Supplementary Materials for 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

In the supplementary material, we first introduce more details about the support and query feature extracion. Then we elaborate the detailed procedure of the initial seeds selection and the detailed implementation of experiments. Finally, we show more qualitative results of our method.

1 More Details of Framework

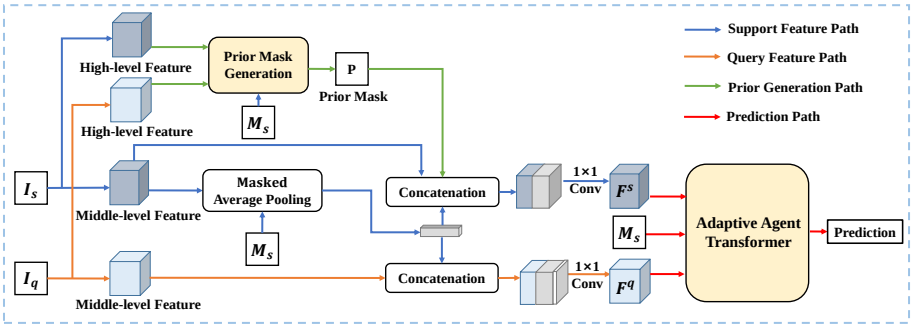

Fig. 1: The network architecture of our approach. We first extract the middle-level features of query and support images. Then the averaged foreground support feature is concatenated with both query and support feature, and the prior mask produced from high-level features is concatenated with the query feature. The flattened support and query features as well as the flattened support mask are fed into the AAFormer to obtain the final prediction.

The overall architecture of our approach is shown in Fig.(1). For backbone feature extractor, as proven in CANet [7] and PFENet [6], directly adopting high-level features that are more class-specific than middle-level features leads to performance degradation in few-shot segmentation. So we apply middle-level features for subsequent feature processing. Specifically, we feed the $\{I_s^k\}_{k=1}^K$ and

⋆ Equal contibution
⋆⋆ Corresponding author

Algorithm 1 The Initialization Process of Agent Tokens

Input: The set $\boldsymbol{X}$ representing the locations of foreground pixels, the set $\boldsymbol{L}$ containing both background pixel locations ($\boldsymbol{B}$) and labelled seed points ($\boldsymbol{P}$), and the number of agent tokens K;

Output: Initial agent tokens $\mathbf{F}^a = \{\mathbf{f}_k^a\}_{k=1}^K$;

1: Initializing the set $\boldsymbol{P} = \{\}$;

2: **for** k in $\{1, 2, \ldots, K\}$ **do**

3: Calculating the distance transform between locations $\boldsymbol{y} \in \boldsymbol{L}$ and a specific foreground location $\boldsymbol{x} \in \boldsymbol{X}$:

$$D(\boldsymbol{x}) = \min_{\boldsymbol{y} \in \boldsymbol{L}} \|\boldsymbol{x} - \boldsymbol{y}\|_2 ;$$

4: Selecting the furthest distance $\boldsymbol{p}^*$:

$$\boldsymbol{p}^* = \arg\max_{\boldsymbol{x}} D(\boldsymbol{x});$$

5: Updating $\boldsymbol{P} = \boldsymbol{P} \cup \{\boldsymbol{p}^*\}$, $\boldsymbol{L} = \boldsymbol{B} \cup \boldsymbol{P}$, and getting the support feature which can be seen as an initial agent token $\mathbf{f}_k^a$ at location $\boldsymbol{p}^*$;

6: **end for**

7: **return** $\mathbf{F}^a$

I_q to shared ResNet50 [3], then we apply a channel-wise concatenation of the features from `block-3` and `block-4`, followed by a 1×1 convolution layer to reduce the dimension to $h \times w \times d$, where the d is the hidden dimension which can be adjusted in experiments. The parameters of the backbone networks are kept unchanged during training for a better generalization of the model as proven in [7,6]. In order to correctly utilize high-level semantic clues, like [6], we compute the similarity between query pixel features $f^q \in F_h^q$ and foreground support pixel features $f^s \in F_h^s \odot M_s$, where the F_h^s and F_h^q are support features and query features from `block-5` of ResNet50 [3], respectively. Then we take the maximum similarity among all support pixels as the foreground probability of query features $c^q \in \mathbb{R}$ as:

$$c^q = \max_{s \in \{1,2,\ldots,hw\}} \left(\cos\left(f^q, f^s\right)\right), \tag{1}$$

where 'cos' denotes the cosine similarity. The similarity map will be normalized to the range of $[0, 1]$ using a min-max normalization then we obtain the prior mask $\mathbf{P}$ that tells the probability of pixels belonging to a target class. The prior mask is then concatenated with the query middle-level feature. We also extract the middle-level mask averaged support feature and then concatenate to both query and support feature for pixel-wise comparison, the concatenation of the features are processed by a 1x1 convolution to obtain the support feature map $\mathbf{F}^s \in \mathbb{R}^{h \times w \times c}$ and query feature map $\mathbf{F}^q \in \mathbb{R}^{h \times w \times c}$, where h, w denote the height, width of the feature map. The $\mathbf{F}^s$ and $\mathbf{F}^q$ as well as $\mathbf{M}_s$ are fed into the AAFormer to obtain the final prediction.

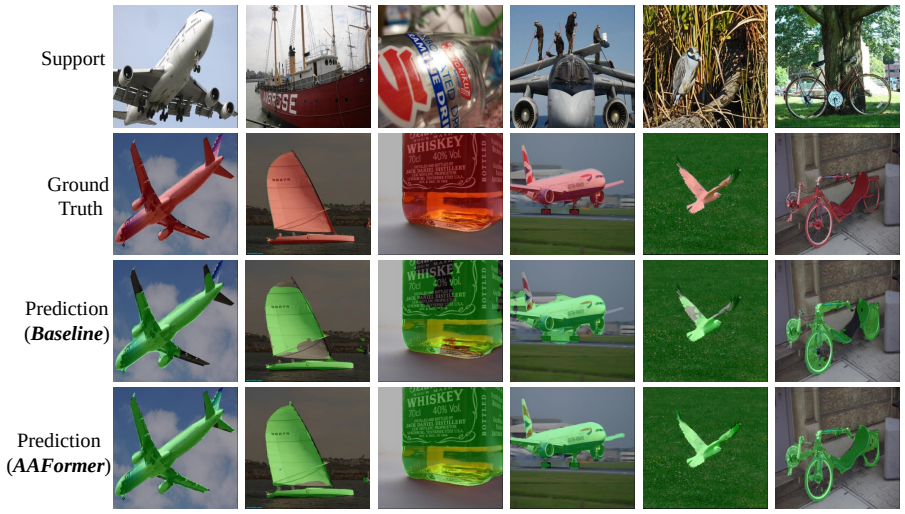

Fig. 2: More qualitative comparison with the baseline. AAFormer can achieve more accurate segmentation.

2 More Details of Initial Agent Tokens

In this section, as a sub-module that can make the agent learning decoder spatial-aware, we elaborate the complete initialization process of agent tokens benefiting from distance transform (DT). Formally, given the support mask, we can obtain the set $\boldsymbol{X}$ representing the locations of foreground pixels, and the set $\boldsymbol{L}$ containing background pixel locations ($\boldsymbol{B}$) and labelled seed points ($\boldsymbol{P}$), i.e., $\boldsymbol{L} = \boldsymbol{B} \cup \boldsymbol{P}$ and initially $\boldsymbol{P} = \{\}$. Then, an Euclidean distance transform is used to iteratively place seed points spaced at the maximum distance from the boundaries and any other seed points to obtain initial agent tokens $\mathbf{F}^a \in \mathbb{R}^{K \times c}$. The detailed process is described in Algorithm 1. In this way, the initialization of seed points which uniformly distributed in the foreground region will results in faster convergence.

3 More Details of Implementation

As data augmentation is significant to mitigate the over-fitting problems, we first resize and crop the input samples to 473×473, then rotate them randomly from $-10°$ to $10°$. We set the number of the cross layers in our agent learning decoder as 1 and as 2 in the agent matching decoder. The hidden dimension of MLP layer is set to three times of input hidden dimension and the input hidden dimension is set to 384 for all transformer blocks. The ϵ of sinkhorn algorithm is set to 0.05 and the default setting of the maximum iteration is 10. Dice loss [5] is adopted to train our model. The parameters of all the transformer blocks are optimized with AdamW optimizer [4] and the learning rate is set to 1×10^{-4} with

the weight decay 1×10^{-2}. We adopt the SGD optimizer with 'poly' policy [1] to decay the learning rate by multiplying $(1 - \frac{current_{iter}}{max_{iter}})^{poewr}$ to optimize the rest of the parameters, the *power* equals to 0.9 and we use the initial learning rate 2.5×10^{-3}, momentum 0.9 and weight decay 1×10^{-4}. Our experiments are conducted on four GeForce RTX GPUs.

4 More Visualizations

We show more qualitative results on Pascal-5^i [2]. We can observe that our baseline model fails to accurately segment the target objects when the background clutters occur in the support images or the appearance of target objects vary a lot. Differently, our method utilizes the agent tokens to inject the contextual information to pixel-level matching, which is more robust to noisy pixels thus yielding more precise segmentation.

References

1. 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)
2. 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)
3. 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)
4. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019)
5. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016)
6. 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)
7. 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)