Delving Deep into Many-to-many Attention for Few-shot Video Object Segmentation

Haoxin Chen[1], Hanjie Wu[1], Nanxuan Zhao[2], Sucheng Ren[1], Shengfeng He[1*]
[1] School of Computer Science and Engineering, South China University of Technology
[2] The Chinese University of Hong Kong

Abstract

This paper tackles the task of Few-Shot Video Object Segmentation (FSVOS), i.e., segmenting objects in the query videos with certain class specified in a few labeled support images. The key is to model the relationship between the query videos and the support images for propagating the object information. This is a many-to-many problem and often relies on full-rank attention, which is computationally intensive. In this paper, we propose a novel Domain Agent Network (DAN), breaking down the full-rank attention into two smaller ones. We consider one single frame of the query video as the domain agent, bridging between the support images and the query video. Our DAN allows a linear space and time complexity as opposed to the original quadratic form with no loss of performance. In addition, we introduce a learning strategy by combining meta-learning with online learning to further improve the segmentation accuracy. We build a FSVOS benchmark on the Youtube-VIS dataset and conduct experiments to demonstrate that our method outperforms baselines on both computational cost and accuracy, achieving the state-of-the-art performance. Code is available at https://github.com/scutpaul/DANet.

1. Introduction

With the number of online videos increasing rapidly, Video Object Segmentation (VOS) attracts more and more attention as an important step to various video applications, such as video retrieval and editing [52]. Based on the user interaction, existing VOS algorithms have two common settings: unsupervised VOS and semi-supervised VOS. As shown in Fig. 1, unsupervised VOS [1, 17, 47, 13, 19, 23] directly segments primary objects in the videos without any human intervention. The objects often localize in salient regions. In contrast, semi-supervised VOS [2, 39, 5, 27] gives the ground truth segmentation of the first frame and propagates the labeled object information into subsequent frames. However, it requires pixel-level annotation of the first frame for each individual video, which limits the scalability for processing massive amount of videos. While interactive VOS [28, 11] further reduces the required human efforts to a few strokes, the provided information may be too coarse for cross-frame segmentation. To trade-off between

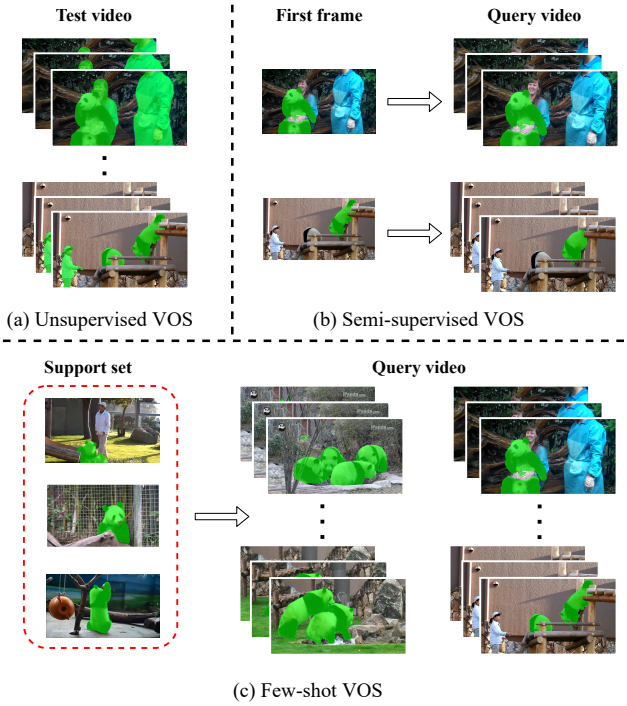

Figure 1: Problem settings for different VOS tasks. (a) Unsupervised VOS segments salient objects without guidance. (b) Semi-supervised VOS segments specified objects in a video given the segmentation mask for the first frame. (c) Few-shot VOS segments objects across videos with the same category as objects in the labeled support set.

*Corresponding author (hesfe@scut.edu.cn).

semantics-aware segmentation and cross-video processing, recent studies [16, 32, 34] leverage new interactions such as natural languages and class labels.

In this paper, we target on Few-Shot Video Object Segmentation (FSVOS), which is still under explored. FSVOS aims to segment new object classes across query videos with only a few annotated support images (Fig. 1). The support images can be selected randomly outside the query videos. FSVOS is able to balance between semantics-aware segmentation and cross-video processing, meeting the demand of applications with surging online videos in real case.

The key to solving FSVOS is using labeled support images for guiding the semantics-aware segmentation. Constructing correlation among support images and query videos is a many-to-many problem. There are two major solutions in recent studies, as shown in Fig. 2. The prototype-based [7, 25, 50] methods convert it into a one-to-many problem by extracting a class prototype (i.e., a global descriptor) from the support images, which inevitably loses structural information of the support images. The attention-based [40, 55, 27, 43] methods fully utilize the labeled support images and learn a many-to-many attention between every support-query image pairs. However, the computational complexity grows exponentially as the number of input images grows.

After delving deep into many-to-many attention, we find that the attention can be decomposed to reduce computational cost and thus introduce the following hypothesis. Given a query q and a pair of key-value k-v, we obtain the attention feature v^A using the typical attention function:

$$v^A = Attention(q, k, v) = Av = \sigma(\frac{q(k)^{\mathrm{T}}}{\sqrt{C_k}})v, \quad (1)$$

where A is the attention matrix, and σ is a softmax operation. A scale factor of $\frac{1}{\sqrt{C_k}}$ is to maintain the stability of the numerical scale.

Attention Decomposition Hypothesis. We hypothesize that the original attention matrix can be replaced with a product of two smaller attention matrices through an agent t. The new attention matrix $\hat{A}$ is expressed as:

$$\hat{A} = A^{qt}A^{tk}, \quad (2)$$

where A^{qt} is the attention between the query q and the agent t, and A^{tk} is the attention between the agent t and the key k, defined as:

$$A^{qt} = \sigma(\frac{q(k^t)^{\mathrm{T}}}{\sqrt{C_k}}); A^{tk} = \sigma(\frac{q^t(k)^{\mathrm{T}}}{\sqrt{C_k}}). \quad (3)$$

We provide a theoretical support for the above hypothesis, and propose a novel Domain Agent Network (DAN) accordingly. We treat a single frame of the query video as the agent for this video domain. Our method is possible to convert the previous exponential growth of computational complexity into a linear one. Due to the addition of invisible channel attention to enhance the query features, our attention module shows better results than the full-rank attention in both theoretical and practical analyses.

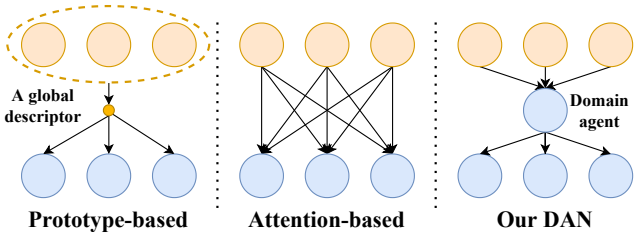

Figure 2: Different solutions for solving many-to-many problem. Orange circles represent the support images, and blue circles represent the query frames. The prototype-based methods covert it into a one-to-many problem by extracting a global descriptor from the support images. The attention-based methods use a full-rank attention to learn many-to-many mapping. Our Domain Agent Attention decomposes the full-rank attention using a domain agent.

Furthermore, we propose a learning strategy for FSVOS by combining meta-learning with online learning. We use meta-learning in the training phase to learn generic object segmentation across categories. While in the testing phase, we use online learning to update the feature representation for the unseen category. We construct a new benchmark for FSVOS based on the Youtube-VIS dataset. We demonstrate the feasibility of our method through ablation studies and compare its performance with several few-shot semantic segmentation methods. Experimental results show that our proposed method outperforms existing methods in terms of both segmentation accuracy and computational cost.

The main contributions of this work are fourfold.

- We delve into the conventional many-to-many attention and prove that the original attention matrix can be replaced by the product of two smaller attention matrices bridged by an agent.
- We work on an under-explored task called few-shot video object segmentation, and propose a novel domain agent network based on theoretical support, balancing the accuracy, computational burden, and speed.
- We present a learning strategy that combines meta-learning and online learning to improve the generalization ability of segmentation and category-specific feature representations.
- We build up the fist FSVOS benchmark and compare our model with existing methods to show its efficiency over both accuracy and the computational cost.

2. Related Work

2.1. Video Object Segmentation

Traditional unsupervised VOS methods rely on heuristic rules, such as point trajectories [1, 8, 26], object proposals [17, 18], and saliency [47] to segment the primary objects in the video. With the development of the deep learning and appearance of large-scale annotated video dataset [30], many works [13, 38, 19, 35, 23, 57] learn to segment from the labeled data under the zero-shot setting, i.e., segmenting objects without human supervision during test time. Recently, Lu *et al*. [24] propose a method for VOS by learning from unlabeled videos.

Caelles *et al*. [2] is the first work on semi-supervised VOS, learning to propagate the labeled object in the first frame into the subsequent frames for the query video. The propagation-based methods [5, 39, 49] use the consistency of the object motion for learning, while the matching-based methods [53, 54, 46] use different approaches to find the best correspondence between the first frame and the query frame for propagation. Besides, other methods use the memory network [27, 22] to store the features of the previous frames for a video, which helps to segment object over time. Furthermore, works in [16, 32, 34] utilize weakly-supervised information, such as natural language and class labels, to segment objects across videos.

2.2. Few-shot Semantic Segmentation

Few-shot semantic segmentation aims to learn segmentation for the novel class only from a few examples. Shaban *et al*.[33] propose a two-branch network consisting of the conditioning branch and the segmentation branch. The feature extracted from the support images by the conditioning branch guides the segmentation of the query images on the segmentation branch. Some methods [7, 25, 44, 50] are based on the idea of prototype from metric learning to solve the problem. Recent works [55, 43] leverage the graph attention operation to obtain the attention feature for guiding the segmentation. Besides, Tian *et al*.[37] propose to adaptively enhance the query features with training-free prior mask to overcome improper usage of high-level information from training classes.

2.3. Many-to-many Attention

Attention mechanisms have recently received much research attention due to the excellent performance. Many-to-many attention is applied in many tasks depending on the use of query and key. Vaswani *et al*. [40] propose to learn self-attention in the feature space. The memory-based VOS methods [27, 32] leverage the many-to-many attention to learn the guidance information from the memory features to the query features. The graph attention methods [41, 55, 43] learn a graph matching attention by modeling the input images as a dichotomous graph. However, directly using these full-rank attentions in FSVOS suffers from expensive computational cost, especially when the number of processing images increases. We instead introduce a method to decompose the full-rank attention with the theoretical guarantee.

2.4. Online Learning

Some methods [2, 42, 29, 20] apply online learning to solve semi-supervised VOS problem for improving performance during the test time. However, it is time-consuming to learn a video-specific representation. Our framework also uses online training to improve performance. The difference from the previous methods is that we fine-tune on each unseen category instead of each video.

3. Theoretical Support for the Hypothesis

Before introducing our domain agent network, we first provide a proof for the attention decomposition hypothesis discussed in the Introduction.

The regular attention [40] is a kind of dot-then-exponentiate function $K(x, y) = exp(xy)$, and the softmax function can be seen as adding the normalization into this non-linear kernel function $K(x, y)$. For a period of time, the studies [31, 15, 9] in approximating non-linear kernels, convert the problem to find the mapping function φ as:

$$K(x, y) = \mathbb{E}[\varphi(x)\varphi(y)]. \tag{4}$$

Recently, the study [4] proposes to use the random feature map [31] for building the map estimator for the softmax-kernel. More concretely, it has been proven that a positive random feature mapping function can be used to approximate the softmax-kernel function without bias, expressed as follows:

$$\varphi(x) = \frac{1}{\sqrt{m}} exp(-\frac{\|x\|^2}{2})(exp(\omega_1^{\mathrm{T}} x), ..., exp(\omega_m^{\mathrm{T}} x)), \tag{5}$$

where deterministic vectors $\omega_1, .., \omega_m \overset{iid}{\sim} \mathcal{N}(0, \mathbf{I}_C)$ are randomly sampled. These randomly sampled vectors are used to map the original features $x \in \mathbb{R}^C$ from C-dimension into a m-dimensional space, where the computed inner product of $\varphi(x) \in \mathbb{R}^m$ and $\varphi(y) \in \mathbb{R}^m$ can approximate the softmax-kernel $SM(x, y)$.

We adapt the above function of Eq. 4 to decompose the full-rank attention matrix into the following form:

$$A = \sigma(\frac{q(k)^{\mathrm{T}}}{\sqrt{C_k}}) = \mathbb{E}[q'(k')^{\mathrm{T}}], \tag{6}$$

where $q', k' \in \mathbb{R}^{L \times m}$ are composed of rows $\varphi(q_i^{\mathrm{T}})^{\mathrm{T}}$, $\varphi(k_i^{\mathrm{T}})^{\mathrm{T}}$ respectively.

In the same way, we can approximate the two attention matrices of Eq. 3 as:

$$A^{qt} = \mathbb{E}[q'((k^t)')^{\mathrm{T}}]; A^{tk} = \mathbb{E}[(q^t)'(k')^{\mathrm{T}}], \tag{7}$$

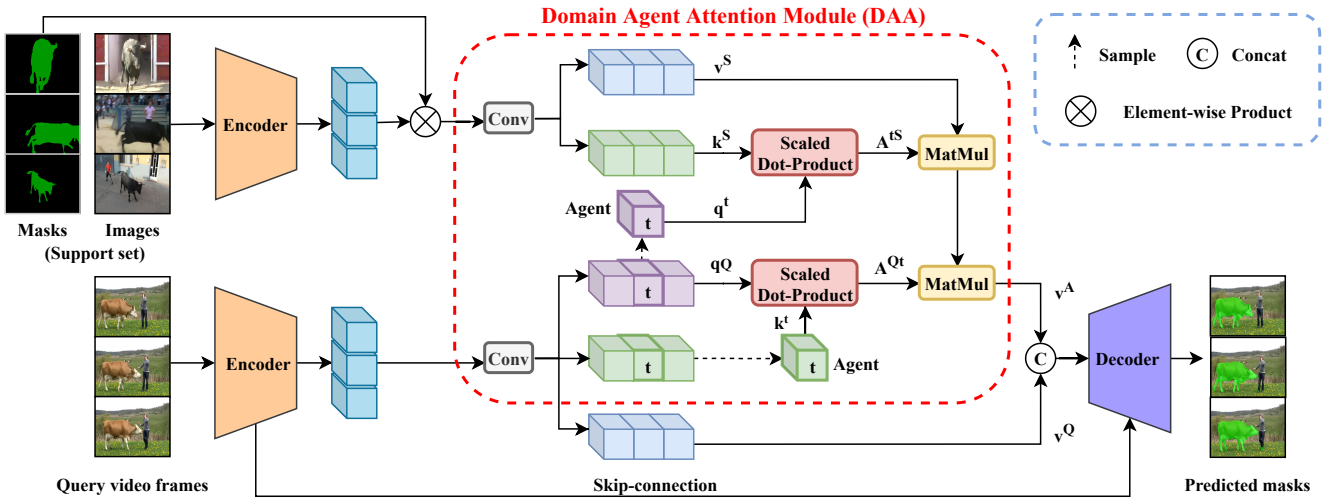

Figure 3: The architecture of our Domain Agent Network. The core of our DAN is the domain agent attention module highlighted in the red box. We decompose the full-rank attention into the product of two smaller ones, connected by an agent. Our DAN enables more efficient computation for the FSVOS task.

where the φ use the same deterministic vectors. After that, we can restate the replaced matrix in Eq. (2) to the following:

$$\hat{A} = \mathbb{E}[q'((k^t)')^{\mathrm{T}}(q^t)'(k')^{\mathrm{T}}]. \tag{8}$$

According to the associativity of matrix multiplication, we can extract the middle multiplication part $((k^t)')^{\mathrm{T}}(q^t)'$ and get the following result:

$$((k^t)')^{\mathrm{T}}(q^t)' = A^t, \tag{9}$$

where $((k^t)')^{\mathrm{T}} \in \mathbb{R}^{m\times L}$ and $(q^t)' \in \mathbb{R}^{L\times m}$. We denote L as the length of the features and m as the number of channels. In contrast to the previous attention matrix A that models correlation between features of pixel to pixel spatially, the correlation matrix $A^t \in \mathbb{R}^{m\times m}$ can be regarded as a channel-wise attention matrix.

Combining the Eq. 8 and Eq. 9, we can get the replaced attention matrix as follows:

$$\hat{A} = \mathbb{E}[q' A^t (k')^{\mathrm{T}}] = \mathbb{E}[\tilde{q'}(k')^{\mathrm{T}}], \tag{10}$$

where the channel attention matrix can be seen as acting on the features of the query frames. In particular, each channel of the query feature is re-weighted by its correlation with the other channels. For each channel i of the computed $\tilde{q'}$, we have $\tilde{q'}_i = \sum_{j=1}^{m}(A^t_{i,j}q'_j)$. Therefore, compared to the original attention matrix, our new matrix approximately adds a channel-wise attention, which may even better than the original one. In the following sections, we will introduce how we incorporate the hypothesis into the model design and prove its efficiency through experiments.

4. Domain Agent Network

In this section, we present our Domain Agent Network (DAN) based on the hypothesis, as shown in Fig. 3.

4.1. Method Overview

The goal of FSVOS is to segment out the objects with the same class as the labeled support images for the query videos. We separate the video dataset into two sets D^{train} and D^{test} based on the class labels, where the D^{train} is used for training and D^{test} is used for testing. There is no overlapping class between D^{train} and D^{test}. Both the training set D^{train} and the testing set D^{test} are composed of several episodes. Each episode contains a support set S_c and a query set Q for the class label c, where the query set $Q = \{x^q_i\}^N_{i=1}$ has a video with N frames. And the support set $S_c = \{x^s_{c,i}, m^s_{c,i}\}^M_{i=1}$ has a set of labeled image-mask pairs under the class label c. The network learns to predict mask $\widehat{Y}_c = \{\widehat{m}^q_{c,i}\}^N_{i=1}$ for each frame in the query set.

Our DAN mainly consists of three components: an encoder network, a domain agent attention module, and a decoder network. As shown in Fig. 3, given the support and the query sets as inputs, two encoders sharing the same weight extract features for the support set and the query set respectively. We then compute two attention matrices: one is the correlation between the domain agent and the support set, and the other is the correlation between the domain agent and the query set. We derive the final attention features by weighting the support features with the above two attention matrices. After concatenating the attention features with the feature values of the query set, we use a decoder to predict the final segmentation masks. We fur-

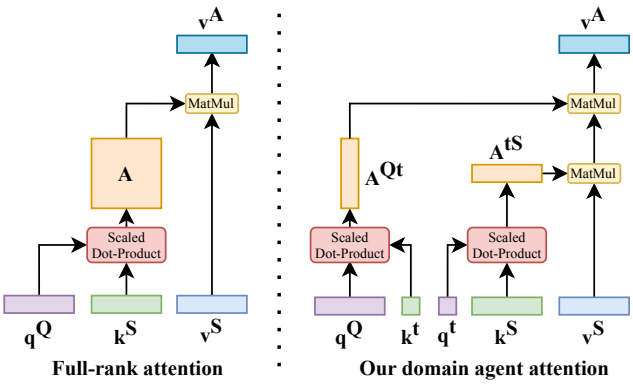

Figure 4: Comparison between conventional full-rank attention and our domain agent attention.

ther propose a learning strategy for FSVOS by combining meta-learning with online learning. We then discuss the architecture of DAN and the learning strategy in details.

4.2. Encoder and Decoder

We use a Siamese-architecture for designing encoders that share the same weight. The previous work [56] finds that the features from high layers are less generalized in the few-shot scenario. We thus use ResNet-50 [10] pretrained on ImageNet [6] without the block-4 as our encoder backbone to obtain the generic feature representation. Since the goal of FSVOS is to identify the objects in the query set that have the same category as those labeled in the support set, we compute features for foreground objects by weighting the support features with the ground truth masks.

After processing by the domain agent module, we concatenate the feature values of the query set v^Q with the attention features v^A before sending into the decoder, denoted as f^a. The decoder then predicts segmentation masks for the query set. Specifically, we design our decoder based on the upsample operation [48] and the skip-connection. We upsample the feature f^a to revert to the size of the input image. Meanwhile, we fuse the lower-level features of the query frames into the upsample features by skip-connections.

4.3. Domain Agent Attention Module

Calculating the correlation among images in the support set and query set often involve many-to-many attention matrix computation. As discussed in Sec. 3, we delve deep into the typical many-to-many attention matrix and decompose it to save computational cost. We have proven that the attention matrix can be replaced by the product of two smaller matrices through an agent. As mentioned in Eq. 10, the replaced matrix essentially adds channel-wise attention. Earlier studies [12, 3] on channel-wise attention directly learn a single dimension ($1 \times C$) weight for reweighting the importance of each channel. A recent study [45] propose to learn a $C \times C$ cross-channel weight, leading to better performance. We use the similar approach by learning the channel-wise attention dynamically to highlight the important feature channels based on the correlation among each channel. Simultaneously, we observe that the frames within a single video are similar. The channel-wise attention learned from one frame could approximate the channel-wise attention for the other frames. Therefore, we use the middle frame as the agent for a video to guarantee the learned channel-wise attention is informative.

Formally, the query features are mapping to queries q^Q, and key-value pairs k^Q - v^Q through a single convolutional layer. Similarly, the support features are mapping to queries q^S and key-value pairs k^S - v^S with the dimensions as follows:

$$\begin{cases} q^Q \in \mathbb{R}^{N \times L_q}, q^S \in \mathbb{R}^{M \times L_q}, L_q = H \times W \times C_q, \\ k^Q \in \mathbb{R}^{N \times L_k}, k^S \in \mathbb{R}^{M \times L_k}, L_k = H \times W \times C_k, \\ v^Q \in \mathbb{R}^{N \times L_v}, v^S \in \mathbb{R}^{M \times L_v}, L_v = H \times W \times C_v, \end{cases} \tag{11}$$

and we set $C_q = C_k$ in our case.

The frame t, sampled from the query video, is called domain agent. Following Eq. 3, the attention matrix A^{Qt} between Q and t and the attention matrix A^{tS} between t and S_c are calculated as follows:

$$A^{Qt} = \sigma(\frac{q^Q (k^t)^{\mathrm{T}}}{\sqrt{C_k}}); A^{tS} = \sigma(\frac{q^t (k^S)^{\mathrm{T}}}{\sqrt{C_k}}), \tag{12}$$

where q^t and k^t are query and key of the agent feature sampled from the q^Q and k^Q. Afterward, we obtain the attention feature v^A, denoted as:

$$v^A = Av^S = A^{Qt} A^{tS} v^S, \tag{13}$$

where we calculate the $A^{tS}v^S$ first to avoid storing and calculating the large matrix A as shown in Fig. 4. In summary, the memory storage and time cost of our attention module are $O((N+M)(HW)^2)$ and $O((N+M)(HW)^2C)$, while the memory storage and time cost of full-rank attention module are $O((NM)(HW)^2)$ and $O((NM)(HW)^2C)$.

4.4. Learning Strategy

In this subsection, we combine two learning methods to learn our DAN, as shown in Fig. 5. Meta-learning aims to learn a generalized meta-learner that enables adaption to a new task with the training in similar tasks. Many recent works [36, 55, 14] exploit meta-learning in few-show tasks to generalize on unseen classes. During the training phase, we use meta-learning to learn generic semantics-aware object segmentation. In each training episode, we sample the support and query set under the same class label. Following the previous works [44, 56, 37], we fix the parameters of the

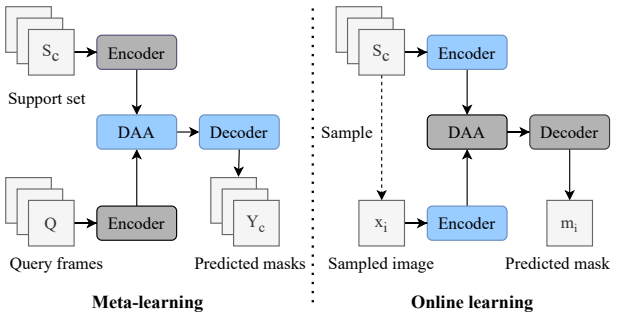

Figure 5: Learning strategy for DAN. We use meta-learning during training and online learning for testing. For each phase, we fix the weights of modules in the grey box, and only train the weights of modules in the blue box.

encoders and only train the domain agent attention module and the decoder.

During the testing phase, DAN inputs a few support images with the labeled class that is unseen in the training time. To better adapt features into the domain of the new class, we propose to use the online learning. As shown in Fig. 5, we fix the domain agent attention module and decoder and only train the encoders. This is because we want to maintain the ability on modeling the correlation between the support and query sets, but learning better input features for the test class. Rather than using online learning on each test video individually, which is time-consuming, we only use the support images to finetune the encoders, shortening the initiation time.

5. Experiments

In this section, we demonstrate the effectiveness of our proposed Domain Agent Network and the importance of the key components through ablation studies in Sec. 5.1. We report the computational cost in Sec. 5.2. Besides, we compare our method with existing few-shot image semantic segmentation models in Sec. 5.3 and show qualitative results in Sec. 5.4.

Dataset and Metrics. Since FSVOS is an underexplored problem, we set up a new benchmark based on the Youtube-VIS dataset [51]. The training set consists of 2,238 YouTube videos with 3,774 instances covering 40 categories. We evenly divide the dataset into four folds and cross-validate over all the folds. Each fold contains 30 categories for training and 10 categories for testing. We follow previous VOS methods [30, 2, 57] for using the region similarity ($\mathcal{J}$) and the contour accuracy ($\mathcal{F}$) to measure the performance. We denote k-shot as using the number of k support images for segmentation. The default setting is 5-shot in our experiments. More specifically, we randomly sample 5 images from a single class as the support set, and consecutive frames from the other videos under the same class as the query set. We run the experiment 5 times for each fold and report the average performance to ensure the confidence of the results.

Methods		Fold-1	Fold-2	Fold-3	Fold-4	Mean
$\mathcal{F}$	FAN	38.7	61.0	59.7	57.6	54.2
	Ours	**40.3**	**62.3**	**60.2**	**59.4**	**55.6**
$\mathcal{J}$	FAN	39.3	64.0	61.2	59.9	56.1
	Ours	**41.5**	**64.8**	**61.3**	**61.4**	**57.2**

Table 1: The effect of our domain agent attention. We denote **FAN** as the method of replacing our domain agent attention with the conventional full-rank attention.

Implementation Details. We use Adam as our optimizer for DAN. We set the learning rate to 1e-5 for meta-learning, and 5e-6 for online learning. We use a combination of cross-entropy loss and IoU loss with $5\mathcal{L}_{CE} + \mathcal{L}_{IoU}$ for meta-learning and only cross-entropy loss $\mathcal{L}_{CE}$ for online learning. We train DAN for 75,000 iterations with a batch size of 4 for meta-learning and 100 iterations with a batch size of 1 for online learning. We set the resolution to (241,425) for the inputs.

5.1. Ablation Study

We compare with two baselines respectively in this subsection. (1) We replace our domain agent attention with the original full-rank attention (**FAN**). (2) We train DAN without online learning, and directly use the model trained by meta-learning for testing.

5.1.1 The Effect of Domain Agent Attention

To avoid the influence of the online learning, we did not conduct finetuning during testing time for both FAN and our method. As shown in Tab. 1, our DAN outperforms traditional FAN by a large margin on both metrics. The accuracy increases by 1.4% for edges and 1.1% for regions. Since both the encoders and the decoder have the same architecture between our DAN and FAN, the performance gap should contribute to the proposed attention module. In other words, our method can better localize on the target objects through the introduced channel-wise attention, underlining the effect of the domain agent attention module. In addition, our approach effectively reduces the memory footprint and computing time, as shown in the Sec. 5.2.

5.1.2 The Effect of Online learning

To validate the effect of our learning strategy, we perform online learning for testing in this experiment. The purpose of online learning is to enhance the feature representation for unseen classes during the training. Since the encoder has pretrained on the ImageNet, covering many categories

Learning strategies	$\mathcal{F}$		$\mathcal{J}$	
	w/o	online	w/o	online
Deer	61.5	**63.6**	64.7	**66.6**
Giraffe	70.8	**73.2**	68.1	**70.2**
Hand	46.2	**50.5**	50.2	**57.0**
Parrot	58.9	**59.3**	64.6	**65.0**
Person	30.0	**36.2**	20.7	**29.4**
Skateboard	24.8	**36.8**	9.6	**16.3**
Surfboard	**49.8**	49.2	22.2	**23.0**
Tennis racket	17.9	**23.7**	10.7	**14.4**
Mean	45.0	**49.1**	38.8	**42.7**

Table 2: The effect of online learning for unseen class over the ImageNet.

of the Youtube-VIS, to prevent the encoder from corrupting well-trained features, we only examine categories that are not presented in the ImageNet.

As shown in Tab. 2, the performance of truly unseen classes increase a lot. For online learning, we set the number of training iterations to 100, which only takes 20s. This is insignificant compared to the time required for processing the entire video sets.

5.2. Computational Cost

Previous experiments have shown that our model outperforms the previous full-rank attention model. As discussed in Sec. 4.3, our attention module can save computational cost theoretically with a linear growth compared to the quadratic one. Therefore, we collect the memory and time usages as the number of inputs growing. To guarantee consistent and comparable results, we test the models on one 2080Ti GPU.

Memory Cost. As shown in Fig. 6, our method uses significantly less memory than FAN. When the support and query set individually contain 40 images, the memory consumption of our model is only 8.79 GiB, while FAN already uses 10.76 GiB when taking 20 images as input per set. After visualizing the tendency of the growth for each method, we can find that FAN grows exponentially, which makes it almost impossible to process more support and query images with limited memory. Instead, our method provides viable solution to process more support images, and more query frames for realistic applications.

Time Cost. To examine the time usage of our model, we test on different number of input query frames, ranging from 5 to 40, under different k-shot settings. As shown in Fig. 7, both FAN and DAN can benefit from processing a larger number of query frames. This is mainly due to the overhead of computing features for support images that can be mitigated as the query frames increasing. As can be seen that our DAN always takes less time for processing, under the benefit of our decomposition of attention matrix.

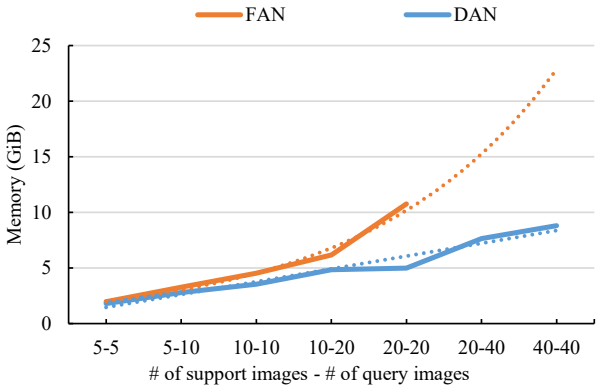

Figure 6: Comparison on memory cost. We show the memory usage along with the number of input images. We draw the tendency of different methods with dot lines.

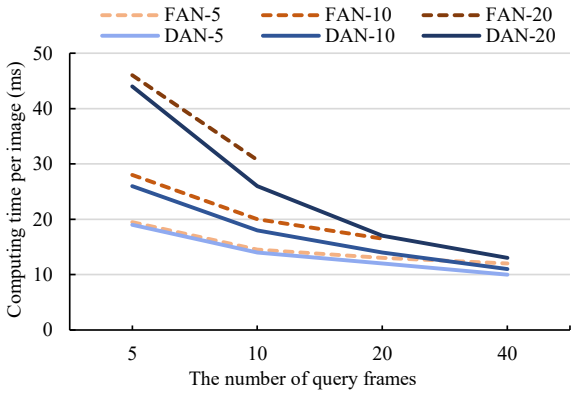

Figure 7: Comparison on time cost. We show the time usage along with the number of query frames in different settings. We use a format of "Method - k" in the legend to indicate method with k support images.

5.3. Comparisons to Existing Methods

The closest problem to ours is few-shot semantic segmentation working on individual images. We thus compare our method with the state-of-the-art image-based methods, which can be adapted to our task easily by processing each frame one by one. Particularly, we compare with (1) PFENet [37], an attention-based method relying on a semantic prior; (2) PPNet [21], a prototype-based method by finding prototypes of support images via K-means; (3) PMMs [50], another prototype-based method using Expectation-Maximization (EM) algorithm for finding prototypes.

As shown in Tab. 3, we generally outperform the above methods, especially on the metric of contour accuracy ($\mathcal{F}$). This is because of our method models this many-to-many

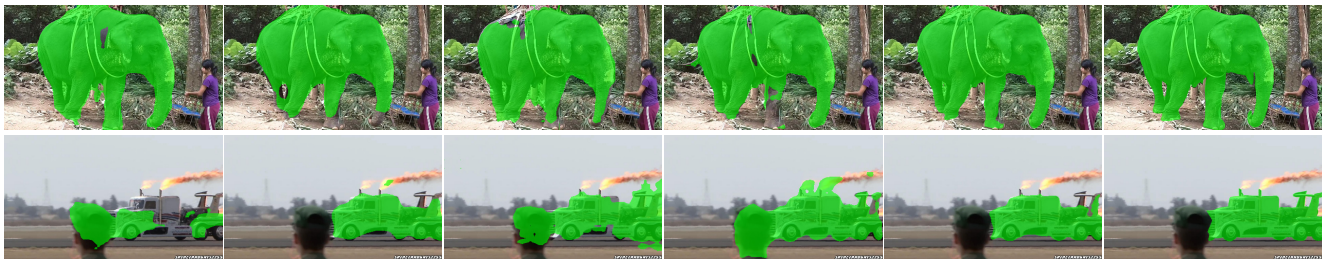

Figure 8: Qualitative comparisons of our method with baseline and applicable methods, including FAN, PMMs [50], PFENet [37], and PPNet [21].

	Methods	Fold-1	Fold-2	Fold-3	Fold-4	Mean
$\mathcal{F}$	PMMs [50]	34.2	56.2	49.4	51.6	47.9
	PFENet [37]	33.7	55.9	48.7	48.9	46.8
	PPNet [21]	35.9	50.7	47.2	48.4	45.6
	Ours	**42.3**	**62.6**	**60.6**	**60.0**	**56.3**
$\mathcal{J}$	PMMs [50]	32.9	61.1	56.8	55.9	51.7
	PFENet [37]	37.8	64.4	56.3	56.4	53.7
	PPNet [21]	**45.5**	63.8	60.4	58.9	57.1
	Ours	43.2	**65.0**	**62.0**	**61.8**	**58.0**

Table 3: Comparisons to existing applicable methods.

attention to contain detailed information and fuse lower-level features to the upsample features by skip-connections. Compared to these methods, our approach models the cross-frame correlation through the agent, allowing the network to embrace certain temporal information.

5.4. Qualitative Results

Comparisons to Other Methods. Fig. 8 visualizes the qualitative results of our method compared to those generated by FAN and methods mentioned in Sec. 5.3. We can see that the segmentation masks generated by our method are more compact with the target objects. In the first row, the segmentation results of PFENet, PPNet and FAN are incomplete. Although PMMs segments on the right object, the boundary of the segmentation is not accurately aligned with the object. In contrast, we have a much better result near the boundary. In the second row, the query frame contains two objects, a person and a car. The segmentation predicted by PMMs, PPNet and FAN mistakenly include a partial region of the person, while our method is more accurate and only segment out the target car.

The Effect of Online Learning. Fig. 9 shows the segmentation results of our method with and without online learning. Though our method is able to localize the target object without online learning, the result is less satisfactory by including some irrelevant objects (e.g., the helmet in the first row) or with some missing regions (e.g., the tennis racket in the second row). After finetuning with online learning, we can notice that the segmentation results become more complete and compact.

(a) w/o online learning (b) w/ online learning

Figure 9: Qualitative comparisons of our method training with and without online learning.

6. Conclusion

In this paper, we propose a novel domain agent network for solving the few-shot video object segmentation task. The key idea is to decompose the typical many-to-many attention matrix into the product of two smaller ones through an agent. We include theoretical proof to demonstrate the validity and efficiency of the decomposition. We also propose a learning strategy by combining meta-learning with online learning to further improve the performance. Through extensive experiments, we demonstrate the benefit of our method both quantitatively and qualitatively.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61972162), and CCF-Tencent Open Research fund.

References

[1] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages 282–295. Springer, 2010. 1, 3

[2] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 3, 6

[3] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *CVPR*, pages 5659–5667, 2017. 5

[4] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *arXiv preprint arXiv:2009.14794*, 2020. 3

[5] Hai Ci, Chunyu Wang, and Y. Wang. Video object segmentation by learning location-sensitive embeddings. In *ECCV*, 2018. 1, 3

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 5

[7] Nanqing Dong and E. Xing. Few-shot semantic segmentation with prototype learning. In *BMVC*, 2018. 2, 3

[8] Katerina Fragkiadaki, Geng Zhang, and Jianbo Shi. Video segmentation by tracing discontinuities in a trajectory embedding. In *CVPR*, pages 1846–1853. IEEE, 2012. 3

[9] Raffay Hamid, Ying Xiao, Alex Gittens, and Dennis DeCoste. Compact random feature maps. In *ICML*, pages 19–27, 2014. 3

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[11] Yuk Heo, Yeong Jun Koh, and Chang-Su Kim. Interactive video object segmentation using global and local transfer modules. In *ECCV*, 2020. 1

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 5

[13] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, pages 2117–2126. IEEE, 2017. 1, 3

[14] Muhammad Abdullah Jamal and Guo-Jun Qi. Task agnostic meta-learning for few-shot learning. In *CVPR*, pages 11719–11727, 2019. 5

[15] Purushottam Kar and Harish Karnick. Random feature maps for dot product kernels. In *AISTATS*, pages 583–591, 2012. 3

[16] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with referring expressions. In *ECCV*, pages 7–12, 2018. 2, 3

[17] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, pages 1995–2002. IEEE, 2011. 1, 3

[18] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, pages 2192–2199, 2013. 3

[19] Siyang Li, Bryan Seybold, Alexey Vorobyov, Alireza Fathi, Qin Huang, and C-C Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, pages 6526–6535, 2018. 1, 3

[20] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018. 3

[21] Yongfei Liu, Xiangyi Zhang, Songyang Zhang, and Xuming He. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 2020. 7, 8

[22] Xinkai Lu, W. Wang, Martin Danelljan, Tianfei Zhou, J. Shen, and L. Gool. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020. 3

[23] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, pages 3623–3632, 2019. 1, 3

[24] Xiankai Lu, Wenguan Wang, Jianbing Shen, Yu-Wing Tai, David J Crandall, and Steven CH Hoi. Learning video object segmentation from unlabeled videos. In *CVPR*, pages 8960–8970, 2020. 3

[25] Khoi Nguyen and Sinisa Todorovic. Feature weighting and boosting for few-shot segmentation. In *ICCV*, 2019. 2, 3

[26] Peter Ochs, Jitendra Malik, and Thomas Brox. Segmentation of moving objects by long term video analysis. In *PAMI*, volume 36, pages 1187–1200. IEEE, 2013. 3

[27] S. Oh, Joon-Young Lee, N. Xu, and S. Kim. Video object segmentation using space-time memory networks. In *ICCV*, pages 9225–9234, 2019. 1, 2, 3

[28] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Fast user-guided video object segmentation by interaction-and-propagation networks. In *CVPR*, pages 5247–5256, 2019. 1

[29] Federico Perazzi, A. Khoreva, Rodrigo Benenson, B. Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, pages 3491–3500, 2017. 3

[30] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 3, 6

[31] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *NeurIPS*, volume 20, pages 1177–1184, 2007. 3

[32] Seonguk Seo, Joon-Young Lee, and Bohyung Han. Urvos: Unified referring video object segmentation network with a large-scale benchmark. In *ECCV*, 2020. 2, 3

[33] Amirreza Shaban, Shray Bansal, Zhen Liu, Irfan Essa, and Byron Boots. One-shot learning for semantic segmentation. In *BMVC*, 2017. 3

[34] Mennatullah Siam, Naren Doraiswamy, Boris N. Oreshkin, Hengshuai Yao, and Martin Jagersand. Weakly supervised few-shot object segmentation using co-attention with visual and semantic embeddings. In *IJCAI*, pages 860–867, 2020. 2, 3

[35] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm

for video salient object detection. In *ECCV*, September 2018. 3

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, June 2018. 5

[37] Zhuotao Tian, Hengshuang Zhao, Michelle Shu, Zhicheng Yang, Ruiyu Li, and Jiaya Jia. Prior guided feature enrichment network for few-shot segmentation. In *PAMI*, 2020. 3, 5, 7, 8

[38] Pavel Tokmakov, Karteek Alahari, and Cordelia Schmid. Learning motion patterns in videos. In *CVPR*, pages 3386–3394, 2017. 3

[39] P. Tokmakov, Alahari Karteek, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, 2017. 1, 3

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 2, 3

[41] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 3

[42] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, volume abs/1706.09364, 2017. 3

[43] Haochen Wang, Xudong Zhang, Yutao Hu, Y. Yang, Xianbin Cao, and Xiantong Zhen. Few-shot semantic segmentation with democratic attention networks. In *ECCV*, 2020. 2, 3

[44] Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. Panet: Few-shot image semantic segmentation with prototype alignment. In *ICCV*, 2019. 3, 5

[45] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *CVPR*, pages 11534–11542, 2020. 5

[46] Q. Wang, L. Zhang, Luca Bertinetto, W. Hu, and P. Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, pages 1328–1338, 2019. 3

[47] Wenguan Wang, Jianbing Shen, and Fatih Porikli. Saliency-aware geodesic video object segmentation. In *CVPR*, pages 3395–3402, 2015. 1, 3

[48] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, pages 7376–7385, 2018. 5

[49] H. Xiao, Jiashi Feng, Guosheng Lin, Y. Liu, and Maojun Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, pages 1140–1148, 2018. 3

[50] Boyu Yang, Chang Liu, Bohao Li, Jianbin Jiao, and Ye Qixiang. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 2020. 2, 3, 7, 8

[51] Linjie Yang, Yuchen Fan, and Ning Xu. Video instance segmentation. In *ICCV*, pages 5188–5197, 2019. 6

[52] Rui Yao, Guosheng Lin, Shixiong Xia, Jiaqi Zhao, and Yong Zhou. Video object segmentation and tracking: A survey. In *TIST*, volume 11, pages 1–47. ACM New York, NY, USA, 2020. 1

[53] J. S. Yoon, François Rameau, J. Kim, Seokju Lee, Seunghak Shin, and In-So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, pages 2186–2195, 2017. 3

[54] Xiaohui Zeng, Renjie Liao, L. Gu, Y. Xiong, S. Fidler, and R. Urtasun. Dmm-net: Differentiable mask-matching network for video object segmentation. In *ICCV*, pages 3928–3937, 2019. 3

[55] 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 *ICCV*, pages 9587–9595, 2019. 2, 3, 5

[56] 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 *CVPR*, pages 5217–5226, 2019. 5

[57] Tianfei Zhou, Jianwu Li, Shunzhou Wang, Ran Tao, and Jianbing Shen. Matnet: Motion-attentive transition network for zero-shot video object segmentation. In *TIP*, volume 29, pages 8326–8338. IEEE, 2020. 3, 6