Doubly Deformable Aggregation of Covariance Matrices for Few-shot Segmentation

Zhitong Xiong [1], Haopeng Li [2], and Xiao Xiang Zhu [1,3]

[1] Data Science in Earth Observation, Technical University of Munich (TUM)
[2] School of Computing and Information Systems, University of Melbourne
[3] Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR)

Abstract. Training semantic segmentation models with few annotated samples has great potential in various real-world applications. For the few-shot segmentation task, the main challenge is how to accurately measure the semantic correspondence between the support and query samples with limited training data. To address this problem, we propose to aggregate the learnable covariance matrices with a deformable 4D Transformer to effectively predict the segmentation map. Specifically, in this work, we first devise a novel hard example mining mechanism to learn covariance kernels for the Gaussian process. The learned covariance kernel functions have great advantages over existing cosine similarity-based methods in correspondence measurement. Based on the learned covariance kernels, an efficient doubly deformable 4D Transformer module is designed to adaptively aggregate feature similarity maps into segmentation results. By combining these two designs, the proposed method can not only set new state-of-the-art performance on public benchmarks, but also converge extremely faster than existing methods. Experiments on three public datasets have demonstrated the effectiveness of our method.[4]

Keywords: deep kernel learning, few-shot segmentation, Gaussian process, similarity measurement, Transformer

1 Introduction

Semantic segmentation at the pixel level [46,41,19,6] is one of the fundamental tasks in computer vision and has been extensively studied for decades. In recent years, the performance of semantic segmentation tasks has been significantly improved due to the substantial progress in deep learning techniques. Large-scale convolutional neural networks (CNNs) [28,8], vision Transformers [3,18], and MLP-based deep networks [15] have greatly improved the ability of visual representation learning, and significantly enhanced the performance of downstream tasks such as semantic segmentation [47,46].

[4] Code: `https://github.com/ShadowXZT/DACM-Few-shot.pytorch`

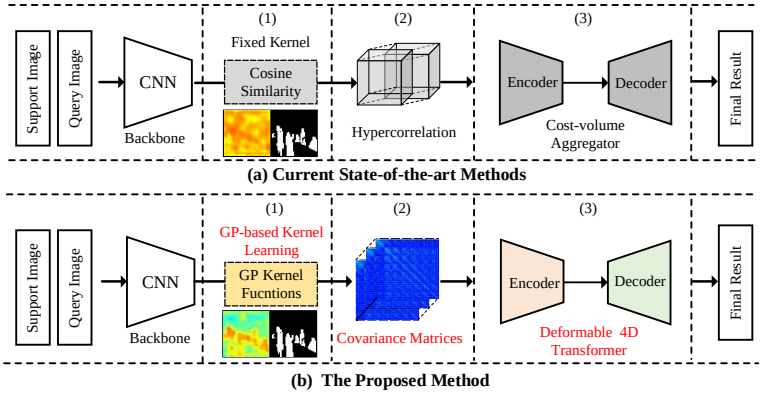

Fig. 1. Comparison of the proposed DACM (b) with the framework of existing state-of-the-art methods (a). There are three main differences. (1) We propose to utilize GP-based kernel functions for similarity measurement. (2) We use covariance matrices instead of cosine similarity as the cost volume. (3) A DDT module is designed for effective cost volume aggregation.

However, these advances currently rely heavily on large-scale annotated datasets that require extensive manual annotation efforts. This is not conducive to real-world computer vision applications, as obtaining pixel-level annotations is expensive and time-consuming. To remedy this issue, the few-shot segmentation task has emerged and attracted more and more research attention [40,37,17,42,10,43]. For few-shot segmentation, only a handful of annotated samples (support samples) for each class are provided for the training of the model, which largely mitigates the reliance on large-scale manual annotations. Although an attempt has been made [1] to handle few-shot segmentation in a transductive way, most existing works model the few-shot segmentation in a meta-learning setting [26,34] to avoid overfitting brought by insufficient training data. As support samples are the most important guidance for the prediction of query samples, the key to few-shot segmentation is to effectively exploit information from the support set.

Prototype-based few-shot segmentation methods attempt to extract a prototype vector to represent the support sample [35,31,40]. Despite its effectiveness, compressing the features of support samples into a single vector is prone to noise and neglects the spatial structure of objects in the support image. To maintain the geometric structure of objects, visual correspondence-based methods [20,10,43] opt to utilize dense, pixel-wise semantic correlations between support and query images. Sophisticated pixel-wise attention-based methods [44] have also been devised to leverage dense correlations between support and query samples. However, directly using pixel-level dense information is a double-edged sword, which brings two new challenges: feature ambiguities caused by repetitive patterns, and background noise. To handle these issues, recent work [20] shows that aggregating dense correlation scores (also called *cost volume*) with 4D convolution can attain outstanding performance for few-shot segmentation.

To better model the interaction among correlation scores, [9] proposes to combine 4D convolutions and 4D swin Transformer [18] in a coarse-to-fine manner. Although recent cost aggregation-based methods [20,9] attain state-of-the-art performance, they still suffer from two limitations: 1) *lack of flexibility in computing correlation scores*, which results in an extremely slow convergence speed; 2) *lack of flexibility in aggregating high-dimensional correlation scores*, which leads to limited performance with high computational cost.

To tackle the first limitation, we propose a Gaussian process (GP) based kernel learning method to learn flexible kernel functions for a more accurate similarity measurement. We find that existing state-of-the-art methods [20,9] mainly focus on the cost volume aggregation module and directly rely on the hypercorrelation tensors constructed by the cosine similarity. However, we argue that directly using the cosine similarity lacks flexibility and cannot faithfully reflect the semantic similarities between support and query pixels. As shown in Fig. 1 (a), the match between the cosine similarity map and the ground truth label is poor. To handle this problem, we design a hard example mining mechanism to dynamically sample hard samples to train the covariance kernel functions. As shown in Fig. 1 (b), using the learned covariance kernels can generate a more reasonable similarity map and greatly improve the convergence speed.

Another limitation is mainly caused by the cost volume aggregation module. Prior works have shown that a better cost volume aggregation method can attain superior few-shot segmentation performance. However, the existing 4D convolution-based method [20] lacks the ability to model longer distance relations between elements in the cost volume. VAT [9] proposes to combine 4D convolutions with 4D swin Transformers [18] for cost volume aggregation. Although outstanding performance can be achieved, it requires a notable GPU memory consumption and suffers from a slow convergence speed. Considering this, we propose a doubly deformable 4D Transformer based aggregation method, with deformable attention mechanisms on both the support and query dimensions of the 4D cost volume input. Compared with 4D convolutions, it can not only model longer-distance interactions between pixels owing to the Transformer network design, but also learn to selectively attend to more informative regions of the support and query information.

To sum up, we propose a novel few-shot segmentation framework using doubly deformable aggregation of covariance matrices (DACM), which aggregates learnable covariance matrices with a doubly deformable 4D Transformer to predict segmentation results effectively. The proposed GP-based kernel learning can learn more accurate similarity measurements for cost volume generation. In what follows, the designed doubly deformable 4D Transformer can effectively and efficiently aggregate the multi-scale cost volume into the final segmentation result. Specifically, our contributions can be summarized as follows:

- Towards a more accurate similarity measurement, a GP-based kernel learning method is proposed to learn flexible covariance kernel functions, with a novel hard example mining mechanism. To our knowledge, we are the first to use

learnable covariance functions instead of cosine similarity for the few-shot segmentation task.
- Towards a more flexible cost volume aggregation, we propose a doubly deformable 4D Transformer (DDT) module, which utilizes deformable attention mechanisms on both the support and query dimensions of the 4D cost volume input. DDT can enhance representation learning by selectively attending to more informative regions.
- By combining these two modules, the proposed DACM method can attain new state-of-the-art performance on three public datasets with an extremely fast convergence speed. We also provide extensive visualization to better understand the proposed method.

2 Related Work

2.1 Few-shot Segmentation

Mainstream methods for few-shot segmentation can be roughly categorized into prototype-based methods [40,37,17,2] and correlation-based methods [42,10,43,36]. Prototype-based methods aim to generate a prototype representation [29] for each class based on the support sample, and then predict segmentation maps by measuring the distance between the prototype and the representations of the query image densely [45,2,37]. How to generate the prototype representation is the core of such methods. For example, a feature weighting and boosting model was proposed for prototype learning in [21]. Considering the limitations of using a single holistic representation for each class, a prototype learner was proposed in [40] to learn multiple prototypes based on the support images and the corresponding object masks. Then, the query images and the prototypes were used to generate the final object masks. To capture diverse and fine-grained object characteristics, Part-aware Prototype Network (PPNet) was developed in [17] to learn a set of part-aware prototypes for each semantic class. A prototype alignment regularization between support and query samples was proposed in Prototype Alignment Net-work (PANet) [37] to fully exploit semantic information from the support images and achieve better generalization on unseen classes. To handle the intra-class variations and inherent uncertainty, probabilistic latent variable-based prototype modeling was designed in [31,35], which leveraged probabilistic modeling to enhance the segmentation performance.

As prototype-based methods suffer from the loss of spatial structure due to average pooling [45], correlation-based methods were proposed to model the pair-wise semantic correspondences densely between the support and query images. For instance, Attention-based Multi-Context Guiding (A-MCG) network [10] was proposed to fuse the multi-scale context features extracted from the support and query branch. Specifically, a residual attention module was introduced into A-MCG to enrich the context information. To retain the structure representations of segmentation data, attentive graph reasoning [43] was exploited to transfer the class information from the support set to the query set, where element-to-element correspondences were captured at multiple semantic levels.

Democratic Attention Networks (DAN) was introduced in [36] for few-shot semantic segmentation, where the democratized graph attention mechanism was applied to establish a robust correspondence between support and query images. As proven in existing state-of-the-art methods, cost volume aggregation plays an important role in the few-shot segmentation task. To this end, 4D convolutions and Transformers were designed in [20,9] and achieved highly competitive few-shot segmentation results.

2.2 Gaussian Process

Recently, the combination of the Gaussian process and deep neural networks has drawn more and more research attention due to the success of deep learning [33,30,23,38]. For instance, scalable deep kernels were introduced in [38] to combine the non-parametric flexibility of GP and the structural properties of deep models. Furthermore, adaptive deep kernel learning [33] was proposed to learn a family of kernels for numerous few-shot regression tasks, which enabled the determination of the appropriate kernel for specific tasks. Deep Kernel Transfer (DKT) [23] was presented to learn a kernel that can be transferred to new tasks. Such a design can be implemented as a single optimizer and can provide reliable uncertainty measurement. GP has been introduced to few-shot segmentation in [11], where they proposed to learn the output space of the GP via a neural network that encoded the label mask. Although there is an attempt to incorporate GP, it is still in its infancy and far from fully exploiting the potential of GP for few-shot segmentation.

3 Methodology

The whole architecture of the proposed DACM framework is illustrated in Fig. 2. Given the input support and query images, multi-level deep features are first extracted by the fixed backbone network pre-trained on ImageNet [25]. Three levels of deep features with different spatial resolutions form a pyramidal design. For each level, average pooling is used to aggregate multiple features into a single one. Then, three different GP models are trained for computing the covariance matrices (4D cost volume) for each level. Next, the proposed Deformable 4D Transformer Aggregator (DTA) is combined with the weight-sparsified 4D convolution for cost volume aggregation. Finally, a decoder is used to predict the final segmentation result for the input query image.

3.1 Preliminaries: Gaussian Process

The covariance kernels of GP specify the statistical relationship between two points at the input space (*e.g.*, support and query features). Namely, the learned covariance matrices can be naturally used to measure the correlations between support and query samples. Formally, the data set consists of N samples of dimension D, *i.e.*, $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^D$ is a data point and y_i is the

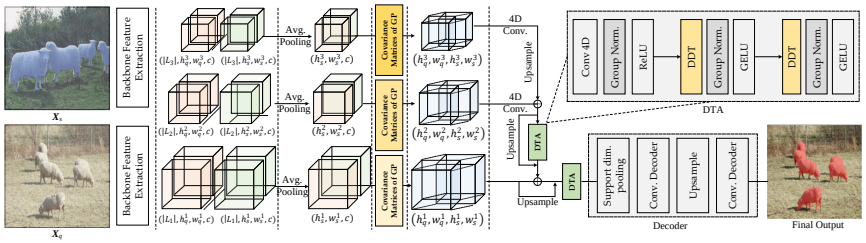

Fig. 2. The whole architecture of our proposed DACM framework. Under a pyramidal design, DACM consists of three parts: 1) feature extraction by the backbone network; 2) cost volume generation based on covariance kernels; 3) cost volume aggregation using DTA modules and 4D convolutions [20].

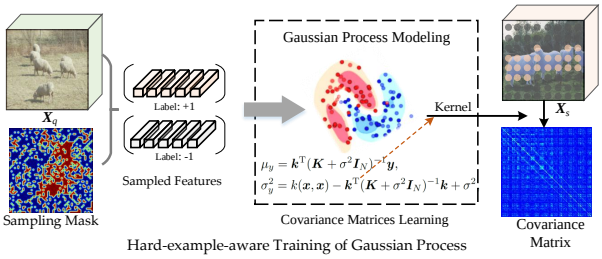

Fig. 3. Illustration of the GP-based kernel learning process. Owing to Gaussian modeling, covariance functions can measure the similarity in a continuous feature space.

corresponding label. The GP regression model assumes that the outputs y_i can be regarded as certain deterministic latent function $f(x_i)$ with zero-mean Gaussian noise ε, *i.e.*, $y_i = f(x_i) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. GP sets a zero-mean prior on f, with covariance $k(x_i, x_j)$. The covariance function k reflects the smoothness of f. The most widely-used covariance function is the Automatic Relevance Determination Squared Exponential (ARD SE), *i.e.*,

$$k(x_i, x_j) = \sigma_0^2 \exp\left\{-\frac{1}{2}\sum_{d=1}^{D}\frac{((x_i)_d - (x_j)_d)^2}{l_d^2}\right\}, \tag{1}$$

where $\sigma^2, \sigma_0^2, \{l_d\}_{d=1}^{D}$ are hyper-parameters. As shown in Fig. 3, after the optimization of hyper-parameters, we can compute the covariance matrices using the learned kernel k.

3.2 Hard Example Mining-based GP Kernel Learning

For few-shot segmentation, the whole dataset is split into the meta-training, meta-validation, and meta-testing subsets. Each *episode* [34,26] in the subsets consists of a support set $\mathcal{S} = \left\{(\boldsymbol{I}_s^i, \boldsymbol{M}_s^i)\right\}_{i=1}^{K}$ and a query image $\boldsymbol{I}_q$ of the same

class, where $\boldsymbol{I}_s^i \in \mathbb{R}^{3\times H\times W}$ is the support image and $\boldsymbol{M}_s^i \in \{0,1\}^{H\times W}$ is its binary object mask for a certain class. K is the number of the support images. We discuss the one-shot segmentation ($K=1$) case in the following sections.

Given the support image-mask pair $(\boldsymbol{I}_s, \boldsymbol{M}_s)$ and the query image-mask pair $(\boldsymbol{I}_q, \boldsymbol{M}_q)$ where $\boldsymbol{I}_s, \boldsymbol{I}_q \in \mathbb{R}^{3\times H\times W}$ and $\boldsymbol{M}_s, \boldsymbol{M}_q \in \{0,1\}^{H\times W}$, a pretrained backbone is used to extract features from $\boldsymbol{I}_s$ and $\boldsymbol{I}_q$, and the feature maps at layer l are denoted as $\boldsymbol{F}_s^l, \boldsymbol{F}_q^l \in \mathbb{R}^{c_l\times h_l\times w_l}$, where c_l, h_l, w_l denote that the features are of channel c_l, height h_l and width w_l. Note that we view the feature vectors at all spatial positions on feature maps of the query images as the training data. As shown in Fig. 3, our motivation is to dynamically select training samples from $\boldsymbol{F}_q^l$ for training the GP at layer l with a hard example mining mechanism. To this end, we propose a hard example aware sampling strategy based on the similarity (cost volume) between support and query samples $\boldsymbol{F}_s^l$ and $\boldsymbol{F}_q^l$. The 4D covariance matrices $\mathcal{C}^l \in \mathbb{R}^{h_l\times w_l\times h_l\times w_l}$, *i.e.*, the cost volume, can be computed with the GP using an initialized covariance kernel by

$$\mathcal{C}^l(i,j) = \mathrm{ReLU}\left(k\left(\frac{\boldsymbol{F}_q^l(i)}{\left\|\boldsymbol{F}_q^l(i)\right\|}, \frac{\boldsymbol{F}_s^l(j)}{\left\|\boldsymbol{F}_s^l(j)\right\|}\right)\right), \tag{2}$$

where i and j denote spatial positions of 2D feature maps. In what follows, we reshape the 4D cost volume into $\mathcal{C}^l \in \mathbb{R}^{h_l\times w_l\times h_l w_l}$ and sum up it along the third dimension. Then we can get a 2D similarity map $\mathcal{S}^l \in \mathbb{R}^{h_l\times w_l}$ of the query image. An example 2D similarity map can be found in Fig. 1. Ideally, we expect the high values in $\mathcal{S}^l$ to totally lie in the ground truth mask. The negative positions with high similarity values in the 2D similarity map $\mathcal{S}^l$ can be viewed as hard examples. Thus, we utilize $\mathcal{S}^l$ to generate a probability map for sampling the training samples from $\boldsymbol{F}_q^l$. Specifically, we can obtain a 2D probability map for sampling training data by

$$\hat{\mathcal{S}}^l = \mathcal{S}^l + \lambda \boldsymbol{M}_q^l, \tag{3}$$

$$p^l = \frac{\hat{\mathcal{S}}^l - \min(\hat{\mathcal{S}}^l)}{\max(\hat{\mathcal{S}}^l) - \min(\hat{\mathcal{S}}^l)} \tag{4}$$

$$t^l \sim \mathrm{Bernoulli}(p^l). \tag{5}$$

In Eq. 3, we add $\mathcal{S}^l$ and the scaled query mask $\boldsymbol{M}_q^l$ with λ to ensure that adequate number of positive samples can be selected for training at the early stage. Then, in Eq. 4, we normalize $\mathcal{S}^l$ to represent the probability that the sample will be selected or not. After the softmax operation, the 2D probability map $p^l \in [0,1]^{h_l\times w_l}$ can be obtained with each position representing the probability of being selected. Finally, a binary mask $t^l \in \{0,1\}^{h_l\times w_l}$ can be sampled from a Bernoulli distribution, where 0 means unselected and 1 means selected.

Such a sampling manner aims to sample query features which are likely to be incorrectly classified during the training stage. Then, we aim to use the sampled hard example-aware features on the query image to optimize the kernel. Suppose the sampled N training data at feature layer l is $\{(x_j^q, y_j^q)\}_{j=1}^N$ ($x_j^q \in \mathbb{R}^{c_l}$). Then

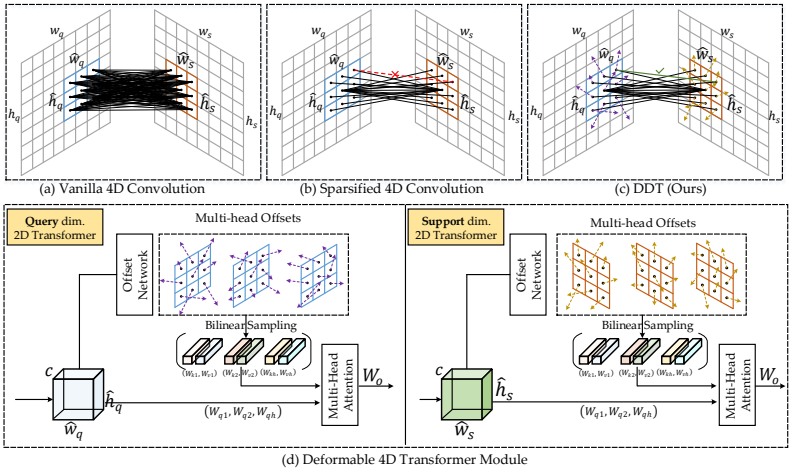

Fig. 4. Illustration of the proposed DDT module. DDT utilizes doubly deformable attention mechanisms on both the support and query dimensions of the 4D cost volume. As shown in (b), the red dashed line indicates that the path is dropped by weight sparsification. While the dropped path can be recovered with the proposed DDT module if it is important, as presented in (c).

these samples are used to learn the hyper-parameters $\left\{\sigma^2, \sigma_0^2, \{l_d\}_{d=1}^{D}\right\}$ of GP for computing the covariance matrices by maximizing the marginal likelihood. Please refer to **§ 4** of the supplementary material for a more detailed definition.

3.3 Doubly Deformable 4D Transformer for Cost Volume Aggregation

To clearly describe the proposed DDT module, we will first introduce the sparsified 4D convolution proposed in [20]. As shown in Fig. 4, weight-sparsified 4D convolution drops majority paths and only considers the activations at positions of either one of 2-dimensional centers. Formally, given 4D position $(\mathbf{u}, \mathbf{u}') \in \mathbb{R}^4$, by only considering its neighbors adjacent to either $\mathbf{u}$ or $\mathbf{u}'$, the weight-sparsified 4D convolution can be approximated by two 2-dimensional convolutions performed on 2D slices of hypercorrelation tensor. Different from prior works, the proposed DDT module aims to 1) utilize deformable attention mechanisms [39,48] on both the support and query dimensions of the 4D input to learn more flexible aggregation; 2) provide the ability to model longer distance relations between elements of the cost volume. Specifically, we can formulate the DDT module for aggregating 4D tensor input $\mathcal{C}$ (we omit layer l for simplicity) as follows:

$$\begin{aligned} \mathrm{DDT}(\mathcal{C}, \mathbf{u}, \mathbf{u}') = \mathrm{SDT}\left(\mathcal{C}, (\mathbf{u}, \mathbf{p}')\right) + \mathrm{QDT}(\mathcal{C}, (\mathbf{p}, \mathbf{u}')), \\ \mathbf{p} := \mathcal{P}\left(\mathbf{u}\right), \mathbf{p}' := \mathcal{P}\left(\mathbf{u}'\right), \end{aligned} \tag{6}$$

where $\mathbf{p}$ and $\mathbf{p}'$ denote the neighbour pixels centered at position $\mathbf{u}$ and $\mathbf{u}'$. SDT denotes the deformable Transformer on the support dimension (2D slice) of the cost volume. Similarly, QDT represents the deformable Transformer on the query dimension (2D slice) of $\mathcal{C}$.

As QDT shares similar computation process with SDT, we will only introduce the computation of SDT for the sake of simplicity. Suppose the 2D slice $\mathcal{C}(\mathbf{u}, \mathbf{p}') \in \mathbb{R}^{c \times h \times w}$ of the cost volume is with channel dimension c, height h and width w. Note that $c = 1$ in this case. We first normalize the coordinates to the range $[-1, +1]$. $(-1, -1)$ indicates the top-left corner and $(+1, +1)$ represents the bottom-right corner. Then, an offset network is utilized to generate n offset maps $\{\Delta\mathbf{p}_i{}'\}_{i=1}^{n}$ using two convolution layers. Each offset map has the spatial dimension (h, w) and is responsible for a head in the multi-head attention module. By adding the original coordinates $\mathbf{p}'$ and $\Delta\mathbf{p}'$, we can obtain the shifted positions for the 2D slice input. In what follows, the differentiable bilinear sampling is used to obtain vectors $\tilde{x} \in \mathbb{R}^{c \times h \times w}$ at the shifted positions by:

$$\tilde{x} = \phi(\mathcal{C}(\mathbf{u}, :), \mathbf{p}' + \Delta\mathbf{p}'). \tag{7}$$

Then, we project the sample feature vectors into keys $k \in \mathbb{R}^{c' \times hw}$ and values $v \in \mathbb{R}^{c' \times hw}$ using learnable W_k and W_v as follows: $k = \tilde{x}W_k$, $v = \tilde{x}W_v$. Where c' is the projected channel dimension. Next, the query tokens $\mathcal{C}(\mathbf{u}, \mathbf{p}')$ in the query features are projected to queries $q \in \mathbb{R}^{c' \times hw}$. For each head of the multi-head attention, we can compute the output $z \in \mathbb{R}^{c' \times h \times w}$ using the self-attention mechanism as follows:

$$q = \mathcal{C}(\mathbf{u}, \mathbf{p}')\, W_q,\ \ z = \text{softmax}\left(qk^{\mathrm{T}}/\sqrt{d}\right) v, \tag{8}$$

where d denotes the dimension of each head, W_q is a learnable matrix. The whole computation process of DDT is illustrated in Fig. 4. To sum up, the proposed light-weight, plug-and-play DDT module utilizes flexible deformable modeling to compensate the dropped informative activations in weight-sparsified 4D convolution. It can also enjoy the long-distance interactions modeling brought by Transformers.

4 Experiments

4.1 Experiment Settings

Datasets Three few-shot segmentation datasets are exploited to evaluate the proposed method: PASCAL-5^i [26], COCO-20^i [14], and FSS-1000 [13]. PASCAL-5^i is re-created from the data in PASCAL VOC 2012 [4] and the augmented mask annotations from [7]. PASCAL-5^i contains 20 object classes and is split into 4 folds. COCO-20^i consists of 80 object classes and is also split into 4 folds. Following prior experimental settings [17,36,21,32] on PASCAL-5^i and COCO-20^i datasets, for each fold i, the data in the rest folds are used for training, and 1,000 episodes randomly sampled from the fold i are used for evaluation. FSS-1000 contains 1,000 classes and is split into 520, 240, and 240 classes for training, validation, and testing, respectively.

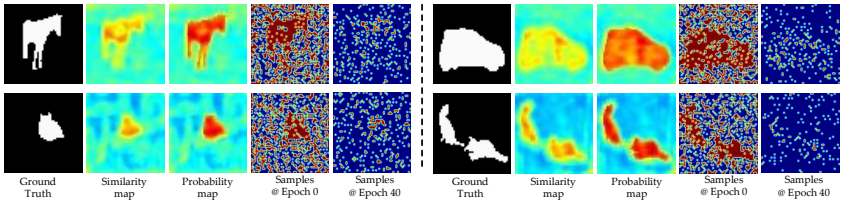

Fig. 5. Some qualitative examples of the proposed hard example-aware sampling strategy for training the GP models.

Evaluation Metrics Mean intersection over union (mIoU) and foreground-background IoU (FB-IoU) are reported for comparisons. mIoU is the average of IoU over all classes in a fold, *i.e.*, $\text{mIoU} = \frac{1}{|\mathcal{C}_{test}|}\sum_{c\in\mathcal{C}}\text{IoU}_c$. As for FB-IoU, the object classes are not considered in this metric. The average of foreground IoU and background IoU is computed, *i.e.*, $\text{FB-IoU} = \frac{1}{2}(\text{IoU}_F + \text{IoU}_B)$. Compared with FB-IoU, mIoU can better reflect the generalization ability of segmentation methods to unseen classes.

Implementation Details The proposed method is implemented using PyTorch [22] and GPyTorch [5]. VGG16 [28], ResNet50 and ResNet101 [8] pre-trained on ImageNet are adopted as the backbone networks. To construct pyramidal features, three levels of features with resolutions of 50×50, 25×25, 13×13 (12×12 for VGG backbone) are used. The GP models and cost volume aggregation models are trained in an end-to-end manner. Adam optimizer [12] is used for the optimization of the covariance matrices aggregation part as well as the GP models. The initial learning rate for the covariance matrices aggregation part is set to $1e-3$, while the learning rate for three GP models is set to $1e-2$. Only **50 epochs** are used for training the proposed DACM on all three datasets. It is worth noting that 50 epochs are significantly less than existing methods that usually require more than 200 epochs for model training.

4.2 Results and Analysis

PASCAL-5^i We compare our method with existing state-of-the-art methods. Table 1 presents the 1-shot and 5-shot segmentation results. "DACM (Ours)" demotes the model presented in Fig. 2. Although DACM is trained with only **50 epochs**, it can still significantly outperform existing state-of-the-art models [20,9] with three different backbones. The comparison results demonstrate that the proposed GP-based kernel learning and DDT module are beneficial for few-shot segmentation task. In addition, we also visualize the sampled locations (described in Eq. 5) during the GP kernel training in Fig. 5 for a better understanding of the designed hard example-aware sampling strategy. It can be seen that hard examples are selected at the later stage of the training process. Some

Table 1. Performance comparisons on PASCAL-5^i dataset in mIoU and FB-IoU.

Backbone	Methods	1-shot Segmentation						5-shot Segmentation						#Learnable
		5^0	5^1	5^2	5^3	Mean	FB-IoU	5^0	5^1	5^2	5^3	Mean	FB-IoU	Params
VGG16 [28]	co-FCN [24]	36.7	50.6	44.9	32.4	41.1	60.1	37.5	50.0	44.1	33.9	41.4	60.2	34.2M
	AMP-2 [27]	41.9	50.2	46.7	34.7	43.4	61.9	40.3	55.3	49.9	40.1	46.4	62.1	15.8M
	PANet [37]	42.3	58.0	51.1	41.2	48.1	66.5	51.8	64.6	59.8	46.5	55.7	70.7	14.7M
	PFENet [32]	56.9	**68.2**	54.4	52.4	58.0	72.0	59.0	69.1	54.8	52.9	59.0	72.3	10.4M
	HSNet [20]	59.6	65.7	59.6	54.0	59.7	73.4	64.9	69.0	64.1	58.6	64.1	76.6	**2.6M**
	DACM (Ours)	**61.8**	67.8	**61.4**	**56.3**	**61.8**	**75.5**	**66.1**	**70.6**	**65.8**	**60.2**	**65.7**	**77.8**	3.0M
ResNet50 [8]	PPNet [17]	48.6	60.6	55.7	46.5	52.8	69.2	58.9	68.3	66.8	58.0	63.0	75.8	31.5M
	PFENet [32]	61.7	69.5	55.4	56.3	60.8	73.3	63.1	70.7	55.8	57.9	61.9	73.9	10.8M
	RePRI [1]	59.8	68.3	62.1	48.5	59.7	—	64.6	71.4	**71.1**	59.3	66.6	—	—
	HSNet [20]	64.3	70.7	60.3	60.5	64.0	76.7	70.3	73.2	67.4	67.1	69.5	80.6	**2.6M**
	CyCTR [44]	67.8	72.8	58.0	58.0	64.2	—	71.1	73.2	60.5	57.5	65.6	—	—
	VAT [9]	67.6	71.2	62.3	60.1	65.3	77.4	72.4	73.6	68.6	65.7	70.0	80.9	3.2M
	DACM (Ours)	66.5	72.6	62.2	61.3	65.7	77.8	72.4	73.7	69.1	**68.4**	70.9	81.3	3.0M
	DACM (VAT)	**68.4**	**73.1**	**63.5**	**62.2**	**66.8**	**78.6**	**73.8**	**74.7**	70.3	68.1	**71.7**	**81.7**	3.3M
ResNet101 [8]	FWB [21]	51.3	64.5	56.7	52.2	56.2	—	54.8	67.4	62.2	55.3	59.9	—	43.0M
	PPNet [17]	52.7	62.8	57.4	47.7	55.2	70.9	60.3	70.0	69.4	60.7	65.1	77.5	50.5M
	DAN [36]	54.7	68.6	57.8	51.6	58.2	71.9	57.9	69.0	60.1	54.9	60.5	72.3	—
	PFENet [32]	60.5	69.4	54.4	55.9	60.1	72.9	62.8	70.4	54.9	57.6	61.4	73.5	10.8M
	RePRI [1]	59.6	68.6	62.2	47.2	59.4	—	66.2	71.4	67.0	57.7	65.6	—	—
	HSNet [20]	67.3	72.3	62.0	63.1	66.2	77.6	71.8	74.4	67.0	68.3	70.4	80.6	**2.6M**
	CyCTR [44]	69.3	72.7	56.5	58.6	64.3	72.9	73.5	74.0	58.6	60.2	66.6	75.0	—
	VAT [9]	68.4	72.5	64.8	64.2	67.5	78.8	73.3	75.2	68.4	69.5	71.6	82.0	3.3M
	DACM (Ours)	68.7	73.5	63.4	64.2	67.5	78.9	72.7	75.3	68.3	69.2	71.4	81.5	3.1M
	DACM (VAT)	**69.9**	**74.1**	**66.2**	**66.0**	**69.1**	**79.4**	**74.2**	**76.4**	**71.1**	**71.6**	**73.3**	**83.1**	3.4M

Table 2. Performance comparisons on COCO-20^i dataset in mIoU and FB-IoU.

Backbone	Methods	1-shot Segmentation						5-shot Segmentation					
		5^0	5^1	5^2	5^3	Mean	FB-IoU	5^0	5^1	5^2	5^3	Mean	FB-IoU
ResNet50 [8]	PMM [40]	29.3	34.8	27.1	27.3	29.6	—	33.0	40.6	30.3	33.3	34.3	—
	RPMM [40]	29.5	36.8	28.9	27.0	30.6	—	33.8	42.0	33.0	33.3	35.5	—
	PFENet [32]	36.5	38.6	35.0	33.8	35.8	—	36.5	43.3	38.0	38.4	39.0	—
	RePRI [1]	32.0	38.7	32.7	33.1	34.1	—	39.3	45.4	39.7	41.8	41.6	—
	HSNet [20]	36.3	43.1	38.7	38.7	39.2	68.2	43.3	51.3	48.2	45.0	46.9	70.7
	CyCTR [44]	38.9	43.0	39.6	39.8	40.3	—	41.1	48.9	45.2	47.0	45.6	—
	VAT [9]	39.0	43.8	42.6	39.7	41.3	68.8	44.1	51.1	50.2	46.1	47.9	72.4
	DACM (Ours)	37.5	44.3	40.6	40.1	40.6	68.9	44.6	52.0	49.2	46.4	48.1	71.6
	DACM (VAT)	**41.2**	**45.2**	**44.1**	**41.3**	**43.0**	**69.4**	**45.2**	**52.2**	**51.5**	**47.7**	**49.2**	**72.9**

comparative visualization results of "DACM (Ours)" model on this dataset are presented in Fig. 6.

The proposed GP-based kernel learning and DDT module can be plugged in any cost volume aggregation-based model. "DACM+VAT" denotes the combination of DACM with VAT by 1) replacing the cosine similarity-based cost volume with our covariance matrices; 2) replacing the last 4D convolution layer with our DDT. We can see that the combined method "DACM (VAT)" can achieve better segmentation performance than others. This demonstrates the superiority of the proposed framework.

COCO-20^i The comparison results on the COCO-20^i dataset are reported in Table 2. The results reveal that "DACM (Ours)" clearly outperform the HSNet baseline under both 1-shot and 5-shot settings. "DACM (VAT)" model achieves

Table 3. Performance Comparisons on the FSS-1000 dataset.

Backbone	Methods	mIoU 1-shot	mIoU 5-shot
ResNet50 [8]	FSOT [16]	82.5	83.8
	HSNet [20]	85.5	87.8
	VAT [9]	89.5	90.3
	DACM (Ours)	**90.7**	**91.6**
ResNet101 [8]	DAN [36]	85.2	88.1
	HSNet [20]	86.5	88.5
	VAT [9]	90.0	90.6
	DACM (Ours)	**90.8**	**91.7**

Table 4. Ablation study on PASCAL-5^i dataset.

Backbone	Methods	mIoU 1-shot	mIoU 5-shot
VGG16 [28]	Baseline [20]	59.6	64.9
	+ GP-KL	60.8	65.2
	+ DDT-1	61.2	65.7
	+ DDT-2	**61.8**	**66.1**
	+ DDT-3	61.6	65.8

Fig. 6. Some visualization examples of the proposed method for few-shot segmentation. We compare DACM with the baseline (HSNet) method.

the best results compared with existing methods. COCO-20^i is a more difficult few-shot segmentation dataset. Basically, taking HSNet [20] as the baseline, DACM can achieve an clear improvement. It is worth noting that DACM requires less epochs for the model training, while it can still outperform VAT [9] and set new state-of-the-art results on the COCO-20^i dataset.

FSS-1000 FSS-1000 is simpler than the other two datasets. The comparison results are shown in Table 3. Under both the 1-shot setting and 5-shot setting, DACM can obtain a clear new state-of-the-art performance in terms of mIoU. This indicates the effectiveness of our method. Comparison results on these three public datasets demonstrate the effectiveness of the proposed DACM method.

4.3 Ablation Study and Analysis

Ablation Study To validate the effects of the proposed two modules, *i.e.*, GP-based kernel learning (GP-KL) module and DDT module, we conduct ablation studies. In addition, we compare the performance of using 1, 2 and 3 DDT layers (DDT-1, DDT-2 and DDT-3) to further study the effects of the number of DDT layers. The mIoU results on the PASCAL-5^i dataset are presented in Table 4.

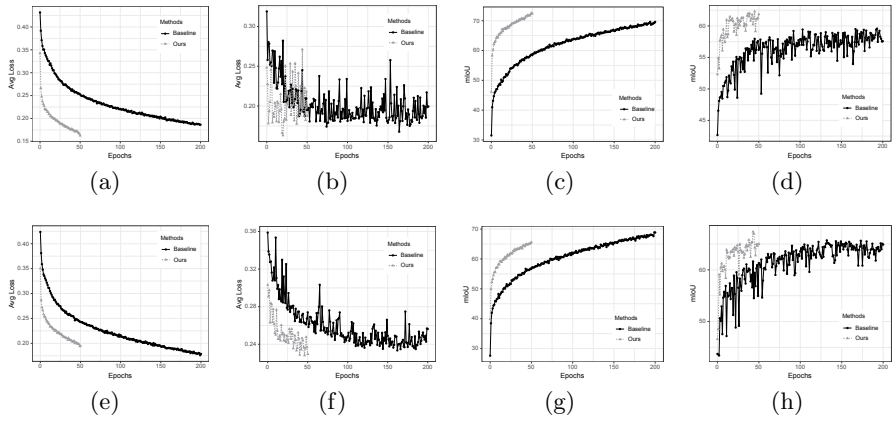

Fig. 7. (a), (b), (e), and (f) show the training/validation loss curves on the fold 0 and fold 1 of PASCAL-5i dataset. (c), (d), (g), and (h) present the training/validation mIoU curves.

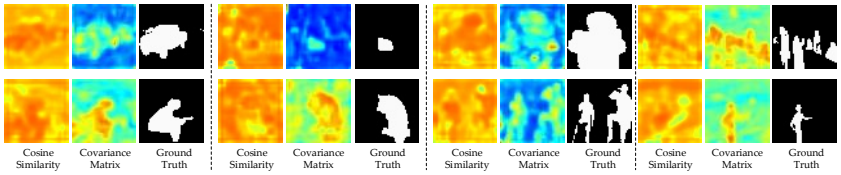

Fig. 8. Visualization of some examples of the covariance matrices. Compared with the commonly used cosine similarity, our method can learn more reasonable similarity measurements for few-shot segmentation.

We can see that using 2 DDT layers can obtain the best result. The reason is that more DDT layers may bring the overfitting problem on several folds of the dataset. We also find that combining the GP-based kernel learning and DDT modules together results in the best few-shot segmentation performance.

Convergence Speed In Fig. 7, we plot the loss curves and mIoU values during the training process on two folds of the PASCAL-5^i dataset. Obviously, our DACM can converge much faster than the baseline method (HSNet). Although trained with only 50 epochs, DACM can obtain much better training and validation mIoU performance than the baseline method. This clearly demonstrate the effectiveness of the proposed modules.

Covariance Matrices Visualization To better understand the proposed DACM method, we further visualize and compare the learned covariance matrices. As shown in Fig. 8, our method can get more reasonable similarity maps than cosine similarity. This also explains why DACM can converge faster than existing

Fig. 9. Visualization of some examples of the learned deformable offsets for both the query and support samples.

methods. Furthermore, we also visualize the learned deformable offsets for the support and query dimensions in Fig. 9. It can be seen that the deformable attention mechanism tends to attend on informative regions to learn more powerful representations.

5 Conclusions

In this paper, we present a few-shot segmentation method by doubly deformable aggregating covariance matrices. The proposed method aggregates learnable covariance matrices with a doubly deformable 4D Transformer to predict segmentation results effectively. Specifically, we make the following contributions. 1) We devise a novel hard example mining mechanism for learning covariance kernels of Gaussian process to enable a more accurate correspondence measurement. 2) We design a doubly deformable 4D Transformer to effectively and efficiently aggregate the multi-scale cost volume into the final segmentation results. 3) By combining these two modules, the proposed method can achieve state-of-the-art few-shot segmentation performance with a fast convergence speed.

6 Acknowledgement

This work is jointly supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. [ERC-2016-StG-714087], Acronym: *So2Sat*), by the Helmholtz Association through the Framework of Helmholtz AI (grant number: ZT-I-PF-5-01) - Local Unit "Munich Unit @Aeronautics, Space and Transport (MASTr)" and Helmholtz Excellent Professorship "Data Science in Earth Observation - Big Data Fusion for Urban Research"(grant number: W2-W3-100), by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI lab "AI4EO – Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond" (grant number: 01DD20001) and by German Federal Ministry of Economics and Technology in the framework of the "national center of excellence ML4Earth" (grant number: 50EE2201C).

References

1. Boudiaf, M., Kervadec, H., Masud, Z.I., Piantanida, P., Ben Ayed, I., Dolz, J.: Few-shot segmentation without meta-learning: A good transductive inference is all you need? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13979–13988 (2021)
2. Dong, N., Xing, E.P.: Few-shot semantic segmentation with prototype learning. In: BMVC. vol. 3 (2018)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
4. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1), 98–136 (2015)
5. Gardner, J.R., Pleiss, G., Bindel, D., Weinberger, K.Q., Wilson, A.G.: Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. arXiv preprint arXiv:1809.11165 (2018)
6. Hao, S., Zhou, Y., Guo, Y.: A brief survey on semantic segmentation with deep learning. Neurocomputing **406**, 302–321 (2020)
7. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: European conference on computer vision. pp. 297–312. Springer (2014)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
9. Hong, S., Cho, S., Nam, J., Kim, S.: Cost aggregation is all you need for few-shot segmentation. arXiv preprint arXiv:2112.11685 (2021)
10. Hu, T., Yang, P., Zhang, C., Yu, G., Mu, Y., Snoek, C.G.: Attention-based multi-context guiding for few-shot semantic segmentation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 8441–8448 (2019)
11. Johnander, J., Edstedt, J., Felsberg, M., Khan, F.S., Danelljan, M.: Dense gaussian processes for few-shot segmentation. arXiv preprint arXiv:2110.03674 (2021)
12. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
13. Li, X., Wei, T., Chen, Y.P., Tai, Y.W., Tang, C.K.: Fss-1000: A 1000-class dataset for few-shot segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2869–2878 (2020)
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
15. Liu, H., Dai, Z., So, D., Le, Q.: Pay attention to mlps. Advances in Neural Information Processing Systems **34** (2021)
16. Liu, W., Zhang, C., Ding, H., Hung, T.Y., Lin, G.: Few-shot segmentation with optimal transport matching and message flow. arXiv preprint arXiv:2108.08518 (2021)
17. Liu, Y., Zhang, X., Zhang, S., He, X.: Part-aware prototype network for few-shot semantic segmentation. In: European Conference on Computer Vision. pp. 142–158. Springer (2020)

18. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
19. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
20. Min, J., Kang, D., Cho, M.: Hypercorrelation squeeze for few-shot segmentation. arXiv preprint arXiv:2104.01538 (2021)
21. Nguyen, K., Todorovic, S.: Feature weighting and boosting for few-shot segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 622–631 (2019)
22. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32**, 8026–8037 (2019)
23. Patacchiola, M., Turner, J., Crowley, E.J., O'Boyle, M., Storkey, A.J.: Bayesian meta-learning for the few-shot setting via deep kernels. Advances in Neural Information Processing Systems **33** (2020)
24. Rakelly, K., Shelhamer, E., Darrell, T., Efros, A., Levine, S.: Conditional networks for few-shot semantic segmentation (2018)
25. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
26. Shaban, A., Bansal, S., Liu, Z., Essa, I., Boots, B.: One-shot learning for semantic segmentation. arXiv preprint arXiv:1709.03410 (2017)
27. Siam, M., Oreshkin, B., Jagersand, M.: Adaptive masked proxies for few-shot segmentation. arXiv preprint arXiv:1902.11123 (2019)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. arXiv preprint arXiv:1703.05175 (2017)
30. Snell, J., Zemel, R.: Bayesian few-shot classification with one-vs-each pólya-gamma augmented gaussian processes. arXiv preprint arXiv:2007.10417 (2020)
31. Sun, H., Lu, X., Wang, H., Yin, Y., Zhen, X., Snoek, C.G., Shao, L.: Attentional prototype inference for few-shot semantic segmentation. arXiv preprint arXiv:2105.06668 (2021)
32. Tian, Z., Zhao, H., Shu, M., Yang, Z., Li, R., Jia, J.: Prior guided feature enrichment network for few-shot segmentation. IEEE Transactions on Pattern Analysis & Machine Intelligence (01), 1–1 (2020)
33. Tossou, P., Dura, B., Laviolette, F., Marchand, M., Lacoste, A.: Adaptive deep kernel learning. arXiv preprint arXiv:1905.12131 (2019)
34. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29**, 3630–3638 (2016)
35. Wang, H., Yang, Y., Cao, X., Zhen, X., Snoek, C., Shao, L.: Variational prototype inference for few-shot semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 525–534 (2021)
36. Wang, H., Zhang, X., Hu, Y., Yang, Y., Cao, X., Zhen, X.: Few-shot semantic segmentation with democratic attention networks. In: Computer Vision–ECCV

2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16. pp. 730–746. Springer (2020)
37. Wang, K., Liew, J.H., Zou, Y., Zhou, D., Feng, J.: PANet: Few-shot image semantic segmentation with prototype alignment. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9197–9206 (2019)
38. Wilson, A.G., Hu, Z., Salakhutdinov, R., Xing, E.P.: Deep kernel learning. In: Artificial intelligence and statistics. pp. 370–378. PMLR (2016)
39. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. arXiv preprint arXiv:2201.00520 (2022)
40. Yang, B., Liu, C., Li, B., Jiao, J., Ye, Q.: Prototype mixture models for few-shot semantic segmentation. In: European Conference on Computer Vision. pp. 763–778. Springer (2020)
41. Yang, Y., Soatto, S.: Fda: Fourier domain adaptation for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4085–4095 (2020)
42. Yang, Y., Meng, F., Li, H., Wu, Q., Xu, X., Chen, S.: A new local transformation module for few-shot segmentation. In: International Conference on Multimedia Modeling. pp. 76–87. Springer (2020)
43. Zhang, C., Lin, G., Liu, F., Guo, J., Wu, Q., Yao, R.: Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9587–9595 (2019)
44. Zhang, G., Kang, G., Yang, Y., Wei, Y.: Few-shot segmentation via cycle-consistent transformer. Advances in Neural Information Processing Systems **34** (2021)
45. Zhang, X., Wei, Y., Yang, Y., Huang, T.S.: Sg-one: Similarity guidance network for one-shot semantic segmentation. IEEE transactions on cybernetics **50**(9), 3855–3865 (2020)
46. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6881–6890 (2021)
47. Zhu, F., Zhu, Y., Zhang, L., Wu, C., Fu, Y., Li, M.: A unified efficient pyramid transformer for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2667–2677 (2021)
48. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)