Dynamic Focus-aware Positional Queries for Semantic Segmentation

Haoyu He[1] Jianfei Cai[1] Zizheng Pan[1] Jing Liu[1]
Jing Zhang[2] Dacheng Tao[2] Bohan Zhuang[1†]

[1] ZIP Lab, Monash University [2] The University of Sydney

Abstract

The DETR-like segmentors have underpinned the most recent breakthroughs in semantic segmentation, which end-to-end train a set of queries representing the class prototypes or target segments. Recently, masked attention [8] is proposed to restrict each query to only attend to the foreground regions predicted by the preceding decoder block for easier optimization. Although promising, it relies on the learnable parameterized positional queries which tend to encode the dataset statistics, leading to inaccurate localization for distinct individual queries. In this paper, we propose a simple yet effective query design for semantic segmentation termed Dynamic Focus-aware Positional Queries (DFPQ), which dynamically generates positional queries conditioned on the cross-attention scores from the preceding decoder block and the positional encodings for the corresponding image features, simultaneously. Therefore, our DFPQ preserves rich localization information for the target segments and provides accurate and fine-grained positional priors. In addition, we propose to efficiently deal with high-resolution cross-attention by only aggregating the contextual tokens based on the low-resolution cross-attention scores to perform local relation aggregation. Extensive experiments on ADE20K and Cityscapes show that with the two modifications on Mask2former, our framework achieves SOTA performance and outperforms Mask2former by clear margins of 1.1%, 1.9%, and 1.1% single-scale mIoU with ResNet-50, Swin-T, and Swin-B backbones on the ADE20K validation set, respectively. Source code is available at https://github.com/ziplab/FASeg.

1. Introduction

Semantic segmentation aims at assigning each pixel in an image with a semantic class label. As the end-to-end Detection Transfomer (DETR) [3,42,49,58] is revolutionizing the paradigm of the object detection task, recent segmentors [2,8,9,54] follow DETR to learn a set of queries representing the class prototypes or target segments and achieve state-of-the-art performance on semantic segmentation.

In DETR-like frameworks, providing the queries with meaningful positional priors and encourage each query to concentrate on specific regions is essential to learn representative queries [28, 43, 49, 58]. In this spirit, masked attention [8] is proposed, which restricts each query to only attend to a foreground region predicted by the previous decoder block with binary masks. Although promising, the positional priors in masked attention may be inaccurate and deteriorate performance for two reasons. First, each query comprises a content query that contains semantic information and a positional query that provides positional information for the likely locations of the target segments. However, masked attention still relies on positional queries that are randomly initialized learnable parameters [3, 40] (Figure 1 (a)), which tend to encode the average statistics across the dataset and cannot reflect the segments with large location variances. Second, since each query only attends to the predicted foreground regions, inaccurate predictions lead to error accumulation across the decoder blocks, especially during an early training stage.

To this end, recent detectors propose to dynamically encode the anchor points into the positional queries to guide queries concentrating around the anchor positions [28, 30, 43] (Figure 1 (b)). The anchor-based query design mitigates the mentioned issues as the positional queries are dynamically generated for each target object, thus providing more accurate positional priors. In addition, it avoids restricting the queries to only attend to the foreground regions with binary masks to mitigate the error accumulation issue. However, the anchor-based queries cannot describe the fine-grained positional priors for semantic segmentation, which has details, edges, and boundaries [5, 6].

Motivated by the observations that attention scores imply the salient regions for token pruning [24, 26], self-supervised learning [4], and semantic segmentation [34,56], in this paper, we propose a simple yet effective query design for semantic segmentation, dubbed Dynamic Focus-aware Positional Queries (DFPQ), which dynamically generates

†Corresponding author. E-mail: bohan.zhuang@gmail.com

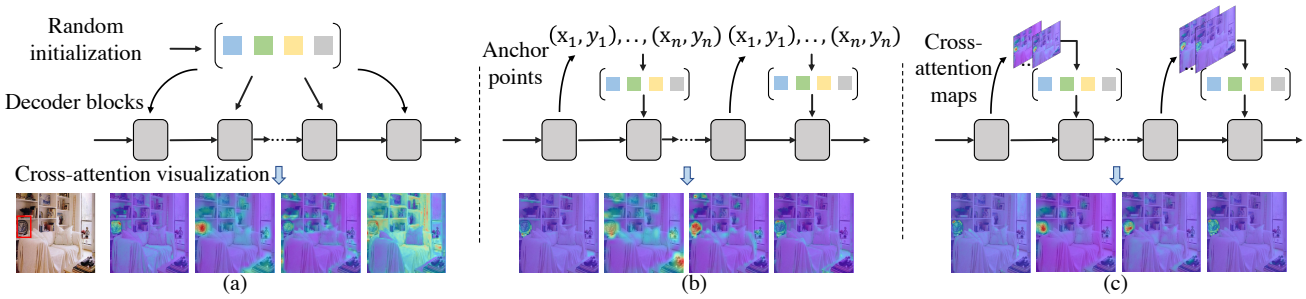


Figure 1. (a) The original randomly initialized positional queries [3] as learnable network parameters, where the positional queries are shared among the Transformer decoder blocks and tend to encode dataset statistics modelling the likely positions for the semantic regions, which leads to inaccurate localization. (b) The anchor-based positional queries [43] are conditional on the bounding box coordinates to give each query positional priors around the anchor. However, the anchor points cannot describe semantic regions, thus still sub-optimal for semantic segmentation. (c) Our dynamic focus-aware queries for semantic segmentation are dynamically generated from the cross-attention scores of the preceding decoder block to provide accurate and fine-grained positional priors, facilitating locating and refining the target segments progressively.

the positional queries conditioned on the cross-attention scores of the preceding decoder block and the positional encodings for the corresponding image features, simultaneously (Figure 1 (c)). In this way, our DFPQ preserves the localization information of the target segments, thereby providing accurate and fine-grained positional priors and facilitating progressively locating and refining the target segments. When implementing the positional encodings with more powerful ones like [10], our DFPQ is further empowered with higher capacity to encode the neighbourhood information for the target segments. Compared to the anchor-based positional queries [28, 43], our DFPQ can cover fine-grained locations for the segmentation details, edges, and boundaries which include rich segmentation cues.

In addition, we propose an efficient method named High-Resolution Cross-Attention (HRCA) to mine details for segmenting small regions from the high-resolution feature maps ($1/4 \times 1/4$ of the original image size). Considering performing cross-attention on high-resolution feature maps requires a formidable amount of memory footprints and computational complexity, *e.g.*, 11G extra floating-point operations with an input resolution of 512×512, we propose to encode token affinity only on the informative areas of high-resolution feature maps that are indicated important in the low-resolution counterparts. In this way, fine-grained details are learned efficiently with affordable memory and computations.

Our main contributions can be summarized as follows:

- We make the pioneering attempt to present a simple yet effective query formulation for semantic segmentation, which provides accurate and fine-grained positional priors to localize the target segments, and mitigates the error accumulation problem while being lightweight with little extra computation.
- We propose an efficient high-resolution cross-attention layer to enrich the segmentation details, which discards the semantically unimportant regions for any target segments in the high-resolution feature maps with affordable memory footprint and computational cost.
- Extensive experiments on ADE20K and Cityscapes datasets demonstrate that simply incorporating our DFPQ and HRCA into Mask2former [8] achieves significant performance gain and outperforms the SOTA methods. For instance, our FASeg outperforms SOTA methods by 1.1%, 1.3%, and 0.9% single-scale mIoU on the ADE20K [57] validation set with ResNet-50, Swin-T, and Swin-B backbones, respectively.

2. Related Work

Semantic segmentation with Transformers. The recent segmentors with Transformers [25, 36, 50, 54] have pushed the horizon for semantic segmentation. In general, these segmentors consist of three modules: a backbone, a neck, and a head. Correspondingly, the recent advances can be roughly split into three orthogonal categories. The first category [14, 46, 48, 50, 52, 53] aims at learning more representative features by improving the backbone, mostly by improving the self-attention mechanism in Transformers. For example, focal self-attention [48] combines both fine-grained and coarse-grained features in a backbone self-attention layer. To provide better multi-scale features with neck, the second category [22, 23, 34, 34, 45] improves the feature pyramid network (FPN) [27] or pyramid scene parsing (PSP) [55] structure. For instance, SegFormer [45] simplifies FPN under the Transformer backbone to achieve a better accuracy-efficiency trade-off, and SegDeformer [34] adds external memory tokens to preserve the global information. The third category implements the head with Transformer and conduct set prediction following the DETR-like end-to-end framework [3]. In DETR-like framework, target segments are represented by a set of queries. Considering the importance of providing positional priors for

the queries [28, 30, 43], masked attention [8] is proposed to restrict the cross-attention only to the local features. Our work also aims at providing better positional priors. In contrast to [8], we follow the recent detectors [28, 43] to provide accurate positional priors with dynamic positional queries rather than the learnable parameterized positional queries [3]. Differently, our DFPQ provides fine-grained positional priors that can cover the locations for fine segmentation details, edges, and boundaries. Very recent work [25] proposes a versatile multi-task head structure to share the mutual information among the segmentation and detection tasks, which however, is not directly comparable to our work.

Positional encodings for Transformers. Both self-attention and cross-attention for Transformers are permutation-equivalent. Therefore, Positional Encodings (PE) play an essential role in introducing the order of the sequence. In general, the positional encodings include: absolute PE that is generated with sinusoidal functions [35, 40] or being entirely learnable parameters [19, 29]; relative PE that encodes distances between the input tokens [12, 32]; and conditional PE, which is dynamically generated, *e.g.*, PEG [10] generates positional encodings with depth-wise convolution conditioned on the local neighbourhood information. In the same spirit, our DFPQ is also dynamically generated by exploring the idea of conditional encoding [38, 47], thus delivering higher segmentation accuracy. Differently, our DFPQ is tailored specifically for DETR-style semantic segmentation to learn positional priors for each target segment. In addition, since our DFPQ is conditional on the PE for the image features, implementing it with the more powerful PEs [10, 19] can further boost the representational capability of our DFPQ. We investigate the effect of different PE strategies in Section 4.2.

When solely pre-training Transformer backbones, the positional encodings are generally seen as a part of the feature embeddings and directly be combined with patch features after patchifying the image [15, 39]. Differently, in the cross-attention layers of DETR-like frameworks, both the image features and the object queries require additional positional information to provide positional priors for aggregating the query-specific context, which we refer the readers to Section 3.1 for details. Recent detectors [28, 43] encode anchor positions into positional queries. In contrast, we design a novel positional query formulation for semantic segmentation to reflect regions of interest instead of anchor points to preserve fine segmentation details.

3. Method

3.1. Preliminary: Cross-attention in DETR

Before introducing our DFPQ, we first revisit the cross-attention layers in DETR-like frameworks [3]. Cross-attention layer is a basic module that updates the object queries by aggregating the image context. Since the cross-attention layer is permutation-invariant, both queries and keys require positional information, which introduces the order and provides positional priors to encourage high attention scores for positionally important regions. Specifically, with N, D, H and W respectively denoting the number of queries, the hidden dimensions, the height, and the width of the image features, we have the image features $\boldsymbol{K}_c$ and their positional encodings $\boldsymbol{K}_p$ and get keys $\boldsymbol{K} = \boldsymbol{K}_c + \boldsymbol{K}_p$, where $\boldsymbol{K} \in \mathbb{R}^{HW \times D}$. We also have the object queries $\boldsymbol{Q} \in \mathbb{R}^{N \times D}$, where each query consists of a content query $\boldsymbol{Q}_c$ and a positional query $\boldsymbol{Q}_p$.

Then, the cross-attention operation can be formulated as

$$\text{Crs-Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D}}\right)\boldsymbol{V}, \quad (1)$$

where $\boldsymbol{V} \in \mathbb{R}^{HW \times D}$ is also the image features in the DETR-like frameworks [3, 9, 54] and we omit all the linear projections and bias terms for simplicity. From Eq. (1), we can interpret the cross-attention as aggregating image context based on the dot-product similarity between $\boldsymbol{Q}$ and $\boldsymbol{K}$. Since both the content parts and the positional parts for $\boldsymbol{Q}$ and $\boldsymbol{K}$ contribute to calculating the attention scores, similarities for both parts are considered. Therefore, content similarity contributes to mining the correlation between the object queries and the image features, while positional similarity provides positional priors for each target segment.

3.2. Dynamic Focus-aware Positional Queries

In this work, we aim to develop positional queries that provide effective positional priors under DETR-like frameworks for semantic segmentation. We argue that generating positional queries conditioned on cross-attention scores has three good properties. First, the cross-attention scores indicate the areas with rich context and may directly reflect the localization information for the target segments [4, 56]. Therefore, when stacking several decoder blocks with cross-attention layers in DETR-like frameworks, the localization information in the preceding block is helpful for progressively locating the target segments in the later blocks, especially when the blocks handle features at different scales [8, 23]. Second, cross-attention scores are dynamically generated. In contrast to the content-agnostic positional queries as learnable parameters in [3], which tend to encode statistics across the dataset and limit models' generalization capability, cross-attention scores are conditional on each target segment reflecting the specific contextual locations, thereby being more accurate. Finally, the cross-attention scores can cover fine-grained segmentation details, edges, and boundaries instead of encoding only a single center or anchor point alike [28, 43].

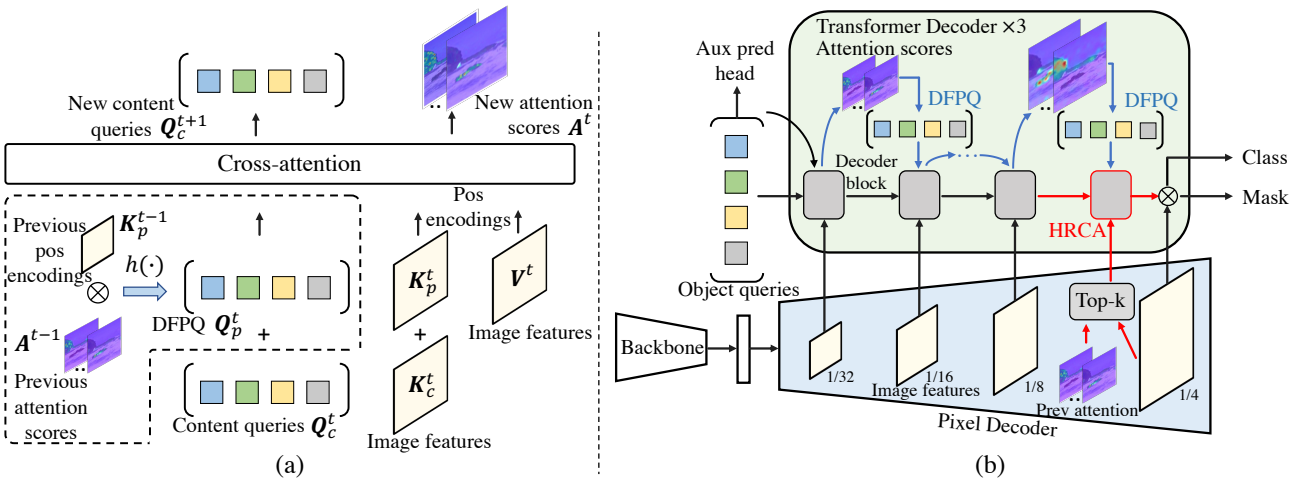


Figure 2. (a) Cross-attention with our dynamic focus-aware positional queries (DFPQ). "pos" is short for positional. We show generating DFPQ in the dashed box, where we multiply the positional encodings for the image features with the cross-attention scores of the preceding decoder block followed by a projection function h to get DFPQ. Here we omit the bias terms for simplicity. (b) The overall framework for our FASeg is built upon Mask2former [8], which employs a Backbone to encode images, a Pixel Decoder to fuse the features under different resolutions, and a Transformer Decoder to learn the representation for each target segment. We first apply our DFPQ in each decoder block to provide more accurate positional priors (marked with blue arrows). Then, we further propose to incorporate our high-resolution cross-attention (HRCA) layers to model the cross-attention between the queries and the high-resolution feature maps (marked with red arrows). Here "Top-k" selects the top-k pixels indicated by the cross-attention scores of the previous Transformer decoder block.

Therefore, we propose to generate the positional queries conditional on the cross-attention scores of the preceding decoder block and the positional encodings for the corresponding image features, simultaneously, as shown in Figure 2 (a). Specifically, since the positional encodings $\boldsymbol{K}_p$ for the image features preserve the positional information, we form our DFPQ by aggregating $\boldsymbol{K}_p$ as indicated by the cross-attention scores $\boldsymbol{A}$ in the cross-attention layer of the preceding decoder block, which can be formulated as

$$\boldsymbol{Q}_p^t = h(\boldsymbol{A}^{t-1}\boldsymbol{K}_p^{t-1} + \boldsymbol{B}), \tag{2}$$

where t is the index of the t-th Transformer decoder block, $\boldsymbol{A}^{t-1} \in \mathbb{R}^{N\times HW}$ is the cross-attention scores from the $(t-1)$-th Transformer decoder block, $\boldsymbol{B} \in \mathbb{R}^{N\times D}$ is learnable network parameters, and h is a two-layered MLP with ReLU non-linearity in between. Note that the bias term $\boldsymbol{B}$ is the original randomly initialized learnable positional queries, which we employ to stable the training in an early training stage. In this way, we dynamically generate DFPQ to provide positional priors for the target segments. It can also cover the fine-grained segmentation cues that are not restricted by anchor points.

Note that as our DFPQ directly aggregates the positional information, implementing DFPQ with different $\boldsymbol{K}_p$ leads to distinct behaviours. When implementing $\boldsymbol{K}_p$ with the absolute sinusoidal function, the resulting DFPQ reflects an anchor point alike [28,43] instead of the target areas. In this case, we implement $\boldsymbol{K}_p$ with conditional positional encodings [10] to further encode the neighbourhood information and preserve the implicit positional priors for localizing the target segments. We empirically investigate the effect of different positional encodings in Section 4.2.

3.3. Efficient High-resolution Cross-attention

As demonstrated by the prior arts [23, 38], high-resolution image features are important for segmenting small regions. However, modelling cross-attention between object queries and high-resolution image features requires an unbearable amount of memory footprints and computational cost. In this case, we propose an efficient High-Resolution Cross-Attention (HRCA) layer to mine details from high-resolution feature maps with affordable memory burden. Specifically, we first select the top-k pixels from the low-resolution image features with the highest cross-attention scores for all object queries. We then map these areas to the high-resolution feature map positions in a top-down manner and only perform cross-attention on these positions. Formally, we first get the low-resolution cross-attention scores $\boldsymbol{A}_l$, and then derive its high-resolution counterpart $\boldsymbol{A}_h = f(\boldsymbol{A}_l)$ with bilinear upsampling operation $f(\cdot)$. We next include the top-k pixels in $\boldsymbol{A}_h$ with the highest scores into set Ω, and the efficient HRCA can be formulated as

$$\text{HRCA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}, \Omega) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}'^T}{\sqrt{D}}\right)\boldsymbol{V}', \tag{3}$$

where $\boldsymbol{K}' = g(\boldsymbol{K}, \Omega)$, $\boldsymbol{V}' = g(\boldsymbol{V}, \Omega)$ and g is the indexing operation. In this way, we only perform cross-attention

on the informative areas for high-resolution feature maps, thereby saving considerable resource consumption.

Our HRCA is closely related to the previous sparse attention methods [1, 37, 41, 51] that only attend to a part of the entire sequence. Differently, our HRCA is specialized to the DETR-like frameworks, which determines the informative pixels based on their contribution to the target segments instead of the other pixels. One similar work to our HRCA is the RCDA module [43], which is a representative sparse cross-attention method that decouples cross-attention into a row-wise and column-wise attention to reduce the memory and computation cost. We include the comparison between our HRCA and RCDA [43] in Section 4.2.

3.4. Focus-aware Segmentation Framework

We first briefly introduce Mask2former [8], which consists of backbone, neck, and head as introduced in Section 2 with the neck and head named "Pixel Decoder" and "Transformer Decoder", respectively. In Mask2former, Pixel Decoder fuses the features at multiple scales following [27,58]. Transformer Decoder cascades three blocks which model the cross-attention between the object queries and the image features of $1/32 \times 1/32$, $1/16 \times 1/16$, and $1/8 \times 1/8$ of the original image resolution, respectively. The Transformer Decoder is repeated three times. We refer readers to [8, 9] for more details.

We develop our FASeg upon the Mask2former [8] framework by simply incorporating our DFPQ and HRCA. The overview of our FASeg is depicted in Figure 2 (b). We first provide more accurate and fine-grained positional priors for Mask2former with DFPQ (Section 3.2). We apply DFPQ in the cross-attention layers for each decoder block to provide good positional priors for aggregating the contextual image features to locate target segments. In this way, we progressively localize the target segments as we go deeper in the decoder blocks. Since there are no cross-attention scores before the first Transformer decoder block, we obtain the DFPQ for the first block by performing average pooling on the predicted foreground mask from the auxiliary prediction head as introduced in [8]. Next, we employ HRCA (Section 3.3) to enrich the segmentation details with affordable peak-time memory footprints and computational complexity. We add a fourth decoder block equipped with HRCA to model cross-attention on the high-resolution feature maps after the cascaded three decoder blocks that are already in Mask2former in a top-down manner. With the two simple modifications, our FASeg achieves solid performance gain over the original Mask2former (See Section 4.1).

4. Experiments

Implementation details. Unless otherwise specified, we adopt the same training settings and implementation details as in Mask2former [8]. For our efficient HRCA in Section 3.3, we select $|\Omega| = \lfloor HW/32 \rfloor$ from the low-resolution feature maps ($1/32 \times 1/32$ of the original image size). By default, we train our models with a batch size of 16 on 8 NVIDIA V100 GPUs. We adopt ResNet [21] and Swin Transformer [29] pre-trained backbones. For ResNet [21], we use the ResNet-50 (R50) variant. For Swin Transformer [29], we use the Swin-T, Swin-B, and Swin-L backbones where Swin-B and Swin-L are pre-trained on ImageNet-22k [13]. Unless specified, we adopt all training settings the same as the default settings of FASeg with R50 [21] backbone on ADE20K `val` [57] with 150 categories for ablation experiments. We conduct the main

Table 1. Performance comparisons with the state-of-the-art semantic segmentation methods on ADE20K `val` [57] with 150 categories. #P and #F indicate the number of parameters (M) and FLOPs (G). We report both single-scale (s.s.) and multi-scale (m.s.) inference results.

Method	Backbone	mIoU s.s. (%)	mIoU m.s. (%)	#P	#F
UperNet [44]	R50	42.1	-	67	238
DeepLab V3+ [6]	R50	44.0	44.9	44	177
SenFormer [2]	R50	44.7	45.2	144	179
Maskformer [9]	R50	44.5	46.7	41	53
PFD [31]	R50	45.6	48.3	74	61
Mask2former [8]	R50	47.2	49.2	44	71
FASeg (ours)	R50	**48.3**	**49.3**	51	72
UperNet [44]	Swin-T	44.4	46.1	60	236
SenFormer [2]	Swin-T	46.0	-	144	179
Maskformer [9]	Swin-T	46.7	48.8	42	55
PFD [31]	Swin-T	48.3	49.6	74	65
Mask2former [8]	Swin-T	47.7	49.6	47	74
FASeg (ours)	Swin-T	**49.6**	**51.3**	54	75
SenFormer [2]	Swin-B	51.8	-	204	242
Maskformer [9]	Swin-B	52.7	53.9	102	195
PFD [31]	Swin-B	54.1	55.3	123	206
Mask2former [8]	Swin-B	53.9	55.1	107	223
FASeg (ours)	Swin-B	**55.0**	**56.0**	113	225
UperNet [44]	Swin-L	52.1	53.5	234	647
SenFormer [2]	Swin-L	53.1	54.2	314	546
Maskformer [9]	Swin-L	54.1	55.6	212	375
PFD [31]	Swin-L	56.0	57.2	242	385
Mask2former [8]	Swin-L	56.1	57.3	215	403
FASeg (ours)	Swin-L	**56.3**	**57.7**	222	405

Table 2. Performance comparisons with the state-of-the-art semantic segmentation methods on Cityscapes `val` [11]. We report single-scale (s.s.) inference results. #P and #F indicate the number of parameters (M) and FLOPs (G).

Method	Backbone	mIoU s.s. (%)	#P	#F
Maskformer [9]	R50	78.5	41	405
Senformer [2]	R50	78.8	144	1,317
DeepLab V3+ [2]	R50	79.0	-	-
Mask2former [8]	R50	79.4	44	526
Maskformer [9]	R101	79.1	60	561
Mask2former [8]	R101	80.1	67	628
SenFormer [2]	R101	80.3	162	1,473
FASeg (ours)	R50	**80.5**	67	533

experiments and ablation studies with the same seeds as Mask2former to seek fair comparisons.

Datasets. We conduct our experiments on ADE20K [57] and Cityscapes [11]. ADE20K [57] is one of the most challenging large-scale datasets for semantic segmentation, which covers 150 fine-grained semantic concepts, where the training set and validation set contain 20,210 and 2,000 images, respectively. Cityscapes [11] is an urban street-view dataset with high-resolution images from 50 cities with 19 semantic classes, which consists of 2,975 images for the training set and 2,725 images for the validation set.

Evaluation metrics. We use single-scale (s.s.) and multi-scale (m.s.) mean Intersection over Union (mIoU) [17] as the evaluation metric. We also compare models in terms of their model size (number of parameters) and computational complexity with Floating-point Operations (FLOPs) to evaluate the efficiency of these models. For ablation studies on HRCA, we also show the training-time GPU memory consumption. For ADE20K [57] and Cityscapes [11], we calculate FLOPs with fixed 512×512 and 1024×2048 image size, respectively.

Compared methods. We compare our method with the SOTA semantic segmentation methods, including DeepLab V3+ [6], UperNet [44], Maskformer [9], SenFormer [2], PFD [31] and Mask2former [8]. Among them, SenFormer [2], PFD [31] and Mask2former [8] are the recent Transformer-based segmentors, where PFD learns a hierarchy of latent queries to enrich the multi-scale information and SenFormer ensembles the multi-scale predictions. We refer the readers to Section 2 for more details.

4.1. Main Results

We compare our FASeg with state-of-the-art semantic segmentation methods on ADE20K `val` [57] and Cityscapes `val` [11]. The results are reported in Tables 1 and 2. We observe that on ADE20K `val` (Table 1), with affordable number of extra parameters and FLOPs, our FASeg consistently outperforms the SOTA methods. Specifically, FASeg achieves 48.3%, 49.6%, 55.0%, and 56.3% mIoU for single-scale inference, outperforming the SOTA methods by 1.1%, 1.3%, 0.9%, and 0.2% on R50, Swin-T, Swin-B, and Swin-L backbones, respectively. The solid performance gain demonstrates the superiority of our FASeg framework. Our FASeg has more improvements with the smaller backbones (*e.g.*, R50, Swin-T, and Swin-B). We conjecture that localizing the contextural features with smaller backbones under inferior representational capability is challenging. Nevertheless, our DFPQ provides more accurate positional priors, which ease the localization difficulty and lead to better results. For the comparisons on Cityscapes `val` in Table 2, we observe that with the R50 backbone, our FASeg outperforms all the SOTA methods under desirable numbers of parameters and FLOPs. Surprisingly, FASeg even outperforms the SOTA methods employing the R101 backbone, which demonstrates the effectiveness of our FASeg. To further investigate the flexibility and potential of our main contribution DFPQ, we show more experiments on instance segmentation in the supplementary material.

We next show some qualitative results in Figure 3 and find that our FASeg provides more accurate predictions with finer details. The improved segmentation results again show the superiority of our DFPQ and HRCA. We include more qualitative results in the supplementary material.

Table 3. Effect of the positional encodings $\boldsymbol{K}_p$ for the image features on ADE20K `val` [57] with 150 categories.

$\boldsymbol{K}_p$	Mask2former [8] mIoU s.s. (%)	FASeg mIoU s.s. (%)
Sinusoidal [3]	47.2	46.9
Learnable absolute [19]	47.0	47.5
Conditional [10]	47.3	48.3

Table 4. Ablation study for FASeg on ADE20K `val` [57] and Cityscapes `val` [11]. #P and #F indicate the number of parameters (M) and FLOPs (G) evaluated on 512×512 images.

DFPQ	HRCA	ADE20K `val` mIoU s.s. (%)	Cityscapes `val` mIoU s.s. (%)	#P	#F
		47.2	79.4	44	71
✓		47.7	80.0	44	71
	✓	47.6	79.8	50	72
✓	✓	48.3	80.5	51	72

Table 5. Performance comparisons between DFPQ and other positional queries variants on ADE20K `val` [57] with 150 categories.

Method	mIoU s.s.(%)
Learnable positional queries	46.9
Pre-defined grid anchor positional queries	46.6
Dynamic anchor positional queries	47.0
Dynamic foreground positional queries	47.8
DFPQ	48.3

4.2. Ablation Study

Effect of $\boldsymbol{K}_p$. We investigate the effect of the positional encodings $\boldsymbol{K}_p$ for the image features on ADE20k `val` with R50 backbone. The results are reported in Table 3. We observe that different $\boldsymbol{K}_p$ have similar performance for Mask2former [8]. However, more powerful $\boldsymbol{K}_p$ leads to much higher performance for our FASeg. For instance, FASeg with conditional positional encodings [10] outperforms Mask2former counterpart and FASeg with sinusoidal positional encodings [3] by 1.0% and 1.4% mIoU, respectively. The reason is that compared to Mask2former, our FASeg additionally aggregates $\boldsymbol{K}_p$ to get DFPQ as explained in Section 3.2. Therefore, more powerful $\boldsymbol{K}_p$ leads to higher representational capability of DFPQ that boosts the performance. We also find that with sinusoidal positional encodings, FASeg has even lower performance than Mask2former as the DFPQ aggregated from sinusoidal positional encodings reflects a single anchor point which can-

not cover the fine-grained segmentation cues.

Effectiveness of DFPQ and HRCA. We investigate the effectiveness of our DFPQ and HRCA on ADE20k `val` and Cityscapes `val` with the ResNet-50 backbone. The results are reported in Table 4. We observe that both DFPQ and HRCA gain clear margins from the vanilla Mask2former [8]. To be specific, integrating DFPQ on Mask2former boosts the performance by 0.5% and 0.6% mIoU on ADE20k `val` and Cityscapes `val`, respectively, with barely any extra parameter and computational cost. It is indicated that DFPQ is lightweight and contributes largely on the performance gain. Employing HRCA on Mask2former leads to 0.4% mIoU gain on both datasets, which however, has 6M more parameters and 1G higher FLOPs. The additional parameters and FLOPs are brought by the extra decoder layers handling high-resolution image features. Finally, our FASeg with both DFPQ and HRCA improves 1.1% mIoU for both ADE20k `val` and Cityscapes `val`, demonstrating the superiority of our FASeg.

DFPQ vs. other positional queries. We investigate the effectiveness of our DFPQ and compare it with other learnable query variants on ADE20K `val` [57]. The results are presented in Table 5. Here we adopt all the other settings the same as our FASeg with the R50 backbone and only differ the positional queries for all the competitors. Specifically, we compare with four settings: 1) learnable parameterized positional queries that are randomly initialized [3]; 2) positional queries as the pre-defined grid anchor points akin to [43]; 3) positional queries dynamically generated from the center of the foreground masks predicted by the previous layer similar to [28]; 4) positional queries dynamically generated from the entire predicted foreground masks. We find that our DFPQ outperforms all the competitors by large margins. For example, our DFPQ achieves 1.7% and 1.3% higher mIoU than the pre-defined grid and dynamic anchor positional queries, respectively. It is suggested that our DFPQ better suits semantic segmentation than the other positional query variants. We also visualize cross-attention maps among the different positional queries in Figure 4. We observe that our DFPQ (Figure 4 (c)) helps generate more compact and consistent cross-attention maps focusing on the target segments than the learnable parameterized positional queries (Figure 4 (a)) and dynamic anchor positional queries (Figure 4 (b)).

HRCA vs. other efficient cross-attention methods. We investigate the effectiveness of our HRCA and compare it with other cross-attention methods on ADE20k `val`. The results are reported in Table 6. For a fair comparison, we only replace HRCA for the other efficient cross-attention methods on our FASeg with the R50 backbone. We compare with three baselines: 1) the vanilla cross-attention that models the entire high-resolution features; 2) our HRCA with randomly sampled top-k pixels to form set Ω that $|\Omega|$ is the same as HRCA; 3) RCDA [43] that the cross-attention is decoupled to a row-wise and column-wise attention as introduced in Section 3.3. We empirically find that compared with the vanilla cross-attention, our HRCA achieves 1.0% mIoU gain while exhibiting 1,108M lower training-time GPU memory and 11G lower FLOPs. Our HRCA also outperforms the two efficient cross-attention methods by large margins. For example, HRCA achieves 0.8% higher mIoU than RCDA with marginally increased training-time memory. The results demonstrate the superiority of our HRCA for efficiently identifying and utilizing contextual tokens in high-resolution features.

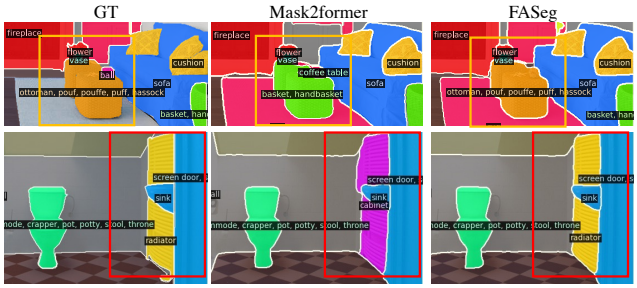


Figure 3. Qualitative results on the ADE20K `val` [57]. Compared to Mask2former [8], our FASeg predicts masks with finer details and yields more accurate predictions.

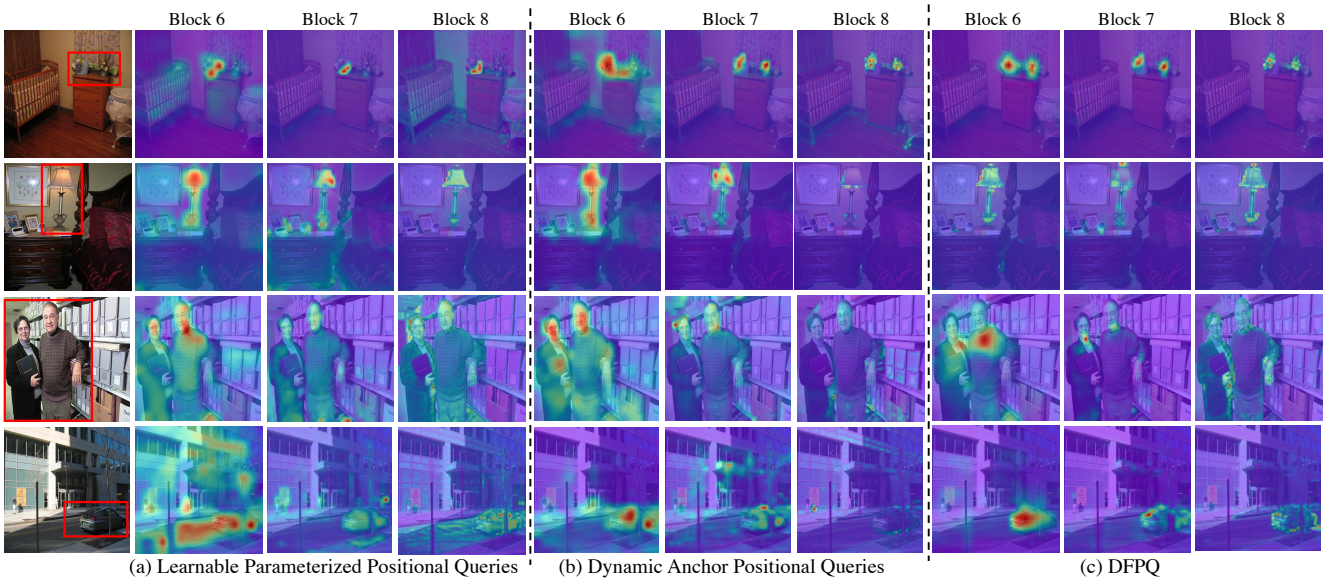
Table 6. Performance comparisons between our HRCA and other efficient cross-attention methods on ADE20K `val` [57] with 150 categories. #F denotes the number of FLOPs (G). The training memory footprint (M) and FLOPs are measured under 512×512 image resolutions with a batch size of 4 on a single GPU.

Method	mIoU s.s. (%)	Training Memory (M)	#F
Vanilla	47.3	7,451	83
Random Ω	46.7	6,343	72
RCDA	47.5	6,082	72
HRCA	48.3	6,343	72

Table 7. Effect of $|\Omega|$ in our efficient HRCA on ADE20K `val` [57] with 150 categories. #F indicates the number of FLOPs (G).

$\lvert\Omega\rvert$	mIoU s.s. (%)	Training memory (M)	#F
HW	47.3	7,451	83
$\lfloor HW/16 \rfloor$	47.7	6,381	72
$\lfloor HW/32 \rfloor$	48.3	6,343	72
$\lfloor HW/64 \rfloor$	48.0	6,317	71

Effect of $|\Omega|$ in HRCA. We then investigate how $|\Omega|$ affects the performance, memory consumption and computational complexity on FASeg with R50 backbone on ADE20k `val`. The results are reported in Table 7. $|\Omega|$ determines the number of contextual tokens used in attention as introduced in Section 3.3. Here we measure the memory consumption by the training-time memory with a batch size of 4 on a single GPU. In the vanilla cross-attention layers, cross-attention attends to the entire feature maps from the encoder, in which case $|\Omega| = HW$. We observe that our HRCA outperforms the vanilla cross-attention by a signifi-



(a) Learnable Parameterized Positional Queries (b) Dynamic Anchor Positional Queries (c) DFPQ

Figure 4. Visualizations of the cross-attention maps for learnable positional queries ([3, 8]), dynamic anchor positional queries (alike [28]) and our DFPQ. We show the visualizations for the normalized cross-attention maps in the last three decoder blocks and indicate the target segments in the red boxes. The cross-attention maps with the learnable positional queries and the dynamic anchor positional queries are often scattered without a clear focus and mix up different segments, while the cross-attention maps with DFPQ are more compact and consistent to reflect the target segments.

Table 8. Effect of applying HRCA to other high-resolution feature scales for FASeg with Swin-B Backbone on ADE20K `val` [57] with 150 categories.

$1/4 \times 1/4$	$1/8 \times 1/8$	mIoU s.s. (%)	Training Memory (M)
✓		55.0	20,418
✓	✓	54.9	19,898
	✓	54.8	17,817

cant margin. We conjecture that the sparse property [16, 18] has reduced the redundancy in high-resolution feature maps in our HRCA and leads to higher performance and efficiency. Since our HRCA achieves the highest performance when $|\Omega| = \lfloor HW/32 \rfloor$, we set $|\Omega| = \lfloor HW/32 \rfloor$ by default for all the other experiments.

Effect of applying HRCA to other high-resolution feature scales. By default, HRCA is applied only to the high-resolution $1/4 \times 1/4$ feature scale. We explore applying HRCA to $1/8 \times 1/8$ and both $1/4 \times 1/4$ and $1/8 \times 1/8$ feature scales for FASeg with Swin-B backbone on ADE20k `val`. We measure the training-time memory consumption with a batch size of 4 on a single GPU and report the results in Table 8. We find that the performance only fluctuates within 0.2% mIoU. In particular, modeling the cross-attention only on the $1/8 \times 1/8$ feature scale with HRCA saves more than 2,000M training-time memory, suggesting the potential for extending HRCA to more high-resolution features to alleviate the memory burden.

5. Conclusion

In this paper, we have explored providing positional priors with positional queries for the DETR-style semantic segmentation. Specifically, we have proposed to dynamically generate the positional queries conditioned on the cross-attention scores of the preceding decoder block and the positional encodings for the corresponding image features, simultaneously. We have found that our novel query design delivers more accurate and fine-grained positional priors facilitating localizing the target segments progressively. To mitigate the training-time memory cost when modeling cross-attention on high-resolution feature maps, we have presented an efficient approach to only aggregate the contextual tokens from the high-resolution feature maps, which is shown to learn low-level details with affordable memory and computations. Finally, we have conducted extensive experiments to demonstrate the effectiveness of our proposed framework on the semantic segmentation task and its potential to be extended to other segmentation tasks.

Limitations and societal impact. Although our HRCA enriches the segmentation details with affordable memory and computations, it still requires more parameters. To this end, we will explore slimming [7, 20] or reusing [33] these blocks to save parameters. Another potential future direction is to explore the explainability of the positional priors generated by our DFPQ. Our technical innovations do not appear to have any negative societal impacts. However, the trained model may deliver unstable or biased predictions with training data that is not reviewed properly.

Acknowledgement. Dr Jing Zhang was supported by Australian Research Council Projects in part by FL170100117 and IH180100002.

References

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. 5

[2] Walid Bousselham, Guillaume Thibault, Lucas Pagano, Archana Machireddy, Joe Gray, Young Hwan Chang, and Xubo Song. Efficient self-ensemble framework for semantic segmentation. *arXiv preprint arXiv:2111.13280*, 2021. 1, 5, 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229, 2020. 1, 2, 3, 6, 7, 8

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021. 1, 3

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017. 1

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 1, 5, 6

[7] Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *NeurIPS*, 34:19974–19988, 2021. 8

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, volume 34, 2021. 1, 3, 5, 6

[10] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. In *ICLR*, 2023. 2, 3, 4, 6

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 5, 6

[12] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 3

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[14] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022. 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[16] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *ICML*, pages 2943–2952, 2020. 8

[17] Mark Everingham, SM Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 6

[18] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019. 8

[19] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*, pages 1243–1252, 2017. 3, 6

[20] Haoyu He, Jing Liu, Zizheng Pan, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Pruning self-attentions into convolutional layers in single path. *arXiv preprint arXiv:2111.11802*, 2021. 8

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[22] Shihua Huang, Zhichao Lu, Ran Cheng, and Cheng He. Fapn: Feature-aligned pyramid network for dense image prediction. In *ICCV*, pages 864–873, 2021. 2

[23] Jitesh Jain, Anukriti Singh, Nikita Orlov, Zilong Huang, Jiachen Li, Steven Walton, and Humphrey Shi. Semask: Semantically masked transformers for semantic segmentation. *arXiv preprint arXiv:2112.12782*, 2021. 2, 3, 4

[24] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, pages 620–640, 2022. 1

[25] Feng Li, Hao Zhang, Huaizhe xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. *arXiv preprint arXiv:2206.02777*, 2022. 2, 3

[26] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022. 1

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 2, 5

[28] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *ICLR*, 2022. 1, 2, 3, 4, 7, 8

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 3, 5

[30] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pages 3651–3660, 2021. 1, 2

[31] Zipeng Qin, Jianbo Liu, Xiaolin Zhang, Maoqing Tian, Aojun Zhou, Shuai Yi, and Hongsheng Li. Pyramid fusion transformer for semantic segmentation. *arXiv preprint arXiv:2201.04019*, 2022. 5, 6

[32] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *NAACL*, pages 464–468, 2018. 3

[33] Zhiqiang Shen, Zechun Liu, and Eric Xing. Sliced recursive transformer. In *ECCV*, pages 727–744, 2022. 8

[34] Hengcan Shi, Munawar Hayat, and Jianfei Cai. Transformer scale gate for semantic segmentation. In *CVPR*, 2023. 1, 2

[35] Vighnesh Shiv and Chris Quirk. Novel positional encodings to enable tree-based transformers. *NeurIPS*, 32, 2019. 3

[36] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 2

[37] Shitao Tang, Jiahui Zhang, Siyu Zhu, and Ping Tan. Quadtree attention for vision transformers. In *ICLR*, 2022. 5

[38] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, pages 282–298, 2020. 3, 4

[39] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021. 3

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1, 3

[41] Pichao Wang, Xue Wang, Fan Wang, Ming Lin, Shuning Chang, Hao Li, and Rong Jin. Kvt: k-nn attention for boosting vision transformers. In *ECCV*, pages 285–302, 2022. 5

[42] Wen Wang, Jing Zhang, Yang Cao, Yongliang Shen, and Dacheng Tao. Towards data-efficient detection transformers. In *ECCV*, pages 88–105, 2022. 1

[43] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022. 1, 2, 3, 4, 5, 7

[44] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 5, 6

[45] 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*, volume 34, 2021. 2

[46] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *NeurIPS*, 34:28522–28535, 2021. 2

[47] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, volume 32, 2019. 3

[48] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *NeurIPS*, 2021. 2

[49] Zhuyu Yao, Jiangbo Ai, Boxun Li, and Chi Zhang. Efficient detr: improving end-to-end object detector with dense prior. *arXiv preprint arXiv:2104.01318*, 2021. 1

[50] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *TPAMI*, 2022. 2

[51] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. In *NeurIPS*, volume 33, pages 17283–17297, 2020. 5

[52] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: learning varied-size window attention in vision transformers. In *ECCV*, pages 466–483, 2022. 2

[53] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *IJCV*, pages 1–22, 2023. 2

[54] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In *NeurIPS*, volume 34, 2021. 1, 2, 3

[55] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 2

[56] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890, 2021. 1, 3

[57] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017. 2, 5, 6, 7, 8

[58] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 5