Adaptive Agent Transformer for Few-shot Segmentation

Yuan Wang[1*], Rui Sun[1*], Zhe Zhang[3,4,5**], and Tianzhu Zhang[1,2,5]

[1] University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
[3] Beijing Institute of Technology
[4] Lunar Exploration and Space Engineering Center of CNSA
[5] Deep Space Exploration Laboratory

Abstract. Few-shot segmentation (FSS) aims to segment objects in a given query image with only a few labelled support images. The limited support information makes it an extremely challenging task. Most previous best-performing methods adopt prototypical learning or affinity learning. Nevertheless, they either neglect to further utilize support pixels for facilitating segmentation and lose spatial information, or are not robust to noisy pixels and computationally expensive. In this work, we propose a novel end-to-end adaptive agent transformer (AAFormer) to integrate prototypical and affinity learning to exploit the complementarity between them via a transformer encoder-decoder architecture, including a representation encoder, an agent learning decoder and an agent matching decoder. The proposed AAFormer enjoys several merits. First, to learn agent tokens well without any explicit supervision, and to make agent tokens capable of dividing different objects into diverse parts in an adaptive manner, we customize the agent learning decoder according to the three characteristics of context awareness, spatial awareness and diversity. Second, the proposed agent matching decoder is responsible for decomposing the direct pixel-level matching matrix into two more computationally-friendly matrices to suppress the noisy pixels. Extensive experimental results on two standard benchmarks demonstrate that our AAFormer performs favorably against state-of-the-art FSS methods.

Keywords: Few-shot Segmentation, Semantic Segmentation, Transformer

1 Introduction

Semantic segmentation is a fundamental task that has achieved conspicuous achievements attributed to the development in deep neural network, especially fully convolutional network (FCN) [21]. However, it is laborious and time-consuming to gather massive pixel-level annotations as training data. To alleviate

[*] Equal contibution
[**] Corresponding author

the data-hunger issue, considerable works [6,16,23,14] have turned their attention to the semi-supervised setting. However, neither fully supervised models nor semi-supervised models generalize well to novel classes with extremely few exemplars. In contrast, humans can easily identify a new object after only seeing it once. Inspired by this, there has been increasing interest recently on few-shot segmentation (FSS) [26] which can quickly adapt to novel categories.

In this work, we tackle the few-shot segmentation problem, where the goal is to segment objects in a given *query* image I_q while only a few *support* images I_s with corresponding annotations M_s are available. Since there are usually large intra-class variations such as scale, pose or background differences between the support and query images, how to fully exploit limited information from support samples for accurate segmentation is thus extremely challenging.

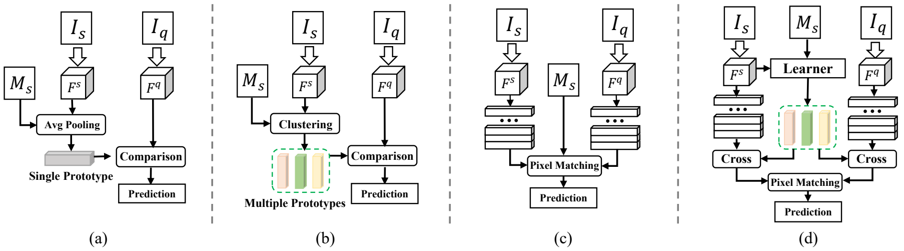

Fig. 1: Different learning formulation for few-shot segmentation. (a) Prototypical learning methods with single prototype. (b) Prototypical learning methods with multiple prototypes. (c) Affinity learning methods. (d) Our proposed AAFormer that absorbs the merits of both prototypical learning and affinity learning methods by modeling adaptive agent tokens for pixel-level matching.

Top-performing FSS methods can be roughly categorized as prototypical learning methods and affinity learning methods. On one hand, prototypical learning methods [33,10,43] adopt masked average pooling to achieve a single prototype in the hope of being robust to noisy pixels, and perform feature comparison between query pixels and a single prototype to segment the desired object, as shown in Fig.1a. However, these methods inevitably drop the spatial information. Moreover, relying solely on the single prototype focusing on global foreground feature fails to capture the diverse object parts, which are crucial to deal with object occlusion and large variations across images. To alleviate these problems, recent works adopt EM algorithm [38] or clustering [20,17] to generate multiple prototypes for better spatial coverage in foreground regions (see Fig.1b). However, these methods only conduct matching between obtained prototypes and query features without explicitly exploring the valuable pixel-level support information, which can actually further contribute to the precise segmentation. On the other hand, affinity learning methods [40,32,42] attempts to directly leverage pixel-to-pixel similarity between support features and query features for segmentation (see Fig.1c). These approaches take advantage of the detailed pixel-level support information and perform well in preserving spatial information. However, direct pixel-level similarity is not only computation prohibitive, but also tends to suffer from the confusion caused by background clutters or noisy pixels because of neglecting contextual information. Overall, The above analysis indicates that prototypical learning methods and affinity learning methods are naturally complementary. The former mines the pixel context information against noisy pixels and is computationally friendly, but fails to further utilize valuable support pixels to facilitate segmentation and loses spatial information, while the latter is just the other way around. Therefore, it is more desirable to integrate these two formulations for exploiting their complementary potential by performing pixel-level matching based on modeling diverse prototypes.

Motivated by the above discussions, we propose an end-to-end Adaptive Agent Transformer (**AAFormer**) to integrate adaptive prototypes as agent into affinity-based FSS (Fig.1d) via a transformer encoder-decoder architecture [30], including a representation encoder, an agent learning decoder and an agent matching decoder. **In the representation encoder,** we propose the self-attention mechanism to capture the full image context information. Specifically, we aggregate pixel-specific global context to each pixel position to obtain robust context-aware pixel features that can represent object appearance well. **In the agent learning decoder,** we distill support information into condensed agent tokens to establish the bridge between the support and query images. To learn the agent tokens well without any explicit supervision, we elegantly design this decoder customized for the following three characteristics. (a) Context awareness. We introduce the masked cross attention mechanism that only attends agent tokens with support pixels restricted to the foreground region. In this way, agent tokens have the ability to further absorb foreground context from support pixels, and adapt to occlusion and large variations across images. (b) Spatial awareness. To make the part masks activated by the agent tokens more compact rather than dispersive, we model the structural spatial information with the support of distance transformation for agent tokens initialization. In addition, we also introduce the position embedding in the agent learning process to make output agent tokens sensitive to spatial location, and guide the part mask by learning a local activation. (c) Diversity. To avoid the multiple agent tokens focusing on the same object part, we impose the equal partition constraint to expand the discrepancy among part masks. In specific, we allocate foreground pixels evenly over agent tokens benefiting from the initial marginal distribution of the optimal transport algorithm, and attain the optimal transport plan which can be regarded as the refined part masks. In this case, agent tokens can decompose different target objects into diverse and complementary parts in an *adaptive* manner. **In the agent matching decoder,** we decompose the massive pixel-level support-query matching matrix into two more manageable matrices based on obtained agent tokens at a light computational cost, and introduce the alignment matrix for filtering out ambiguous matching caused by noisy pixels. In specific, direct support-query matching is substituted by support-agent matching and agent-query matching. With a limited number of agent tokens, AAFormer efficiently performs pixel-level matching with a drastically reduced complexity compared to previous one. Besides, the alignment matrix guided by context-rich agent tokens can filter out the matching weights between support and query pixels that do not belong to the same object part. In this case, the noisy pixels will be suppressed while the true correspondences enjoy higher weights.

The contributions of our method could be summarized as follows:

- We propose an Adaptive Agent Transformer (AAFormer) for the few-shot segmentation in a unified framework. Specifically, we design the representation encoder to acquire global context-aware pixel features, the agent learning decoder to condense support information into agent tokens for bridging the support and query images, and the agent matching decoder to decompose the direct pixel-level matching matrix into two more computationally-friendly matrices for suppressing the noisy pixels.
- To the best of our knowledge, this is the first work to absorb the merits of both prototypical learning and affinity learning formulation by modeling adaptive agent tokens for pixel-level matching. To learn agent tokens well without any explicit supervision, and to make agent tokens capable of dividing different objects into diverse parts in an adaptive manner, we further customize the agent learning decoder according to the three characteristics of context awareness, spatial awareness and diversity.
- Extensive experimental results with two different backbones on two challenging benchmarks demonstrate that our AAFormer performs favorably against state-of-the-art FSS methods.

2 Related Work

2.1 Semantic Segmentation

Semantic segmentation is a task of assigning each pixel in a given image into a category label, most promising segmentation methods are based on the Fully Convolutional Network (FCN) [21]. Later, many remarkable breakthroughs come from the enlargement of the receptive field. For example, Deeplab [4,5] integrates dilated convolutions combined with pyramid pooling module [44] into the FCN architecture. In addition to CNN based models, some recent works [35,27,7,8,45,2] have applied transformer-based architectures for semantic segmentation [28], and has resulted in comparable performance. For instance, Mask2former [7] treats semantic segmentation as a binary mask prediction task based on set prediction mechanism proposed by DETR [3]. However, these methods usually require massive pixel-level annotations as training data and cannot generalize to novel classes with only a few labelled images. In this paper, we focus on few-shot segmentation to overcome these limitations.

2.2 Few-Shot Segmentation

Few-shot segmentation [26] tackles a challenging task of segmenting novel class query images with only a few labeled support images available. Existing FSS methods can be roughly categorized into two categories: prototypical learning methods and affinity learning methods. For prototypical learning, most methods [10,41,38,20,43,39] adopt masked average pooling to achieve a single prototype and perform feature comparison with query pixels to segment the desired object. For example, PANet [33] performs a prototype alignment regularization that encourages the prototypes to contain more consistent information. However, these methods are prone to inevitably drop the spatial information [17]. To alleviate these problems, recent works [38,20,17] attempt to generate multiple prototypes by EM algorithm or clustering for better spatial coverage in foreground regions. For instance, Zhang *et al.* [39] encode the uncovered support feature for initial prediction as a extra auxiliary prototype to reduce information loss. However, these methods neglect exploring the valuable pixel-level support information, which can actually further contribute to the precise segmentation.

Different from prototypical learning, affinity learning methods [32,40,42,39] attempt to directly leverage pixel-level support-query matching for segmentation. For example, PFENet [29] constructs the class-agnostic prior mask to guide the segmentation by calculating the maximum support-query similarity in high-level features. CyCTR [42] introduces the cycle-consistent attention operation to aggregate beneficial support pixel-level features. However, direct pixel-level similarity is not only computation prohibitive, but also tends to suffer from the confusion caused by background clutters or noisy pixels because of neglecting contextual information. Apart from existing methods, our method absorbs the merits of both prototypical learning and affinity learning methods by modeling adaptive agent tokens for pixel-level matching with a transformer encoder-decoder architecture.

3 The Proposed Approach

3.1 Problem Definition

Widely used episodic meta-training [31] is adopted in few-shot segmentat. Specifically, we denote the training set as $\mathcal{D}_{train}$ and the testing set as $\mathcal{D}_{test}$, the categories of the two sets $\mathbb{C}_{train}$ and $\mathbb{C}_{test}$ are disjoint ($\mathbb{C}_{train} \cap \mathbb{C}_{test} = \emptyset$). To train the model, a set of episodes are sampled from $\mathcal{D}_{train}$, each of which is composed of a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. In the K-shot setting, $\mathcal{S} = \{(I_s^k, M_s^k)\}_{k=1}^K$, where the I_s^k and the M_s^k are the i-th support image and its corresponding ground-truth binary mask. Meanwhile, $\mathcal{Q} = (I_q, M_q)$, where the I_q and M_q are the query image of the same class in $\mathcal{S}$ and its ground-truth, respectively. In each episode, the model makes prediction on the I_q of $\mathcal{Q}$ conditioned on the $\mathcal{S}$, and M_q is provided to supervise the training process.

3.2 Overview

As illustrated in Fig.2, the proposed AAFormer mainly includes three modules, the representation encoder, the agent learning decoder and the agent matching decoder. Among them, the representation encoder is applied to consider the global context to learn robust features that effectively represent object appearance. The agent learning decoder is responsible for adaptively absorbing contextual information into agent tokens, and makes these regions discovered by learnt agent tokens compact and diverse. The agent matching decoder is used to equip each query pixel with beneficial support information to facilitate the classification. The details are as follows.

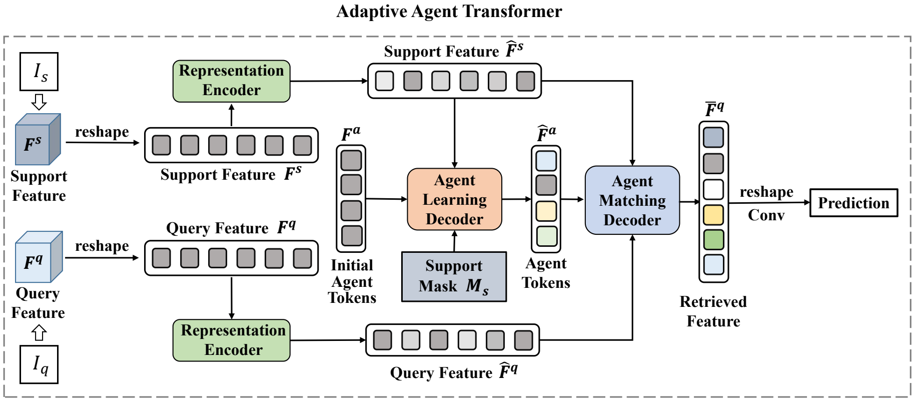

Fig. 2: Framework of our proposed Adaptive Agent Transformer (AAFormer). There are three modules in the AAFormer, i.e., a representation encoder, an agent learning decoder and an agent matching decoder.

3.3 Representation Encoder

We design the representation encoder to acquire robust context-aware pixel features that can represent object appearance well. Given the backbone features $\mathbf{F}^s \in \mathbb{R}^{h\times w\times c}$ and query feature map $\mathbf{F}^q \in \mathbb{R}^{h\times w\times c}$ obtained from the pretrained ResNet [13]. To cope with the difference in the distribution of targets on scale and pose, we adopt self-attention mechanism in the representation encoder to capture the long-range context information. Sepcificly, we flatten the spatial dimensions of $\mathbf{F}^s$ and $\mathbf{F}^q$ as 1D sequences. Then we obtain the queries, keys and values from $\mathbf{F}^s \in \mathbb{R}^{hw\times c}$ and $\mathbf{F}^q \in \mathbb{R}^{hw\times c}$. Note that we denote the superscript as $*$ and $* \in \{s, q\}$ for brevity. Formally,

$$\mathbf{Q}^* = \mathbf{F}^*\mathbf{W}^{\mathcal{Q}}_*, \quad \mathbf{K}^* = \mathbf{F}^*\mathbf{W}^{\mathcal{K}}_*, \quad \mathbf{V}^* = \mathbf{F}^*\mathbf{W}^{\mathcal{V}}_*, \tag{1}$$

where $\mathbf{W}^{\mathcal{Q}}_* \in \mathbb{R}^{c\times c_k}, \mathbf{W}^{\mathcal{K}}_* \in \mathbb{R}^{c\times c_k}, \mathbf{W}^{\mathcal{V}}_* \in \mathbb{R}^{c\times c_v}$ are linear projections. Then we can calculate the attention weight matrix $\mathbf{S} \in \mathbb{R}^{hw\times hw}$ with the scaled dot-product attention and the output context-aware pixel features are computed through the following equation:

$$\hat{\mathbf{F}}^* = \text{Attention}(\mathbf{Q}^*, \mathbf{K}^*, \mathbf{V}^*) = \text{Softmax}(\frac{\mathbf{Q}^*(\mathbf{K}^*)^\mathsf{T}}{\sqrt{\mathrm{d_k}}})\mathbf{V}^*. \tag{2}$$

Among which $\sqrt{\mathrm{d_k}}$ is a scaling factor for stabilizing the training and T denotes the transpose operation. Following the standard transformer [30], the Eq. (2) is implemented with the multi-head mechanism and the feed-forward network (FFN) is further applied to obtain the final output $\hat{\mathbf{F}}^s$ and $\hat{\mathbf{F}}^q$. In this way, The obtained pixel features are supported by its global context so that are more robust to background clutters and can better represent object appearance.

3.4 Agent Learning Decoder

Agent learning decoder aims to condense support information into a set of agent tokens for bridging the support and query images. We first elaborate the initialization of the agent tokens which can accelarate the training and make the agent tokens **spatial-aware**. Specifically, Following [15], Euclidean distance transform is used to iteratively select a set of seed points that far away from each other as well as the boundaries, please refer to the **Supplementary Materials** for specific practices. We then adopt the features at the chosen seed points that distribute uniformly in the masked region as initial agent tokens denoted by $\mathbf{F}^a \in \mathbb{R}^{K\times c}$, where the K is the number of agent tokens.

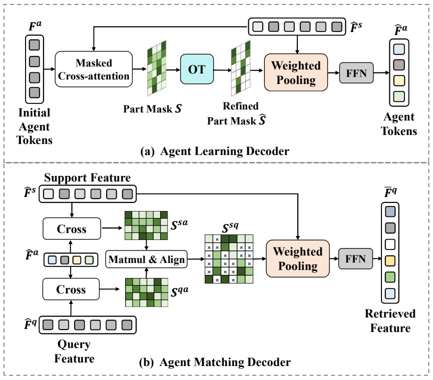

Fig. 3: Illustration of the Agent Learning Decoder(1-st row) and the Agent Matching Decoder (2-nd row).

In order to make agent tokens **context-aware**, we introduce the masked cross-attention between the agent tokens and support features to efficiently aggregate the relevent forground contextual information into corresponding agent tokens. Concretely, we first calculate a masked attention weight matrix:

$$\mathbf{S} = \text{Softmax}(\frac{\mathbf{Q}^{\mathrm{a}}(\mathbf{K}^{\mathrm{s}})^\mathsf{T}}{\sqrt{\mathrm{d_k}}} + \boldsymbol{\mathcal{M}}), \quad \mathbf{Q}^{\mathrm{a}} = \mathbf{F}^{\mathrm{a}}\mathbf{W}^{\mathcal{Q}}_{\mathrm{a}}, \quad \mathbf{K}^{\mathrm{s}} = \mathbf{F}^{\mathrm{s}}\mathbf{W}^{\mathcal{K}}_{\mathrm{s}}, \tag{3}$$

where the additional attention mask $\boldsymbol{\mathcal{M}}$ at feature location (m, n) is

$$\boldsymbol{\mathcal{M}} = \begin{cases} 0, & \text{if} \quad \mathbf{N}(m,n) = 1 \\ -\infty, & \text{otherwise} \end{cases}, \tag{4}$$

among which the $\mathbf{N} \in \{0,1\}^{K \times hw}$ denote the duplication of $\mathbf{M} \in \mathbb{R}^{1 \times hw}$ from $\mathbb{R}^{1 \times hw}$ to $\mathbb{R}^{K \times hw}$ and the $\mathbf{M}$ is the flattened support mask. The masked cross-attention only attends within the foreground region of the support mask for agent tokens, which not only makes agent tokens rich in forground context, but also leads to faster convergence [7].

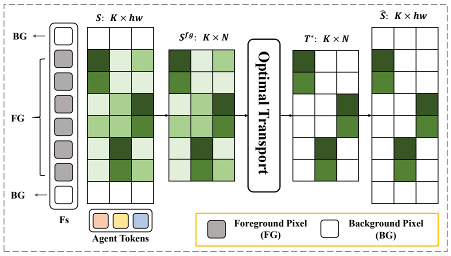

Fig. 4: Process of obtaining the refined part mask via OT algorithm.

We found that without constraining the agent learning process, multiple agent tokens tend to focus on the same area. For the purpose of learning more **diverse** agent tokens, we further constrain the attention matrix to evenly allocate the forground pixels to different agent tokens. Concretly, we model the pixels allocating as the Optimal Transport (OT) problem. The goal of the OT problem is to find a transportation plan $\mathbf{T}^{*}$ at a global minimal transportation cost, which can be solved elegantly using Sinkhorn algorithm with linear programming [9]. As illustrated in Fig.4., we are intrested in condensing the foreground support features into different agent tokens. The cost matrix is defined as $(1 - \mathbf{S}^{fg})$, where the $\mathbf{S}^{fg} \in \mathbb{R}^{K \times N}$ is the matrix of similarity between agent tokens and forground support features, and the N is the amount of the foreground support pixels specified by support mask. The higher similarity in $\mathbf{S}^{fg}$ leads to a lower corresponding transport cost. We denote the transport plan as $\mathbf{T} \in \mathbb{R}^{K \times N}$ and the optimization function is as follows:

$$\max_{\mathbf{T} \in \mathcal{T}} \operatorname{Tr}\left(\mathbf{T}^{\mathsf{T}}(1 - \mathbf{S}^{fg})\right) + \epsilon H(\mathbf{T}), \quad H(\mathbf{T}) = -\sum_{ij} \mathbf{T}_{ij} \log \mathbf{T}_{ij}, \tag{5}$$

where $H(\mathbf{T})$ is the entropy function, and ϵ is the parameter that controls the smoothness of the mapping and is set to be 0.05 in our experiments. We impose the *equal partition* constraints on $\mathbf{T}$:

$$\mathcal{T} = \left\{ \mathbf{T} \in \mathbb{R}_{+}^{K \times N} \mid \mathbf{T}\mathbb{1} = \frac{1}{K} \cdot \mathbb{1}, \mathbf{T}^{\mathsf{T}}\mathbb{1} = \frac{1}{N} \cdot \mathbb{1} \right\}, \tag{6}$$

where $\mathbb{1}$ denotes the vector of all ones in the appropriate dimension. Eq.(6) enforces that each agent token is assigned the same number of forground pixels thus preventing a trivial solution where all pixels are assigned to a single agent token. So that different agent tokens responsible for different areas that are mutual complementary. As shown in Fig.4, we zero-pad the $\mathbf{T}^{*}$ to result in the refined part mask $\hat{\mathbf{S}}$. The final output agent tokens are acquired from the weighted sum of $\mathbf{V}^{s}$:

$$\hat{\mathbf{F}}^{a} = \mathrm{FFN}(\hat{\mathbf{S}})\mathbf{V}^{\mathrm{s}} \tag{7}$$

Benefiting from ALD, the output agent tokens decompose different target objects into diverse and complementary parts in an adaptive manner.

3.5 Agent Matching Decoder

Agent matching decoder is designed to equip query pixels with salutary information from support features in a robust and efficient way. Different from previous methods [32,42] that perform dense similarity calculation directly of two branches, we decompose the massive pixel-level support-query matrix into two more mangeable matrices and introduce an extra alignment matrix for filtering out ambiguous matching. Formally:

$$\mathbf{S}^{as} = \frac{\mathbf{Q}^a(\mathbf{K}^s)^\mathsf{T}}{\sqrt{d_k}}, \quad \mathbf{Q}^a = \hat{\mathbf{F}}^a\mathbf{W}^{\mathcal{Q}}_a, \quad \mathbf{K}^s = \hat{\mathbf{F}}^s\mathbf{W}^{\mathcal{K}}_s, \tag{8}$$

$$\mathbf{S}^{qa} = \frac{\mathbf{Q}^q(\mathbf{K}^a)^\mathsf{T}}{\sqrt{d_k}}, \quad \mathbf{Q}^q = \hat{\mathbf{F}}^q\mathbf{W}^{\mathcal{Q}}_q, \quad \mathbf{K}^a = \hat{\mathbf{F}}^a\mathbf{W}^{\mathcal{K}}_a, \tag{9}$$

$$\mathbf{S}^{qs} = \mathrm{Softmax}(\mathbf{S}^{\mathrm{sa}}\mathbf{S}^{\mathrm{aq}} + \boldsymbol{\mathcal{A}}), \tag{10}$$

where the $\mathbf{W}^{\mathcal{K}}_*, \mathbf{W}^{\mathcal{Q}}_*$ and $* \in \{s, a, q\}$ denote the linear projection, and the aligning matrix $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{hw \times hw}$ is obtained by

$$\boldsymbol{\mathcal{A}}(i,j) = \begin{cases} \quad 0, \text{ if } \mathrm{argmax_t}\, \mathbf{S}^{\mathrm{as}}(\mathrm{t,i}) = \mathrm{argmax_t}\, \mathbf{S}^{\mathrm{qa}}(\mathrm{j,t}) \\ -\infty, \text{ otherwise} \end{cases}, \tag{11}$$

where $(i,j) \in \{1, 2, \ldots, hw\}$ and $t \in \{1, 2, \ldots, K\}$. In this way, $\boldsymbol{\mathcal{A}}$ filters out these attention weights between the support and query pixels that do not belong to the same agent token. We inplant this support-query correlation into the multi-head attention mechnism within the decoder, and given the support-query attention matrix $\mathbf{S}^{qs}$ we can retrieve the corresponding support features via the weighted sum of $\mathbf{V}^s$ and a FFN:

$$\bar{\mathbf{F}}^q = \mathrm{FFN}((\mathbf{S}^{sq})\mathbf{V}^{\mathrm{q}}), \quad \mathbf{V}^{\mathrm{q}} = \hat{\mathbf{F}}^{\mathrm{s}}\mathbf{W}^{\mathcal{V}}, \tag{12}$$

The obtained $\bar{\mathbf{F}}^q$ is reshaped back to spatial dimensions and processed by a small convolution block to result in the final prediction. The convlution block consists of one 3 × 3 convolution, one *ReLU* activation and one 1 × 1 convolution. The proposed alignment matrix injects the contextual information into the pixel-wise matching to filter out the matching weights between support and query pixels that do not belong to the same object part. Besides, the decomposition in Eq.(10) converts the computation complexity from $o(c(hw)^2)$ to $o(K(hw)^2)$, where the c is the hidden dimension of decoder and $K \ll c$, which makes our approach more efficient.

4 Experiments

4.1 Dataset and Evaluation Metric

Dataset. We evaluate our approach on two widely used few-shot segmentation datasets, Pascal-5^i [11] and COCO-20^i [18]. For Pascal-5^i, which consists of the

Table 1: Comparison with other state-of-the-art methods for 1-shot and 5-shot segmentation on Pascal-5^i. The mIoU of each fold and the FB-IoU of four folds are reported. Best results in bold.

Method	Backbone	mIoU(1-shot)					FB-IoU	mIoU(5-shot)					FB-IoU
		5^0	5^1	5^2	5^3	Mean	(1-shot)	5^0	5^1	5^2	5^3	Mean	(5-shot)
PANet[ICCV2019] [33]	Vgg-16	42.3	58.0	51.1	41.2	48.1	66.5	51.8	64.6	59.8	46.5	55.7	70.7
FWB[ICCV2019] [24]		47.0	59.6	52.6	48.3	51.9	-	50.9	62.9	56.5	50.1	55.1	-
SG-One[TCYB2020] [43]		40.2	58.4	48.4	38.4	46.3	63.1	41.9	58.6	48.6	39.4	47.1	65.9
PMM[ECCV2020] [38]		47.1	65.8	50.6	48.5	53.0	-	50.0	66.5	51.9	47.6	54.0	-
ASR[CVPR2021] [19]		50.2	66.4	54.3	51.8	55.7	-	53.7	68.5	55.0	54.8	58.0	-
CANet[CVPR2019] [41]	Res-50	52.5	65.9	51.3	51.9	55.4	66.2	55.5	67.8	51.9	53.2	57.1	69.6
PGNet[ICCV2019] [40]		56.0	66.9	50.6	50.4	56.0	69.9	57.7	68.7	52.9	54.6	58.5	70.5
PPNet[ECCV2020] [20]		47.8	58.8	53.8	45.6	51.5	-	58.4	67.8	64.9	56.7	62.0	-
PMM[ECCV2020] [38]		55.2	66.9	52.6	50.7	56.3	-	56.3	67.3	54.5	51.0	57.3	-
PFENet[TPAMI2020] [29]		61.7	69.5	55.4	56.3	60.8	73.3	63.1	70.7	55.8	57.9	61.9	73.9
SCLNet[CVPR2021] [39]		63.0	70.0	56.5	57.7	61.8	71.9	64.5	70.9	57.3	58.7	62.9	72.8
ASGNet[CVPR2021] [17]		58.8	67.9	56.8	53.7	59.3	69.2	63.7	70.6	64.2	57.4	63.9	74.2
MMNet[ICCV2021] [34]		62.7	70.2	57.3	57.0	61.8	-	62.2	71.5	57.5	**62.4**	63.4	-
RePRI[CVPR2021] [1]		60.2	67.0	**61.7**	47.5	59.1	-	64.5	70.8	**71.7**	60.3	66.8	-
CWT[ICCV2021] [22]		56.3	62.0	59.9	47.2	56.4	-	61.3	68.5	68.5	56.6	63.7	-
SAGNN[CVPR2021] [36]		64.7	69.6	57.0	57.2	62.1	73.2	64.9	70.0	57.0	59.3	62.8	73.3
ASR[CVPR2021] [19]		55.2	70.4	53.4	53.7	58.2	72.9	59.4	71.9	56.9	55.7	61.0	74.1
CMN[ICCV2021] [37]		64.3	70.0	57.4	**59.4**	62.8	72.3	65.8	70.4	57.6	60.8	63.7	72.8
CyCTR[NIPS2021] [42]		67.8	72.8	58.0	58.0	64.2	-	71.1	73.2	60.5	57.5	65.6	-
AAFormer (Ours)	Res-50	**69.1**	**73.3**	59.1	59.2	**65.2**	**73.8**	**72.5**	**74.7**	62.0	61.3	**67.6**	**76.2**
FWB[ICCV2019] [24]	Res-101	51.3	64.5	56.7	52.2	56.2	-	54.9	67.4	62.2	55.3	59.9	-
DAN[ECCV2020] [32]		54.7	68.6	57.8	51.6	58.2	62.3	57.9	69.0	60.1	54.9	60.5	63.9
PFENet[TPAMI2020] [29]		60.5	69.4	54.4	55.9	60.1	72.9	62.8	70.4	54.9	57.6	61.4	73.5
ASGNet[CVPR2021] [17]		59.8	67.4	55.6	54.4	59.3	71.7	64.6	71.3	64.2	57.3	64.4	75.2
RePRI[CVPR2021] [1]		59.6	68.6	**62.2**	47.2	59.4	-	66.2	71.4	67.0	57.7	65.6	-
CWT[ICCV2021] [22]		56.9	65.2	61.2	48.8	58.0	-	62.6	70.2	**68.8**	57.2	64.7	-
CyCTR[NIPS2021] [42]		69.3	72.7	56.5	58.6	64.3	72.9	73.5	74.0	58.6	60.2	66.6	75.0
AAFormer (Ours)	Res-101	**69.9**	**73.6**	57.9	**59.7**	**65.3**	**74.9**	**75.0**	**75.1**	59.0	**63.2**	**68.1**	**77.3**

Pascal VOC 2012 dataset with extra annotations from SBD [12], 20 categories are divided into 4 folds with 5 classes per fold for cross-validation, as done in [26]. For COCO-20^i, we follow the the data split protocol in [24] to separate the 80 classes evenly into 4 folds, where each fold contains 60 classes for training and the remaining 20 classes for testing. During inference, 1,000 episodes are randomly sampled from the test split.

Evaluation Metric. Following the previous practices [29,33,43,41,42], we adopt two evaluation metrics, i.e., mean intersection-over-union (mIoU) and foreground-background IoU (FB-IoU). We mainly focus on the mIoU metric as it reflects the average result over all classes thus alleviating the performance bias of scarce classes.

4.2 Implementation Details

Our models are trained 200 epochs with batch size 4 for Pascal-5^i and 50 epochs with batch size 24 for COCO-20^i. We adopt the ImageNet [25] pretrained ResNet50 and Resnet101 [13] as the backbone to extract features in our experiments for fair comparison. Given the support feature $\mathbf{F}^s \in \mathbb{R}^{h\times w\times c}$ and query feature $\mathbf{F}^q \in \mathbb{R}^{h\times w\times c}$, we set the number of the cross layers in our agent learning decoder to 1, and set 2 in the representation encoder. Please see the **supplementary material** for more implementation details.

Table 2: Comparison with other state-of-the-art methods for 1-shot and 5-shot segmentation on COCO-20^i. The mIoU of each fold and the FB-IoU of four folds are reported. Best results in bold.

Method	Backbone	mIoU(1-shot)					FB-IoU	mIoU(5-shot)					FB-IoU
		20^0	20^1	20^2	20^3	Mean	(1-shot)	20^0	20^1	20^2	20^3	Mean	(5-shot)
PPNet[ECCV2020] [20]	Res-50	28.1	30.8	29.5	27.7	29.0	-	39.0	40.8	37.1	37.3	38.5	-
PMM[ECCV2020] [38]		29.5	36.8	28.9	27.0	30.6	-	33.8	42.0	33.0	33.3	35.5	-
MMNet[ICCV2021] [34]		34.9	41.0	37.2	37.0	37.5	-	37.0	40.3	39.3	36.0	38.2	-
RePRI[CVPR2021] [1]		31.2	38.1	33.3	33.0	34.0	-	38.5	46.2	40.0	43.6	42.1	-
ASR[CVPR2021] [19]		30.6	36.7	32.7	35.4	33.9	-	33.1	39.5	34.2	36.2	35.8	-
CMN[ICCV2021] [37]		37.9	**44.8**	38.7	35.6	39.3	61.7	42.0	**50.5**	41.0	38.9	43.1	63.3
CyCTR[NIPS2021] [42]		38.9	43.0	39.6	39.8	40.3	-	41.1	48.9	45.2	47.0	45.6	-
FWB[ICCV2019] [24]	Res-101	17.0	18.0	21.0	28.9	21.2	-	19.1	21.5	23.9	30.1	23.7	-
PFENet[TPAMI2020] [29]		34.3	33.0	32.3	30.1	32.4	58.6	38.5	38.6	38.2	34.3	37.4	61.9
SCLNet[CVPR2021] [39]		36.4	38.6	37.5	35.4	37.0	-	38.9	40.5	41.5	38.7	39.9	-
CWT[ICCV2021] [22]		30.3	36.6	30.5	32.2	32.4	-	38.5	46.7	39.4	43.2	42.0	-
SAGNN[CVPR2021] [36]		36.1	41.0	38.2	33.5	37.2	60.9	40.9	48.3	42.6	38.9	42.7	63.4
AAFormer (Ours)	Res-50	**39.8**	44.6	**40.6**	**41.4**	**41.6**	**67.7**	**42.9**	50.1	**45.5**	**49.2**	**46.9**	**68.2**

4.3 Comparison with State-of-the-art Methods

Pascal-5^i. In Tabel 1, we compare our proposed AAFormer with the state-of-the-art few-shot segmentation methods. We consistently observe that our AAFormer outperforms all previous models under both 1-shot and 5-shot settings, which strongly proves the effectiveness of our method. For fair comparison, we report results with the ResNet-50 and Resnet-101 backbones. Specifically, Our approach achieves 65.3% and 68.1% in the 1-shot and 5-shot settings with the ResNet-101 backbone that significantly outperforms the recent prototypical learning methods (e.g., ASGNet), achieving a large margin of 6.0% and 3.7% in mIoU. This is because the prototypic learning methods only leverage the correlation between the prototypes and query features without considering the pixel-level support features, while the agent matching decoder in our method further explores the pixel-wise support information and contributes to accurate segmentation. With the more lightweight ResNet50 backbone, the performance of AAFormer is also in the lead. Compared with the best affinity learning method (CyCTR), our method has a clear lead and obtains 1.0% mIoU gain in the 1-shot setting and 2.0% in the 5-shot setting.

COCO-20^i. In Tabel 2 we report the comparison on COCO-20^i, which is much more difficult than Pascal-5^i with more complex cases, such as drastic object appearance differences, messy scenes and severe occlusion. In the absence of careful parameter adjustments, our AAFormer also achieves superior results than the existing best performing method (CyCTR), i.e., obtains 1.3% and 1.3% mIoU gain in the 1-shot and 5-shot settings. This demostrates the the stability of our method. We analyze that the performance can also benefit from the proposed representation encoder, which can capture the full image context information for better representing the object appearance to deal with complex cases. While conducting feature processing on the raw backbone features tends to be confused by the background clutters or other interferent.

4.4 Ablation Study and Analysis

To look deeper into our method, we perform a series of ablation studies to analyze each component of our AAFormer, including the representation encoder (REnc), the agent matching decoder (AMD) and the agent learning decoder (ALD). Note that we remove all modules except the encoder with two residual blocks to conduct direct pixel-level matching between the support and query images as our baseline.

Table 3: Ablation study results. Experiments are conducted on Pascal-5^0 for 1-shot setting with ResNet-50.

REnc	AMD	ALD	mIoU
			62.2
✓			64.9
✓	✓		67.4
✓	✓	✓	**69.1**

(a) Ablation of model components.

Init.	Update	mIoU
DT	K-Means	67.4
DT	ALD(w/o OT)	68.3
DT	ALD(w/ OT)	**69.1**
Learnable	ALD(w/ OT)	68.5

(b) Ablation of agent initialization and update.

Effectiveness of the Representation Encoder. As shown in Tabel 3a, The introduction of the representation encoder achieves a certain performance lift compared with the baseline, e.g., 2.7% in mIoU. The improvements can be mainly ascribed to the proposed representation encoder that can effectively capture robust context information for representing object appearance well even in complex cases.

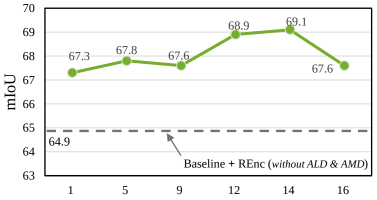

Fig. 5: Comparisons of performance with different number of agent tokens.

Effectiveness of the Agent Matching Decoder. From the comparison between the 2-nd and the 3-rd row of Table 3a, we observe that the agent matching decoder significantly improves the performance, e.g.,2.5% in mIoU. Note that the agent tokens in the 3-rd row are produced from the K-Means clustering. We conclude that this performance gain comes from the alignment matrix in the agent learning decoder, which can filter out the matching weights between support and query pixels that do not belong to the same object part for more accurate segmentation.

Effectiveness of the Agent Learning Decoder. With the utilization of the agent learning decoder, further improvements can be observed, e.g., 2.4% mIoU. This proves that our agent learning decoder can adaptively learn spatial-aware and diverse agent tokens to absorb the local context of support pixels and make the agent matching decoder more effective in reducing background noises.

Analysis of the Agent Learning Process. To explore effectiveness of different ways to learn agents, we evaluate multiple combinations of initialization and update of agent tokens. Naively, we first initialize the agents with the help of distance transformation (DT), and then update them by K-Means algorithm. The way is similar to [17] and the result is displayed in the first row of Table 3b.

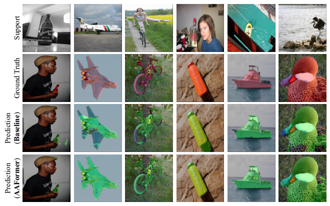

Fig. 6: Qualitative comparison with the baseline. AAFormer can achieve more accurate segmentation.

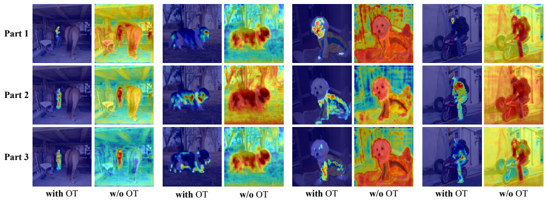

Fig. 7: Visualization of the learned agent tokens with OT and without OT. As we can see, these agent tokens adaptively decompose the object into different parts benefiting from OT.

Then we replace the K-Means algorithm by the agent learning decoder (without OT algorithm) to update the initial agent tokens. We can observe that there is a 0.9% mIoU improvement, showing our proposed ALD can better make the agents represent foreground context than traditional K-Means algorithm. And the performance improvement of 0.8% mIoU can be obtained by adding OT algorithm to the ALD, which indicates that the equal partition constraint brought by OT is beneficial to make the agents learn to adaptively decompose different target objects into diverse and complementary parts. When we use learnable tokens as the initialization of the agents, the mIoU is degraded by 0.6%, which is not surprising as query tokens will lose the structural spatial information compared to DT initialization. Despite, its performance is approximate to DT initialization with ALD (without OT), once again validating that our ALD can produce powerful agent tokens.

Hyperparameter Evaluations. In Fig. 5, we conduct quantitative experiment to analyze how many agent tokens K are better for segmentation. We can observe that the performance continues to grow until $K = 14$, which means that it is sufficient for agent learning decoder by mining fourteen different object parts.

4.5 Visualizations

Visualization of Learned Agent Tokens. We visualize the part masks activated by agent tokens to qualitatively evaluate the effect of the optimal transport algorithm. As shown in Fig.7, we can observe that without OT, multiple prototypes tend to focus on the same part containing large background noises. And thanks to the equal partition constraint from OT, the agent tokens successfully divide different target objects into diverse and complementary parts in an adaptive manner. For example, the three part masks in object *dog* (in the fifth column) focus on the head, body, and limbs respectively.

Visualization of Part Correspondence. Under the extreme challenge, we visualize part masks which come from the same class of support-query image pairs and are activated by a specific agent token, as shown in Fig.8. As we can see, intrinsic semantic correspondence is established between the pair of part masks obtained from the same agent token. For example, in the object *human* (the second column), the part *head* of the query image can accurately match with the corresponding part masks of the support image, even though the objects in the two images have different poses. This proves that our ALD module enables diverse agent tokens to activate on the same latent semantic regions. In this way, agent tokens can adapt to occlusions, different poses and different scales across images.

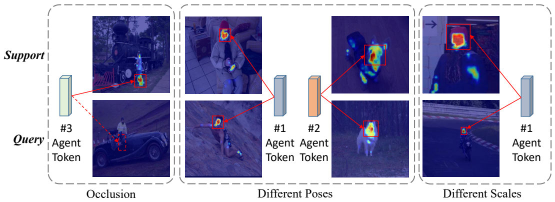

Fig. 8: Visualization of part correspondence. As we can see, the pair of part masks obtained from the same agent token have intrinsic semantic consistency.

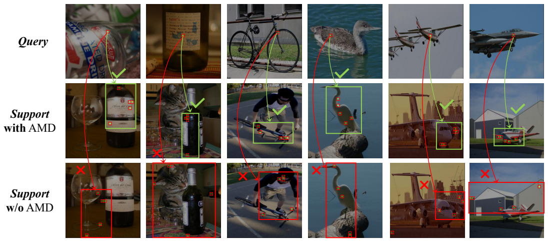

Fig. 9: Visualization of pixel correspondence. The green and red arrows point to the top five support pixels that match the query pixel with and without AMD module, respectively.

Visualization of Pixel Correspondence. To vividly present the effect of our AMD module, we visualize differences in pixel correspondences according to whether AMD exits. As shown in Fig.9, with the utilization of AMD module, the top five pixels corresponding to the query points tend to line in the foreground of support images. While these ones will contain large background noises without the AMD module to perform direct pixel matching. This is in line with the design idea of AMD module, i.e., filtering out the unreasonable matching weights.

5 Conclusion

In this paper, we propose a novel adaptive agent transformer (AAFormer) to integrate prototypical and affinity learning to exploit the complementarity between them via a transformer encoder-decoder architecture. Extensive experimental results on two standard benchmarks demonstrate that AAFormer performs favorably against state-of-the-art FSS methods.

Acknowledgments

This work was partially supported by the National Nature Science Foundation of China (Grant 62022078, 12150007, 62121002), and National Defense Basic Scientific Research Program (JCKY2020903B002).

References

1. Boudiaf, M., Kervadec, H., Masud, Z.I., Piantanida, P., Ben Ayed, I., Dolz, J.: Few-shot segmentation without meta-learning: A good transductive inference is all you need? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13979–13988 (2021)
2. Bousselham, W., Thibault, G., Pagano, L., Machireddy, A., Gray, J., Chang, Y.H., Song, X.: Efficient self-ensemble framework for semantic segmentation. arXiv preprint arXiv:2111.13280 (2021)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
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., 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)
6. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2613–2622 (2021)
7. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. arXiv preprint arXiv:2112.01527 (2021)
8. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. Advances in Neural Information Processing Systems **34** (2021)
9. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. Advances in neural information processing systems **26** (2013)
10. Dong, N., Xing, E.P.: Few-shot semantic segmentation with prototype learning. In: BMVC. vol. 3 (2018)
11. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
12. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: European conference on computer vision. pp. 297–312. Springer (2014)
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. Hu, H., Wei, F., Hu, H., Ye, Q., Cui, J., Wang, L.: Semi-supervised semantic segmentation via adaptive equalization learning. Advances in Neural Information Processing Systems **34** (2021)
15. Irving, B.: maskslic: regional superpixel generation with application to local pathology characterisation in medical images. arXiv preprint arXiv:1606.09518 (2016)
16. Koh, J.Y., Nguyen, D.T., Truong, Q.T., Yeung, S.K., Binder, A.: Sideinfnet: A deep neural network for semi-automatic semantic segmentation with side information. In: European Conference on Computer Vision. pp. 103–118. Springer (2020)

17. Li, G., Jampani, V., Sevilla-Lara, L., Sun, D., Kim, J., Kim, J.: Adaptive prototype learning and allocation for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8334–8343 (2021)
18. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
19. Liu, B., Ding, Y., Jiao, J., Ji, X., Ye, Q.: Anti-aliasing semantic reconstruction for few-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9747–9756 (2021)
20. Liu, Y., Zhang, X., Zhang, S., He, X.: Part-aware prototype network for few-shot semantic segmentation. In: European Conference on Computer Vision. pp. 142–158. Springer (2020)
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
22. Lu, Z., He, S., Zhu, X., Zhang, L., Song, Y.Z., Xiang, T.: Simpler is better: Few-shot semantic segmentation with classifier weight transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8741–8750 (2021)
23. Luo, W., Yang, M.: Semi-supervised semantic segmentation via strong-weak dual-branch network. In: European Conference on Computer Vision. pp. 784–800. Springer (2020)
24. Nguyen, K., Todorovic, S.: Feature weighting and boosting for few-shot segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 622–631 (2019)
25. 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)
26. Shaban, A., Bansal, S., Liu, Z., Essa, I., Boots, B.: One-shot learning for semantic segmentation. arXiv preprint arXiv:1709.03410 (2017)
27. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7262–7272 (2021)
28. Sun, R., Li, Y., Zhang, T., Mao, Z., Wu, F., Zhang, Y.: Lesion-aware transformers for diabetic retinopathy grading. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10938–10947 (2021)
29. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence (01), 1–1 (2020)
30. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
31. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)
32. Wang, H., Zhang, X., Hu, Y., Yang, Y., Cao, X., Zhen, X.: Few-shot semantic segmentation with democratic attention networks. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. pp. 730–746. Springer (2020)
33. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: Panet: Few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9197–9206 (2019)

34. Wu, Z., Shi, X., Lin, G., Cai, J.: Learning meta-class memory for few-shot semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 517–526 (2021)
35. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. Advances in Neural Information Processing Systems **34** (2021)
36. Xie, G.S., Liu, J., Xiong, H., Shao, L.: Scale-aware graph neural network for few-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5475–5484 (2021)
37. Xie, G.S., Xiong, H., Liu, J., Yao, Y., Shao, L.: Few-shot semantic segmentation with cyclic memory network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7293–7302 (2021)
38. Yang, B., Liu, C., Li, B., Jiao, J., Ye, Q.: Prototype mixture models for few-shot semantic segmentation. In: European Conference on Computer Vision. pp. 763–778. Springer (2020)
39. Zhang, B., Xiao, J., Qin, T.: Self-guided and cross-guided learning for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8312–8321 (2021)
40. Zhang, C., Lin, G., Liu, F., Guo, J., Wu, Q., Yao, R.: Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9587–9595 (2019)
41. Zhang, C., Lin, G., Liu, F., Yao, R., Shen, C.: Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5217–5226 (2019)
42. Zhang, G., Kang, G., Wei, Y., Yang, Y.: Few-shot segmentation via cycle-consistent transformer. arXiv preprint arXiv:2106.02320 (2021)
43. Zhang, X., Wei, Y., Yang, Y., Huang, T.S.: Sg-one: Similarity guidance network for one-shot semantic segmentation. IEEE Transactions on Cybernetics **50**(9), 3855–3865 (2020)
44. 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)
45. 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)