue to reduce the parameter count but also lead to a drop in accuracy. Therefore, choosing a 1/2 compression ratio as the default setting strikes a good balance between model accuracy and computational efficiency.

Table 4. Comparison of compression levels in HCLEA on ADE-150 mIoU.

Compression Level	mIoU	Param.(M)
1/1	26.85	12.3
1/2	26.75	8.6
1/4	26.69	8.1
1/8	26.62	7.7

In the HCLEA module design, the local enhancement component applies convolutional operations along with feature concatenation to improve the model's ability to capture detailed local patterns. By assessing the model's performance across different configurations, we analyze the effect of the local enhancement module on segmentation accuracy.

Table 5. Effects of local enhancement components on ADE-150 mIoU.

Concat?	Conv?	mIoU
no	no	26.6
yes	no	26.8
yes	yes	26.9

We perform an ablation study to assess the role of the local enhancement module in HCLEA.. As shown in Table 5, we compare three configurations: the full local enhancement module, removal of the convolution operation, and complete removal of the local enhancement module. The complete HCLEA achieves the highest mIoU of 26.9. Removing the convolution decreases performance to 26.8, while eliminating the entire module results in a further drop to 26.6. These results highlight the importance of local enhancement, particularly the convolution operation, in capturing fine-grained details. While some enhancement effect remains without convolution, the absence of effective local modeling leads to performance degradation. The complete removal causes a more significant decline, confirming the module's contribution to robust segmentation in complex scenes.

Importance of Adaptive Feature Fusion Block. The AFFB is one of the core innovations in our model. We expect that the AFFB module, through dynamically calculating feature weights, will enable more flexible and effective feature fusion, thereby improving the overall performance of the model.

Table 6. Comparison of AFFB module usage on ADE-847 dataset.

use AFFB?	mIoU	FPS
yes	9.7	52.4
no	9.5	53.4

We evaluate the impact of the AFFB module on the ADE-847 dataset. As shown in Table 6, the model with AFFB achieves a higher mIoU of 9.7 compared to 9.5 without it. While the addition of AFFB slightly reduces inference speed, the difference is negligible. These results demonstrate that AFFB improves segmentation performance by enabling adaptive feature fusion. Unlike fixed fusion strategies, AFFB dynamically adjusts weight distribution based on input diversity, enhancing the model's ability to handle fine-grained and complex features.

Table 7. Comparison of AFFB implementation methods.

Implementation Method	mIoU	FPS
AVG	9.8	35.0
Conv	9.7	34.5

In **Table 7**, we compare two weight calculation methods in the AFFB: Global Average Pooling (GAP) and convolution operations. The experimental results show that using GAP for weight calculation slightly outperforms the convolution-based approach in mIoU, achieving 9.8, while the convolution-based method reaches 9.7. In terms of inference speed, the GAP method achieves 35.0 FPS, while the convolution method reaches 34.5 FPS. This indicates that GAP has lower computational overhead and effectively captures global features. Therefore, when dealing with large-scale data and scenarios rich in global information, it offers a better balance between performance and speed. Although convolution operations have an advantage in capturing local features, their higher computational cost results in slightly slower inference speed compared to GAP.

4.5 Discussion of Limitations

Although the HCLEA and AFFB proposed in this paper demonstrate excellent performance in open-vocabulary semantic segmentation tasks, there are still some limitations. First, while HCLEA's compression strategy effectively reduces computational complexity, its design is relatively simple. In the future, more flexible compression schemes could be explored to further enhance feature representation capabilities. Additionally, the AFFB module prioritizes accuracy during inference, which leads to a slight sacrifice in inference speed. Future research could focus on improving inference speed while maintaining accuracy. Lastly, the model has primarily been tested on standard datasets, and further validation is needed in more complex real-world application scenarios to verify its generalization ability.

5 Conclusion

In this work, we introduced the HSAN framework, which integrates the Hybrid Compression and Local Enhancement Attention (HCLEA) mechanism alongside the Adaptive Feature Fusion Block (AFFB) to address open-vocabulary semantic segmentation.

Compared with existing approaches, our method incorporating HCLEA and AFFB delivers superior segmentation results. The HCLEA module achieves a strong balance between computational cost and the ability to extract both global and detailed features, while AFFB improves fusion adaptability via dynamic weighting. Extensive evaluations conducted on multiple benchmarks confirm that the proposed method performs competitively against leading segmentation models. Furthermore, ablation experiments were carried out to assess the individual impact of each module on the system's overall effectiveness.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

. References

1. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. IEEE, Boston (2015).
2. Ronneberger, O., Fischer, P., and Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, LNCS, vol. 9351, pp. 234–241. Springer, Heidelberg (2015).
3. Li, X., Zhang, J., Yang, Y., et al.: SFNet: Faster and accurate semantic segmentation via semantic flow. International Journal of Computer Vision 132(2), 466–489 (2024).
4. Lin, G., Milan, A., Shen, C., et al.: RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1925–1934. IEEE, Honolulu (2017).
5. Cheng, B., Misra, I., Schwing, A.G., et al.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1290–1299. IEEE/CVF, New Orleans (2022).
6. Radford, A., Kim, J.W., Hallacy, C., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763. PMLR, Virtual (2021).
7. Jia, C., Yang, Y., Xia, Y., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International Conference on Machine Learning, pp. 4904–4916. PMLR, Virtual (2021).
8. Li, B., Weinberger, K.Q., Belongie, S., et al.: Language-driven semantic segmentation. arXiv preprint arXiv:2201.03546 (2022).
9. Liang, F., Wu, B., Dai, X., et al.: Open-vocabulary semantic segmentation with mask-adapted CLIP. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7061–7070. IEEE/CVF, Vancouver (2023).
10. Xu, M., Zhang, Z., Wei, F., et al.: Side adapter network for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2945–2954. IEEE/CVF, Vancouver (2023).
11. Bucher, M., Vu, T.H., Cord, M., et al.: Zero-shot semantic segmentation. Advances in Neural Information Processing Systems 32 (2019).
12. Xian, Y., Choudhury, S., He, Y., et al.: Semantic projection network for zero-and few-label semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8256–8265. IEEE/CVF, Long Beach (2019).

13. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from CLIP. In: European Conference on Computer Vision, pp. 696–712. Springer, Cham (2022).
14. Ding, J., Xue, N., Xia, G.S., et al.: Decoupling zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11583–11592. IEEE/CVF, New Orleans (2022).
15. Xu, M., Zhang, Z., Wei, F., et al.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: European Conference on Computer Vision, pp. 736–753. Springer, Cham (2022).
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
17. Shaker, A., Maaz, M., Rasheed, H., et al.: Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17425–17436. IEEE/CVF, Paris (2023).
18. Han, D., Ye, T., Han, Y., et al.: Agent attention: On the integration of softmax and linear attention. In: European Conference on Computer Vision, pp. 124–140. Springer, Cham (2024).
19. Ma, X., Zhou, C., Kong, X., et al.: Mega: Moving average equipped gated attention. arXiv preprint arXiv:2209.10655 (2022).
20. Xiao, T., Singh, M., Mintun, E., et al.: Early convolutions help transformers see better. Advances in Neural Information Processing Systems 34, 30392–30400 (2021).
21. Mehta, S., Rastegari, M.: MobileViT: Light-weight, general-purpose, and mobile-friendly vision transformer. arXiv preprint arXiv:2110.02178 (2021).
22. Wu, H., Xiao, B., Codella, N., et al.: CvT: Introducing convolutions to vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 22–31. IEEE/CVF, Virtual (2021).
23. Graham, B., El-Nouby, A., Touvron, H., et al.: LeViT: A vision transformer in convnet's clothing for faster inference. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12259–12269. IEEE/CVF, Virtual (2021).
24. Srinivas, A., Lin, T.Y., Parmar, N., et al.: Bottleneck transformers for visual recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16519–16529. IEEE/CVF, Virtual (2021).
25. Pan, J., Bulat, A., Tan, F., et al.: EdgeViTs: Competing light-weight CNNs on mobile devices with vision transformers. In: European Conference on Computer Vision, pp. 294–311. Springer, Cham (2022).
26. Chu, X., Tian, Z., Wang, Y., et al.: Twins: Revisiting the design of spatial attention in vision transformers. Advances in Neural Information Processing Systems 34, 9355–9366 (2021).
27. Wang, W., Xie, E., Li, X., et al.: PVT v2: Improved baselines with pyramid vision transformer. Computational Visual Media 8(3), 415–424 (2022).
28. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and stuff classes in context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1209–1218. IEEE, Salt Lake City (2018).
29. Zhou, B., Zhao, H., Puig, X., et al.: Scene parsing through ADE20K dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 633–641. IEEE, Honolulu (2017).
30. Mottaghi, R., Chen, X., Liu, X., et al.: The role of context for object detection and semantic segmentation in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 891–898. IEEE, Columbus (2014).
31. Everingham, M., Eslami, S.M.A., Van Gool, L., et al.: The PASCAL visual object classes challenge: A retrospective. International Journal of Computer Vision 111, 98–136 (2015).

32. Xu, J., De Mello, S., Liu, S., et al.: GroupViT: Semantic segmentation emerges from text supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18134–18144. IEEE/CVF, New Orleans (2022).
33. Ghiasi, G., Gu, X., Cui, Y., et al.: Scaling open-vocabulary image segmentation with image-level labels. In: European Conference on Computer Vision, pp. 540–557. Springer, Cham (2022).
34. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE, Las Vegas (2016).
35. Tan, M., Le, Q.: EfficientNet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning, pp. 6105–6114. PMLR, Long Beach (2019).
36. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Advances in Neural Information Processing Systems 30 (2017).
37. Zhu, X., Su, W., Lu, L., et al.: Deformable DETR: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020).