Few-Shot Segmentation via Cycle-Consistent Transformer

Anonymous Author(s)
Affiliation
Address
email

Abstract

Few-shot segmentation aims to train a segmentation model that can fast adapt to novel classes with few exemplars. The conventional training paradigm is to learn to make predictions on query images conditioned on the features from support images. Previous methods only utilized the semantic-level prototypes of support images as the conditional information. These methods cannot utilize all pixel-wise support information for the query predictions, which is however critical for the segmentation task. In this paper, we focus on utilizing pixel-wise relationships between support and target images to facilitate the few-shot semantic segmentation task. We design a novel **Cy**cle-**C**onsistent **Tr**ansformer (CyCTR) module to aggregate pixel-wise support features into query ones. CyCTR performs cross-attention between features from different images, *i.e.* support and query images. We observe that there may exist unexpected irrelevant pixel-level support features. Directly performing cross-attention may aggregate these features from support to query and bias the query features. Thus, we propose using a novel cycle-consistent attention mechanism to filter out possible harmful support features and encourage query features to attend to the most informative pixels from support images. Experiments on all few-shot segmentation benchmarks demonstrate that our proposed CyCTR leads to remarkable improvement compared to previous state-of-the-arts. Specifically, on Pascal-5^i and COCO-20^i datasets, we achieve 66.6% and 45.6% mIoU for 5-shot segmentation, outperforming previous state-of-the-art by 3.6% and 7.1% respectively.

1 Introduction

Recent years have witnessed great progress in semantic segmentation [17, 3, 39]. The success can be largely attributed to large amounts of annotated data [40, 15]. However, labeling dense segmentation masks are very time-consuming. Semi-supervised segmentation [13, 34, 33] has been broadly explored to alleviate this problem, which assumes a large amount of unlabeled data is accessible. However, semi-supervised approaches may fail to generalize to novel classes with very few exemplars. In the extreme low data regime, few-shot segmentation [22, 30] is introduced to train a segmentation model that can quickly adapt to novel categories.

Most few-shot segmentation methods follow a learning-to-learn paradigm where predictions of query images are made conditioned on the features and annotations of support images. The key to the success of this training paradigm lies in how to effectively utilize the information provided by support images. Previous approaches extract semantic-level prototypes from support features and follow a metric learning [24, 5, 30] pipeline extending from PrototypicalNet [23]. According to the granularity of utilizing support features, these methods can be categorized into two groups, as illustrated in Figure. 1: 1) Class-wise mean pooling [30, 38, 37] (Figure 1(a)). Support features within regions of different categories are averaged to serve as prototypes to facilitate the classification of query pixels.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

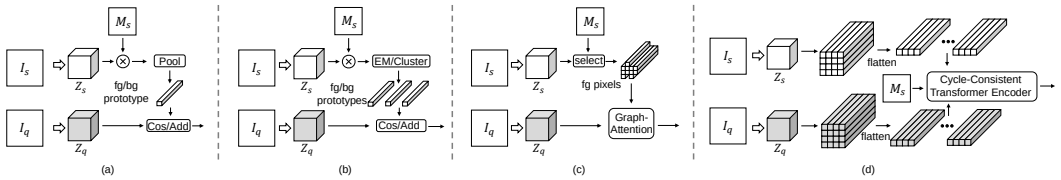

Figure 1: Different learning frameworks for few-shot segmentation, from the perspective of ways to utilize support information. (a) Class-wise mean pooling based method. (b) Clustering based method. (c) Foreground pixel attention method. (d) Our Cycle-Consistent Transformer (CyCTR) framework that enables all beneficial support pixel-level features (foreground and background) to be considered.

2) Clustering [16, 35, 36, 29](Figure 1(b)). Recent works attempt to generate multiple prototypes via EM algorithm or K-means clustering [35, 16], in order to extract more abundant information from support images. These prototype-based methods need to "compress" support information into different prototypes (*i.e.* class-wise or cluster-wise), which may lead to various degrees of loss of beneficial support information and thus harm segmentation on query image. Rather than using prototypes to abstract the support information, [36, 29] (Figure 1(c)) propose to employ the attention mechanism to extract information from support foreground pixels for segmenting query. However, such methods ignore all the background support pixels that can be beneficial for segmenting query image. And incorrectly consider partial foreground support pixels that are quite different from the query ones, leading to sub-optimal results.

In this paper, we focus on equipping each query pixel with relevant information from support images to facilitate the query pixel classifications. Inspired by the transformer architecture [27] which performs feature aggregation through attention, we design a novel **Cy**cle-**C**onsistent **Tr**ansformer (CyCTR) module (Figure 1(d)) to aggregate pixel-wise support features into query ones. Specifically, our CyCTR consists of two types of transformer blocks: the self-alignment block and the cross-alignment block. The self-alignment block is employed to encoder the query image features by aggregating its relevant context information, while the cross-alignment aims to aggregate the pixel-wise features of support images into the pixel-wise features of query image. Different from self-alignment where Query[1], Key and Value come from the same embedding, cross-alignment takes features from query images as Query, and those from support images as Key and Value. In this way, CyCTR provides abundant pixel-wise support information for pixel-wise features of query images to make predictions.

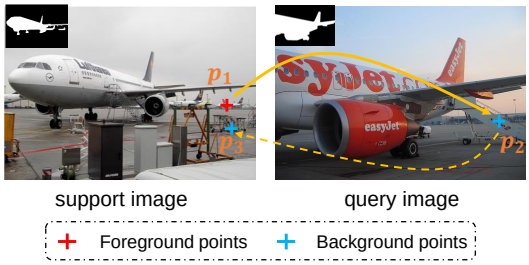

Figure 2: The motivation of our proposed method. Many pixel-level support features are quite different from the query ones, and thus may confuse the attention. We incorporate cycle-consistency into attention to filter such confusing support features. Note that the confusing support features may come from foreground and background.

Moreover, we observe that due to the differences between support and query images, *e.g.*, scale, color and scene, only a small proportion of support pixels can be beneficial for the segmentation of query image. In other words, in the support image, some pixel-level information may confuse the attention in the transformer. Figure 2 provides a visual example of a support-query pair together with the support mask. The confusing support pixels may come from both foreground pixels and background pixels. For instance, point p_1 in the support image located in the plane afar, which is indicated as foreground by the support mask. However, the nearest point p_2 in the query image (*i.e.* p_2 has the largest feature similarity with p_1) belongs to a different category, *i.e.* background. That means, there exists no query pixel which has both high similarity and the same semantic label with p_1. Thus, p_1 is likely to be harmful for segmenting "plane" and should be ignored when performing the attention. To overcome this issue, in CyCTR, we propose to equip the cross-alignment block with a novel cycle-consistent attention operation. Specifically, as shown in Figure 2, starting from one

[1]To distinguish from the phrase "query" in few-shot segmentation, we use "Query" with capitalization to note the query sequence in the transformer.

spatial point of the support features, we find its nearest neighbor in the query features. In turn, this nearest neighbor finds the most similar support feature. If the starting and the end support features come from the same category, a cycle-consistency relationship is established. We incorporate such an operation into attention to force query features only attend to cycle-consistent support features to extract information. In this way, the support pixels that are far away from query ones are not considered. Meanwhile, cycle-consistent attention enables us to more safely utilize the information from background support pixels, without introducing much bias into the query features.

In a nutshell, our contributions are summarized as follows: (1) We tackle few-shot segmentation from the perspective of providing each query pixel with relevant information from support images through pixel-wise alignment. (2) We propose a novel cycle-consistent transformer (CyCTR) to aggregate the pixel-wise support features into the query ones. In CyCTR, we observe that many support features may confuse the attention and bias pixel-level feature aggregation, and propose incorporating cycle-consistent operation into the attention to deal with this issue. (3) Our CyCTR achieves state-of-the-art results on two few-shot segmentation benchmarks, *i.e.*, Pascal-5^i and COCO-20^i. Extensive experiments validate the effectiveness of each component in our CyCTR.

2 Related Work

2.1 Few-Shot Segmentation

Few-shot segmentation [22] is established to perform segmentation with very few exemplars. Recent approaches formulate few-shot segmentation from the view of metric learning [24, 5, 30]. For instance, [5] first extends ProtoNets [23] to perform few-shot segmentation. PANet [30] simplifies the framework with an efficient prototype learning framework. SG-One [38] leverage the cosine similarity map between the single support prototype and query features to guide the prediction. CANet [37] replaces the cosine similarity with an additive alignment module and iteratively refines the network output. PFENet [25] further designs an effective feature pyramid module and leverages a prior map to achieve better segmentation performance. Recently, [35, 16, 36] point out that only a single support prototype is insufficient to represent a given category. Therefore, they attempt to obtain multiple prototypes to represent the support objects with the EM algorithm as prototypes and the prototypes are compared with query image through cosine similarity [16, 35]. Besides, [36, 28] attempt to use graph attention networks [28] to utilize all foreground support pixel features. However, due to the large differences between support and query images, not all pixels are credible for final query segmentation and our method is designed to overcome this issue.

2.2 Transformer

Transformer and self-attention are firstly introduced in the fields of machine translation and natural language processing [4, 27], and are receiving increasing interests recently in the computer vision credit to their long-range modeling capability. Previous works utilize self-attention as additional module on top of existing convolutional networks, *e.g.,* Nonlocal [31] and CCNet [12]. ViT [6] and its following work [26] demonstrate the pure transformer architecture can achieve state-of-the-art for image recognition. On the other hand, DETR [2] builds up an end-to-end framework with a transformer encoder-decoder on top of backbone networks for object detection. And its deformable vairents [43] improves the performance and training efficiency.

2.3 Cycle-consistency Learning

Our work is partially inspired by cycle-consistency learning [42, 7] that is explored in various computer vision areas. For instance, in image translation, CycleGAN [42] uses cycle-consistency to align image pairs. It is also effective in learning 3D correspondence [41], consistency between video frames [32] and association between different domains [14]. These works typically constructs cycle-consistency loss between aligned targets (*e.g.*, images). However, the simple training loss cannot be directly applied to few-shot segmentation because the test categories are unseen from the training process and no finetuning is involved during testing. In this work, we combine the ideology of cycle-consistency with transformer and show its substantial potential in few-shot segmentation.

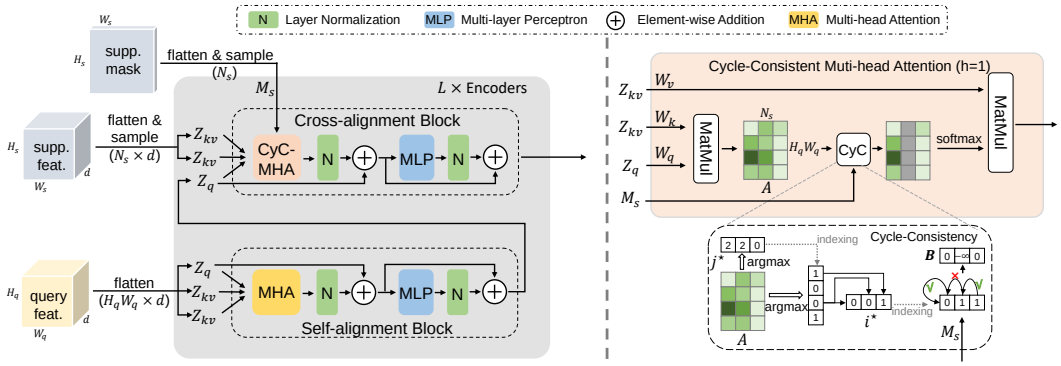

Figure 3: Framework of our proposed cycle-consistent transformer (CyCTR). (a). Each encoder of CyCTR consists of two transformers blocks, the cross-alignment block is for receiving information from support images and the self-alignment block is for keeping the feature coherence within the query feature map. In the cross-alignment block, we introduce the multi-head cycle-consistent attention (shown in right), with number of head $h = 1$ for presentation simplicity. The aggregation operation is guided by the cycle-consistency relationships among query and support features.

3 Methodology

3.1 Problem Setting

Few-shot segmentation aims at training a segmentation model that can segment novel objects with very few annotated samples. Specifically, given dataset D_{train} and D_{test} with category set C_{train} and C_{test} respectively, where $C_{train} \cap C_{test} = \emptyset$, the model trained on D_{train} is directly used to test on D_{test}. In line with previous works [25, 30, 37], episode training is adopted in this work for few-shot segmentation. Each episode is composed of k support images I_s and a query image I_q to form a k-shot episode $\{\{I_s\}^k, I_q\}$. Then the training set and test set are represented by $D_{train} = \{\{I_s\}^k, I_q\}^{N_{train}}$ and $D_{test} = \{\{I_s\}^k, I_q\}^{N_{test}}$, where N_{train} and N_{test} is the number of episodes for training and test set. During training, both support masks M_s and query masks M_q are available for training images, and only support masks are accessible during testing.

3.2 Revisiting of Transformer

Following the general form in [27], a transformer block is composed of alternating layers of multiheaded attention (MHA) and multi-layer perceptron (MLP). LayerNorm (LN) [1] and residual connection [10] are applied at the end of each block. Specially, an attention layer is formulated as

$$\text{Atten}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

where $[Q; K; V] = [W_q Z_q; W_k Z_{kv}; W_v Z_{kv}]$, in which Z_q is the input Query sequence, Z_{kv} is the input Key/Value sequence, $W_q, W_k, W_v \in \mathbb{R}^{d \times d}$ is the learnable parameters, d is the hidden dimension of the input sequences and we assume all sequences have the same dimension d by default. The attention layer computes the normalized attention weight $AT = \text{softmax}(\frac{QK^T}{\sqrt{d}})$ by attending each Query element to all Key elements and aggregating items from Value sequence through matrix multiplication. When $Z_q = Z_{kv}$, it functions as self-attention mechanism.

The multihead attention layer is an extention of attention layer, which performs h attention operations and concatenate results together. Typically, it divide input dimensions d into h groups to obtain

$$\text{MHA}(Q, K, V) = [\text{head}_1, ..., \text{head}_\text{h}], \tag{2}$$

where $\text{head}_\text{m} = \text{Atten}(Q_m, K_m, V_m)$ and the inputs $[Q_m, K_m, V_m]$ are the mth group from $[Q, K, V]$ with dimension d/h.

3.3 Cycle-Consistent Transformer

Our framework is illustrated in Figure 3(a). Generally, an encoder of our cycle-consistent transformer (CyCTR) consists of a self-alignment transformer block for encoding the query features and a cross-alignment transformer block to enable the query features to attend to the informative support features. The whole CyCTR module consists of stacking L encoders.

Specifically, for the given query feature $X_q \in \mathbb{R}^{H_q \times W_q \times d}$ and support feature $X_s \in \mathbb{R}^{H_s \times W_s \times d}$, we first flatten them into 1D sequences (with shape $HW \times d$) as inputs for transformer, in which a *token* is represented by the feature $z \in \mathbb{R}^d$ at one pixel location. The self-alignment block only takes the flattened query feature as input to ensure the self-alignment. As context information of each pixel has been proved beneficial for segmentation [3, 39], we adopt the self-alignment block to pixel-wise features of query image to aggregate their global context information. We don't pass support images through this self-alignment block, as we mainly focus on the segmentation performance of query images and passing through the support images which don't coordinate with the query mask may do harm to the self-alignment on query images.

In contrast, the cross-alignment block performs attention between query and support pixel-wise features to aggregate relevant support features into query ones. It takes the flattened query feature and samples a subset of support feature (the sampling procedure is discussed latter) with size $N_s \leq H_s W_s$ as Key/Value sequence Z_{kv}. With these two blocks, it is expected to better encoder the query features to facilitate the subsequent pixel-wise classification.

3.3.1 Cycle-Consistent Attention

According to the aforementioned discussion, the pure pixel-level attention may be confused by excessive harmful support features. To alleviate this issue, as shown in Figure 3(b), a cycle-consistent attention operation is proposed. We first go through the proposed approach for 1-shot case for presentation simplicity and then discuss it in the few-shot setting. Formally, an affinity map $A = \frac{QK^T}{\sqrt{d}}$, $A \in \mathbf{R}^{H_q W_q \times N_s}$ is first calculated to measure the correspondence relationships between all query and support tokens. Then, for a single token position j ($j \in \{0, 1, ..., N_s\}$) in the support feature, its most similar point $i^\star$ in the query feature is obtained by

$$i^\star = \underset{i}{\operatorname{argmax}}\, A_{(i,j)}, \tag{3}$$

where $i \in \{0, 1, ..., H_q W_q\}$ is the index of the flattened query feature. Since the query mask is not accessible, the label of token $i^\star$ is unkown. However, we can map it back into support features in the similar way to find its most similar feature point $j^\star$ in support sequence:

$$j^\star = \underset{j}{\operatorname{argmax}}\, A_{(i^\star,j)}. \tag{4}$$

Given the support label $M_s \in \mathbb{R}^{N_s}$, we call a cycle-consistency established if $M_{s(j)} = M_{s(j^\star)}$. The cycle-consistency relationships are constructed for all tokens in the support sequence. A possible way to utilize the cycle-consistency is to maximize the similarity between matched points [42, 14]. However, in few-shot segmentation, the goal is to fast adapt the model to novel categories rather than improving to fit to training categories. Thus, we incorporate the cycle-consistency into the attention operation to encourage the cycle-consistent cross-attention. First, by traversing all support tokens, an additive bias $B \in \mathbb{R}^{N_s}$ is obtained by

$$B_j = \begin{cases} 0, & \text{if} M_{s(j)} = M_{s(j^\star)} \\ -\infty, & \text{if} M_{s(j)} \neq M_{s(j^\star)} \end{cases},$$

where $j \in \{0, 1, ..., N_s\}$. Then, for a single query token $Z_{q(i)} \in \mathbb{R}^d$ at location i, it aggregate support information by

$$\text{CyCAtten}(Q_i, K_i, V_i) = \text{softmax}(A_{(i)} + B)V, \tag{5}$$

where $i \in \{0, 1, ..., H_q W_q\}$ and A is obtained by $\frac{QK^T}{\sqrt{d}}$. In the forward process, B is element-wise added with the affinity $A_{(i)}$ for $Z_{q(i)}$ to aggregate support features. In this way, the attention weight for the cycle-inconsistent support feature tends to be zero, implying that these irrelevant information will not be considered. Besides, the cycle-consistent attention implicitly encourages the consistency

between the most relative features among query and support features through backpropagation, which can reinforce the model to produce consistent feature representations

Furthermore, aggregating the most relevant support features from support to query would cause the query feature incoherent. The reason cames from the pixel-level aggregation where different pixels would have different affinities across query and support. Although using a self-attention operation (*e.g.,* in the self-alignment block) can alleviate this problem to some extent, it also suffer from the similar problem, *i.e.* fusing background features with foreground features within query tokens. Inspired by our cycle-consistent attention, it would be better to recieve information from more credible location than aggregating from all pixels. Due to the lack of query mask M_q, it is impossible to establish the self-cycle-consistency relationship for query feature iteself. Inspired by DeformableAttention [43], the credit points can be obtained via a learnable way as $\Delta = f(Q+\mathrm{Coord})$ and $A^{'} = g(Q+\mathrm{Coord})$, where $\Delta \in \mathbb{R}^{H_p W_p \times P}$ is the predicted credit points, in which each element $\delta \in \mathbb{R}^P$ in Δ represents the relative offeset from each piexl and P represents the number of points to aggregate feature. And $A^{'} \in \mathbb{R}^{H_q W_q \times P}$ is the attention weights. $\mathrm{Coord} \in \mathbb{R}^{H_q W_q \times d}$ is the positional encoding [20] to make the prediction be aware of absolute position, $f(\cdot)$ and $g(\cdot)$ are two fully connected layers that predict the offsets and attention weights. Therefore, the self-attention within the self-alignment transformer block is operating as

$$\mathrm{PredAtten}^{'}(Q_r, V_r) = \sum_{g}^{P} \mathrm{softmax}(A^{'})_{(r,g)} V_{r+\Delta_{(r,g)}}, \tag{6}$$

where $r \in \{0, 1, ..., H_q W_q\}$ is the index of the flattened query feature.

Broadly speaking, the cycle-consistent transformer effectively avoids being biased by possible harmful features so that facilitate the training of few-shot segmentation.

Mask-guided sparse sampling and K-shot Setting: Our proposed cycle-consistency transformer can be easily extended to K-shot setting where $K > 1$. The only difference is the input support features. When more than one support feature maps are provided, all support features are flattened and concatenated together as input. As the transformer is operating at the pixel-level, a sampling strategy is required to reduce the number of tokens in support sequence. In this work, we apply a simple mask-guided sampling strategy to achieve this goal. Concretely, given the k-shot support sequence $Z_s \in \mathbb{R}^{kH_s W_s \times d}$ and the flattened support masks $M_s \in \mathbb{R}^{kH_s W_s}$, the sampled support sequence is obtained by uniformly sample $N_{fg} <= \frac{N_s}{2}$ tokens from the foreground regions and the rest $N_s - N_{fg}$ tokens are sampled from the background regions in all support images. This reduces the computation complexity, making it independent to the spatial size of support image. Furthermore, this strategy balance the foreground-background proportion and also implictly considers the spatial size of each object region in different support images.

3.4 Overall Framework

Following previous works [25, 30, 37], both query and support image are first feed into a shared backbone network (*e.g.,* ResNet [10]) with ImageNet [21] pretrain to obtain general image features. Similar to [25], middle-level features are processed by a 1x1 convolution to reduce the hidden dimension and high-level feature are used to generate a prior map[2] that concatenated with the middle-level feature. To enable the pixel-wise comparison, we also concatenate the mask averaged support feature to both query and support feature and processed by a 1x1 convolution before input into the transformer. The final segmentation restuls is obtained by reshaping the output sequence back to spatial dimensions and predicted by a small convolution head that is consisted by one 3x3 convolution, one ReLU activation and a 1x1 convolution. As for the baseline of our method, we use two residual blocks [10] instead of the transformer to encode query feature.

4 Experiments

4.1 Dataset and Evaluation Metric

We conduct experiments on two commonly used few-shot segmentation datasets, Pascal-5^i [8] (which is combined with SBD [9] dataset and COCO-20^i [15], to evaluate our method. For Pascal-5^i, 20

[2]The prior map is generated with reduced high-level features, details can be found in the Appendix.

Table 1: Comparison with other state-of-the-arts for 1-shot and 5-shot segmentation on PASCAL-5^i using the mIoU (%) as evaluation metric. Best results in bold.

Method	Backbone	1-shot					5-shot				
		5^0	5^1	5^2	5^3	Mean	5^0	5^1	5^2	5^3	Mean
PANet [30]	Vgg-16	42.3	58.0	51.1	41.2	48.1	51.8	64.6	59.8	46.5	55.7
FWB [19]		47.0	59.6	52.6	48.3	51.9	50.9	62.9	56.5	50.1	55.1
SG-One [38]		40.2	58.4	48.4	38.4	46.3	41.9	58.6	48.6	39.4	47.1
RPMM [35]		47.1	65.8	50.6	48.5	53.0	50.0	66.5	51.9	47.6	54.0
CANet [37]	Res-50	52.5	65.9	51.3	51.9	55.4	55.5	67.8	51.9	53.2	57.1
PGNet [36]		56.0	66.9	50.6	50.4	56.0	57.7	68.7	52.9	54.6	58.5
RPMM [35]		55.2	66.9	52.6	50.7	56.3	56.3	67.3	54.5	51.0	57.3
PPNet [16]		47.8	58.8	53.8	45.6	51.5	58.4	67.8	**64.9**	56.7	62.0
PFENet [25]		61.7	69.5	55.4	56.3	60.8	63.1	70.7	55.8	**57.9**	61.9
CyCTR (Ours)	Res-50	**67.8**	**72.8**	**58.0**	**58.0**	**64.2**	**71.1**	**73.2**	60.5	57.5	**65.6**
FWB [19]	Res-101	51.3	64.5	56.7	52.2	56.2	54.9	67.4	**62.2**	55.3	59.9
DAN [29]		54.7	68.6	**57.8**	51.6	58.2	57.9	69.0	60.1	54.9	60.5
PFENet [25]		60.5	69.4	54.4	55.9	60.1	62.8	70.4	54.9	57.6	61.4
CyCTR (Ours)	Res-101	**69.3**	**72.7**	56.5	**58.6**	**64.3**	**73.5**	**74.0**	58.6	**60.2**	**66.6**

Table 2: Comparison with other state-of-the-arts for 1-shot and 5-shot segmentation on COCO-20^i using the mIoU (%) as evaluation metric. Best results in bold.

Method	Backbone	1-shot					5-shot				
		20^0	20^1	20^2	20^3	Mean	20^0	20^1	20^2	20^3	Mean
FWB [19]	Res-101	19.9	18.0	21.0	28.9	21.2	19.1	21.5	23.9	30.1	23.7
PPNet [16]	Res-50	28.1	30.8	29.5	27.7	29.0	39.0	40.8	37.1	37.3	38.5
RPMM [35]	Res-50	29.5	36.8	29.0	27.0	30.6	33.8	42.0	33.0	33.3	35.5
PFENet [25]	Res-101	34.3	33.0	32.3	30.1	32.4	38.5	38.6	38.2	34.3	37.4
CyCTR (Ours)	Res-50	**38.9**	**43.0**	**39.6**	**39.8**	**40.3**	**41.1**	**48.9**	**45.2**	**47.0**	**45.6**

classes are separated into 4 splits where for each split 15 classes is used for training and 5 classes for test. During test, 1000 episodes are randomly sampled for evaluation. In COCO-20^i, we follow the data split settings in FWB [19] to divide 80 classes evenly into 4 splits, 60 classes for training and test on 20 classes. Detailed data split setting can be found in the Supplementary Material. Following common practice [25, 30, 38], the mean intersection over union (mIoU) is adopted as the evaluation metric, which is the averaged value of IoU of all test classes. We also report the foreground-background IoU (FB-IoU) for reference.

4.2 Implementation Details

In our experiments, the training strategies follow the same setting in [25]: training 50 epochs for COCO-20^i and 200 epochs for Pascal-5^i, images are resized and cropped to 473×473 for both datasets and using Random rotation from $-10°$ to $10°$ as data augmentation. We use ImageNet [21] pretrained ResNet [10] as the backbone network and its parameters (including batchnorms) are freezed. For the proposed cycle-consistent transformer, we set the number of support tokens N_s to 600 for 1-shot and 5×600 for 5-shot setting. The number of sampled tokens is obtained according to the averaged number of foreground pixels among Pascal-5^i training set. For the self-attention block, the number of points P to aggregate features for all pixel locations are set to 9. For other hyper-parameters in transformer blocks, we set the hidden dimension of MLP layer to three times of input hidden dimension and the input hidden dimension is 384 for the best model. The number of head for all attention layers are set to 1 for simplicity. For parameters in transformer blocks, the learning rate is set to 1×10^{-4} with the weight decay 1×10^{-2} and is optimized with AdamW [18] optimizer following other transformer works [2, 6, 26]. For the rest parameters, we use the initial learning rate 2.5×10^{-3}, momentum 0.9 and weight decay 1×10^{-4} and SGD optimizer with poly learning rate decay [3] and batch size 4. All experiments are carried out on four Tesla V100 GPUs.

4.3 Comparisons with State-of-the-Art Methods

In Table 1 and Table 2, we compare our method with other state-of-the-art few-shot segmentation approaches on Pascal-5^i and COCO-20^i respectively. It can be seen that our approach achieves new

Table 4: Ablation studies of effectiveness of each component in our cycle-consistent transformer.

self-alignment	cross-alignment	CyCTR (pred)	CyCTR (fg. only)	CyCTR	mIoU (%)
					58.8
✓					61.6
✓	✓				61.2
✓	✓	✓			61.9
✓	✓		✓		62.0
✓	✓			✓	62.8

Table 3: Comparison with other methods using FB-IoU (%) on Pascal-5^i for 1-shot and 5-shot segmentation.

Method	Backbone	FB-IoU (%)	
		1-shot	5-shot
A-MCG [11]	Res-101	61.2	62.2
DAN [29]	Res-101	71.9	72.3
PFENet [25]	Res-101	72.9	73.5
CyCTR (Ours)	Res-101	72.9	**75.0**

state-of-the-art performance on both Pascal-5^i and COCO-20^i. Specifically, on Pascal-5^i, to make fair comparisons with other methods, we report results with both ResNet-50 and ResNet-101. Our CyCTR achieves 64.3% and 66.6% with ResNet-101 backbone for 1-shot and 5-shot, outperforming previous state-of-the-art results by significant margins of 4.2% and 5.2%, respectively. It is worth noting that different from the state-of-the-art PFENet [25] that tends to degrade with ResNet-101, our method can obtain slight improvement with a larger backbone network. For COCO-20^i results in Table 2, our method also outperforms other methods by a large margin due to the capability of the transformer to fit more complex data. Besides, Table 3 shows the comparison using FB-IoU on PASCAL-5^i for 1-shot and 5-shot segmentation, our method also achieves the new state-of-the-art results.

4.4 Ablation Studies

To provide a deeper understanding of our proposed method, ablation studies are carried out on Pascal-5^i 1-shot setting and results are reported in terms of mIoU.

4.4.1 Component-wise Ablations

We perform ablation studies in regard to each component of our CyCTR in Table 4. The first line is the result of the baseline, where the encoder consists of two residual blocks rather than the transformer. For all ablations, the hidden dimension is set to 128 and two transformer encoders are used. The mIoU results are averaged over four splits. Firstly, we only use the self-alignment block that only encodes query features. The support information of this case only comes from the concatenated single support vector similar to [37]. It can already achieve decent results, showing that the transformer encoder is effective for modeling context for few-shot segmentation. Then, we utilize the cross-alignment block but only with the vanilla attention operation in Equation 1. The mIoU is degraded by 0.4%, which is not surprising as discussed in our paper that some support features at pixel-level may be harmful for query segmentation. By using the cycle-consistent attention, the performance is improved by a large margin, *i.e.* 1.6% mIoU compared to the vanilla attention. Besides, our cycle-consistent transformer aggregates pixel-level information from both foreground and background support features after sampling, which is not well explored in previous works due to large amounts of unexpected support information at pixel-level. It can be seen in Table 4 that the support background information is also important for few-shot segmentation and provides 0.8% mIoU improvement.

Besides, we construct the cycle-consistency within the cross-alignment block while using the predicted aggregation strategy in the self-alignment block. One may curious that whether the predicted manner can also work well for aggregating support features to query. Therefore, in Table 4, we use predicted aggregation in Equation 6 instead of the cycle-consistent attention in the cross-alignment block. It does show a positive impact for aggregating useful from support but is 0.9% worse than the proposed cycle-consistent attention. The reason lies in the dramatically changing support images during few-shot segmentation testing. The cycle-consistency is better than the learnable way as it can globally consider the varying conditional information from both query and support.

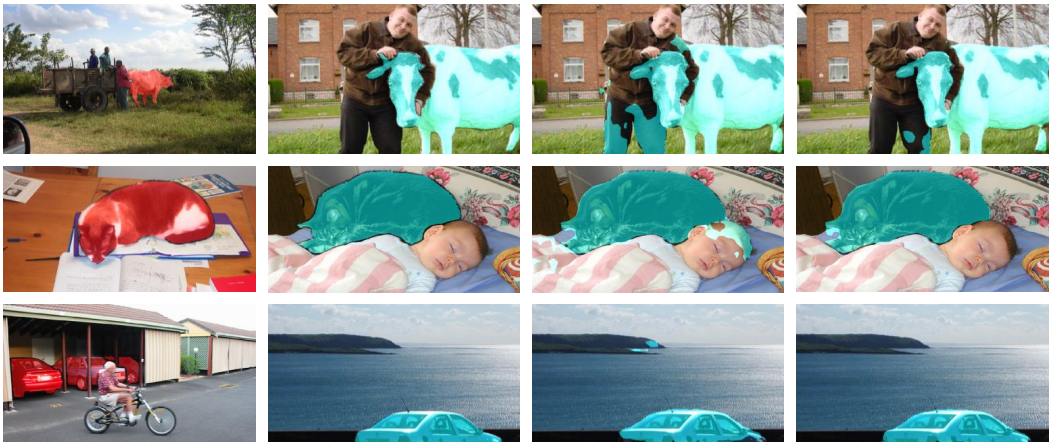

Figure 4: Qualitative results on Pascal-5^i. From left to right, each column shows the examples of: Support image with mask region in red; Query image with ground truth mask region in blue; Result produced by the model without cycle-consistency in CyCTR; Result produced by our method.

4.4.2 Effect of Model Capacity

We can stack more encoders or increase the hidden dimension of encoders to increase its capacity and validate the effectiveness of our CyCTR. The results with different numbers of encoders (denoted as L) or hidden dimensions (denoted as d) are shown in Table 5a and 5b. While increasing L or d within a certain range, CyCTR achieves better results. Optimal L and d exist due to the limited amounts of data.

Table 5: Effect of varying (a) number of encoders L and (b) hidden dimensions d. When varying L, d is fixed to 128; while varying d, L is fixed to 2.

#Encoder	mIoU (%)
1	61.4
2	**62.8**
3	61.8

(a)

#Dim	mIoU (%)
256	63.3
384	**64.2**
512	63.6

(b)

4.5 Qualitative results

In Figure 4, we show some qualitative results generated by our model on Pascal-5^i. It can be seen that our cycle-consistent attention in the transformer block can improve the segmentation quality by suppressing false positives. For instance, without cycle-consistency, the model misclassifies trousers as “cow” in the first row, baby’s hair as “cat” in the second row, and a fraction of mountain as “car” in the third row, while our model rectifies these part as background. However, in the first row, our CyCTR still segments part of the trousers as "cow" and the right boundary of the segmentation mask is slightly worse than the model without cycle-consistency. The reason comes from the extreme differences between query and support, *i.e.* the support image indicates a "cattle" but the query image contains a milk cow. The cycle-consistency may over-suppress the positive region, which may be a potential direction to further investigate to improve our method.

5 Conclusion

In this paper, we design a CyCTR module to deal with the few-shot segmentation problem. Different from previous practices that either adopt semantic-level prototype(s) from support images or only use foreground support features to encode query features, our CyCTR utilizes all pixel-level support features and can effectively eliminate aggregating confusing and harmful support features with the proposed novel cycle-consistency attention. We conduct extensive experiments on two popular benchmarks, and our CyCTR outperforms previous state-of-the-art methods by a significant margin. We hope this work can motivate researchers to design more effective algorithms to utilize pixel-level support features to advance the future pixel-level few-shot learning research.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[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] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Nanqing Dong and Eric P Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, volume 3-4, 2018.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1801–1810, 2019.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[9] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[11] Tao Hu, Pengwan Yang, Chiliang Zhang, Gang Yu, Yadong Mu, and Cees GM Snoek. Attention-based multi-context guiding for few-shot semantic segmentation. In *Association for the Advancement of Artificial Intelligence (AAAI)*, volume 33, pages pp. 8441–8448, 2019.

[12] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019.

[13] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.

[14] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander G Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In *NeurIPS*, 2020.

[15] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision (ECCV)*, 2014.

[16] 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.

[17] 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.

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[19] 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.

[20] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[22] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *British Machine Vision Conference (BMVC)*, 2018.

[23] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems (NeurIPS)*, 2017.

[24] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[25] Z Tian, H Zhao, M Shu, Z Yang, R Li, and J Jia. Prior guided feature enrichment network for few-shot segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[26] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[29] Haochen Wang, Xudong Zhang, Yutao Hu, Yandan Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *European Conference on Computer Vision (ECCV)*, 2020.

[30] 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.

[31] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[32] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2566–2576, 2019.

[33] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1568–1576, 2017.

[34] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7268–7277, 2018.

[35] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Qixiang Ye. Prototype mixture models for few-shot semantic segmentation. In *European Conference on Computer Vision*, pages 763–778. Springer, 2020.

[36] Chi Zhang, Guosheng Lin, Fayao Liu, Jiushuang Guo, Qingyao Wu, and Rui Yao. Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9587–9595, 2019.

[37] 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.

[38] Xiaolin Zhang, Yunchao Wei, Yi Yang, and Thomas S Huang. Sg-one: Similarity guidance network for one-shot semantic segmentation. *IEEE Transactions on Cybernetics*, 50(9):3855–3865, 2020.

[39] 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 (CVPR)*, 2017.

[40] 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.

[41] Tinghui Zhou, Philipp Krahenbuhl, Mathieu Aubry, Qixing Huang, and Alexei A Efros. Learning dense correspondence via 3d-guided cycle consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 117–126, 2016.

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[43] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] see the end of Sec. 1.
   (b) Did you describe the limitations of your work? [Yes] we discuss the failure case in Sec. 4, and show it in Figure 4.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] compared with previous works, our one does not introduce additional negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A] this work does not contain any theoretical assumptions.
   (b) Did you include complete proofs of all theoretical results? [N/A] this work does not contain any theoretical proofs.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] we include the details for reproduction in Sec. 4.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see Sec. 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] we follow the usual format used in previous works [25, 30, 37] to report and compare the results.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] see Sec. 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] we cite the creators or original papers of all the related code, data, and models used in our work.
   (b) Did you mention the license of the assets? [No] all the assets are free for research study and widely used in previous works.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] but the code of our method will be made publicly available as soon as this work is accepted.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] all the datasets used in this work are publicly available, free for research , and commonly used in previous related works.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] all the datasets are commonly used in previous related works.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] we did not use crowdsourcing or conduct research with human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] we did not use crowdsourcing or conduct research with human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] we did not use crowdsourcing or conduct research with human subjects.