Beyond Pixel and Object: Part Feature as Reference for Few-Shot Video Object Segmentation

Naisong Luo*, Guoxin Xiong*, Tianzhu Zhang†

MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China
{lns6, xgx}@mail.ustc.edu.cn, tzzhang@ustc.edu.cn

Abstract

Few-Shot Video Object Segmentation (FSVOS) aims to achieve accurate segmentation of video sequences supported by limited annotated images. In this work, we analyze the deficiencies inherent in the use of object prototypes and pixel features as references in previous methods. Then we shed light on that part features, with the ability to adapt to appearance variations and resist noise, are advantageous as representative reference features for aligning support images and query videos. Therefore, we propose a Part Agent Learning Network (PALN) to leverage part features from two aspects. First, we elaborately employ Optimal Transport algorithm with equal partition constraint to make part agents capable of dividing support objects into diverse parts in an adaptive manner. Second, we design a dedicated cache mechanism to learn temporal part agents as lightweight historic target representation to exploit temporal consistency. With the aid of these learned part agents, our PALN can effectively achieve support-query alignment and temporal alignment for accurate segmentation of query videos. Extensive experimental results on two challenging benchmarks demonstrate that our method performs favorably against state-of-the-art FSVOS methods.

Introduction

Video object segmentation (VOS) (Oh et al. 2019; Cheng and Schwing 2022; Yang, Wei, and Yang 2021; Sun et al. 2023) is a fundamental vision content understanding task. Despite the tremendous success, current deep learning based VOS methods heavily rely on dense video annotations, which are laborious and time-consuming to obtain. To reduce the need for human annotations, few-shot video object segmentation (FSVOS) (Siam et al. 2021; Chen et al. 2021) has been proposed and attracts increasing interest, which aims to achieve accurate segmentation of video sequences supported by limited annotated images.

For FSVOS methods, during the testing phase, given the *support* set (*i.e.*, a few images with masks of target objects), the model need to identify corresponding objects in *query* set (*i.e.*, unlabeled video sequence). And these objects may be new classes that have not been seen during the training phase, imposing stricter requirements on the model generalization. Therefore, during the training phase, existing FSVOS methods adopt the meta-learning (Finn, Abbeel, and Levine 2017) paradigm that simulates the testing procedure by sampling some *support* images and *query* videos from the training data, enabling the model to learn conditional segmentation capability, while enhancing the generalization for unseen categories. Since there are usually significant differences between *support* and *query*, such as scale, pose, background variations and different structure (*i.e.*, video has an

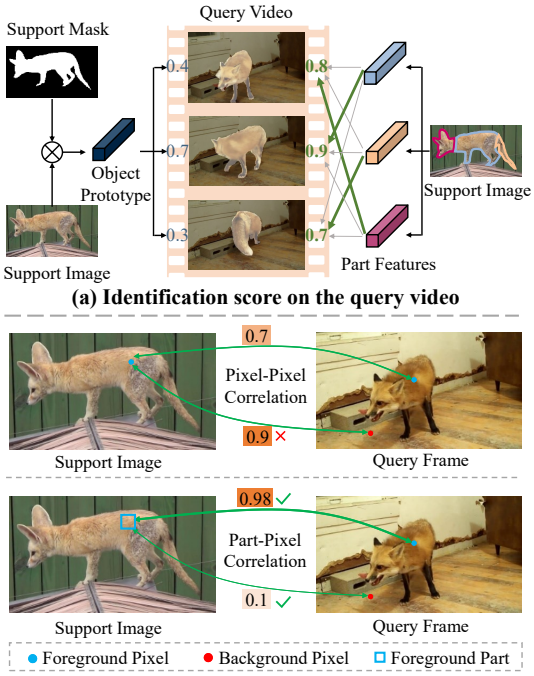

Figure 1: Illustration of our motivation. (a) Identification score is used to measure the adaptability to appearance variations in query video and part features are more suitable for recognizing moving targets. (b) The support-query correlation is measured using cosine similarity, and the part features have the ability to resist noise compared to the pixel feature, achieving more accurate correlation.

*These authors contributed equally.
†Corresponding author.

additional temporal dimension), how to fully exploit representative reference features to align support images and the query video is thus extremely challenging.

Top-performing FSVOS methods can be roughly categorized as prototypical learning methods and affinity learning methods. On the one hand, prototypical learning methods (Liu et al. 2023; Siam, Derpanis, and Wildes 2022; Tang et al. 2023) adopt masked average pooling on support feature map to achieve a compact vector (called prototype) to encode the object semantics as the reference of the query segmentation. However, the target in the query video tends to constantly change the pose, such as the *fox* turn in Figure 1(a). The object prototype struggles to handle every situation coherently. On the other hand, affinity learning methods (Chen et al. 2021; Luo et al. 2024) attempts to directly leverage pixel-pair similarity between support features and query features for segmentation. These methods take advantage of the detailed pixel feature as the diverse target representations. However, direct pixel correlation tends to suffer from the confusion caused by noisy pixels, background clutters and intra-class differences because of the lack of contextual information. As illustrated in Figure 1(b) where the target is *fox*, the support-query similarity (*i.e.*, 0.7) between foreground pixel and foreground pixel is less than the similarity (*i.e.*, 0.9) between the foreground pixel and confusing background pixel, and the erroneous correlation can lead to misclassification of the query pixel. Overall, the above analysis indicates that neither object prototype nor pixel feature are suitable as the representative reference feature.

Motivated by the above discussions, we propose to extract the part feature as the representative reference. The part feature represents a local region of the target with a granularity between the pixel feature and the object prototype. After an in-depth analysis, we find two key properties of part feature which are urgent for FSVOS. (1) **Adaptation to appearance variations**. Intuitively, humans can quickly distinguish the target in motion by capturing partial cues, despite the variations of the target appearance during the video sequence due to factors such as movement, rotation, and lighting changes. In order to quantitatively compare the discrimination of the object prototype and the part feature, we define the identification score, which is the average of the cosine similarity between the support prototype/part and all the query pixel features in the foreground region. As illustrated in Figure 1(a), for each query frame, we can always match a part feature with high identification score that exceeds the score of the object prototype, indicating the superiority of part features in recognizing moving targets within the video. (2) **Resistance to correlation noise**. With the region context, part feature is naturally resistant to noisy pixels. As demonstrate in Figure 1(b), the part feature enhances the correct correlation (*i.e.*, 0.7→0.98) and rectifies the incorrect correlation (*i.e.*, 0.9→0.1). Generally, erroneous correlation will confuse the aggregation of target information and thus produce inaccurate segmentation. Therefore, the part feature plays a critical role in suppressing the ambiguous correlation and further establishing reliable support-query alignment.

In this paper, we propose an end-to-end **Part Agent Learning Network (PALN)** including a part agent encoder, a support-query alignment decoder and a temporal alignment decoder to learn the part feature as representative reference feature for support-query and temporal alignment. Specifically, (1) to learn the part feature, it is not difficult to consider pooling all pixel features within a local region into a single vector. However, relying solely on boundary-derived fragments (such as SLIC (Achanta et al. 2012)) may result in a gap between the obtained fragments and the actual semantic regions. Therefore, directly clustering based on semantic features is a better approach. Nevertheless, existing clustering algorithms like K-Means lack the ability to adjust the distribution of cluster centers, which can lead to multiple centroids focusing on the same object part. To alleviate this problem, we resort to Optimal Transport algorithm in the **part agent encoder** and impose the equal partition constraint to expand the discrepancy among part masks to obtain the part agents. In this way, the part agents can decompose different target objects into diverse and complementary parts in an adaptive manner. (2) Compared to object prototypes and pixel features, part features can establish more reliable correlation with the query video. Therefore, we employ the attention mechanism in the **support-query alignment decoder** to integrate the learned part agents into the query features. (3) Due to the intrinsic temporal consistency within the query video, the target representation from historical frames can serve as the supplementary of support for identifying target object in current frame. Therefore, we design a dedicated cache mechanism in the **temporal alignment decoder** that utilizes the maximum orthogonality principle to extract part features from the previous frame (referred to as temporal part agents). The learned temporal part agents exhibit characteristics of reliability, representativeness and diversity. Compared to existing VOS methods that rely on memory of dense features, our lightweight inference process is more friendly for episodic meta-training, thereby enhancing generalization to unseen categories.

The contributions of our method can be summarized as follows: (1) We analyze the bottlenecks that exist in previous methods from the perspective of object prototypes and pixel features, and shed light on the potential of part features as representative reference features for FSVOS. (2) We propose a Part Agent Learning Network (PALN) to encode part features for both support-query alignment and temporal alignment in a unified framework. Specifically, we elaborately leverage Optimal Transport algorithm with equal partition constraint to compress part agents from support images. Additionally, we design a dedicated cache mechanism to learn the temporal part agent from historical frames. (3) Extensive experimental results on two challenging benchmarks demonstrate that our method performs favorably against state-of-the-art FSVOS methods.

Related Work

Few-Shot Video Object Segmentation. VOS aim to predict the foreground object mask in every video frame given an object mask at the first frame in the semi-supervised setting. Deep learning based VOS methods demonstrate the effectiveness of modeling temporal information in video segmentation tasks. For instance, STM (Oh et al. 2019) constructs

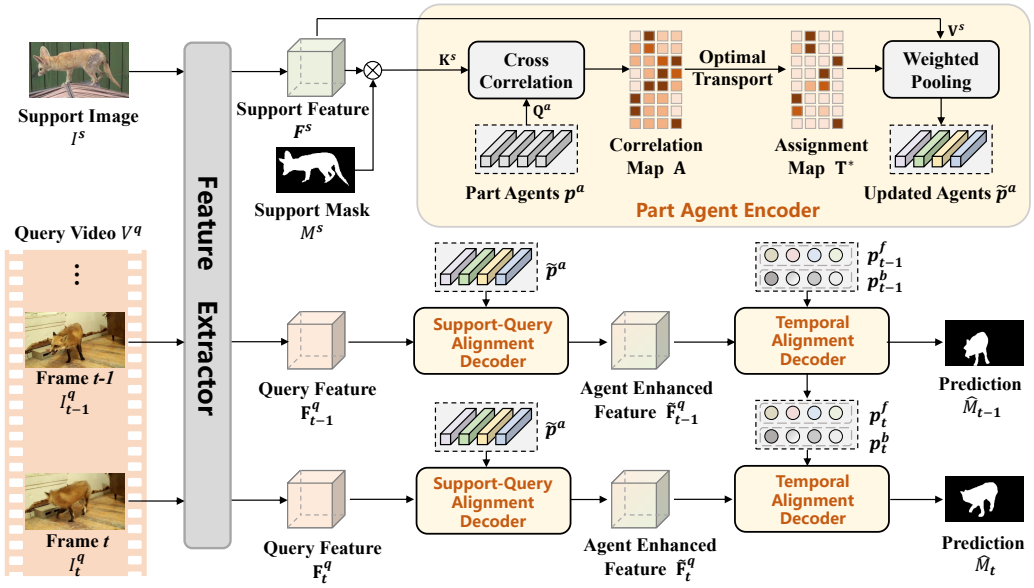

Figure 2: Illustration of the proposed PALN. Our framework mainly consists of three modules. (1) The part agent encoder aims at condensing support information into a set of part agents as target representations. (2) The support-query alignment decoder is responsible for conveying the support information of part agents to the query. (3) The temporal alignment decoder aims to learn the temporal part agents from previous frame to assist the segmentation of current frame.

a memory bank for each object and the non-local attention mechanism is applied to retrieve information from memory to facilitate object segmentation in the current frame. STCN (Cheng, Tai, and Tang 2021) improves the matching metric function and efficiency on this base. XMem (Cheng and Schwing 2022) proposes to use sensory memory, working memory, and long-term memory to build long-range temporal information. Recently, some methods introduced the global interaction of the transformer to the segmentation task, opening up a new segmentation paradigm (Cheng, Schwing, and Kirillov 2021; Sun et al. 2021; Cheng et al. 2022; Luo et al. 2023; Mai et al. 2023; Pan et al. 2023, 2024; Xiong et al. 2024), *e.g.*, Cuite (Cheng et al. 2024) leverages object queries as a high-level summary of the target object in the transformer decoder and achieve excellent results on long video segmentation. To reduce dependence on annotated data, FSVOS aims at segmenting objects in the query videos with a certain class given limited labeled support images of the same category. Previous FSVOS methods can be roughly categorized as affinity learning methods and prototypical learning methods. For affinity learning, DAN (Chen et al. 2021) performs dense matching between support and query feature and considers one single frame of the query video to bridge the support and query. For prototypical learning, VIPMT (Liu et al. 2023) proposes to compress the features of support and video frames into a compact vector (prototype) to prompt query segmentation. HPAN (Tang et al. 2023) uses K-Means to cluster the support features into multiple prototypes. Combining the strengths of previous methods, we innovatively propose to model the part features in a learnable, adaptive and lightweight manner to extract both support and temporal information for FSVOS.

Few-Shot Image Segmentation. FSIS aims to perform dense segmentation for the novel class given only a few annotated examples. Mainstream few-shot segmentation methods can be roughly divided into two categories according to the paradigm to excavate target information from support: prototype-based methods and affinity-based methods. For the former, most methods (Shaban et al. 2017; Tian et al. 2020; Wu et al. 2021; Xie et al. 2021a; Yang et al. 2020; Liu et al. 2020; Li et al. 2021; Zhang, Xiao, and Qin 2021) condense the masked support features into single or multiple prototypes for feature comparison or aggregation. For example, SG-One (Shaban et al. 2017) calculates the cosine similarity between a single support prototype and query features to guide the prediction. PFENet (Tian et al. 2020) and the following works (Wu et al. 2021; Xie et al. 2021a) make use of a class-wise prototype to produce a training-free prior mask to guide the segmentation. While for the affinity-based methods (Zhang et al. 2021; Wang et al. 2022; Hong et al. 2022; Wang et al. 2020; Xie et al. 2021b; Wang, Sun, and Zhang 2023; Wang, Luo, and Zhang 2023), fine-grained pixel-wise relationships between support and query features are further considered to retain the details. Although current FSIS methods make great progress in matching query and support, they are not competent in the video segmentation tasks due the the lack of temporal consistency mining.

Method

As shown in Figure 2, our PALN mainly consists of three modules. (1) The part agent encoder aims at condensing support information into a set of part agents as target representations. (2) The support-query alignment decoder is responsible for conveying the support information of part agents to

the query. (3) The temporal alignment decoder aims to effectively exploit the temporal consistency and learn the temporal part agents to assist the segmentation of current frame.

Problem Definition

The few-shot video segmentation is conducted on a set of episodes during testing. Both the training set $\mathbb{D}_{train}$ and the testing set $\mathbb{D}_{test}$ are composed of several episodes, where the labels are disjoint, *i.e.*, $\mathbb{C}_{train} \cap \mathbb{C}_{test} = \varnothing$. Each episode contains a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. Specifically, the N-shot setting means that the episode is composed of N support images, *i.e.*, $\mathcal{S} = \{(I_l^s, M_l^s)\}_{l=1}^N$, where y_l^s denotes corresponding binary mask. And $\mathcal{Q} = \{(I_l^q, M_l^q)\}_{l=1}^T$ contains a query video V^q with T frames and the associated ground truth masks in the same class with the support set $\mathcal{S}$. The ultimate goal is to segment a query video $V^q \in \mathcal{Q}$ given a few labeled images from $\mathcal{S}$. We use $N = 1$ to describe our method.

Part Agent Encoder

We design the part agent encoder to condense support information into a set of part agents as desired representative reference features. Specifically, given the support image I^s and a query frame I^q, we use a backbone network as feature extractor to obtain the support and query features $\mathbf{F}^s, \mathbf{F}^q \in \mathbb{R}^{h \times w \times c}$, where h, w and c denote height, width and the number of channels, respectively. With corresponding target mask M^s, we filter out the background pixels from the support features:

$$\tilde{\mathbf{F}}^s = \mathbf{F}^s \otimes \mathcal{I}(M^s), \tag{1}$$

where $\otimes$ is the Hadamard product, and $\mathcal{I}$ denotes the operation that resizes M^s to the same shape as $\mathbf{F}^s$. Then we initialize a set of part agents $\boldsymbol{p}^a \in \mathbb{R}^{N_a \times c}$ and employ them to calculate the similarity with the support features $\tilde{\mathbf{F}}^s$, where N_a denotes the number of part agents. We first project the part agents and support features to obtain *queries* $\mathbf{Q}^a$, *keys* $\mathbf{K}^s$ and *values* $\mathbf{V}^s$ as:

$$\mathbf{Q}^a = \boldsymbol{p}^a \mathbf{W}^Q, \quad \mathbf{K}^s = \tilde{\mathbf{F}}^s \mathbf{W}^K, \quad \mathbf{V}^s = \tilde{\mathbf{F}}^s \mathbf{W}^V, \tag{2}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{c \times c}$ are linear projections. And the correlation map can be derived by:

$$\mathbf{A} = \mathrm{softmax}\left(\frac{\mathbf{Q}^a (\mathbf{K}^s)^\top}{\sqrt{c}}\right), \tag{3}$$

where $\sqrt{c}$ is a scaling factor.

To assign the group of semantically consistent pixels to the same agent to form the part, we formulate the pixel-agent assignment as a discrete form of Optimal Transport (OT) problem (Villani 2009). The goal of the OT problem is to find a optimal transport plan $\mathbf{T}^*$ at a global minimal transport cost. In FSVOS, the correlation $\mathbf{A}$ can be regarded as the opposite of the transportation cost between support pixels and agents. Thus optimizing the OT problem is equal to maximize the total correlation, *i.e.*, push each pixel as far as possible to the most similar agent, formulated as:

$$\mathbf{T}^* = \arg\max_{\mathbf{T} \in \mathcal{T}} \mathrm{Tr}\left(\mathbf{T}^\top \mathbf{A}\right) + \epsilon H(\mathbf{T}), \tag{4}$$

where $H(\mathbf{T}) = -\sum_{ij} \mathbf{T}_{ij} \log \mathbf{T}_{ij}$ is the entropy function, and ϵ is the parameter that controls the smoothness of the mapping. $\mathcal{T}$ is the feasible range of the transport plan $\mathbf{T}$, defined as:

$$\mathcal{T} = \left\{\mathbf{T} \in \mathbb{R}_+^{N_a \times hw} \mid \mathbf{T}\mathbb{1} = \mathbb{1}, \mathbf{T}^\top \mathbb{1} = \mathbb{1}\right\}, \tag{5}$$

where $\mathbb{1}$ denotes the vector of all ones in the appropriate dimension. The equal partition constraints in Eq.(5) enforces that each agent is assigned the same number of foreground pixels thus preventing a trivial solution where all pixels are assigned to a single agent, as $\mathbf{T}^*$ in the Figure 2. So that different agent responsible for different part areas that are complementary. The optimal transport plan $\mathbf{T}^*$ in Eq.(4) can be obtained via a fast variant of Sinkhorn-Knopp (Cuturi 2013) which is differentiable.

We apply the weighted pooling with new assignment map $\mathbf{T}^*$ followed by a feed-foreard network (FFN) (Vaswani et al. 2017) to update the part agents:

$$\tilde{\boldsymbol{p}}^a = \mathrm{FFN}(\mathbf{T}^* \mathbf{V}^s) \tag{6}$$

The updated part agents naturally differentiate themselves, eliminating the need for additional explicit supervision, such as diversity loss.

Support-Query Alignment Decoder

The support-query alignment decoder aims at conveying the support information to the query and the part agent is a bridge responsible for aligning the features on both sides. Concretely, we first feed the query features $\mathbf{F}^q$ into deformable self-attention layers (Zhu et al. 2020) to incorporate context information from other pixels. Afterwards, similar to Eq.(2), we use linear projections to project the query features $\mathbf{F}^q$ to the *queries* $\mathbf{Q}^q$, and project the part agents $\tilde{\boldsymbol{p}}^a$ to the *keys* $\mathbf{K}^a$ and the *values* $\mathbf{V}^a$. Then the support information from part agents can be aggregated by:

$$\tilde{\mathbf{F}}^q = \mathrm{FFN}\left(\mathrm{softmax}\left(\frac{\mathbf{Q}^q (\mathbf{K}^a)^\top}{\sqrt{c}}\right)\mathbf{V}^a\right). \tag{7}$$

Temporal Alignment Decoder

The temporal alignment decoder is responsible to effectively exploit the temporal consistency between query video frames to mine extra guidance. We desire to learn the temporal part agent from historical frames. To achieve this, we design a cache mechanism which involves the previous frames. Specifically, at frame t, we preserve the query features $\mathbf{F}_{t-1}^q \in \mathbb{R}^{h \times w \times c}$ of previous frame $t-1$ as well as its predicted target probability $\hat{M}_{t-1} \in \mathbb{R}^{h \times w}$ predicted by the network. We first measure the uncertainty U using the entropy of $\hat{M}_{t-1}$:

$$U = -\hat{M}_{t-1} \log \hat{M}_{t-1}. \tag{8}$$

Then $N_f = 0.3 \times h \times w$ foreground feature points with low uncertainty are extracted from $\mathbf{F}_{t-1}^q$ to form the candidate set $\{\boldsymbol{p}_k\}_{k=1}^{N_f}$. We initial location set $\boldsymbol{L} = \{n_0\}$ and agent set $\boldsymbol{P} = \{\boldsymbol{p}_{n_0}\}$ by randomly selecting a n_0. Afterwards,

Methods	Query	Fine-Tune	1-shot					5-shot				
			1	2	3	4	**Mean**	1	2	3	4	**Mean**
PMM (Yang et al. 2020)	Image	✗	20.5	43.3	32.8	31.2	32.0	32.9	61.1	56.8	55.9	51.7
PFENet (Tian et al. 2020)	Image	✗	18.7	45.2	33.6	30.0	31.9	37.8	64.4	56.3	56.4	53.7
PPNet (Liu et al. 2020)	Image	✗	22.1	44.7	33.9	33.6	33.6	45.5	63.8	60.4	58.9	57.1
RePRI (Boudiaf et al. 2021)	Image	✗	41.0	65.2	58.4	57.8	55.6	45.8	68.6	59.3	64.2	59.5
NTRENet (Liu et al. 2022a)	Image	✗	36.7	63.1	58.9	54.6	53.3	39.0	66.4	61.7	61.2	57.1
SSP (Fan et al. 2022)	Image	✗	41.5	60.4	57.0	51.3	52.6	46.7	64.3	59.3	54.5	56.2
VAT (Hong et al. 2022)	Image	✗	38.2	61.9	56.1	52.3	52.1	42.6	62.8	57.0	56.7	54.8
IPMT (Liu et al. 2022b)	Image	✗	40.7	64.1	58.8	55.5	54.8	43.8	65.8	61.0	60.7	57.8
DANet (Chen et al. 2021)	Video	✗	39.1	61.8	57.4	56.7	53.8	41.5	64.8	61.3	61.4	57.2
DANet (Chen et al. 2021)	Video	✓	39.3	62.6	57.9	57.4	54.3	43.2	65.0	62.0	61.8	58.0
TTI (Siam, Derpanis, and Wildes 2022)	Video	✗	43.1	61.1	53.4	57.7	53.8	48.2	69.0	62.8	63.1	60.8
HPAN (Tang et al. 2023)	Video	✗	42.6	66.2	59.0	57.8	56.4	47.2	70.2	66.0	65.6	62.2
HPAN (Tang et al. 2023)	Video	✓	44.0	66.6	59.4	58.4	57.1	50.2	70.5	66.2	67.0	63.5
VIPMT (Liu et al. 2023)	Video	✗	45.3	66.7	59.6	58.1	57.4	50.6	70.9	68.8	66.5	64.2
PALN (ours)	Video	✗	**46.9**	**71.2**	**62.7**	**62.9**	**60.9**	**50.8**	**73.4**	**69.7**	**67.0**	**65.2**

Table 1: Results of 4-fold cross-validation with metric $\mathcal{J}$ on YouTube-FSVOS using one support image (1-shot) and five support images (5-shot). All competitors adopt ResNet-50 as the backbone for fair comparison. The best results are shown in **bold**.

we calculating the cosine similarity between each candidates and the selected agent:

$$\boldsymbol{S}(k) = \max_{n \in \boldsymbol{L}} \text{cosine}(\boldsymbol{p}_k, \boldsymbol{p}_n). \tag{9}$$

Among the N_f points, the location of the most orthogonal agent in candidates is picked up:

$$n_i = \arg\min_k \boldsymbol{S}(k), \quad s.t. \quad n_i \notin \boldsymbol{L}. \tag{10}$$

In this way, each of the selected pixel feature is far away from each other in the candidate set. Then we update the location set $\boldsymbol{L}$ and the agent set $\boldsymbol{P}$ based on the selected location n_i:

$$\boldsymbol{L} = \boldsymbol{L} \cup \{n_i\}, \quad \boldsymbol{P} = \boldsymbol{P} \cup \{\boldsymbol{p}_{n_i}\}. \tag{11}$$

Repeat the above steps until N_a pixel features are selected. In this way, we can obtain the temporal part agent of foreground $\boldsymbol{p}_t^f = \text{FFN}(\boldsymbol{P}) \in \mathbb{R}^{N_a \times c}$. Employing the same steps, we can acquire the temporal part agent of background $\boldsymbol{p}_t^b$ by picking the background feature points to form the candidate set. Since these agents have high confidence and are sparsely distributed in the feature space, they are inherently diverse and can be used as part representations of the target.

Then we interact them with the query features $\mathbf{F}_t^q$ to activate target-relevant regions in the current frame. Considering that the foreground and background exhibit different temporal properties, we adopt different alignment schemes. Concretely, on the one hand, we utilize $\boldsymbol{p}^f$ (we omit the temporal subscript t for simplicity) to excite the foreground of the current frame, as:

$$\mathcal{M}^f_{x,y,n} = \sum_{k=1}^{c} \mathbf{F}^q_{x,y,k} \cdot \boldsymbol{p}^f_{n,k}, \tag{12}$$

where x, y, k, and n are the indexes of height, width, channel and prototypes, respectively. On the other hand, we use $\boldsymbol{p}^f$ to suppress the background of the current frame, as:

$$\mathcal{M}^b_{x,y,n} = \sum_{k=1}^{c} (\mathbf{F}^q_{x,y,k} - \boldsymbol{p}^b_{n,k})^2. \tag{13}$$

Finally we fuse the temporal activation map $\mathcal{M}^f, \mathcal{M}^b \in \mathbb{R}^{h \times w \times N_a}$ with $\tilde{\mathbf{F}}^q$ to predict the target mask $\hat{M}_t$:

$$\hat{M}_t = \text{Conv}(\tilde{\mathbf{F}}^q \oplus \mathcal{M}^f \oplus \mathcal{M}^b), \tag{14}$$

where $\oplus$ denotes the concatenation operation and Conv denotes the segmentation head that includes a 3×3 convolution, a ReLU activation, and a 1×1 convolution.

Training and Inference

During training, we use the episode-based sequential training strategy to maintain consistency between the training and inference process. specifically, at each training episode, we sample sequential L frames as the query video and N images with the same category as support images. The raw network output $\hat{M}$, which is a probability map of being a foreground object, is directly used for learning the temporal part agent for the next frame. For the first frame of a query video (*i.e.*, $t = 1$), we set the $\mathcal{M}^f, \mathcal{M}^b$ to the zero matrix. The dropout prevents the network from overrelying on temporal information and ignoring the contribution of support. The prediction at each frame will be supervised by the provided ground truth M^q:

$$\mathcal{L}_{seg}(\hat{M}) = \mathcal{L}_{focal}(\hat{M}, M^q) + \mathcal{L}_{dice}(\hat{M}, M^q), \tag{15}$$

where $\mathcal{L}_{focal}$ denotes the focal loss (Lin et al. 2017) and $\mathcal{L}_{dice}$ denotes the dice loss (Milletari, Navab, and Ahmadi 2016). During validation, our model can be applied directly for the inference of query videos with unseen classes without additional test-time fine-tune. Therefore our method introduces no extra computational cost at validation time.

Experiments

Datasets and Evaluation Setting

Following the previous work of FSVOS, we conduct experiments on the YouTube-FSVOS (Chen et al. 2021) and MiniVSPW (Siam, Derpanis, and Wildes 2022) benchmark.

Methods	Query	1-shot					5-shot					FPS
		1	2	3	4	Mean	1	2	3	4	Mean	
DANet (Chen et al. 2021)	Video	23.1	36.5	25.0	26.4	27.8	24.0	37.2	27.1	28.6	29.2	52
TTI (Siam, Derpanis, and Wildes 2022)	Video	23.4	37.0	24.1	27.1	27.9	25.2	37.1	25.0	29.6	29.2	39
HPAN (Tang et al. 2023)	Video	23.9	37.3	30.7	26.6	29.6	25.7	40.8	32.9	28.2	31.9	43
VIPMT (Liu et al. 2023)	Video	23.8	38.8	29.1	28.5	30.1	26.2	42.2	31.6	29.4	32.4	38
PALN (ours)	Video	**25.0**	**39.9**	**34.1**	**30.7**	**32.4**	**27.6**	**42.4**	**36.4**	**31.3**	**34.4**	48

Table 2: Results of 4-fold cross-validation with metric $\mathcal{J}$ on MiniVSPW using one support image (1-shot) and five support images (5-shot). All competitors adopt ResNet-50 as the backbone for fair comparison. FPS indicates speed of inference.

Methods	YouTube-FSVOS		MiniVSPW	
	1-shot	5-shot	1-shot	5-shot
STM (Oh et al. 2019)	57.8	62.1	27.9	30.4
STCN (Cheng, Tai, and Tang 2021)	57.8	62.3	28.0	30.7
XMem (Cheng and Schwing 2022)	58.4	62.7	28.5	30.8
Cutie (Cheng et al. 2024)	58.7	63.3	28.6	31.0
PALN (ours)	**60.9**	**65.2**	**32.4**	**34.4**

Table 3: Comparison with state-of-the-art VOS methods.

PAE	SQAD	TAD	1-shot	5-shot
	✓		53.9	58.7
	✓	✓	56.4	61.4
✓	✓		58.1	62.2
✓	✓	✓	**60.9**	**65.2**

Table 4: Ablation of model components.

Figure 3: Qualitative results of our proposed PALN.

YouTube-FSVOS is built based on YouTube-VIS (Yang, Fan, and Xu 2019) dataset, which encompasses 2,238 videos in total with 3,774 annotated instances covering 40 categories. The dataset is divided into four folds, and each fold contains 30 categories for training and the rest 10 categories for testing. We conduct cross-validation over all four folds. MiniVSPW is built based on VSPW (Miao et al. 2021) dataset, which includes 20 categories. To better evaluate the generalization of the FSVOS methods, MiniVSPW provides longer sequences than YouTube-FSVOS. It is also split to four folds for cross-validation. Following the previous FSVOS and VOS practice (Chen et al. 2021; Perazzi et al. 2016), we adopt the region similarity $\mathcal{J}$ as the evaluation metric, which is the Jaccard index defined as the intersection-over-union (IoU) of the estimated segmentation and the ground-truth mask.

Implementation Details

The feature extractor is implemented by ResNet-50 (He et al. 2016) which is pretrained on ImageNet (Russakovsky et al. 2015) for fair comparisons. We conduct experiments in both 1-shot and 5-shot settings (*i.e.*, one and five labeled support images are provided, respectively). For 5-shot setting, We produce part agents for each support image separately. We set the number of part agents $N_a = 8$. During the training, our model is trained using the Adam (Kingma and Ba 2014) optimizer with an initial learning rate of 0.0001.

Comparison with State-of-the-art Methods

Quantitative Results. We compare our PALN with various state-of-the-art methods on 4 folds of YouTube-FSVOS and MiniVSPW datasets in the 1-shot and 5-shot settings, respectively. In addition to the FSVOS method, we also compare the VOS method and the FSIS method whose query is image instead of video. All the methods adopt ResNet-50 backbone for fair comparisons. For the YouTube-FSVOS dataset, as demonstrated in Table 1, our PALN outperforms all FSVOS and FSIS methods, *i.e.*, improvements of 3.5% and 1.0% $\mathcal{J}$ mean score in 1-shot and 5-shot, respectively. And our method consistently leads the accuracy in all folds. Note that our method does not require test-time fine-tuning. In Table 2, we conduct the comparison on MiniVSPW to verify the generalization abiliity of our method. Our PALN still achieve gains of 2.3% and 2.0% on $\mathcal{J}$ in 5-shot and 1-shot, respectively in the more challenging scenarios. From the comparison of FPS, it can be observed that our method achieves high performance while maintaining fast inference speed. To compare with the VOS methods, we use PALN for predicting the first frame and the following VOS methods are responsible for propagating the segmentation to the full video. Table 3 demonstrates that our method significantly exceeds these VOS methods in the few-shot setting.

Qualitative Results. In Figure 3, we provide some visual comparison cases. Compared with previous FSVOS methods, our PALN can effectively discriminate distractors, yield more lucid boundary and more complete segmentation.

Reference	Method	1-shot	5-shot
object prototypes	-	53.3	60.4
pixel features	-	56.4	61.4
part features	SLIC	58.8	63.2
	K-Means	59.7	64.9
	PAE (ours)	**60.9**	**65.2**

Table 5: Comparisons of different representative reference.

Strategy	1-shot	5-shot
top-N_a	59.0	63.8
random sample	60.1	64.3
cache mechanism (ours)	**60.9**	**65.2**

Table 6: Comparisons of different strategies of temporal part agents learning.

Agents	1-shot	5-shot
w/o temporal part agents	58.1	62.2
background agents	58.8	63.1
foreground agents	59.3	63.8
temporal part agents (ours)	**60.9**	**65.2**

Table 7: Ablation of the foreground and background agents in the temporal alignment decoder.

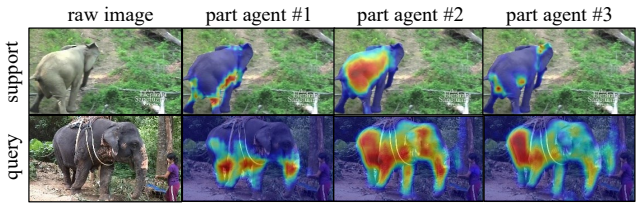

Figure 4: Activation maps of the learned part agents.

Ablation Study

We conduct comprehensive ablation studies on YouTube-FSVOS in terms of $\mathcal{J}$ metric in both 1-shot and 5-shot settings to verify the effectiveness of our proposed modules.

Analysis of Model Components. In Table 4, we denote the part agent encoder as **PAE**, the support-query alignment decoder as **SQAD** and the temporal alignment decoder as **TAD**. Since the alignment of support and query is necessary in FSVOS, we keep only the SQAD, while replacing part agents with pixel features, and removing TAD as our baseline, as the first row of the table. The performance of the baseline is comparable to the affinity learning method DANet (Chen et al. 2021). We validate the importance of each component by adding them in turn to the baseline. (1) Compared to the baseline, the performance boost brougnt by TAD demonstrates the effectiveness of the assistance of temporal part agents, which involve temporal consistency. (2) The utilization of PAE brings obvious improvements (*e.g.*, 3.9%), indicating that the part features can effectively address bottlenecks in FSVOS methods. (3) The introduction of both modules achieves remarkable accuracy gains compared with solely using the PAE or TAD. The improvement indicates that the collaboration of support and temporal information is extremely beneficial for FSVOS.

Effectiveness of the part features. In Table 5, the part features consistently achieve higher score in both settings than object prototypes and pixel features as reference. Among the methods to model part features, the part agents learned by the part agent encoder perform best. This is because the part agents do not have a semantic gap and and do not require additional supervision to decompose objects into diverse and complementary parts. These properties exactly solve the problems in SLIC (Achanta et al. 2012) and K-Means.

Effectiveness of the cache mechanism. In Table 6, we conduct experiments with different sampling strategies to obtain the temporal part agents in the temporal alignment decoder: "top-N_a" – selecting N_a pixels with the highest scores from the candidates; "random sample" – randomly sampling N_a agents from candidates. From the results, "top-N_a" performs poorly, which can be attributed to the unrepresentative nature of the clustered agents which lack the ability to model parts . "random sample" alleviates this problem to some extent but demonstrates from the side that our proposed cache mechanism can empower the agents with reliability, representativeness and diversity.

Effectiveness of the temporal part agents. In Table 7, we conduct experiments to verify the effectiveness of the foreground and background agents in the temporal alignment decoder. Compared to the method without temporal part agents, the background agents bring an average of 0.9% gain and the foreground agents leads to 1.4% performance improvement. It demonstrates that both foreground and background prototypes carry information useful for exploring temporal consistency, and the foreground has a more pronounced role. Adding both of them leads to huge performance improvement, showing that complete temporal part agents are required to fully model the temporal information.

Visualization analysis of part agents. In Figure 4, we use some examples of part agents to activate the support and query features. According to the activation map, different part agents can adaptively focus on different parts of the target object in the support. And the activation region is different and complementary. For the query features, the agents focuses on part regions of the target and rarely respond to background regions.

Conclusion

In this paper, we analyze the bottlenecks of previous methods from the perspectives of object prototypes and pixel features. Furthermore, we highlight the potential of part features as representative reference features, demonstrating their excellent properties of adaptation to appearance variations and resistance to correlation noise in the context of FSVOS. Therefore, we propose the Part Agent Learning Network (PALN), which harnesses part features from support images and historical query frames to achieve precise segmentation. Extensive experiments validate its effectiveness and superiority.

Acknowledgments

This work was supported by the National Defense Basic Scientific Research Program (Grant JCKY2021601B013).

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11): 2274–2282.

Boudiaf, M.; Kervadec, H.; Masud, Z. I.; Piantanida, P.; Ben Ayed, I.; and Dolz, J. 2021. 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*, 13979–13988.

Chen, H.; Wu, H.; Zhao, N.; Ren, S.; and He, S. 2021. Delving Deep Into Many-to-Many Attention for Few-Shot Video Object Segmentation. In *CVPR*, 14040–14049.

Cheng, B.; Misra, I.; Schwing, A. G.; Kirillov, A.; and Girdhar, R. 2022. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1290–1299.

Cheng, B.; Schwing, A.; and Kirillov, A. 2021. Per-pixel classification is not all you need for semantic segmentation. *Advances in neural information processing systems*, 34: 17864–17875.

Cheng, H. K.; Oh, S. W.; Price, B.; Lee, J.-Y.; and Schwing, A. 2024. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3151–3161.

Cheng, H. K.; and Schwing, A. G. 2022. XMem: Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVIII*, 640–658. Springer.

Cheng, H. K.; Tai, Y.-W.; and Tang, C.-K. 2021. Rethinking Space-Time Networks with Improved Memory Coverage for Efficient Video Object Segmentation. *arXiv preprint arXiv:2106.05210*.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26: 2292–2300.

Fan, Q.; Pei, W.; Tai, Y.-W.; and Tang, C.-K. 2022. Self-support few-shot semantic segmentation. In *European Conference on Computer Vision*, 701–719. Springer.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hong, S.; Cho, S.; Nam, J.; Lin, S.; and Kim, S. 2022. Cost aggregation with 4d convolutional swin transformer for few-shot segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, 108–126. Springer.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, G.; Jampani, V.; Sevilla-Lara, L.; Sun, D.; Kim, J.; and Kim, J. 2021. Adaptive Prototype Learning and Allocation for Few-Shot Segmentation. In *CVPR*, 8334–8343.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Liu, N.; Nan, K.; Zhao, W.; Liu, Y.; Yao, X.; Khan, S.; Cholakkal, H.; Anwer, R. M.; Han, J.; and Khan, F. S. 2023. Multi-grained Temporal Prototype Learning for Few-shot Video Object Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18862–18871.

Liu, Y.; Liu, N.; Cao, Q.; Yao, X.; Han, J.; and Shao, L. 2022a. Learning non-target knowledge for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11573–11582.

Liu, Y.; Liu, N.; Yao, X.; and Han, J. 2022b. Intermediate prototype mining transformer for few-shot semantic segmentation. *Advances in Neural Information Processing Systems*, 35: 38020–38031.

Liu, Y.; Zhang, X.; Zhang, S.; and He, X. 2020. Part-aware prototype network for few-shot semantic segmentation. In *ECCV*, 142–158. Springer.

Luo, N.; Pan, Y.; Sun, R.; Zhang, T.; Xiong, Z.; and Wu, F. 2023. Camouflaged Instance Segmentation via Explicit De-Camouflaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17918–17927.

Luo, N.; Wang, Y.; Sun, R.; Xiong, G.; Zhang, T.; and Wu, F. 2024. Exploring the Better Correlation for Few-Shot Video Object Segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*.

Mai, H.; Sun, R.; Zhang, T.; Xiong, Z.; and Wu, F. 2023. DualRel: Semi-Supervised Mitochondria Segmentation From a Prototype Perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19617–19626.

Miao, J.; Wei, Y.; Wu, Y.; Liang, C.; Li, G.; and Yang, Y. 2021. Vspw: A large-scale dataset for video scene parsing in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4133–4143.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 565–571. Ieee.

Oh, S. W.; Lee, J.-Y.; Xu, N.; and Kim, S. J. 2019. Video object segmentation using space-time memory networks. In *ICCV*, 9226–9235.

Pan, Y.; Luo, N.; Sun, R.; Meng, M.; Zhang, T.; Xiong, Z.; and Zhang, Y. 2023. Adaptive template transformer for mitochondria segmentation in electron microscopy images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21474–21484.

Pan, Y.; Sun, R.; Luo, N.; Zhang, T.; and Zhang, Y. 2024. Exploring Reliable Matching with Phase Enhancement for Night-time Semantic Segmentation. *arXiv preprint arXiv:2408.13838*.

Perazzi, F.; Pont-Tuset, J.; McWilliams, B.; Van Gool, L.; Gross, M.; and Sorkine-Hornung, A. 2016. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 724–732.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Shaban, A.; Bansal, S.; Liu, Z.; Essa, I.; and Boots, B. 2017. One-shot learning for semantic segmentation. *arXiv preprint arXiv:1709.03410*.

Siam, M.; Derpanis, K. G.; and Wildes, R. P. 2022. Temporal Transductive Inference for Few-Shot Video Object Segmentation. *arXiv preprint arXiv:2203.14308*.

Siam, M.; Doraiswamy, N.; Oreshkin, B. N.; Yao, H.; and Jagersand, M. 2021. Weakly supervised few-shot object segmentation using co-attention with visual and semantic embeddings. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 860–867.

Sun, R.; Li, Y.; Zhang, T.; Mao, Z.; Wu, F.; and Zhang, Y. 2021. Lesion-aware transformers for diabetic retinopathy grading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10938–10947.

Sun, R.; Wang, Y.; Mai, H.; Zhang, T.; and Wu, F. 2023. Alignment before aggregation: trajectory memory retrieval network for video object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1218–1228.

Tang, Y.; Chen, T.; Jiang, X.; Yao, Y.; Xie, G.-S.; and Shen, H.-T. 2023. Holistic Prototype Attention Network for Few-Shot VOS. *arXiv preprint arXiv:2307.07933*.

Tian, Z.; Zhao, H.; Shu, M.; Yang, Z.; Li, R.; and Jia, J. 2020. Prior guided feature enrichment network for few-shot segmentation. *TPAMI*, (01): 1–1.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Villani, C. 2009. *Optimal transport: old and new*, volume 338. Springer.

Wang, H.; Zhang, X.; Hu, Y.; Yang, Y.; Cao, X.; and Zhen, X. 2020. Few-shot semantic segmentation with democratic attention networks. In *ECCV*, 730–746. Springer.

Wang, Y.; Luo, N.; and Zhang, T. 2023. Focus on query: Adversarial mining transformer for few-shot segmentation. *Advances in Neural Information Processing Systems*, 36: 31524–31542.

Wang, Y.; Sun, R.; and Zhang, T. 2023. Rethinking the Correlation in Few-Shot Segmentation: A Buoys View. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7183–7192.

Wang, Y.; Sun, R.; Zhang, Z.; and Zhang, T. 2022. Adaptive Agent Transformer for Few-Shot Segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, 36–52. Springer.

Wu, Z.; Shi, X.; Lin, G.; and Cai, J. 2021. Learning meta-class memory for few-shot semantic segmentation. In *ICCV*, 517–526.

Xie, G.-S.; Liu, J.; Xiong, H.; and Shao, L. 2021a. Scale-Aware Graph Neural Network for Few-Shot Semantic Segmentation. In *CVPR*, 5475–5484.

Xie, G.-S.; Liu, J.; Xiong, H.; and Shao, L. 2021b. Scale-aware graph neural network for few-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5475–5484.

Xiong, G.; Wang, Y.; Li, Z.; Yang, W.; Zhang, T.; Zhou, X.; Zhang, S.; and Zhang, Y. 2024. Aggregation and Purification: Dual Enhancement Network for Point Cloud Few-shot Segmentation. In Elkind, E., ed., *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*. International Joint Conferences on Artificial Intelligence Organization.

Yang, B.; Liu, C.; Li, B.; Jiao, J.; and Ye, Q. 2020. Prototype mixture models for few-shot semantic segmentation. In *ECCV*, 763–778. Springer.

Yang, L.; Fan, Y.; and Xu, N. 2019. Video instance segmentation. In *ICCV*, 5188–5197.

Yang, Z.; Wei, Y.; and Yang, Y. 2021. Collaborative Video Object Segmentation by Multi-Scale Foreground-Background Integration. *TPAMI*.

Zhang, B.; Xiao, J.; and Qin, T. 2021. Self-Guided and Cross-Guided Learning for Few-Shot Segmentation. In *CVPR*, 8312–8321.

Zhang, G.; Kang, G.; Wei, Y.; and Yang, Y. 2021. Few-Shot Segmentation via Cycle-Consistent Transformer. *arXiv preprint arXiv:2106.02320*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.