Supplementary material: SEAFORMER: SQUEEZE-ENHANCED AXIAL TRANSFORMER FOR MOBILE SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

A ARCHITECTURE DETAILS AND VARIANTS

A.1 SETTINGS AND VARIANTS

SeaFormer backbone contains 6 stages, corresponding to the shared STEM and context branch in Figure 2 in the main paper. When conducting the image classification experiments, a pooling layer and a linear layer are added at the end of the context branch.

Table 1 details the family of our SeaFormer configurations with varying capacities. We construct SeaFormer-Tiny, SeaFormer-Small, SeaFormer-Base and SeaFormer-Large models with different scales via varying the number of SeaFormer layers and the feature dimensions. We use input image size of 512×512 by default. For variants except SeaFormer-Large, SeaFormer layers are applied in the last two stages for superior trade-off between accuracy and efficiency. For SeaFormer-Large, we apply the proposed SeaFormer layers in each stage of the context branch.

A.2 EFFECTIVE AND EFFICIENCY OF SEA ATTENTION

To verify the effectiveness and efficiency of SEA attention based on our designed pipeline, we experiment with convolution, Global attention, Local attention, Axial attention and three convolution enhanced attention methods including our SEA attention, ACmix and MixFormer. The ablation experiments are organized in seven groups. Since the resolution of computing attention is relatively small, the window size in Local attention, ACmix, and MixFormer is set to 4. We adjust the channels when applying different attention modules to keep the FLOPs aligned and compare their performance and latency. The results are illustrated in Table 2.

As demonstrated in the table, SEA attention outperforms the counterpart built on other efficient attentions. Compared with global attention, SEA attention outperforms it by +1.2% Top1 accuracy on ImageNet-1K and +1.6 mIoU on ADE20K with less FLOPs and lower latency. Compared with similar convolution enhanced attention works, ACmix and MixFormer, our SEA attention obtains better results on ImageNet-1K and ADE20K with similar FLOPs but lower latency. The results indicate the effectiveness and efficiency of SEA attention module.

B PASCAL CONTEXT PERFORMANCE

We evaluate performance on Pascal Context *val* set over 59 categories and 60 categories. **PASCAL Context** dataset has 4998/5105 images for *train* and *test*, covering 59 semantic labels and 1 background.

Following TopFormer Zhang et al. (2022), we train the models for 80,000 iterations on PASCAL Context dataset. The same data augmentation strategy and batch size are adopted for a fair comparison. The initial learning rate is 0.0002 and the weight decay is 0.01. A poly learning rare scheduled with factor 1.0 is used.

Table 3 demonstrates that SeaFormer-S is +1.4% mIoU higher (45.08% vs.43.68%) than TopFormer-S with lower latency.

	Resolution	SeaFormer-Tiny	SeaFormer-Small	SeaFormer-Base	SeaFormer-Large
Stage1	H/2 × W/2	[Conv, 3, 16, 2] [MB, 3, 1, 16, 1]	[Conv, 3, 16, 2] [MB, 3, 1, 16, 1]	[Conv, 3, 16, 2] [MB, 3, 1, 16, 1]	[Conv, 3, 32, 2] [MB, 3, 3, 32, 1]
Stage2	H/4 × W/4	[MB, 3, 4, 16, 2] [MB, 3, 3, 16, 1]	[MB, 3, 4, 24, 2] [MB, 3, 3, 24, 1]	[MB, 3, 4, 32, 2] [MB, 3, 3, 32, 1]	[MB, 3, 4, 64, 2] [MB, 3, 4, 64, 1]
Stage3	H/8 × W/8	[MB, 5, 3, 32, 2] [MB, 5, 3, 32, 1]	[MB, 5, 3, 48, 2] [MB, 5, 3, 48, 1]	[MB, 5, 3, 64, 2] [MB, 5, 3, 64, 1]	[MB, 5, 4, 128, 2] [MB, 5, 4, 128, 1]
Stage4	H/16 × W/16	[MB, 3, 3, 64, 2] [MB, 3, 3, 64, 1]	[MB, 3, 3, 96, 2] [MB, 3, 3, 96, 1]	[MB, 3, 3, 128, 2] [MB, 3, 3, 128, 1]	[MB, 3, 4, 192, 2] [MB, 3, 4, 192, 1] [Sea, 3, 8]
Stage5	H/32 × W/32	[MB, 5, 3, 128, 2] [Sea, 2, 4]	[MB, 5, 4, 160, 2] [Sea, 3, 6]	[MB, 5, 4, 192, 2] [Sea, 4, 8]	[MB, 5, 4, 256, 2] [Sea, 3, 8]
Stage6	H/64 × W/64	[MB, 3, 6, 160, 2] [Sea, 2, 4]	[MB, 3, 6, 192, 2] [Sea, 3, 6]	[MB, 3, 6, 256, 2] [Sea, 4, 8]	[MB, 3, 6, 320, 2] [Sea, 3, 8]

Table 1: Architectures for semantic segmentation. [Conv, 3 ,16, 2] denotes regular convolution layer with kernel of 3, output channel of 16 and stride of 2. [MB, 3, 4, 16, 2] means MobileNetV2 Sandler et al. (2018) block with kernel of 3, expansion ratio of 4, output channel of 16 and stride of 2. [Sea, 2, 4] refers to SeaFormer layers with number of layers of 2 and heads of 4.

Method	Params	FLOPs	Latency	Top1	mIoU
Conv	1.6M	0.59G	38ms	66.3	32.8
Local	1.3M	0.60G	48ms	65.9	32.8
Axial	1.6M	0.63G	44ms	66.9	33.7
Global	1.3M	0.61G	43ms	66.7	34.2
ACmix	1.3M	0.60G	54ms	66.0	33.1
MixFormer	1.3M	0.60G	50ms	66.8	33.8
SeaFormer	1.7M	0.60G	**40ms**	**67.9**	**35.8**

Table 2: Performance of different self-attention modules on our designed pipeline on ImageNet-1K and ADE20K datasets.

Backbone	Decoder	F(G)	mIoU(60/59)
MBV2-s16	DeepLabV3+	22.24	38.59/42.34
ENet-s16	DeepLabV3+	23.00	39.19/43.07
MBV3-s16	LR-ASPP	2.04	35.05/38.02
TopFormer-T	Simple Head	0.53	36.41/40.39
SeaFormer-T	Light Head	0.51	**37.27/41.49**
TopFormer-S	Simple Head	0.98	39.06/43.68
SeaFormer-S	Light Head	0.98	**40.20/45.08**
TopFormer-B	Simple Head	1.54	41.01/45.28
SeaFormer-B	Light Head	1.57	**41.77/45.92**

Table 3: Results on Pascal Context *val* set. `F` means `FLOPs`. We omit the latency as the input resolution is almost the same as that in table 1.

Backbone	Decoder	F(G)	mIoU
MBV2-s8	PSPNet	52.94	30.14
ENet-s16	DeepLabV3+	27.10	31.45
MBV3-s16	LR-ASPP	2.37	25.16
TopFormer-T	Simple Head	0.64	28.34
SeaFormer-T	Light Head	0.62	**29.24**
TopFormer-S	Simple Head	1.18	30.83
SeaFormer-S	Light Head	1.15	**32.82**
TopFormer-B	Simple Head	1.83	33.43
SeaFormer-B	Light Head	1.81	**34.07**

Table 4: Results on COCO-Stuff *test* set. `F` means `FLOPs`. We omit the latency in this table as the input resolution is the same as that in table 1.

C COCO-STUFF PERFORMANCE

We compare SeaFormer with the previous approaches on COCO-Stuff *val* set. **COCO-Stuff** dataset augments COCO dataset with pixel-level stuff annotations. 10K complex images are selected from COCO. The *train* and *test* set contain 9K/1K images.

Following TopFormer Zhang et al. (2022), we train the models for 80,000 iterations on COCO-Stuff dataset. The same data augmentation strategy and batch size are adopted for a fair comparison. The initial learning rate is 0.0002 and the weight decay is 0.01. A poly learning rare scheduled with factor 1.0 is used.

Table 4 reveals that SeaFormer-S is +2.0% mIoU higher (32.82% vs.30.83%) than TopFormer-S with less computation cost and lower latency.

D OBJECT DETECTION PERFORMANCE

To further demonstrate the generalization ability of our proposed SeaFormer backbone on other downstream tasks, we conduct object detection task on COCO dataset.

D.1 SETUP

We use RetinaNet Lin et al. (2017) (one-stage) as the detection framework and follow the standard settings to use SeaFormer as backbone to generate e feature pyramid at multiple scales. All models are trained on train2017 split for 12 epochs (1×) from ImageNet pretrained weights.

D.2 RESULTS

From the table 5 We can observe that our SeaFormer achieves superior results on detection task which further demonstrates the strong generalization ability of our method.

E ADDITIONAL ABLATION STUDY

In addition to the ablation study in the submission paper, we investigate the effect of fusion method in fusion block.

E.1 THE INFLUENCE OF FUSION BLOCK DESIGN

Setup We set four fusion methods, including "Add directly", "Multiply directly", "Sigmoid add" and "Sigmoid multiply". **X** directly means features from context branch and spatial branch **X** directly. Sigmoid **X** means feature from context branch goes through a sigmoid layer and **X** feature from spatial branch.

Backbone	AP	FLOPs	Params
ShuffleNetv2 Ma et al. (2018)	25.9	161G	10.4M
SeaFormer-T	**31.5**	**160G**	10.9M
MF151	34.2	161G	14.4M
MV3	27.2	162G	12.3M
SeaFormer-S	**34.6**	**161G**	13.3M
MF214	35.8	**162G**	15.2M
MF294	36.6	164G	16.1M
SeaFormer-B	**36.7**	164G	18.1M
ResNet50 He et al. (2016)	36.5	239G	37.7M
PVT-Tiny Wang et al. (2021)	36.7	221G	23.0M
ConT-M Yan et al. (2021)	37.9	217G	27.0M
SeaFormer-L	**39.8**	**185G**	24.0M

Table 5: Results on COCO object detecion. MF denotes Mobile-Former Chen et al. (2022). MV3 denotes MobileNetV3 Howard et al. (2019).

Fusion method	mIoU
Add directly	35.2
Multiply directly	35.2
Sigmoid add	34.8
Sigmoid multiply	**35.8**

Table 6: Ablation study on fusion method on ADE20K *val* set.

Results From the Table 6 we can see that replacing sigmoid multiply with other fusion methods hurts performance. Sigmoid multiply is our optimal fusion block choice.

F Performance under different precision of the models

Following TopFormer, we measure the latency in the submission papere on a single Qualcomm Snapdragon 865, and only an ARM CPU core is used for speed testing. No other means of acceleration, e.g., GPU or quantification, is used. We provide a more comprehensive comparison to demonstrate the necessity of our proposed method. We test the latency under different precision of the models. From the table 7, it can be seen that whether it is full precision or half precision the performance of SeaFormer is better than that of TopFormer.

G Visualization

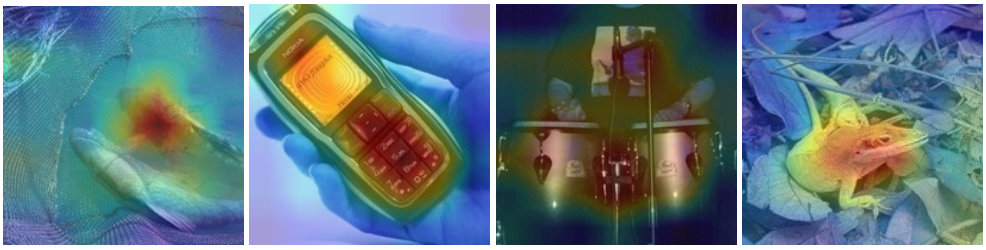
G.1 Attention heatmap

To demonstrate the effectiveness of detail enhancement in our squeeze-enhanced Axial attention (SEA attention), we ablate our model by removing the detail enhancement. We visualize the attention heatmaps of the two models in Figure 1. Without detail enhancement, attention heatmaps from

Model	mIoU	FP32	FP16
TopFormer-T	34.6	43ms	23ms
SeaFormer-T	**35.8**	**40ms**	**22ms**
TopFormer-S	37.0	74ms	41ms
SeaFormer-S	**39.4**	**67ms**	**36ms**
TopFormer-B	39.2	110ms	60ms
SeaFormer-B	**41.0**	**106ms**	**56ms**
SeaFormer-L	43.7	367ms	186ms

Table 7: Performance comparison on ADE20K *val* set under different precision.

(a) Squeeze Axial attention heatmaps



(b) Squeeze-enhanced Axial attention heatmaps

Figure 1: The visualization of attention heatmaps from the model consisting of squeeze Axial attention without detail enhancement (*first row*) and SeaFormer (*second row*). Heatmaps are produced by averaging channels of the features from the last attention block, normalizing to [0, 255] and upsampling to the image size.

solely SA attention appears to be axial strips while our proposed SEA attention is able to activate the semantic local region accurately, which is particularly significant in the dense prediction task.

G.2 Prediction results

We show the qualitative results and compare with the alternatives on the ADE20K validation set from two different perspectives. First we compare with a mobile-friendly rival TopFormer Zhang et al. (2022) with similar FLOPs and latency in Figure 2. Besides, we compare with the Transformer-based counterpart SegFormer-B1 Xie et al. (2021) in Figure 3. In particular, our SeaFormer-L has lower computation cost than the SegFormer-B1. As shown in both figures, we demonstrate better segmentation results than both the mobile counterpart and Transformer-based approach.

H Limitations and societal impact

The mobile-friendly segmentation is deeply related to the industrial application on edge computation platforms, while few academic attempts are made to meet the requirement of the industry. We test our method on a Qualcomm Snapdragon 865 processor (Fig.1 main paper) and shows superior results to the alternatives. We believe our work can lead to expected and unexpected innovations in both academia and industry.

However, our system is not perfect yet and hence not fully trustworthy in real-world deployment. Also, the current system is not exhaustively evaluated and tested due to limited resources. We focus on the mobile semantic segmentation and image classification tasks. New mobile-friendly method for more downstream tasks and extended to GPU systems will be studied in the future.

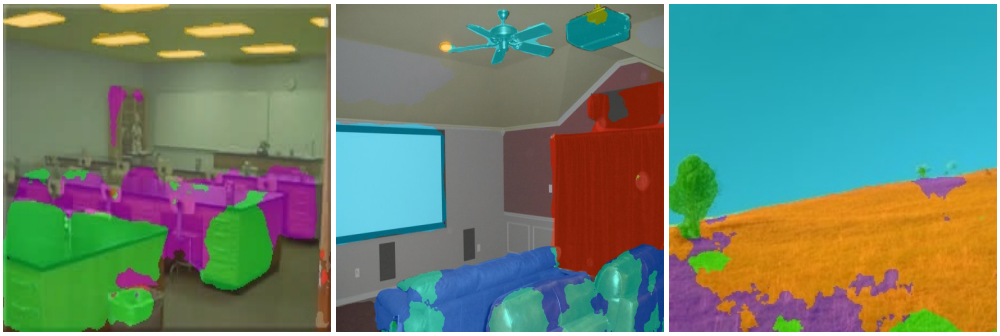
(a) Ground Truth

(b) TopFormer-B Zhang et al. (2022)

(c) SeaFormer-B (Ours)

Figure 2: Visualization of prediction results on ADE20K *val* set.

(a) Ground Truth



(b) SegFormer-B1 Xie et al. (2021)

(c) SeaFormer-L (Ours)

Figure 3: Visualization of prediction results on ADE20K *val* set.

REFERENCES

Yinpeng Chen, Xiyang Dai, Dongdong Chen, Mengchen Liu, Xiaoyi Dong, Lu Yuan, and Zicheng Liu. Mobile-former: Bridging mobilenet and transformer. In *CVPR*, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.

Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv preprint*, 2021.

Wenqiang Zhang, Zilong Huang, Guozhong Luo, Tao Chen, Xinggang Wang, Wenyu Liu, Gang Yu, and Chunhua. Shen. Topformer: Token pyramid transformer for mobile semantic segmentation. In *CVPR*, 2022.