Full-scale Selective Transformer for Semantic Segmentation

Fangjian Lin[1,2,3,*], Sitong Wu[2,*], Yizhe Ma[1], and Shengwei Tian[1**]

[1] School of Software, Xinjiang University, Urumqi, China
[2] Baidu VIS, Beijing, China
[3] Institute of Deep Learning, Baidu Research, Beijing, China
wusitong98@gmail.com, {linfangjian01, mayizhe01, tianshengwei}@163.com

Abstract. In this paper, we rethink the multi-scale feature fusion from two perspectives (scale-level and spatial-level) and propose a full-scale selective fusion strategy for semantic segmentation. Based on such strategy, we design a novel segmentation network, named Full-scale Selective Transformer (FSFormer). Specifically, our FSFormer adaptively selects partial tokens from all tokens at all scales to construct a token subset of interest for each scale. Therefore, each token only interacts with the tokens within its corresponding token subset of interest. The proposed full-scale selective fusion strategy can not only filter out the noisy information propagation but also reduce the computational costs to some extent. We evaluate our FSFormer on four challenging semantic segmentation benchmarks, including PASCAL Context, ADE20K, COCO-Stuff 10K, and Cityscapes, outperforming the state-of-the-art methods. We evaluate our FSFormer on four challenging semantic segmentation benchmarks, including PASCAL Context, ADE20K, COCO-Stuff 10K, and Cityscapes, outperforming the state-of-the-art methods.

Keywords: Semantic segmentation · Transformer · Full-scale feature fusion

1 Introduction

Semantic segmentation aims to predict a semantic label for each pixel in the image, which plays an important role for various applications such as autonomous driving [10] and medical analysis [28]. However, precisely recognize every pixel is still challenging as the objects vary across a wide range of scales. Since FPN [22], a typical and natural solution for this problem is to leverage both high-resolution feature maps with more detail information in shallow layers and high-level feature maps with richer semantics in deep layers via multi-scale feature fusion.

Many works [37, 20, 22, 17, 5, 42, 21, 43, 34, 46, 40, 32] have explored how to fuse multi-scale features. We rethink multi-scale feature fusion from two perspectives, scale-level and spatial-level. The former refers to the fusion strategy

* Equal contributions
** Corresponding author

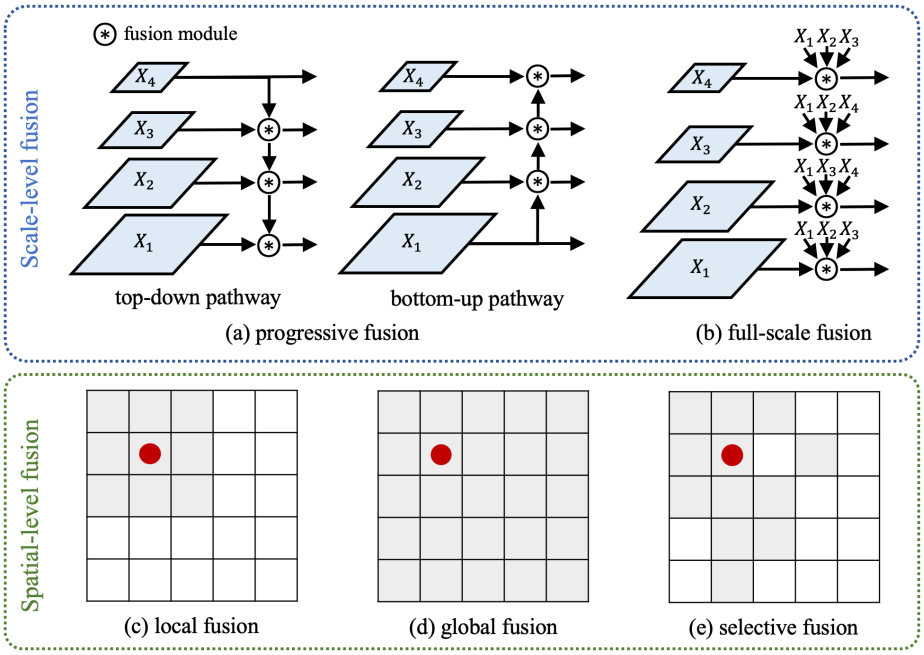

Fig. 1: Comparisons with different scale-level and spatial level feature fusion strategies. The gray shadow area in (c)(d)(e) represents the allowed interaction region of the query (denoted by red point).

across different scales, for example, for one scale, which scales can it interact with. And the latter refers to the interaction range of each token in the spatial dimension. On the one hand, previous scale-level fusion involves two main strategies, that is, progressive fusion and full-scale fusion. As shown in Figure 1(a), progressive fusion has two typical pathway (top-down and bottom-up), where the token at one scale can only interact with the tokens at its adjacent scale. By contrast, in full-scale fusion (Figure 1(b)), each token at one scale can interact with all the tokens at any scale. It has been proved that full-scale fusion has more advantages [19]. On the other hand, spatial-level fusion is a more popular topic. Benefited from the development of convolution, local fusion has been dominant for a long time. As shown in Figure 1(c), each token can only aggregate information from its neighbourhoods. Since the attention mechanism and Transformer architecture become show promising prospects, global fusion achieves more and more attention. As shown in Figure 1(d), each token can exchange information with all the tokens.

In order to accommodate both scale-level fusion and spatial-level fusion, we explore the full-scale global fusion using attention mechanism. Specifically, each token can interact with all the tokens at any scale. Although full-scale and global fusion strategies provide larger interaction range, they introduce more computation burden. Therefore, how to balance the trade-off between performance and

Fig. 2: Visualization of the attention map for full-scale global fusion. (a) is the input image. (b e) show the attention map of the query token at $1^{st} \sim 4^{th}$ scale located at the same position as the red point in (a). The attention map with red, green, orange and blue border corresponds to 1/32, 1/16, 1/8 and 1/4 key tokens respectively.

computational costs is a valuable problem. Through visualization, we found that the attention map of full-scale global fusion presents a sparse property. As shown in Figure 2, we sample a position in the image (denoted by a red point), and visualize the attention map corresponding to the query token located at such position in each scale. It can be found that the high attention weights only lie in partial region, while other area has the relatively small attention weights. This demonstrates that although each token has the opportunity to aggregate information with all the tokens, it has its own interaction range of interest. Note that the effective interaction range is not just simply local or global. In addition, compared with the last four columns of Figure 2, the token at the same spatial location of the feature map at different scales present a different interaction region of interest pattern. Based on this observation, we believe that the range of feature fusion is the more accurate the better, rather than the larger the better. Making each token only interact with other tokens within its region of interest may be a breakthrough to filter out the noisy information and reduce the computational costs.

In this paper, we propose a Full-scale Selective Transformer (FSFormer) for semantic segmentation. The core idea is to perform interaction among tokens via the proposed full-scale selective fusion strategy. Specifically, for each scale, our FSFormer adaptively select partial tokens from all tokens at all the scales to construct a token subset of interest. Each token only interact with the tokens within its corresponding token subset of interest, which is shared by the tokens belonging to the same scale. Such full-scale selective fusion strategy can not only filter out the noisy information propagation but also reduce the computational costs to some extent. To verify the effectiveness, we evaluate our FSFormer on four widely-used semantic segmentation benchmarks, including PASCAL Context [25], ADE20K [45], COCO-Stuff 10K [3], and Cityscapes [10], achieving 58.91%, 54.43%, 49.80%, and 84.46% mIoU respectively, outperforming the state-of-the-art methods.

2 Related Work

Multi-scale Features Fusion. There are various works exploring how to fuse multi-scale features for semantic segmentation. Inspired by FPN [22] that employed a top-down pathway and lateral connections for progressively fusing multi-scale features for object detection, Semantic-FPN [17] and SETR-MLA [43] extended this architecture to fuse multi-scale features for semantic segmentation. Based on this top-down fusion, ZigZagNet [21] proposed top-down and bottom-up propagations to aggregate multi-scale features, while FTN [32] proposed Feature Pyramid Transformer for multi-scale feature fusion. Differently, PSPNet [42] and DeepLab series [5, 4, 6] fused multi-scale features via concatenation at the channel dimension. Different from these methods that fused features on the local region, ANN [46] proposed an Asymmetric Fusion Non-local Block for fusing all features at one scale for each feature (position) on another scale, while FPT [40] proposed Grounding Transformer to ground the "concept" of the higher-level features to every pixel on the lower-level ones. Different from these methods that fuse features from preset subset for queries, we explore how to dynamically select informative subset from the whole multi-scale feature set and fuse them for each query feature.

Transformer-based semantic segmentation. Since Alexey *et al.* [11] introduced Visual Transformer (ViT) for image classification, it has attracted more and more attentions to explore how to use Transformer for semantic segmentation. These methods focused on exploring the various usages of Transformer, including extracting features [43, 27, 35] from input image, learning class embedding [36, 31], or learning mask embedding [9]. For example, SETR [43] treated semantic segmentation as a sequence-to-sequence prediction task and deployed a pure transformer (i.e., without convolution and resolution reduction) to encode an image as a sequence of patches for feature extraction. DPT [27] reassembled the bag-of-words representation provided by ViT into image-like features at various resolutions, and progressively combined them into final predictions. Differently, Trans2Seg [36] formulated semantic segmentation as a problem of dictionary look-up, and designed a set of learnable prototypes as the query of Transformer decoder, where each prototype learns the statistics of one category. SegFormer [35] used Transformer-based encoder to extract features and the lightweight MLP-decoder to predict pixel by pixel. Segmenter [31] employed a Mask Transformer to learn a set of class embedding, which was used to generate class masks. Recent MaskFormer [9] proposed a simple mask classification model to predict a set of binary masks, where a transformer decoder was used to learn mask embedding. Different from these works, we explore how to use Transformer to fuse multi-scale features.

3 Method

3.1 Overview

The overall framework of our FSFormer is shown in Figure 3. Given the input image $\mathcal{I} \in \mathbb{R}^{3\times H\times W}$, the backbone first maps it into multi-scale features $\{X_i\}_{i=1}^{4}$,

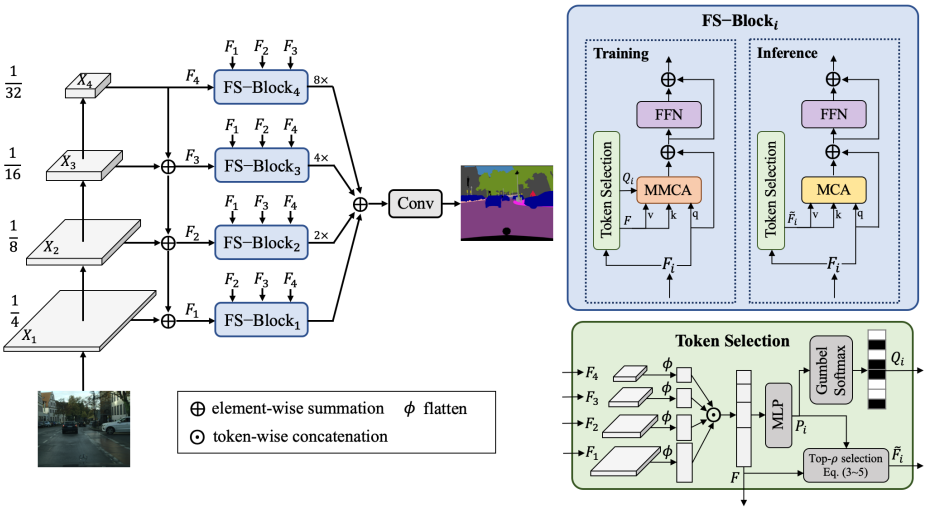

Fig. 3: The overall framework of our FSFormer, whose core component is Full-scale Selective Fusion Block (FS-Block).

where $X_i \in \mathbb{R}^{2^{i-1}C \times \frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}}}$. H and W denotes the height and width respectively. i indicates the scale index and C is the basic channel number. Then, a top-down pathway injects the high-level semantics into all scales to produce enhanced multi-scale representations $\{F_i\}_{i=1}^{4}$, where $F_i \in \mathbb{R}^{D \times \frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}}}$, and D is the channel number of decoder. Next, a full-scale selective block is employed on each scale feature $\{F_i\}$ for context modeling. Finally, we up-sample multi-scale features to the same resolution, followed by an element-wise summation and a simple convolution head (including a 3×3 and a 1×1 convolution) to predict final segmentation result.

3.2 Full-scale Selective Fusion Block

Previous works [22, 20, 42, 19, 40, 33] have shown that fusing multi-scale features from multiple scales are critical for improving semantic segmentation, since the objects in the scene often present a variety of scales. High-resolution features in shallow layers contain more spatial details than low-resolution ones in deeper layers, while the latter contains richer semantics. Besides, small-scale objects have no precise locations in the higher-level since the multiple down-sample operations, while large-scale objects have weak semantics at the lower-level since the insufficient receptive fields.

With regard to the scale-level fusion, fully-scale fusion [19], where each token at one scale has the ability to aggregate information from all the tokens at any scale, shows more advantages than the progressive fusion [22] (each scale can only fuse information from its adjacent scale). According to the spatial-level fusion, convolution-based local fusion takes the dominant position, before the

superior global fusion implemented by recent popular attention mechanisms [12] and Transformer architectures [11, 33]. Based on our attention map visualization under full-scale fusion (Figure 2), we found that although the global attention allows each token to fuse information from all the tokens, each token has its own interaction token subset of interest. Thus, the selective spatial-level fusion strategy may provide a better trade-off between performance and computational costs.

As the core component of our FSFormer, full-scale selective fusion block (FS-Block) aims to combine the full-scale fusion and selective spatial fusion. Specifically, as shown in Figure 3, for each token, it first predicts a token subset of interest from all the tokens at all scales through a token selection module. Note that the tokens at each scale share the same subset. Then, each token only aggregates information from the tokens within its corresponding token subset of interest via a transformer-based module.

Token Selection module. The token selection module is designed to adaptively select a token subset of interest for each scale according to the image content. Figure 3 shows the pipeline of the token selection module in i-th FS-Block. Given the multi-scale features $\{F_i\}_{i=1}^4$, it first concatenate them along token dimension, after a flatten operation,

$$F = \text{Concat}\Big(\phi(F_1), \phi(F_2), \phi(F_3), \phi(F_4)\Big) \in \mathbb{R}^{L\times D}, \tag{1}$$

where $L = \sum_{i=1}^{4} \frac{HW}{2^{2i+2}}$, and ϕ denotes the flatten operation upon the spatial dimension. Then, we employ a MLP module to dynamically predict the interest scores $P_i \in [0,1]^L$ of all tokens for scale i,

$$P_i = \text{Softmax}\big(\text{MLP}(F)\big), \tag{2}$$

where $P_i^j, j \in [0, 1, ..., L-1]$ represents the interest score of the j-th token $F_j \in \mathbb{R}^D$ to the tokens F_i at scale i. Next, given a pre-defined selection ratio $\rho \in (0, 1]$, we select the ρL tokens with top-ρ interest scores P_i from the whole token set F, resulting the token subset of interest $\widetilde{F}_i \in \mathbb{R}^{\rho L\times D}$ for scale i. The process can be formulated as follows:

$$\theta_i = \Theta(P_i), \tag{3}$$

$$Q_i^{\theta_i^j} = \begin{cases} 1, & 0 \le j < \rho L \\ 0, & \rho L \le j < L \end{cases} \tag{4}$$

$$\widetilde{F}_i = F[Q_i = 1], \tag{5}$$

where the Argsort operation Θ (in descending order) is first employed on P_i to obtain the sorted indexes $\theta_i \in [0, L-1]$. θ_i is further used to generate a binary mask Q_i, which indicate which tokens are selected. $Q_i^j = 1$ means the j-th token

is selected into the token subset of interest for scale i, otherwise not selected. $[\delta]$ in Eq. (5) means fetching operation by the given condition δ.

However, such hard selection process is non-differentiable. To overcome this problem, we apply the gumbel-softmax technique [15] to generate the binary mask Q_i from the soft probability distribution P_i,

$$Q_i = \text{Gumbel-softmax}(P_i) \in \{0, 1\}^L. \tag{6}$$

The gumbel-softmax is differentiable, thus enables the end-to-end optimization during training.

Full-scale Selective Fusion. Inspired by the success of Transformer architecture [11], we utilize a transformer layer (including an attention module and feed-forword network (FFN)) for the context modeling. Specifically, we extend the multi-head self attention into multi-head cross attention (MCA) to enable the different sources of query, key and value, which is required for our full-scale selective fusion. MCA is responsible for token-wise interaction, whose forward pass can be formulated as follows:

$$X^{\text{MCA}} = \mathbf{MCA}(X_q, X_k, X_v) = \frac{X_q X_k^T}{\sqrt{D}} \cdot X_v, \tag{7}$$

where X_q, X_k and X_v denote query, key and value embedding respectively. D is the channel number of X_q, X_k and X_v. FFN is in charge of channel-wise projection. We use the same structure of FFN as [11], which contains a layer normalization [2] and a multi-layer perceptron (MLP) module.

Figure 3 illustrates the detailed structure of the i-th FS-Block for inference and training, respectively. During inference, MCA takes the i-th scale tokens F_i as query, and the selected token subset of interest $\widetilde{F}_i$ generated by the token selection module as key and value, *i.e.*, $F_i^{\text{MCA}} = \mathbf{MCA}(F_i, \widetilde{F}_i, \widetilde{F}_i) \in \mathbb{R}^{H_i W_i \times L}$. Thus, each token at scale i has the ability to interact with all the tokens within its corresponding interested token subset $\widetilde{F}_i$, ranging from all the scales.

However, during training, the token subset of interest is sampled by gumbel-softmax, resulting in a non-uniform number of tokens for samples within a batch, which prevents the parallel computing. To overcome this issue, we introduce a masked attention mechanism, named masked multi-head cross attention (MMCA), to not only parallelize the computation but also cut down the interactions between each query token and its uninterested tokens. The MMCA takes the i-th scale tokens F_i, full-scale tokens F and selection mask Q_i as inputs, and output F_i^{MMCA} with the same size as F_i.

$$F_i^{\text{MMCA}} = \mathbf{MMCA}(F_i, F, Q_i) \in \mathbb{R}^{H_i W_i \times D}. \tag{8}$$

Specifically, it first compute the non-selective full-scale fusion via the multi-head cross attention between F_i and F,

$$A = \frac{F_i F^T}{\sqrt{D}} \in \mathbb{R}^{H_i W_i \times L}. \tag{9}$$

Then, we generate the binary selection mask $M_i \in \{0,1\}^{H_iW_i \times L}$ for all tokens at scale i by repeating $Q_i \in \{0,1\}^L$ H_iW_i times, since all the tokens belonging to i-th scale share the same token subset of interest. Note that the mask M_i is shared by all heads. Next, the effects of uninterested tokens in the attention map are filtered out by the following masking mechanism,

$$\widetilde{A}_{ij} = \frac{\exp(A_{ij})M_{ij}}{\sum_{k=1}^{L} M_{ik}}. \tag{10}$$

Note that the mask M_i is shared by all heads. Eq. (10) does not change the size of attention map, thus $\widetilde{A}$ has the same size with A. Finally, such masked attention map $\widetilde{A}$ is multiplied with the whole token set F to generate the final tokens,

$$F_i^{\text{MMCA}} = \widetilde{A}F \in \mathbb{R}^{H_iW_i \times D}. \tag{11}$$

Token Reduction for efficiency. According to Eq. (9), the computational complexity of our MMCA is $O(H_iW_iL)$, which causes heavy computation burden when token number is large (*i.e.,* high-resolution feature maps). In order to improve its efficiency, we further design a meta-learning based projection mechanism to squeeze the query token sequence to a shorter one. Specifically, we perform a projection matrix $R_i \in \mathbb{R}^{N_i \times N_i^{'}}$ on query tokens $F_i \in \mathbb{R}^{N_i \times D}$ to compress the sequence length of query embedding,

$$\widehat{F}_i = R_i^T F_i \in \mathbb{R}^{N_i^{'} \times D}, \tag{12}$$

where $N_i = H_iW_i$ is the original sequence length of F_i. $N_i^{'} = \frac{N_i}{r}$, where r is the reduction ratio. Considering the projection matrix requires the ability to perceive the image content, we dynamically generate R_i through a MLP layer Φ conditioned on the query tokens F_i,

$$R_i = \Phi(F_i). \tag{13}$$

Then, the squeezed query $\widehat{F}_i$ and full-scale tokens F are passed through MMCA as Eq. (8).

$$\widehat{F}_i^{\text{MMCA}} = \mathbf{MMCA}(\widehat{F}_i, F, Q_i) \in \mathbb{R}^{N_i^{'} \times D}. \tag{14}$$

Finally, we re-project the $\widehat{F}_i^{\text{MMCA}}$ back to the original sequence length N_i,

$$F_i^{\text{MMCA}} = R_i \widehat{F}_i^{\text{MMCA}} \in \mathbb{R}^{N_i \times D}. \tag{15}$$

3.3 Loss Function

We now describe the training objectives of our FSFormer. We adopt the widely-used cross-entropy loss for the final predicted probability of each pixel,

$$\mathcal{L}_{\text{ce}} = \sum_{n=1}^{N} \text{CrossEntropy}(y_n, \widehat{y}_n), \tag{16}$$

where y_n and $\widehat{y}_n$ denote the ground-truth one-hot label and predicted probability distribution of n-th pixel.

Similar to previous works [23, 34], we also apply a lightweight segmentation head (1×1 convolution) on the stage 3 output of backbone to project the channel dimension to class number. An auxiliary loss $\mathcal{L}_{\text{aux}}$ is employed on the output of such segmentation head. $\mathcal{L}_{\text{aux}}$ is also implemented by cross-entropy loss.

In addition, in order to constrain the ratio of the selected tokens of interest to a predefined value $\rho \in (0, 1]$, we utilize an MSE loss to regularize the predicted interest scores $\widehat{P}_i$ in Eq. (2),

$$\mathcal{L}_{\text{reg}} = \frac{1}{S}\sum_{i=1}^{S} \|\rho - \frac{1}{L}\sum_{j=1}^{L}(P_i^j)\|^2, \tag{17}$$

where i is the scale index, and S equals to 4 in our experiments.

Overall, the total loss function consists of three terms:

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \alpha\mathcal{L}_{\text{reg}} + \beta\mathcal{L}_{\text{aux}}, \tag{18}$$

where α and β are hyper-parameters. Following previous work [42, 41, 38], we set the weight β of auxiliary loss to 0.4. We ablate the α in the experiment section.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on four widely-used public benchmarks: *ADE20K* [45] is a very challenging benchmark including 150 categories, which is split into 20000 and 2000 images for training and validation. *Cityscapes* [10] carefully annotates 19 object categories of urban scene images. It contains 5K finely annotated images, split into 2975 and 500 for training and validation. *COCO-Stuff 10K*[3] is a large scene parsing benchmark, which has 9000 training images and 1000 testing images with 182 categories (80 objects and 91 stuffs). *PASCAL Context*[25] is an extension of the PASCAL VOC 2010 detection challenge. It contains 4998 and 5105 images for training and validation, respectively. Following previous works, we evaluate the most frequent 60 classes (59 categories with background).

Backbone. For fair comparisons with other methods, we employ the well-known ResNet-101 [13] and Swin Transformer [23] as backbone. All the backbones are pre-trained on ImageNet-1K [29].

Hyper-parameters. The channel D of features F_i is set to 256, the weight α of is set to 0.4, and the target ratio ρ is set to 0.6. The head number of MCA is 8.

Training. We follow the previous works [38, 43, 31, 33] to set up the training strategies for fair comparisons. The data augmentation consists of three steps: (i) random horizontal flip, (ii) random scale with the ratio between 0.5 and 2, (iii) random crop (480×480 for PASCAL Context, 512×512 for ADE20K

Table 1: Comparison with the state-of-the-art methods on ADE20K-*val*, Cityscapes *val* COCO-Stuff 10K-*test* and PASCAL Context-*val*. m.s.: multi-scale inference. "†" means that larger input resolution is used (640×640 for ADE20K and 1024×1024 for Cityscapes). "R101" is short for ResNet-101.

Method	Encoder	mIoU (m.s.)			
		ADE20K	Cityscapes	COCO-Stuff 10K	PASCAL
PSPNet [42]	R101	45.35	80.04	38.86	47.15
DeepLabV3+ [7]	R101	46.35	82.03	-	48.26
EncNet [41]	R101	44.65	76.97	-	-
ANN [46]	R101	45.24	81.30	-	52.80
OCRNet [39]	R101	-	81.87	39.50	54.80
DANet [12]	R101	45.02	82.02	39.70	52.60
CCNet [14]	R101	45.04	80.66	-	-
GFFNet [19]	R101	45.33	81.80	39.20	54.20
FPT [40]	R101	45.90	82.20	-	-
RecoNet [8]	R101	45.54	-	41.50	54.80
MaskFormer [9]	R101	47.20	81.40	39.80	-
FSFormer (ours)	R101	46.56	82.13	41.73	55.23
SETR [44]	ViT-L	50.28	82.15	45.80	55.83
MCIBI [16]	ViT-L	50.80	-	44.89	-
Segmenter [31]†	ViT-L	53.60	81.30	-	59.00
SegFormer [35]†	MiT-B5	51.80	84.00	46.70	-
UperNet [34]	Swin-L	51.17	-	47.71	57.29
UperNet [34]†	Swin-L	53.50	-	-	-
FSFormer (ours)	Swin-L	53.33	83.64	49.80	58.91
FSFormer (ours)†	Swin-L	54.43	84.46	-	-

and COCO-Stuff 10K, and 768×768 for Cityscapes). We use AdamW [24] as the optimizer with 0.01 weight decay. The initial learning rate is 0.00006 for ADE20K and Cityscapes, and 0.00002 on PASCAL Context and COCO-Stuff 10K. The training process contains 160k iterations for ADE20K, 60k iterations for COCO-Stuff 10k, and 80k iterations for Cityscapes and PASCAL Context. The batch size is set to 8 for Cityscapes, and 16 for other datasets. We initialize the encoder by the ImageNet-1K [29] pre-trained parameters, and other parts randomly. Synchronized BN [26] is used to synchronize the mean and standard-deviation of BN [30] across multiple GPUs. All the experiments are implemented with PyTorch [1] and conducted on 8 NVIDIA V100 GPUs.

Evaluation. The performance is measured by the widely-used mean intersection of union (mIoU) for all experiments. For the multi-scale inference, we follow previous works [23, 43] to average the the predictions of our model at multiple scales [0.5, 0.75, 1.0, 1.25, 1.5, 1.75].

4.2 Comparisons with the state-of-the-arts

ADE20K val. Table 1 reports the comparison with the state-of-the-art methods on the ADE20K validation set. Equipped with Swin-L as backbone, our FSFormer is +2.16% mIoU higher (53.33% vs. 51.17%) than UperNet. Recent methods [31, 23] show that using a larger resolution (640×640) can bring more improvements. When a larger resolution (640×640) is adopted, our FSFormer outperforms UperNet by +0.93% (54.43% vs. 53.50%) under the same Swin-L backbone. In addition, our FSFormer(Swin-L) is +2.63% mIoU higher than SegFormer(MiT-B5) (54.43% vs. 51.80%). Although Segmenter [31] uses the stronger ViT-L[11] backbone than Swin-L, our FSFormer also show a +0.73% mIoU advantage than Segmenter. These results demonstrate that the effectiveness of our method.

Cityscapes val. Table 1 shows the comparative results on Cityscapes validation set. Our FSFormer is +1.49% superior than SETR [43] (83.64% vs. 82.15%). According to [35], a higher input resolution of 1024×1024 can bring further performance gain. Thus, we also train our model under such resolution. It can be seen that our FSFormer(Swin-L) outperforms SegFormer(MiT-B5) and Segmenter(ViT-L) by 0.46% and 3.16% mIoU. When using the widely-used ResNet-101 as backbone, our FSFormer achieves 82.13% mIoU, which is +0.26% and +0.73% higher than the well-known OCRNet [38] and the promising MaskFormer [9], respectively.

COCO-Stuff 10K test. As shown in Table 1, our FSFormer achieves 49.80% mIoU, outperforming UperNet by 2.09% under Swin-L backbone. Compared with MCIBI with a stronger ViT-L backbone, our FSFormer presents a +4.91% mIoU superiority. Besides, equipped with ResNet-101 as backbone, our FSFormer achieves 41.73% mIoU, which is +0.23% higher than the previous best RecoNet.

PASCAL Context val. Table 1 compares our method with the state-of-the-arts on PASCAL Context validation set. our FSFormer is +0.43% mIoU higher than RecoNet with ResNet-101 as backbone (55.23% vs. 54.80%). With Swin-L as backbone, our FSFormer achieves 58.91% mIoU, outperforming UperNet by +1.62%. Compared with the methods with using stronger ViT-L as backbone, our FSFormer(Swin-L) is +3.08% mIoU higher than SETR and achieves comparable performance with Segmenter (58.91% vs. 59.00%).

4.3 Ablation study

In this sub-section, we study the effect of key designs and hyper-parameters of our approach. All the ablation studies are conducted under Swin-T [23] backbone on PASCAL Context dataset.

Effect of key designs. We ablate the effect of two key designs (full-scale fusion and token selection) in Table 2. The baseline model denotes the single-scale fusion without token selection, *i.e.,* each token at one scale can only interact

Table 2: Ablation on the effect of full-scale fusion and token selection. s.s.: single-scale inference.

Baseline	Full-scale Fusion	Token Selection	FLOPs	Params	mIoU (s.s.)
✓			54.0G	33M	46.75
✓	✓		75.5G	38M	48.70
✓	✓	✓	73.8G	39M	49.33

with tokens within the same scale, which achieves only 46.75% mIoU. The full-scale fusion brings an obvious improvement (+1.95%), reaching 48.70% mIoU. Benefited from the token selection operation, the FLOPs is reduced by 1.7G and the performance further increase by +0.63%, achieving 49.33% mIoU.

Compare with different token selection manners. To verify the effectiveness of our adaptive token selection strategy, we compare it with two simple and intuitive manners: (i) random selection, (ii) uniform selection. As shown in Table 4 (a), these two fixed selection manners lead to about 1.5% performance decrease, and approximately 2% lower than our adaptive token selection manner. This shows that the token subset of interest need to be adaptively selected according to the image content.

Token selection ratio. The token selection ratio ρ represents the proportion of tokens of interest selected from tokens at all scales, that is an indicator of the size of the token subset of interest. A larger token selection ratio means each token can interact with more tokens, while also cause more computation burden and may introduce noise information. Where, $\rho = 1.0$ means that no selection is performed, that is, each token can interact with all the tokens at any scale. Figure 4 (d) shows the performance when the token selection ratio ρ varies within [0.1, 1.0]. It can be seen that the performance changes with token selection ratio in a unimodal pattern within a range of about 1%. Specifically, the mIoU increases from 48.40% over $\rho = 0.1 \sim 0.6$, peaking at 49.33%, and then falls back to 48.70% at $\rho = 1.0$. Note that the token selection ratio is not the larger the better, which may attribute to the noisy information aggregation caused by excessive interaction range. The best mIoU is achieved at a token selection ratio of 0.6, thus we set ρ to 0.6 by default. The results demonstrates the necessity of selecting a token subset of interest during feature fusion.

Weight of regularization loss for token selection ratio. As mentioned in Section 3.3, we apply a regularization loss to constrain the ratio of selected tokens of interest to a predefined value ρ. Figure 4 (c) shows the effect of different weights (ranged between 0 and 1) for this regularization loss. It can be seen that $\alpha = 0.4$ outperforms its counterparts, achieving the best performance with 49.33% mIoU. Thus, we set $\alpha = 0.4$ by default.

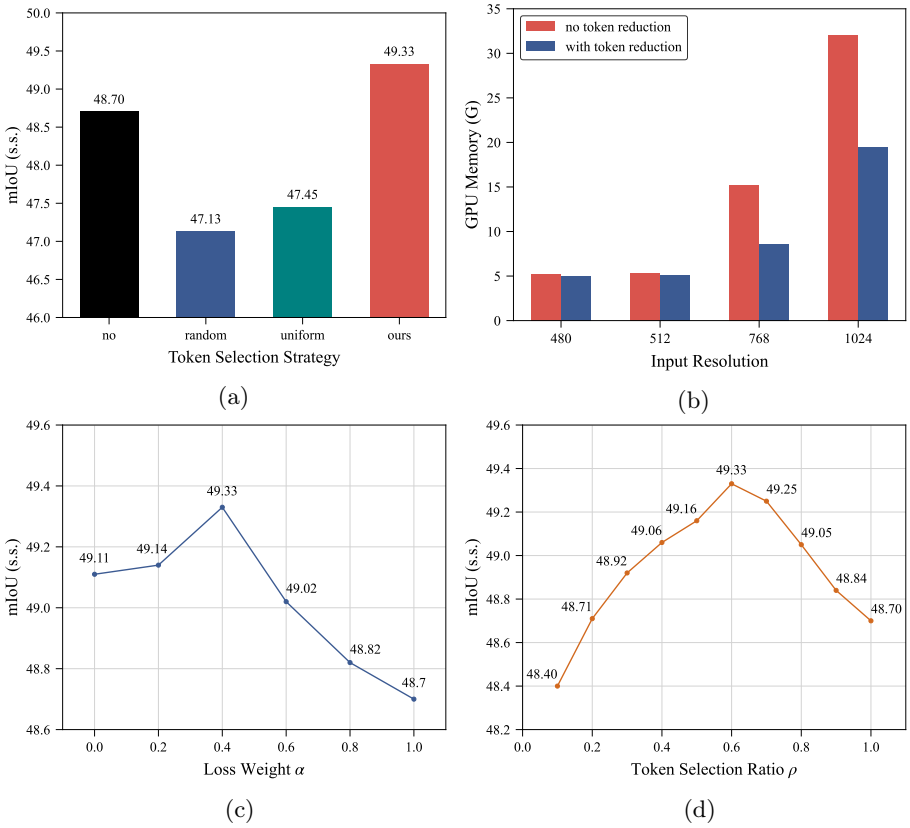

Fig. 4: Effect of (a) different token selection strategies, (b) token reduction, (c) different weights of regularization loss for token selection ratio, and (d) different token selection ratio. "s.s." denotes single-scale inference.

Effect of token reduction. Here, we study the effect of our token reduction strategy on both performance and GPU memory. As shown in Figure 4(b), our token reduction can effectively relieve the GPU memory burden. Note that the larger the input resolution, the more obviously the memory burden will be reduced. Furthermore, such token reduction strategy can bring a slight +0.27% mIoU gain (49.33% vs. 49.06%).

4.4 Further Analysis

Comparisons with other multi-scale fusion decoders. To further verify the effectiveness of our full-scale selective fusion, we compare our FSFormer with other decoders with different multi-scale fusion strategies in Table 3. The results demonstrate the superiority of our method. Compared with other progressive local fusion methods (SETR, Semantic FPN and UperNet), our FSFormer out-

Table 3: Comparisons with other decoders with multi-scale feature fusion on (a) performance under different backbones on PASCAL Context *val* set, and (b) computational costs. We report the FLOPs and Params of decoders, relative to the backbone. The input resolution is set to 480 ×480. "s.s." denotes single-scale inference.

(a)

Encoder	SETR[43]	Semantic FPN[18]	UperNet[34]	GFFNet[19]	FTN[33]	**FSFormer**
Swin-S	50.48	50.49	51.67	51.76	52.14	**52.58**
Swin-B	51.51	51.48	52.52	52.58	52.81	**53.12**
Swin-L	56.62	56.78	56.87	56.90	57.29	**57.63**

(b)

	SETR[43]	Semantic FPN[18]	UperNet[34]	GFFNet[19]	FTN[33]	**FSFormer**
FLOPs	13G	112G	187G	85G	39G	**52G**
Params	3M	54M	37M	17M	25M	**12M**

performs the best one (*i.e.,* UperNet) among them by +0.91% and +0.76% under Swin-T and Swin-L. Compared with the full-scale local fusion decoder, GFFNet, our FSFormer has +0.82%, +0.54% and +0.73% gains in mIoU with Swin-T, Swin-B and Swin-L as backbone respectively. Compared with FTN, a transformer-based progressive global fusion decoder, our FSFormer is +0.44%, +0.31% and +0.34% higher than FTN under different Swin Transformer backbones.

5 Conclusion

In this paper, we first rethink the multi-scale feature fusion from two perspectives (scale-level and spatial-level), and then propose a full-scale selective fusion strategy for semantic segmentation. Based on the proposed fusion mechanism, we design a Full-scale Selective Transformer (FSFormer) for semantic segmentation. Specifically, our FSFormer adaptively select partial tokens from all tokens at all the scales to construct a token subset of interest for each scale. Therefore, each token only interact with the tokens within its corresponding token subset of interest. The proposed full-scale selective fusion strategy can not only filter out the noisy information propagation but also reduce the computational costs to some extent. Extensive experiments on PASCAL Context, ADE20K, COCO-Stuff 10K, and Cityscapes have shown that our FSFormer can outperform the state-of-the-art methods in semantic image segmentation, demonstrating that our FSFormer can achieve better results than previous multi-scale feature fusion methods.

References

1. Adam Paszke, Sam Gross, Soumith Chintala, G Chanan, E Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L Antiga, A Lerer, et.al.: Automatic differentiation in pytorch. In: Advances in neural information processing systems Workshop (2017)
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
3. 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 (2018)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(4), 834–848 (2018). https://doi.org/10.1109/TPAMI.2017.2699184
6. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587 (2017)
7. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
8. Chen, W., Zhu, X., Sun, R., He, J., Li, R., Shen, X., Yu, B.: Tensor low-rank reconstruction for semantic segmentation. In: European Conference on Computer Vision. pp. 52–69. Springer (2020)
9. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. arXiv preprint arXiv:2107.06278 (2021)
10. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. ICLR (2021)
12. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Huang, Z., Wang, X., Huang, L., Huang, C., Wei, Y., Liu, W.: Ccnet: Criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 603–612 (2019)
15. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016)
16. Jin, Z., Gong, T., Yu, D., Chu, Q., Wang, J., Wang, C., Shao, J.: Mining contextual information beyond image for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7231–7241 (2021)

17. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6399–6408 (2019)
18. Kirillov, A., Girshick, R., He, K., Dollár, P.: Panoptic feature pyramid networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6399–6408 (2019)
19. Li, X., Zhao, H., Han, L., Tong, Y., Tan, S., Yang, K.: Gated fully fusion for semantic segmentation. In: AAAI (2020)
20. Lin, D., Ji, Y., Lischinski, D., Cohen-Or, D., Huang, H.: Multi-scale context intertwining for semantic segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 603–619 (2018)
21. Lin, D., Shen, D., Shen, S., Ji, Y., Lischinski, D., Cohen-Or, D., Huang, H.: Zigzagnet: Fusing top-down and bottom-up context for object segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7490–7499 (2019)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
23. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
25. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: 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 (2014)
26. Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., Yu, G., Sun, J.: Megdet: A large mini-batch object detector. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6181–6189 (2018)
27. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. arXiv preprint arXiv:2103.13413 (2021)
28. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
30. Sergey Ioffe, Christian Szegedy: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37. p. 448–456 (2015)
31. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7262–7272 (October 2021)
32. Wu, S., Wu, T., Lin, F., Tian, S., Guo, G.: Fully transformer networks for semantic image segmentation. arXiv preprint arXiv:2106.04108 (2021)
33. Wu, S., Wu, T., Lin, F., Tian, S., Guo, G.: Fully transformer networks for semantic image segmentation. arXiv preprint arXiv:2106.04108 (2021)

34. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 418–434 (2018)
35. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. arXiv preprint arXiv:2105.15203 (2021)
36. Xie, E., Wang, W., Wang, W., Sun, P., Xu, H., Liang, D., Luo, P.: Segmenting transparent object in the wild with transformer. arXiv preprint arXiv:2101.08461 (2021)
37. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)
38. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. arXiv preprint arXiv:1909.11065 (2019)
39. Yuan, Y., Chen, X., Wang, J.: Object-contextual representations for semantic segmentation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. pp. 173–190. Springer (2020)
40. Zhang, D., Zhang, H., Tang, J., Wang, M., Hua, X., Sun, Q.: Feature pyramid transformer. In: European Conference on Computer Vision. pp. 323–339. Springer (2020)
41. Zhang, H., Dana, K., Shi, J., Zhang, Z., Wang, X., Tyagi, A., Agrawal, A.: Context encoding for semantic segmentation. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7151–7160 (2018)
42. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)
43. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. arXiv preprint arXiv:2012.15840 (2020)
44. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890 (2021)
45. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic understanding of scenes through the ade20k dataset. International Journal of Computer Vision **127**(3), 302–321 (2019)
46. Zhu, Z., Xu, M., Bai, S., Huang, T., Bai, X.: Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 593–602 (2019)