Addressing Background Context Bias in Few-Shot Segmentation through Iterative Modulation

Lanyun Zhu[1]* Tianrun Chen[2] Jianxiong Yin[3] Simon See[3] Jun Liu[1]†
Singapore University of Technology and Design [1] Zhejiang University [2] NVIDIA AI Tech Centre [3]

Abstract

Existing few-shot segmentation methods usually extract foreground prototypes from support images to guide query image segmentation. However, different background contexts of support and query images can cause their foreground features to be misaligned. This phenomenon, known as background context bias, can hinder the effectiveness of support prototypes in guiding query image segmentation. In this work, we propose a novel framework with an iterative structure to address this problem. In each iteration of the framework, we first generate a query prediction based on a support foreground feature. Next, we extract background context from the query image to modulate the support foreground feature, thus eliminating the foreground feature misalignment caused by the different backgrounds. After that, we design a confidence-biased attention to eliminate noise and cleanse information. By integrating these components through an iterative structure, we create a novel network that can leverage the synergies between different modules to improve their performance in a mutually reinforcing manner. Through these carefully designed components and structures, our network can effectively eliminate background context bias in few-shot segmentation, thus achieving outstanding performance. We conduct extensive experiments on the PASCAL-5^i and COCO-20^i datasets and achieve state-of-the-art (SOTA) results, which demonstrate the effectiveness of our approach.

1. Introduction

Image segmentation [2–6, 10, 11, 24, 35, 39, 46, 49–51] is a crucial task in computer vision. In recent years, significant progresses have been made in this field, which are primarily attributed to the development of deep learning models [4, 25, 46, 50] trained on large-scale datasets [7, 47]. However, obtaining sufficient labeled data to train a segmentation model is time-consuming and labor-intensive, since it usually takes more than 10 minutes to annotate only one image for getting its ground truth label. To address this issue, Few-Shot Segmentation (FSS) has been proposed as an alternative solution, which aims to segment a class using a very small number of annotated images, thus reducing the need for costly data labeling.

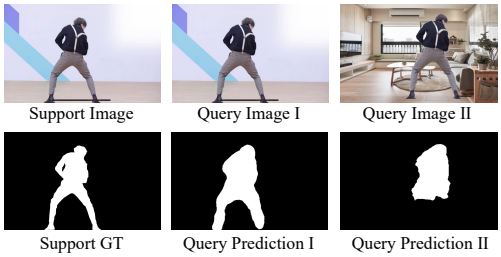

Figure 1. **An example of background context bias in few-shot segmentation.** When the query image shares the same background as the support image (Query Image I), the segmentation is high-quality; but when the query image has a different background (Query Image II), the segmentation is undesirable.

Currently, the prevailing methods for few-shot segmentation [1, 19, 23, 32, 34, 41, 48] typically use the meta-learning and episodic training strategies, in which the model is trained to segment a query image based on a few support images and their ground truth maps in the target class. To extract useful information that can represent the general properties of a class, these methods usually extract one [8, 16, 28] or a few [17, 41] prototypes from the foreground region of the support images. These prototypes are then used to segment query images based on feature concatenation [16, 17] or distance calculation [13, 26]. For this paradigm to be successful, it is necessary to assume that support and query images posses the similar or aligned foreground features. Only when this assumption holds true, the support prototypes can capture the query foreground properties accurately, thus enabling them to guide the segmentation process effectively.

However, we contend that this assumption may not always be true, since different backgrounds between support and query images may cause misalignment of their foreground features. Specifically, the prototypes for few-shot segmentation are typically derived from CNN or trans-

*Email: lanyun_zhu@mymail.sutd.edu.sg
†Corresponding Author

former features, which have a large receptive field that allows background context to be transmitted into foreground. As a result, different backgrounds can affect support and query foreground features differently, leading to feature misalignment that limits support prototypes' ability to guide query image segmentation. Fig. 1 shows an example of this problem. In our experiment, even though the support and query images contain the identical foreground object, the query image's segmentation results become undesired when the object is placed in different environments in support and query images. A similar problem of background context bias has been reported in other domains such as person ReID [33], but to the best of our knowledge, it has not been explicitly emphasized and addressed by researchers in the task of few-shot segmentation. Consequently, it remains a critical yet unresolved challenge that requires attention.

To mitigate the research gap, in this work, we propose a novel network for few-shot segmentation, which can effectively alleviate background context bias and demonstrate improved performance. Specifically, in our method, we employ query context to modulate support features. Through this modulation, query background information that influences the query foreground can be incorporated into the support prototypes, aligning them more closely with the query foreground features and therefore improving their ability to guide query image segmentation. To ensure the effectiveness of this modulation, we further investigate how to extract background context with stronger representation ability. Concretely, to ensure that the extracted context can adequately capture the influence of the background on the foreground, we model the input-to-output evolution of query foreground features within a deep network, and utilize this evolution as a basis for extracting context that is used in the modulation process. We also propose an information cleansing method to prevent noise from accumulating during the modulation process. By integrating these components through an iterative structure, we create a novel network that can leverage the synergies between different modules to improve their performance in a mutually reinforcing manner. Through these carefully designed components and structures, our network can effectively eliminate background context bias in few-shot segmentation, thus achieving outstanding performance as demonstrated by experiments.

We perform extensive experiments on common datasets including PASCAL-5^i and COCO-20^i and report state-of-the-art (SOTA) performance. Our contributions can be summarized as follows: (1) Firstly, we investigate the background context bias problem in FSS, which we find is a critical yet unresolved issue. (2) Secondly, we introduce an iterative approach to address context bias. Each iteration of our method involves a query prediction step for query segmentation, a support modulation step to enhance the guidance effectiveness of support features, and an information cleansing step to prevent the accumulation of noisy information. (3) Thirdly, our proposed approach achieves state-of-the-art (SOTA) performance on few-shot segmentation.

2. Related Work

Few-shot segmentation (FSS) aims to segment a class using a very small number of annotated images. As FSS requires fewer training data, it holds substantial value in practical applications, thus attracting considerable attention from researchers [1, 9, 15, 16, 19, 23, 29, 32, 34, 41]. Current methods typically use the meta-learning and episodic training strategies, in which one [8, 16, 28] or a few [17, 41] foreground prototypes are extracted from the support images and used to segment query images through feature concatenation [16, 17] or distance calculation [13, 26]. These methods, however, suffer from misalignment of foreground features caused by the different backgrounds. In this paper, we propose the first framework that can alleviate this problem through iterative modulation. In our method, we extract background context to modulate support foreground features. Some other FSS methods [8, 21, 41] also take background into consideration. However, they either employ background prototypes to eliminate background regions in query predictions [21], or segment interfering objects belonging to other categories in the background [41], without further considering how background context affects foreground features and the resultant issue of misaligned features. In few-shot segmentation, we design the first method to address this problem. [22, 44] also builds models using an iterative process, but their internal structures and functionalities are entirely different from our method. By using context modulation and information cleansing, we construct a completely new iterative model that is explicitly designed to deal with background context bias. In contrast, without problem-tailored designs, [22, 44] cannot address this problem, thus yielding worse results than our method. Biased attention is another technique related to our approach. In our method, biased attention is built based on confidence variations between two predictions, which is specifically designed to meet the requirements of our framework, enabling effective noise extraction and elimination. Due to this design, our method is different from other biased attention methods such as masked attention in [6, 22], which cannot eliminate noisy information.

3. Task Definition

FSS seeks to perform segmentation given only a small number of annotated images. The target is to train an FSS model on the training set $\mathcal{D}_{train}$ and evaluate it on the test set $\mathcal{D}_{test}$, where the two datasets are disjoint with respect to object classes. To achieve this, we follow previous works

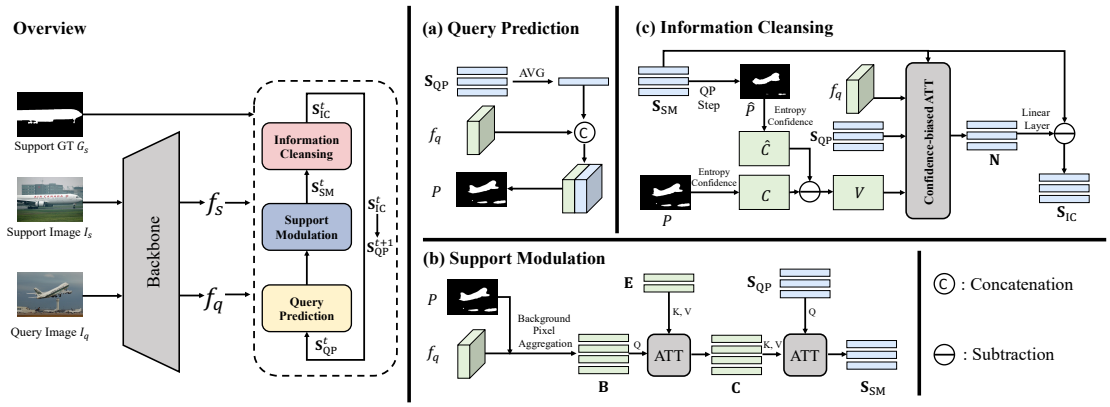

Figure 2. **Overall structure and different components of our network.** First, the backbone generates f_s and f_q for the support image and query image respectively. Next, an iterative structure is designed to fully utilize the features for segmentation, which consists of T iterations, with each iteration containing three successive steps: (a) Query Prediction, (b) Support Modulation, and (c) Information Cleansing. The output from the Information Cleansing step is input into the Query Prediction step of the subsequent iteration. Segmentation result from the Query Prediction step in the last iteration serves as the final prediction at the inference stage. The structure of the confidence-biased ATT in (c) is shown in Fig.4.

[8, 17] under the meta-learning setting and execute episodic training to optimize our FSS model. Specifically, each set is partitioned into multiple episodes, where each episode consists of a support set $\{I^k_s, G^k_s\}^K_{k=1}$ and a query set $\{I_q, G_q\}$, with $I^* \in \mathbb{R}^{H\times W\times 3}$ and $G^* \in \mathbb{R}^{H\times W}$ representing the image and the corresponding ground truth respectively. G^* is a binary mask indicating pixels within the target class, which, for convenience, are denoted as **foreground** in the subsequent sections, and other pixels outside the target class are denoted as **background**. During training, we iteratively sample an episodic from $\mathcal{D}_{train}$ to train a model that predicts G_q based on $\{I^k_s, G^k_s\}^K_{k=1}$ and I_q. Once the training is completed, the model is evaluated on $\mathcal{D}_{test}$. For the convenience of introduction, in the following sections, we describe our method under the 1-shot setting where only one support image is available (K=1). In Supp, we elaborate on how to extend the method to the K-shot setting.

4. Method

4.1. Overview

Fig. 2 provides an overview of our method. First, a backbone network produces features f_s and f_q for support and query images respectively. Next, an iterative structure is designed to fully utilize the features for segmentation, which consists of T iterations, with each iteration containing three successive steps: Query Prediction (QP), Support Modulation (SM), and Information Cleansing (IC). Considering the t-th iteration of the structure, the operation is performed as follows. Firstly, in the QP step, query images are segmented under the guidance of a support foreground feature $\mathbf{S}^t_{\mathrm{QP}}$. Then, in the SM step, query context is extracted and used to enhance $\mathbf{S}^t_{\mathrm{QP}}$'s effectiveness for guiding query segmentation. The enhanced $\mathbf{S}^t_{\mathrm{SM}}$ is obtained in this step. After that, in the IC step, a biased attention adjusts $\mathbf{S}^t_{\mathrm{SM}}$ from the SM step to cleanse information, getting the processed $\mathbf{S}^t_{\mathrm{IC}}$. The three steps are performed iteratively, where the updated $\mathbf{S}^t_{\mathrm{IC}}$ from the previous iteration guides the QP step in the subsequent iteration ($\mathbf{S}^{t+1}_{\mathrm{QP}} \leftarrow \mathbf{S}^t_{\mathrm{IC}}$). Query segmentation result from the last iteration serves as the final prediction at the inference stage.

The motivation to design an iterative manner is based on our observation that each of the three steps can have an influence on one another, so by successively updating the feature at each step, the network can be forced to refine itself towards an optimal solution in a recurrent optimization manner. Specifically, in every iteration, an improved support foreground feature $\mathbf{S}^t_{\mathrm{QP}}$ ($\mathbf{S}^{t-1}_{\mathrm{IC}}$), which is modulated from the SM and IC steps of the previous iteration, can guide the generation of a more accurate query prediction in the QP step. By using this query prediction with higher accuracy, the subsequent SM and IC steps can perform better, thus resulting in an improved $\mathbf{S}^t_{\mathrm{IC}}$. $\mathbf{S}^t_{\mathrm{IC}}$ is further utilized for query prediction in the following iteration ($\mathbf{S}^{t+1}_{\mathrm{QP}} \leftarrow \mathbf{S}^t_{\mathrm{IC}}$). In this way, a recurrent optimization scheme is created that can utilize the iterative structure to continuously refine the query prediction results. In the subsequent sections, we describe each step of our method in detail.

4.2. Query Prediction Step

As shown in Fig. 2 (a), in this step, query images are segmented under the guidance of a support foreground feature $\mathbf{S}_{\mathrm{QP}}$. In the first iteration, $\mathbf{S}_{\mathrm{QP}}$ is initialized using the support backbone features f_s. Specifically, each foreground pixel's feature in f_s is treated as a token, and these tokens are aggregated to obtain $\mathbf{S}_{\mathrm{QP}}$. As a combination of features from all foreground pixels, $\mathbf{S}_{\mathrm{QP}}$ can reflect the general properties of support foreground, making it possible to

guide the segmentation of query foreground belonging to the same category. For the remaining iterations, $\mathbf{S}_{\mathrm{QP}}$ is updated as the output from the IC step of the previous iteration. Given $\mathbf{S}_{\mathrm{QP}}$ and the query backbone features f_q, query segmentation is carried out with the process as follows:

$$P = \phi_p\left(\mathrm{CAT}\left(\mathrm{AVG}\left(\mathbf{S}_{\mathrm{QP}}\right), f_q\right)\right), \tag{1}$$

where AVG denotes the average over all tokens in $\mathbf{S}_{\mathrm{QP}}$, CAT refers to the channel-wise concatenation, ϕ_p is a two-layered 1×1 convolutions that generates the prediction P.

4.3. Support Modulation Step

As a result of background context bias between support and query images, their foreground features are misaligned, which makes the QP step alone insufficient to ensure accurate query image segmentation. To address this problem, we propose a support modulation (SM) step, which uses background context from the query image to modulate $\mathbf{S}_{\mathrm{QP}}$, thereby minimizing the feature gap and increasing its effectiveness in guiding query image segmentation. Specifically, we denote the number of foreground pixels and background pixels in the query image by N_f and N_b, and as shown in Fig. 2 (b), we implement the SM step as follows. Firstly, we extract an evolution feature $\mathbf{E} \in \mathbb{R}^{N_f \times C}$ that reflects how the query foreground representation changes from input to output layers within the backbone network (details are illustrated below). Subsequently, we concatenate the features of all background pixels[1] in f_q to generate a background representation $\mathbf{B} \in \mathbb{R}^{N_b \times C}$. By correlating $\mathbf{E}$ with $\mathbf{B}$, we then capture background context $\mathbf{C}$ by:

$$\mathbf{C} = \mathrm{ATT}\left(\mathbf{Q}_{\mathbf{B}}, \mathbf{K}_{\mathbf{E}}, \mathbf{V}_{\mathbf{E}}\right), \tag{2}$$

where ATT is a cross-attention module in a QKV manner. Finally, we use query context to modulate $\mathbf{S}_{\mathrm{QP}}$ through another cross-attention:

$$\mathbf{S}_{\mathrm{SM}} = \mathbf{S}_{\mathrm{QP}} + \mathrm{ATT}\left(\mathbf{Q}_{\mathbf{S}_{\mathrm{QP}}}, \mathbf{K}_{\mathbf{C}}, \mathbf{V}_{\mathbf{C}}\right). \tag{3}$$

The resulted $\mathbf{S}_{\mathrm{SM}}$ is the output of the SM step. This step includes an innovative design that extracts background context via the evolution feature $\mathbf{E}$, which has not been used previously but has demonstrated success in our experiments. $\mathbf{E}$ is a feature that describes how foreground features change from the input layer to the output layer. Using it is motivated by the analysis of network inputs and outputs. Specifically, in a deep neural network, the input image is context-independent, so its foreground contains no background information; on the other hand, the output features are heavily influenced by long-range context due to the network's high-receptive-field property, so its foreground contains substantial background context. Consequently, by

[1]To generate $\mathbf{B}$, the background region of the query image is estimated from the prediction P of the QP step.

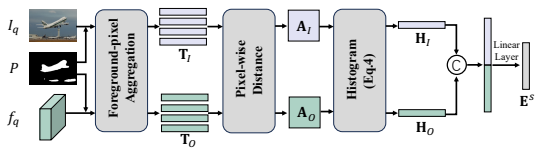

Figure 3. Generation of the structure-wise evolution feature $\mathbf{E}^s$.

modeling the change from input to output, $\mathbf{E}$ can capture the influence of the background on the foreground within the network, thus enabling it to perform modulation effectively. To capture rich and multi-level information for the effective context extraction, we sum two types of evolution features to obtain $\mathbf{E}$, namely pixel-wise evolution feature $\mathbf{E}^p$ and structure-wise evolution feature $\mathbf{E}^s$ as follow:

Pixel-wise Evolution Feature. First, we extract $\mathbf{E}^p$, which provides the evolution information for each pixel in the query foreground. For this, we pass the input query image and its backbone output features through two individual 1×1 convolutions. Through this process, we get a context-independent input representation F_I and a context-influenced output representation F_O, respectively. After that, by identifying the query foreground region using P generated by the QP step, we generate foreground tokens $\mathbf{F}_I \in \mathbb{R}^{N_f \times C}$ and $\mathbf{F}_O \in \mathbb{R}^{N_f \times C}$, which aggregate the features of all foreground pixels in F_I and F_O, respectively. $\mathbf{F}_I$ and $\mathbf{F}_O$ are concatenated along the channel dimension and are passed through a two-layered MLP to produce a feature matrix $\mathbf{E}^p \in \mathbb{R}^{N_f \times C}$. In this way, by modeling the interaction between the context-independent input image and the context-influenced output feature, $\mathbf{E}^p$ extracts pixel-wise evolution features that can be used to model query context.

Structure-wise Evolution Feature. As shown in Fig. 3, in addition to $\mathbf{E}^p$, motivated by the demonstrated importance of structural information in the segmentation task [14, 18], we further introduce a Structure-wise Evolution Feature denoted by $\mathbf{E}^s$. Firstly, We aggregate all foreground pixels from the query image I_q and its backbone output f_q, getting input tokens $\mathbf{T}_I \in \mathbb{R}^{N_f \times 3}$ and output tokens $\mathbf{T}_O \in \mathbb{R}^{N_f \times C}$ respectively. Next, we calculate pixel-wise distances in $\mathbf{T}_I$ and $\mathbf{T}_O$ to get affinity maps $\mathbf{A}_I \in \mathbb{R}^{N_f \times N_f}$ and $\mathbf{A}_O \in \mathbb{R}^{N_f \times N_f}$. Specifically, each item $\mathbf{A}_I^{i,j}$ on $\mathbf{A}_I$ is computed as the cosine similarity between $\mathbf{T}_I^i$ and $\mathbf{T}_I^j$, where $\mathbf{T}_I^i$ refers to the i-th pixel on $\mathbf{T}_I$. $\mathbf{A}_O$ is produced from $\mathbf{T}_O$ similarly. We flatten $\mathbf{A}_I$ and $\mathbf{A}_O$ to the shape $\mathbb{R}^{N_f^2}$, and then introduce histogram features $\mathbf{H}_I \in \mathbb{R}^L$ and $\mathbf{H}_O \in \mathbb{R}^L$ to capture statistical information from them. For this, the continuous range [0,1] is subdivided into L discrete bins $\{I_l\}_{l=1}^L$ with $I_l = [(l-1)/L, l/L]$, then each dimension $\mathbf{H}_I^l$ on $\mathbf{H}_I$ is calculated as the total number of dimensions in $\mathbf{A}_I$ with values falling into the interval I_l.

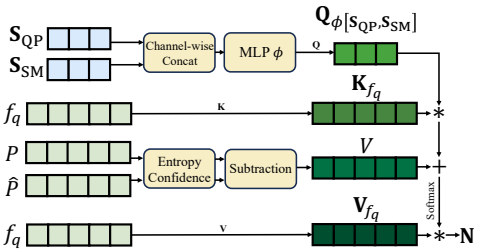

Figure 4. Illustration figure of the confidence-biased attention used in the Information Cleansing Step. P, $\hat{P}$ and f_q are flattened along the spatial dimension before the attention.

Formally,

$$\mathbf{H}_I^l = \sum_{i=1}^{N_f^2} \mathbb{1}\left(\frac{l-1}{L} < \mathbf{A}_I^i < \frac{l}{L}\right),\ l \in [0, L-1]. \quad (4)$$

For $\mathbb{1}$ to be differentiable, we use a spire-shaped judge function, which is described in Supp. Using the same approach, we can get $\mathbf{H}_O$ from $\mathbf{A}_O$. Finally, we normalize and concatenate $\mathbf{H}_I$ and $\mathbf{H}_O$, followed by a two-layered MLP to get the Structure-wise Evolution Feature $\mathbf{E}^s \in \mathbb{R}^C$.

In the end, $\mathbf{E} \in \mathbb{R}^{N_f \times C}$ is generated by adding $\mathbf{E}^s \in \mathbb{R}^C$ to each pixel of the pixel-wise evolution feature $\mathbf{E}^p \in \mathbb{R}^{N_f \times C}$, which can then be used to modulate $\mathbf{S}$ through Eq. 2 and Eq. 3.

Discussion: Why to Use Affinity Maps and Histograms. To generate $\mathbf{E}^s$, we employ affinity maps followed by histograms to extract structural information. Using affinity maps is motivated by their strong ability to represent image structures [16, 20]. Histograms are used for two reasons. Firstly, histograms can capture structural information like contrast and smoothness [12], so they are helpful for representing the structure features of the foreground region at a particular network layer. Secondly, histograms facilitate the extraction of features from affinity maps $\mathbf{A}_I \in \mathbb{R}^{N_f \times N_f}$ and $\mathbf{A}_O \in \mathbb{R}^{N_f \times N_f}$, which are unfixed in size due to the variable number of foreground pixels (N_f) in different images. Using histograms, $\mathbf{A}_I$ and $\mathbf{A}_O$ can be converted into fixed-shaped representations, thus making it possible to extract evolution features from them through linear layers that require a fixed number of input and output channels.

4.4. Information Cleansing Step

To extract context, in the SM step, we use the prediction P from the QP step as a foreground-background indicator to create features like $\mathbf{B}$ in Eq.2. However, P is a coarse prediction with some incorrectly segmented pixels, so using it directly can introduce noise into these features. Consequently, through the processes of the SM step, this noise can be propagated to the output $\mathbf{S}_{\mathrm{SM}}$, finally impeding its effectiveness in guiding query prediction. To overcome this issue, we propose an additional step called Information Cleansing (IC), which removes the noise from $\mathbf{S}_{\mathrm{SM}}$, thus producing a cleaner $\mathbf{S}_{\mathrm{IC}}$ that can guide query image segmentation more effectively. The structure of this step is shown in Fig. 2 (c). Firstly, we introduce a confidence-biased attention as shown in Fig.4 to capture the accumulated noisy information, which is inspired by recent semi-supervised learning study [38] demonstrating that noisy information can lead to lower prediction certainty. Specifically, we replace $\mathbf{S}_{\mathrm{QP}}$ in Eq. 1 with $\mathbf{S}_{\mathrm{SM}}$ and perform the QP step. By doing so, in addition to prediction P from $\mathbf{S}_{\mathrm{QP}}$, we get an intermediate prediction $\hat{P}$ from $\mathbf{S}_{\mathrm{SM}}$. Next, we calculate the entropy confidence for each pixel on P and $\hat{P}$, resulting in confidence maps C and $\hat{C}$ with the same shape as P and $\hat{P}$. A confidence variance map V is then calculated as the difference between C and $\hat{C}$, i.e., $V = \hat{C} - C$. We use V as a bias term, adding it to the softmax matrix of a vanilla attention to derive a modified attention that is formulated as:

$$\mathbf{N} = \mathrm{softmax}\left(\mathbf{Q}_{\phi[\mathbf{S}_{\mathrm{QP}},\mathbf{S}_{\mathrm{SM}}]}\mathbf{K}_{f_q} + V\right)\mathbf{V}_{f_q}, \quad (5)$$

where $\phi[\mathbf{S}_{\mathrm{QP}}, \mathbf{S}_{\mathrm{SM}}]$ denotes passing the concatenation of $\mathbf{S}_{\mathrm{QP}}$ and $\mathbf{S}_{\mathrm{SM}}$ through a two-layered MLP. The feature produced by this operation reflects the evolution from $\mathbf{S}_{\mathrm{QP}}$ to $\mathbf{S}_{\mathrm{SM}}$ through the SM step. V and f_q are flatten along the spatial dimension before the attention. For a pixel (i, j) on V, a higher value of $V^{i,j}$ indicates a sharper decline in its prediction confidence, and vice versa. By adding V to the softmax matrix, the attention is encouraged to focus on pixels with reduced confidences. In this way, the attention can capture noisy information $\mathbf{N}$ accumulated by the SM step. Eventually, we remove the noisy information from $\mathbf{S}_{\mathrm{SM}}$ by:

$$\mathbf{S}_{\mathrm{IC}} = \mathbf{S}_{\mathrm{SM}} - \phi\left(\mathbf{N}\right), \quad (6)$$

where ϕ is a linear layer. The generated $\mathbf{S}_{\mathrm{IC}}$ is the output of the IC step, and also the input for the QP step in the next iteration ($\mathbf{S}_{\mathrm{QP}}^{t+1} \leftarrow \mathbf{S}_{\mathrm{IC}}^{t}$).

4.5. Optimization

We optimize the model with the following loss function:

$$\mathcal{L} = \sum_{t=1}^{T} L_{CE}\left(P_t, G_q\right) + \lambda \sum_{t=1}^{T-1} L_{CE}\left(\hat{P}_t, G_q\right), \quad (7)$$

where T refers to the number of iterations. G_q is the ground truth of the query image. P_t denotes the query prediction from the QP step in the t-th iteration. $\hat{P}_t$ denotes the intermediate prediction from the IC step in the t-th iteration. L_{CE} refers to the cross-entropy loss function. λ is a hyper-parameter.

5. Experiments

5.1. Datasets

We evaluate our method on two widely-used datasets: PASCAL-5^i and COCO-20^i. The PASCAL-5^i dataset contains images from PASCAL VOC 2012, with annotations

Backbone	Method	Conference	1-shot					5-shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
ResNet50	NTRENet [21]	CVPR2022	65.4	72.3	59.4	59.8	63.2	66.2	72.8	61.7	62.2	65.7
	BAM[16]	CVPR2022	69.0	73.6	67.5	61.1	67.8	70.6	75.1	70.8	67.2	70.9
	AAFormer[37]	ECCV2022	69.1	73.3	59.1	59.2	65.2	72.5	74.7	62.0	61.3	67.6
	SSP[8]	ECCV2022	60.5	67.8	66.4	51.0	61.4	67.5	72.3	**75.2**	62.1	69.3
	IPMT[22]	NeurIPS2022	72.8	73.7	59.2	61.6	66.8	73.1	74.7	61.6	63.4	68.2
	ABCNet[36]	CVPR2023	68.8	73.4	62.3	59.5	66.0	71.7	74.2	65.4	67.0	69.6
	HDMNet [30]	CVPR2023	71.0	75.4	68.9	62.1	69.4	71.3	76.2	71.3	68.5	71.8
	MIANet[42]	CVPR2023	68.5	75.8	67.5	63.2	68.7	70.2	77.4	70.0	**68.8**	71.7
	MSI[27]	ICCV2023	71.0	72.5	63.8	**65.9**	68.5	73.0	74.2	70.5	66.6	71.1
	SCCAN[40]	ICCV2023	68.3	72.5	66.8	59.8	66.8	72.3	74.1	69.1	65.6	70.3
	ABCB (Ours)	CVPR2024	**72.9**	**76.0**	**69.5**	64.0	**70.6**	**74.4**	**78.0**	73.9	68.3	**73.6**
ResNet101	NTRENet[21]	CVPR2022	65.5	71.8	59.1	58.3	63.7	67.9	73.2	60.1	66.8	67.0
	DCAMA[31]	ECCV2022	62.5	70.8	64.5	56.4	63.5	70.0	73.8	66.8	65.0	68.9
	VAT[13]	ECCV2022	68.1	71.7	64.8	63.3	67.0	72.6	74.1	69.5	69.5	71.4
	ABCNet[36]	CVPR2023	65.3	72.9	65.0	59.3	65.6	71.4	75.0	68.2	63.1	69.4
	MSI[27]	ICCV2023	**73.1**	73.9	64.7	68.8	70.1	73.6	76.1	68.0	71.3	72.2
	SCCAN[40]	ICCV2023	70.9	73.9	66.8	61.7	68.3	73.1	76.4	70.3	66.1	71.5
	ABCB (Ours)	CVPR2024	73.0	**76.0**	**69.7**	**69.2**	**72.0**	**74.8**	**78.5**	**73.6**	**72.6**	**74.9**

Table 1. Performance comparison with other methods on PASCAL-5^i.

Backbone	Method	Conference	1-shot					5-shot				
			Fold-0	Fold-1	Fold-2	Fold-3	Mean	Fold-0	Fold-1	Fold-2	Fold-3	Mean
ResNet50	NTRENet[21]	CVPR2022	36.8	42.6	39.9	37.9	39.3	38.2	44.1	40.4	38.4	40.3
	BAM[16]	CVPR2022	43.4	50.6	47.5	43.4	46.2	49.3	54.2	51.6	49.6	51.2
	SSP[8]	ECCV2022	35.5	39.6	37.9	36.7	47.4	40.6	47.0	45.1	43.9	44.1
	MM-Former[45]	NeurIPS2022	40.5	47.7	45.2	43.3	44.2	44.0	52.4	47.4	50.0	48.4
	ABCNet[36]	CVPR2023	42.3	46.2	46.0	42.0	44.1	45.5	51.7	52.6	46.4	49.1
	MIANet[42]	CVPR2023	42.5	53.0	47.8	47.4	47.7	45.8	58.2	51.3	51.9	51.7
	MSI[27]	ICCV2023	42.4	49.2	49.4	46.1	46.8	47.1	54.9	54.1	51.9	52.0
	SCCAN[40]	ICCV2023	40.4	49.7	49.6	45.6	46.3	47.2	57.2	**59.2**	52.1	53.9
	ABCB (Ours)	CVPR2024	**44.2**	**54.0**	**52.1**	**49.8**	**50.0**	**50.5**	**59.1**	57.0	**53.6**	**55.1**
ResNet101	NTRENet[21]	CVPR2022	38.3	40.4	39.5	38.1	39.1	42.3	44.4	44.2	41.7	43.2
	SSP[8]	ECCV2022	39.1	45.1	42.7	41.2	42.0	47.4	54.5	50.4	49.6	50.2
	IPMT[22]	NeurIPS2022	40.5	45.7	44.8	39.3	42.6	45.1	50.3	49.3	46.8	47.9
	ABCNet[36]	CVPR2023	36.5	35.7	34.7	31.4	34.6	40.1	40.1	39.0	35.9	38.8
	MSI[27]	ICCV2023	44.8	54.2	52.3	48.0	49.8	49.3	58.0	56.1	52.7	54.0
	SCCAN[40]	ICCV2023	42.6	51.4	50.0	48.8	48.2	49.4	61.7	61.9	55.0	57.0
	ABCB (Ours)	CVPR2024	**46.0**	**56.3**	**54.3**	**51.3**	**51.5**	**51.6**	**63.5**	**62.8**	**57.2**	**58.8**

Table 2. Performance comparison with other methods on COCO-20^i.

Method	mIoU
Baseline (Backbone + QP)	61.48
Baseline + SM	68.20
Baseline + SM + IC	72.92

Table 3. Ablation of different components in our method.

Method	mIoU
Ours	72.92
Ours w/o $\mathbf{E}^p$	68.85
Ours w/o $\mathbf{E}^s$	70.29

Table 4. Ablation of $\mathbf{E}^p$ and $\mathbf{E}^s$ used in support modulation step

Number of Iterations (T)	mIoU	MACs (G)
1	61.48	225.8
2	69.82	241.3
3	72.92	257.0
4	72.99	272.7
5	73.05	288.2

Table 5. Ablation for the number of iterations.

Iteration (t)	mIoU
1	63.08
2	70.76
3	72.92

Table 6. The mIoUs of predictions from different iterations.

extended from the SDS to include 20 categories. The COCO-20^i dataset is based on the MSCOCO dataset and includes 80 categories. To be consistent with previous works, we divided the overall categories into four folds and conduct experiments in a cross-validation manner, i.e., to use three folds for training and the remaining one for testing.

5.2. Implementation Details

We employ ResNet50 and ResNet101 pretrained on ImageNet as the network backbone with the structure in [26] to enhance feature effectiveness, followed a correlation modules in [30] to fuse information. L discrete bins is used to generate histograms for the structure-wise evolution feature, where L is set to 16. T indicating the number of iterations for the iterative structure is 3. The model is trained for 250 epochs on PASCAL-5^i and 70 epochs on COCO-20^i. λ in Eq. 7 is set to 0.2. We use SGD as the optimizer with momentum and weight decay set to 0.9 and 0.0001, respectively. We set the initial learning rate to 0.002 and batch size to 16 for PASCAL-5^i, and 0.005 with batch size 8 for COCO-20^i. We adopt 'poly' policy as the learning rate decay strategy, where the learning rate for each iteration equals to initial rate multiplied by $\left(1 - \frac{iter}{max\_iter}\right)^{0.9}$. For the input images, we employ random scaling and horizontal flipping for data augmentation, and then crop images to the size 473×473 for PASCAL-5^i and 641×641 for COCO-20^i to get the training samples. The experiments are implemented using Pytorch on NVIDIA Tesla V100 GPUs.

5.3. Comparison to State-of-the-art

To evaluate the effectiveness of our method, we compare it with other state-of-the-art methods under different backbones, including ResNet50 and ResNet101, and on a variety of few-shot settings, including 1-shot and 5-shot. For each setting, we report the results of using different folds as the test set as well as their mean result. All the compared methods are published in the past 2 years.

The results on PASCAL-5^i are shown in Table. 1. Under both 1-shot and 5-shot settings, our method can significantly outperform existing methods for both ResNet50 and ResNet101 backbones. To be specific, for the ResNet101 backbone, our method achieves 72.0% and 74.9% mIoUs on the 1-shot and 5-shot settings, outperforming the second-place method by 1.9% and 2.7%, respectively. The results demonstrate that our method can work well and achieve outstanding performance for both 1-shot and multi-shot segmentation. It is worth noting that some of the compared methods [8, 16, 40] also make use of image background information. Our method differs from them by addressing the problem of background context bias for the first time, thus

achieving the best performance.

The results on COCO-20^i are shown in Table. 2. Based on ResNet101, our approach performs better than the second-place method by 1.7% and 1.8%, reaching 51.5% and 58.8% mIoUs on the 1-shot and 5-shot settings respectively. Compared to PASCAL-5^i, COCO-20^i is more challenging due to its more complicated backgrounds. Despite these challenges, our method still shows significant advantages benefiting from the context-based modulation, as demonstrated by the excellent results in various settings.

5.4. Ablation Study

We conduct several ablation studies to verify the effectiveness of our designs. The experiments in this section are performed on PASCAL-5^i fold-0 with the ResNet50 backbone. Due to the paper length limitation, more ablation study results are presented in **Supp**.

Ablation of Different Components. Our method is structured as an iterative process with three successive steps: Query Prediction (QP), Support Modulation (SM), and Information Cleansing (IC). We conduct experiments to evaluate the effectiveness of each component, and the results are shown in Table. 3. A network only comprising a backbone and a QP step is used as the baseline, which achieves 61.48% mIoU. Incorporating the SM step on the baseline results in a performance boost, achieving a mIoU of 68.20%, 6.72% higher than the baseline. The further usage of IC step improves mIoU to 72.92%. The results demonstrate that each component can contribute to performance improvement in our method.

Ablation of Support Modulation step. To improve the effectiveness of support foreground features, we use context information extracted from query images for modulation. To capture a comprehensive context, we generate two query evolution features: pixel-wise evolution feature $\mathbf{E}^p$ and structure-wise evolution feature $\mathbf{E}^s$, which capture the evolution of pixels and global structures respectively. To verify the necessity of these representations, we conduct experiments and present the results in Table 4. By removing $\mathbf{E}^p$ and $\mathbf{E}^s$, performance is decreased by 4.07% and 2.63% respectively. These results suggest that both features are important to capture a comprehensive evolution feature, which is crucial for ensuring an effective context extraction.

Ablation of Number of Iterations. An iterative structure is employed in our method with T iterations. We perform experiments to determine the best choice for the hyperparameter T and present the results in Table 5, which shows the mIoUs and MACs for different settings. As shown in the table, the mIoU improves from 61.48% to 72.92% as T increases from 1 to 3. When T is higher than 3, further increasing T does not significantly improve performance, but rather increases computation burden. Therefore, we choose $T = 3$ as the optimal number of iterations.

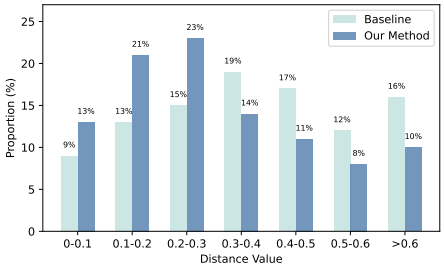

Figure 5. Statistical distribution of distances between the support foreground features and query foreground features across all episodes. Baseline refers to backbone with a single QP step.

5.5. Predictions at Different Iterations

To gain a deeper evaluation of our iterative structure, we analyze the segmentation results from different iterations. The total number of iterations T is set to 3. The results are presented in Table.5, which shows the mIoUs of predictions generated at different iterations (t=1,2,3). As t increases, the segmentation results continue to improve. In the first iteration, we obtain suboptimal predictions because the feature misalignment is caused by the different backgrounds, which results in the low mIoU scores. By using the iterative structure, the performance is improved significantly, resulting in a 9.84% increase in mIoU from the first to the final iteration.

5.6. Mitigation of the Feature Misalignment

To demonstrate that our method can eliminate foreground feature misalignment caused by background context bias, for all episodes of the test set, we compute the cosine distance between foreground average features of support and query images. This evaluation is conducted on both the baseline and our method for comparison. The distribution of these distance values is shown in Fig.5, where the horizontal axis represents different distance intervals, and the vertical axis indicates the proportion of episodes falling into each interval. Compared to the baseline, when our method is used, there are more episodes with lower foreground feature distances. This shows that our method can effectively narrow foreground feature distances between support and query images, thereby enhancing the effectiveness of support foreground features in guiding query segmentation.

5.7. Visualizations

In order to illustrate the effectiveness and advantages of our method, in Fig.6 and Fig.7, we present the following two types of visualizations:

Visualization Comparisons with SOTA methods. In Fig.6, we provide prediction visualizations of different methods when the support and query images have significantly different backgrounds. The compared methods include CANet [43], SSP [8], and IPMT [22]. Under this challenging scenario, all the compared methods encounter the background context bias issue, which results in unsatisfactory predictions. In contrast, our method effectively

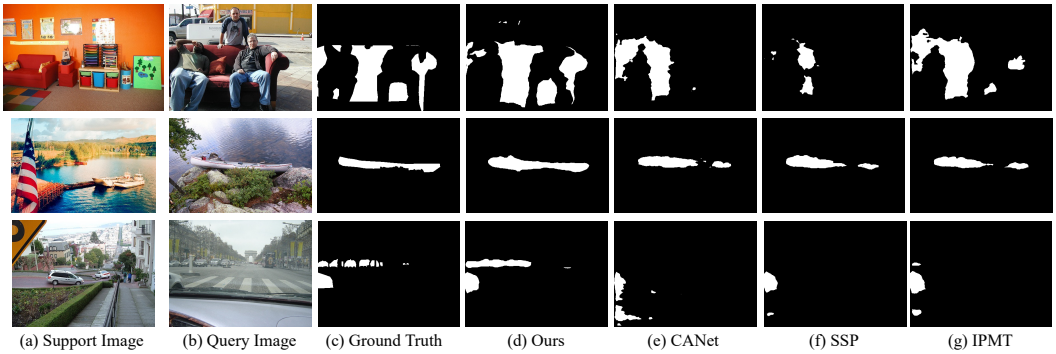

Figure 6. Prediction visualizations of different methods when the support and query images have significantly different backgrounds.

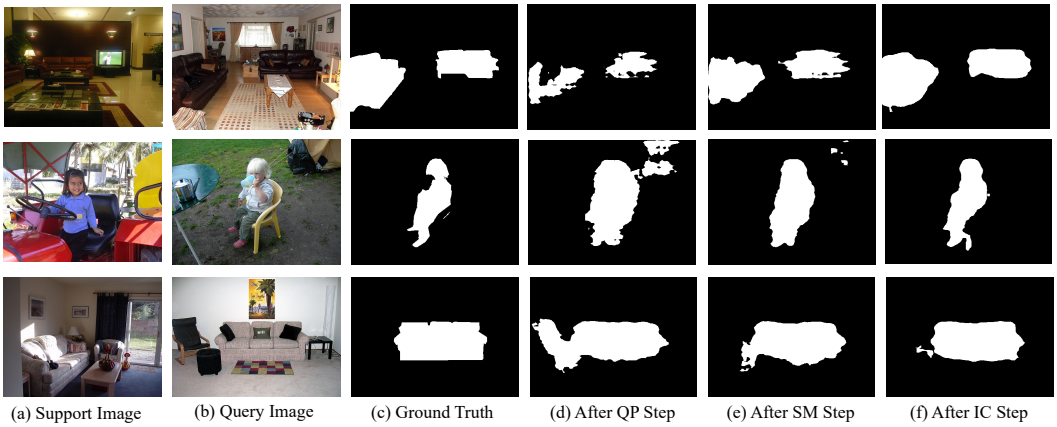

Figure 7. Prediction visualizations by using the output features after each of the QP, SM, and IC steps in an iteration for query guidance.

addresses this issue and produces significantly better predictions. These results demonstrate that our method can achieve robust results when support and query images have different backgrounds and outperforms the other methods.

Prediction Visualizations of Using Features After Each Step. In Fig.7, we present the prediction results by using the support features $\mathbf{S}_{\mathrm{QP}}$, $\mathbf{S}_{\mathrm{SM}}$ and $\mathbf{S}_{\mathrm{IC}}$ after each of the QP, SM, and IC steps in an iteration as the guidance feature used in Eq.1. It can be observed that the prediction gradually refines after each step, which demonstrates the benefit of using each step proposed in our method.

5.8. Computation Cost and Parameter Number

Our method is both effective and efficient. Compared to the baseline, which comprises a backbone network followed by a single QP step, our methods only increase computation and memory usage slightly. Specifically, on the PASCAL-5^i dataset with the ResNet50 backbone, the baseline consumes 210.3G MACs of computation, requiring 27.6M parameters. Our method consumes 257.0G MACs of computation, requiring 33.5M parameters. Compared to the baseline, our method significantly improves mIoU (+16.50%) while only increasing computation and memory by 22.2% and 21.4%, respectively. We also calculate the parameter usage of each step in our method. Specifically, the QP step, SM step, and IC step require 0.6M, 3.9M, and 2.0M parameters, respectively. It is worth noting that different iterations share the same parameters, thus avoiding the huge memory cost caused by the iterative structure.

6. Conclusion

This paper presents a novel method to address background context bias in few-shot segmentation. In our method, we employ an iterative structure involving three successive steps: Query Prediction, Support Modulation, and Information Cleansing. This structure can address the misalignment of foreground features and reduce the noise accumulation, creating a recurrent optimization scheme that can continuously refine the segmentation results. Experiments on PASCAL-5^i and COCO-20^i demonstrate the high effectiveness of our method in achieving SOTA segmentation results. We believe that our research provides valuable insights and advancements in the field of few-shot segmentation, which can contribute to the development of segmentation algorithms with higher accuracy and robustness.

Acknowledgement This research is supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-PhD-2021-08-006), MOE AcRF Tier 2 projects MOE-T2EP20222-0009 and MOE-T2EP20123-0014.

References

[1] Malik Boudiaf, Hoel Kervadec, Ziko Imtiaz Masud, Pablo Piantanida, Ismail Ben Ayed, and Jose Dolz. 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*, pages 13979–13988, 2021. 1, 2

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014. 1

[3] 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. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[5] Tianrun Chen, Lanyun Zhu, Chaotao Deng, Runlong Cao, Yan Wang, Shangzhan Zhang, Zejian Li, Lingyun Sun, Ying Zang, and Papa Mao. Sam-adapter: Adapting segment anything in underperformed scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3367–3375, 2023.

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 1, 2

[7] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[8] Qi Fan, Wenjie Pei, Yu-Wing Tai, and Chi-Keung Tang. Self-support few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 701–719. Springer, 2022. 1, 2, 3, 6, 7

[9] Lin Geng Foo, Hossein Rahmani, and Jun Liu. Ai-generated content (aigc) for various data modalities: A survey, 2023. 2

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 1

[11] Xiao Fu, Shangzhan Zhang, Tianrun Chen, Yichong Lu, Lanyun Zhu, Xiaowei Zhou, Andreas Geiger, and Yiyi Liao. Panoptic nerf: 3d-to-2d label transfer for panoptic urban scene segmentation. In *2022 International Conference on 3D Vision (3DV)*, pages 1–11. IEEE, 2022. 1

[12] Rafael C Gonzales and Paul Wintz. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 1987. 5

[13] Sunghwan Hong, Seokju Cho, Jisu Nam, Stephen Lin, and Seungryong Kim. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *European Conference on Computer Vision*, pages 108–126. Springer, 2022. 1, 2, 6

[14] Deyi Ji, Haoran Wang, Mingyuan Tao, Jianqiang Huang, Xian-Sheng Hua, and Hongtao Lu. Structural and statistical texture knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16876–16885, June 2022. 4

[15] Siyu Jiao, Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, and Humphrey Shi. Mask matching transformer for few-shot segmentation. In *Advances in Neural Information Processing Systems*. 2

[16] Chunbo Lang, Gong Cheng, Binfei Tu, and Junwei Han. Learning what not to segment: A new perspective on few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8057–8067, 2022. 1, 2, 5, 6

[17] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8334–8343, 2021. 1, 2, 3

[18] Zihan Lin, Zilei Wang, and Yixin Zhang. Continual semantic segmentation via structure preserving and projected feature alignment. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 345–361. Springer, 2022. 4

[19] Weide Liu, Chi Zhang, Guosheng Lin, and Fayao Liu. Crnet: Cross-reference networks for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4165–4173, 2020. 1, 2

[20] Xinyu Liu, Beiwen Tian, Zhen Wang, Rui Wang, Kehua Sheng, Bo Zhang, Hao Zhao, and Guyue Zhou. Delving into shape-aware zero-shot semantic segmentation. *arXiv preprint arXiv:2304.08491*, 2023. 5

[21] Yuanwei Liu, Nian Liu, Qinglong Cao, Xiwen Yao, Junwei Han, and Ling Shao. Learning non-target knowledge for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11573–11582, 2022. 2, 6

[22] Yuanwei Liu, Nian Liu, Xiwen Yao, and Junwei Han. Intermediate prototype mining transformer for few-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 35:38020–38031, 2022. 2, 6, 7

[23] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 142–158. Springer, 2020. 1, 2

[24] Zhikang Liu and Lanyun Zhu. Label-guided attention distillation for lane segmentation. *Neurocomputing*, 438:312–322, 2021. 1

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[26] Juhong Min, Dahyun Kang, and Minsu Cho. Hypercorrelation squeeze for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vi-*

sion, pages 6941–6952, 2021. 1, 2, 6

[27] Seonghyeon Moon, Samuel S Sohn, Honglu Zhou, Sejong Yoon, Vladimir Pavlovic, Muhammad Haris Khan, and Mubbasir Kapadia. Msi: Maximize support-set information for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19266–19276, 2023. 6

[28] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 622–631, 2019. 1, 2

[29] Atsuro Okazawa. Interclass prototype relation for few-shot segmentation. In *European Conference on Computer Vision*, pages 362–378. Springer, 2022. 2

[30] Bohao Peng, Zhuotao Tian, Xiaoyang Wu, Chengyao Wang, Shu Liu, Jingyong Su, and Jiaya Jia. Hierarchical dense correlation distillation for few-shot segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23641–23651, 2023. 6

[31] Xinyu Shi, Dong Wei, Yu Zhang, Donghuan Lu, Munan Ning, Jiashun Chen, Kai Ma, and Yefeng Zheng. Dense cross-query-and-support attention weighted mask aggregation for few-shot segmentation. In *European Conference on Computer Vision*, pages 151–168. Springer, 2022. 6

[32] Mennatullah Siam, Boris N Oreshkin, and Martin Jagersand. Amp: Adaptive masked proxies for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5249–5258, 2019. 1, 2

[33] Maoqing Tian, Shuai Yi, Hongsheng Li, Shihua Li, Xuesen Zhang, Jianping Shi, Junjie Yan, and Xiaogang Wang. Eliminating background-bias for robust person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5794–5803, 2018. 2

[34] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9197–9206, 2019. 1, 2

[35] Yan Wang, Jian Cheng, Yixin Chen, Shuai Shao, Lanyun Zhu, Zhenzhou Wu, Tao Liu, and Haogang Zhu. Fvp: Fourier visual prompting for source-free unsupervised domain adaptation of medical image segmentation. *IEEE Transactions on Medical Imaging*, 2023. 1

[36] Yuan Wang, Rui Sun, and Tianzhu Zhang. Rethinking the correlation in few-shot segmentation: A buoys view. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192, 2023. 6

[37] Yuan Wang, Rui Sun, Zhe Zhang, and Tianzhu Zhang. Adaptive agent transformer for few-shot segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 36–52. Springer, 2022. 6

[38] Zhenyu Wang, Ya-Li Li, Ye Guo, and Shengjin Wang. Combating noise: semi-supervised learning by region uncertainty quantification. *Advances in Neural Information Processing Systems*, 34:9534–9545, 2021. 5

[39] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 1

[40] Qianxiong Xu, Wenting Zhao, Guosheng Lin, and Cheng Long. Self-calibrated cross attention network for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 6

[41] Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. Mining latent classes for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8721–8730, 2021. 1, 2

[42] Yong Yang, Qiong Chen, Yuan Feng, and Tianlin Huang. Mianet: Aggregating unbiased instance and general information for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7131–7140, 2023. 6

[43] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. 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*, pages 5217–5226, 2019. 7

[44] Gengwei Zhang, Guoliang Kang, Yi Yang, and Yunchao Wei. Few-shot segmentation via cycle-consistent transformer. *Advances in Neural Information Processing Systems*, 34:21984–21996, 2021. 2

[45] Gengwei Zhang, Shant Navasardyan, Ling Chen, Yao Zhao, Yunchao Wei, Honghui Shi, et al. Mask matching transformer for few-shot segmentation. *Advances in Neural Information Processing Systems*, 35:823–836, 2022. 6

[46] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1

[47] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 1

[48] Lanyun Zhu, Tianrun Chen, Deyi Ji, Jieping Ye, and Jun Liu. Llafs: When large-language models meet few-shot segmentation. *arXiv preprint arXiv:2311.16926*, 2023. 1

[49] Lanyun Zhu, Tianrun Chen, Jianxiong Yin, Simon See, and Jun Liu. Continual semantic segmentation with automatic memory sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3082–3092, 2023. 1

[50] Lanyun Zhu, Deyi Ji, Shiping Zhu, Weihao Gan, Wei Wu, and Junjie Yan. Learning statistical texture for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12537–12546, 2021. 1

[51] Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 593–602, 2019. 1