Learning Position and Target Consistency for Memory-based Video Object Segmentation

Li Hu, Peng Zhang, Bang Zhang, Pan Pan, Yinghui Xu, Rong Jin
Machine Intelligence Technology Lab, Alibaba Group
{hooks.hl, futian.zp, zhangbang.zb, panpan.pp, renji.xyh, jinrong.jr}@alibaba-inc.com

Abstract

This paper studies the problem of semi-supervised video object segmentation(VOS). Multiple works have shown that memory-based approaches can be effective for video object segmentation. They are mostly based on pixel-level matching, both spatially and temporally. The main shortcoming of memory-based approaches is that they do not take into account the sequential order among frames and do not exploit object-level knowledge from the target. To address this limitation, we propose to Learn position and target Consistency framework for Memory-based video object segmentation, termed as LCM. It applies the memory mechanism to retrieve pixels globally, and meanwhile learns position consistency for more reliable segmentation. The learned location response promotes a better discrimination between target and distractors. Besides, LCM introduces an object-level relationship from the target to maintain target consistency, making LCM more robust to error drifting. Experiments show that our LCM achieves state-of-the-art performance on both DAVIS and Youtube-VOS benchmark. And we rank the 1st in the DAVIS 2020 challenge semi-supervised VOS task.

1. Introduction

Video object segmentation(VOS) is a fundamental computer vision task, with a wide range of applications including video editing, video composition and autonomous driving. In this paper, we focus on the task of semi-supervised video object segmentation. Given a video and the ground truth object mask of the first frame, semi-supervised VOS predicts the segmentation masks of the objects specified by the ground truth mask in the first frame for the remaining frames. In video sequences, the target object will undergo large appearance changes due to continuous motion and variable camera view. And it may disappear in some frames due to occlusion between different objects. Furthermore, there are also similar instances of same categories that are difficult to distinguish, making the problem even harder. Therefore, semi-supervised VOS is extremely challenging despite the provided annotation in the first frame.

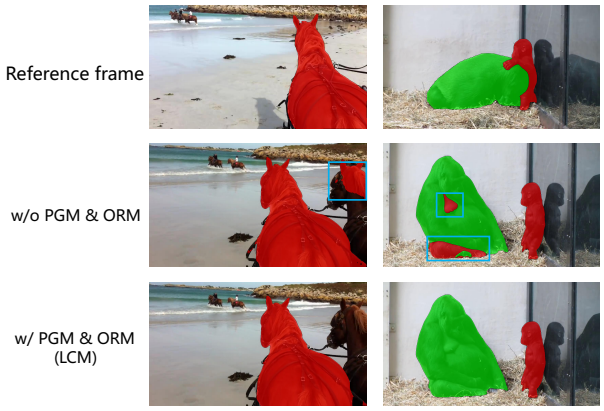

Figure 1. Typical memory-based approaches rely on pixel-level similarity, which leads to errors in prediction, as show in second row. The proposed Position Guidance Module(PGM) helps the network track the motion trajectory(bottom left). And the object-aware Object Relation Module(ORM) prevents the network from making fragmented segmentation pieces(bottom right).

The fundamental problem of VOS lies in how to make full use of the spatio-temporally structured information contained in video frames. Memory-based approaches are recently proposed with significant performance improvements in popular VOS benchmarks, e.g. DAVIS[33, 34] and Youtube-VOS[46]. Space-Time Memory network(STM)[29] is the first memory-based semi-supervised VOS method, developing a memory mechanism to store information from all previous frames for the query frame to read. It differs from other matching-based methods as it expands its search range to the entire space-time domain and perform dense matching in the feature space. However, memory-based methods only consider pixel-level matching and tend to retrieve all pixels with high matching score in the query image. It may fail when a non-target region share similar visual appearance with the target re-

gions as illustrated in Figure 1. Recently, KMN[36] introduces memory-to-query matching to improve STM. But the solution remains pixel similarity matching which cannot deal with appearance changes and deformation. In order to tackle the aforementioned issues, we propose to improve memory-based methods from two aspects: 1) Position consistency. The movement of objects usually follows a certain trajectory, which serves as an important instruction to guide segmentation. 2) Target consistency. The overall embedding feature for the tracked target should maintain object-level consistency throughout the entire video.

Propagation-based methods[48, 45, 24] introduce to directly utilize the prediction from previous frames for better segmentation. Inspired by these works, we propose to apply previous positional information as a guidance for memory-based methods to maintain position consistency. Typical matching-based methods[17, 6] only consider pixel-level feature without the context information from the entire object. Inspired by some works in tracking[2] and one/few-shot detection[10, 14], we propose to integrate object-level feature into memory-based network to maintain target consistency.

To this end, we propose a novel framework to Learn position and target Consistency for Memory-based video object segmentation(LCM). Taking advantage of STM, LCM performs pixel-level matching mechanism to retrieve target pixels based on similarity and stores previous information in a memory pool. This procedure is named Global Retrieval Module(GRM). Besides, LCM learns a local embedding named Position Guidance Module(PGM) to fully utilize the position consistency and guides the segmentation by learning a location response. To maintain target consistency, LCM introduces Object Relation Module(ORM). As the target object is annotated in the first frame of a video, the object relationship from the first value embedding is encoded to the query frame, which serves as a consistent fusion for context feature during the entire video sequence. Figure 1 illustrates the effectiveness of our LCM against typical errors in memory-based methods.

Our contributions can be summarized as follows:

- We propose a novel Position Guidance Module to compute a location response to maintain position consistency in memory-based methods.
- We propose Object Relation Module to effectively fuse object-level information for maintaining consistency of the target object.
- We achieve state-of-the-art performance on both DAVIS and Youtube-VOS benchmark and rank the 1st in the DAVIS 2020 challenge semi-supervised VOS task.

2. Related Works

Top-down methods for VOS. Top-down methods tackle video object segmentation with two processes. They first conduct detection methods to obtain proposals for target objects and then predict mask results. PReMVOS[27] utilizes Mask RCNN[12] to generate coarse mask proposals and conducts refinement, optical flow and re-identification to achieve a high performance. DyeNet[21] applies RPN[35] to extract proposals and uses Re-ID Module to associate proposal with recurrent mask propagation. TANDTTM[18] proposes Temporal Aggregation Network and Dynamic Template Matching to combine RPN with videos and select correct RoIs. Top-down methods rely heavily on the pre-trained detectors and the pipelines are usually too complicated to conduct end-to-end training.

Propagation-based methods for VOS. Propagation-based methods utilize the information from previous frames. MaskTrack[31] directly concatenates previous mask with current image as the input. RGMP[45] also concatenates previous masks and proposes a siamese encoder to utilize the first frame. OSMN[48] designs a modulator to encode spatial and channel modulation parameters computed from previous results. AGSS-VOS[24] uses current image and previous results and combines instance-specific branch and instance-agnostic branch with attention-guided decoder. In general, previous frame is similar in appearance to the current frame, but it cannot handle occlusion and error drifting. And previous works usually conduct implicit feature fusion which is lack of interpretability.

Matching-based methods for VOS. Matching-based methods perform pixel-level matching between template frame and current frame. PML[6] proposes a embedding network with triplet loss and nearest neighbor classifier. VideoMatch[17] conducts soft matching with foreground and background features to measure similarity. FEELVOS[39] proposes global and local matching according to the distance value. CFBI[49] applies background matching together with an instance-level attention mechanism. The main inspiration of our work is STM[29] which proposes to use all previous frames by storing information as memory. KMN[36] applies Query-to-Memory matching to improve original STM with kernelized memory read. Matching-based methods ignore the temporal information especially positional relationship. And they miss the knowledge from the overall target object.

Attention mechanism. Attention is widely adopted in machine learning including natural language process and computer vision. Non-local[42] network computes attention response at a position as a weighted sum of the features at all positions, capturing global and long-term information. [53] proposes a generalized attention formulation for modeling spatial attention. Many semantic segmentation works[19, 52, 50] utilize attention to build context in-

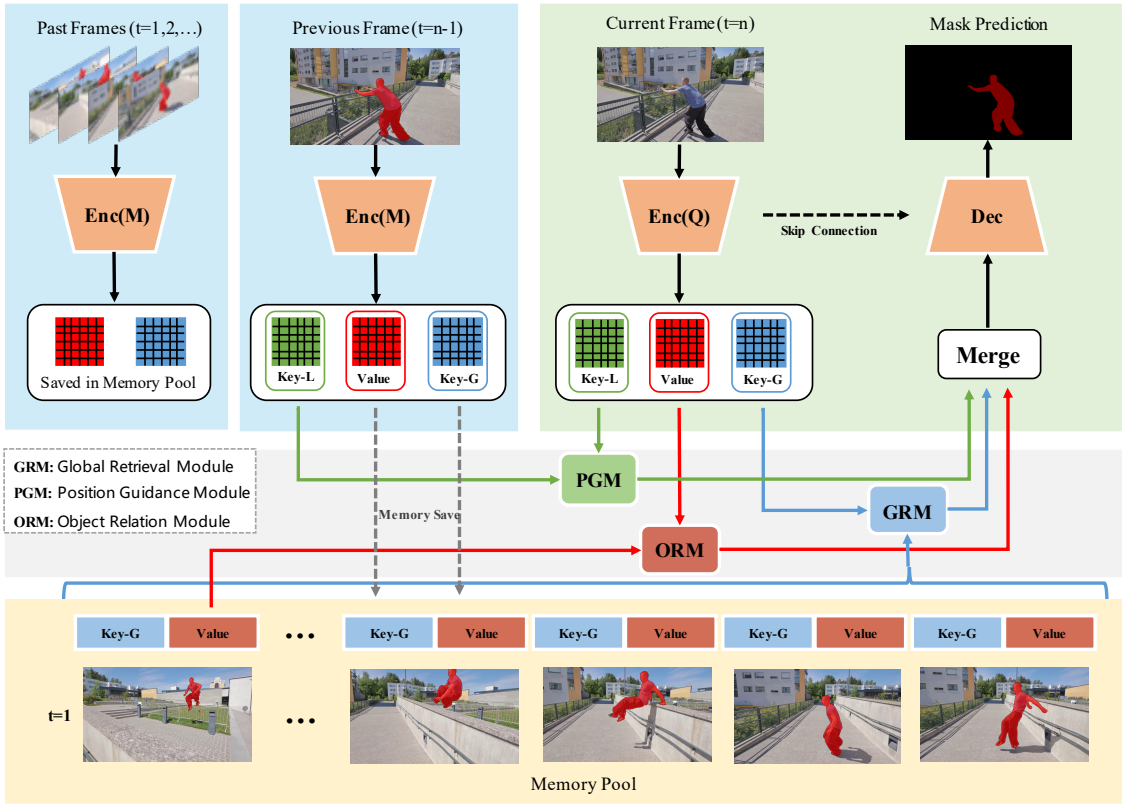

Figure 2. The overview of our LCM. The information of past frames are stored in memory pool. Global Retrieval Module(GRM) conducts pixel-level matching between query and memory pool. Position Guidance Module(PGM) encodes information from previous frame. Object Relation Module(ORM) fuses feature from first value embedding.

formation for every pixels. [14] emphasizes the features of the query and images via co-attention and co-excitation.

3. Methods

We first present the overview of our LCM in section 3.1. In section 3.2, we describe the Global Retrieval Module. Then we introduce the proposed Position Guidance Module and Object Relation Module in section 3.3 and section 3.4. Finally, the detail of training strategy is in section 3.5.

3.1. Overview

The overall architecture of LCM is illustrated in Figure 2. LCM uses a typical Encoder-Decoder architecture to conduct segmentation. For a query image, the query encoder produces three embeddings, i.e. $Key\text{-}G$, $Key\text{-}L$ and $Value$. The embeddings are fully exploited in three submodules: Global Retrieval Module(GRM), Position Guidance Module(PGM) and Object Relation Module(ORM). First, GRM is designed the same as Space-Time Memory Network(STM)[29]. It calculates a pixel-level feature correlation between the current frame and memory pool. The $Key\text{-}G$ and $value$ from previous frames are stored in memory pool via the memory encoder. Second, we propose PGM, which learns a feature embedding for both current frame and previous adjacent frame. Obviously previous frame contains similar positional information with current frame. Thus we build a positional relationship between these two frames which enhances positional constrain for the retrieved pixels. Moreover, to merge object-level information into pixel-level matching procedure and to prevent the accumulative error in memory pool, we propose ORM. The information of objects in the first frame will be maintained during entire sequential inference. Finally, we introduce the training strategy of our LCM. In the following section, we will further present a specific description.

3.2. Global Retrieval Module

Global Retrieval Module(GRM) highly borrows the implementation of Space-time Memory Network(STM)[29]. As illustrated in Figure 2, Previous frames together with its mask predictions are encoded through the memory encoder meanwhile current frame is encoded through the query encoder. We use the ResNet-50[13] as backbone for both encoders. For the tth frame, the output feature maps are

defined as $r^M \in \mathbb{R}^{H\times W\times C}$ and $r^Q \in \mathbb{R}^{H\times W\times C}$. For previous frames, the memory global key $k^M \in \mathbb{R}^{H\times W\times C/8}$ and memory value $v^M \in \mathbb{R}^{H\times W\times C/2}$ are embedded through two separated 3×3 convolutional layers from r^M. Then both embeddings are stored in memory pool and are concatenated along the temporal dimension, which are defined as $k_p^M \in \mathbb{R}^{T\times H\times W\times C/8}$ and $v_p^M \in \mathbb{R}^{T\times H\times W\times C/2}$. For query image, the query global key $k^Q \in \mathbb{R}^{H\times W\times C/8}$ will be embedded from r^Q. The Global Retrieval Module retrieves the matched pixel feature based on the similarity of the global key between query and memory pool by the following formulation:

$$s(i,j) = \frac{exp(k_p^M(i) \odot k^Q(j)^\intercal)}{\sum_i exp(k_p^M(i) \odot k^Q(j)^\intercal)} \tag{1}$$

where i and j are the pixel feature indexes of memory pool and the query. $\odot$ represents the matrix inner production, and function s denotes the $softmax$ operation, determining the location of the most similar pixel feature in memory pool for the query. Then the retrieved value feature is calculated as:

$$y^{GRM}(j) = \sum_i s(i,j) \odot v_p^M(i) \tag{2}$$

Global Retrieval Module encourages the query to search for the pixel-level appearance feature with high similarity along both spatial and temporal dimension. The main contribution of this module is its high recall. However, such mechanism does not fully utilize the characteristics of video object segmentation. The calculation of the correlation map is equally conducted with all features in memory pool without position consistency. As a consequence, the network tends to learn where to find the similar area but not correctly tracking the target object. The following proposed modules aim to solve above problems.

3.3. Position Guidance Module

In video object segmentation, the motion trajectory of an object is continuous and the recent frames usually contain the cues of approximate location of the target. When conducting Global Retrieval, all pixels with high similarity will be matched. Thus, if some small areas or other objects besides the tracked one have similar appearance feature, the Global Retrieval Module often incorrectly retrieves them as illustrated in Figure 1. Thus, the positional information from recent frames should be effectively used.

Here we introduce Position Guidance Module(PGM) which encodes previous adjacent frame to learn position consistency. As shown in Figure 2, in addition to output global key, we also propose to extract local key from the $res4$ feature map for local position addressing. Specifically, another 3×3 convolutional layer is applied for both query embedding and previous adjacent memory embedding to

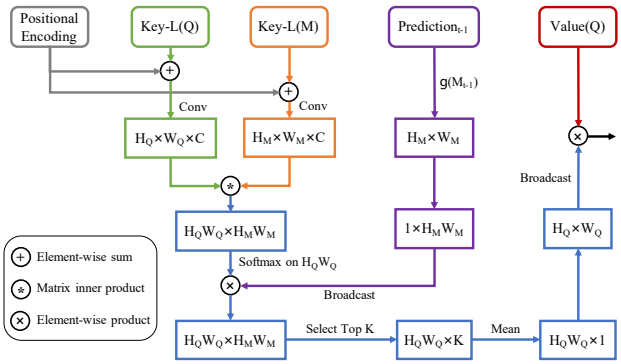

Figure 3. Implementation of Position Guidance Module.

output query local key $k_L^Q \in \mathbb{R}^{H\times W\times C/8}$ and memory local key $k_L^M \in \mathbb{R}^{H\times W\times C/8}$.

The implementation of Position Guidance Module is depicted in Figure 3. The global key is learned to encode visual semantics for matching robust to appearance variations as described in STM. In comparison, the local key is designed to not only address feature similarity but also encode positional correspondence. Since the matrix operation for these embeddings is position-invariant, we supplement them with 2D positional encodings[30, 4] to maintain location cues. We use sine and cosine functions with different frequencies to define a fixed absolute encoding associated with the corresponding position, formulating it as $pos(i)$. Positional encodings are added to both local keys followed by a 1×1 convolutional layer f_n. We depict the process as follows.

$$p^M(i) = f_n(k_L^M(i) + pos(i)) \tag{3}$$

$$p^Q(j) = f_n(k_L^Q(j) + pos(i)) \tag{4}$$

Then we reshape p^M and p^Q and apply matrix inner product to get the embedding S with size of $HW\times HW$. Softmax operation is applied on the query dimension to form a response distribution for each location in the previous frame. Meanwhile we use the previous predicted mask to reduce the response of non-object areas. The calculation is defined as:

$$S(i,j) = \frac{exp(p^Q(j) \odot p^M(i)^\intercal)}{\sum_j exp(p^Q(j) \odot p^M(i)^\intercal)} * g(M_{t-1}) \tag{5}$$

where $g(x) = \frac{exp(x)}{e}$ prevents the response from the location of background close to zero since the previous prediction is not always correct. Next we select the top-K values on the memory dimension and average them to get the position map of size $H\times W$. Experimentally, we set $K = 8$. The selected locations in the memory map determine a significant position association with corresponding query location. And the location with high response value in the position map represents the area where objects are most likely

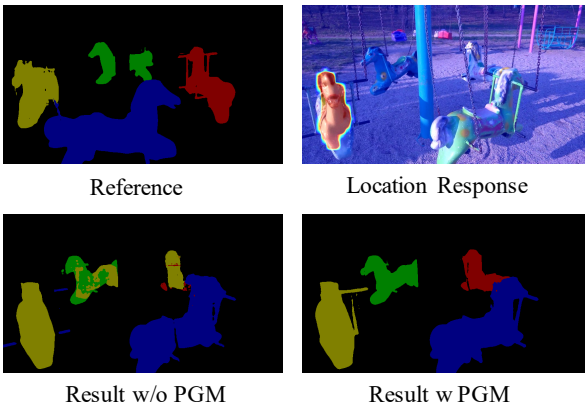

Figure 4. The effectiveness of PGM.

to appear in the query image. Finally, this position map serves as a spatial attention map and we conduct element-wise product between the position map and the query value v^Q:

$$y^{PGM}(j) = \frac{\sum_i topK\{S(i,j)\}}{K} * v^Q(j) \tag{6}$$

To demonstrate the effectiveness of PGM, we illustrate the typical case in Figure 4. Without PGM, pixels of similar objects are likely to be retrieved due to the high appearance similarity. As a comparison, PGM promotes a better discrimination between target and distractors. We normalize the learned location response in PGM to a heatmap. The result shows that PGM learns a response distribution which not only considers the similarity of the appearance features between objects, but also correctly determines the location area of the target.

3.4. Object Relation Module

In video object segmentation, it is critical to utilize object-level feature of the target, which is not covered by above mechanism. The matching-based pixel retrieval is a bottom-up approach and lack of context information. During video inference, the accumulative error often brings noisy (Key-G, $value$) pairs into memory pool and will mislead the subsequent pixel matching process and position guidance as shown in middle right of Figure 1. To tackle above problems, it is essential to additionally utilize the first frame as it always provides intact and reliable masks. Specifically, we propose Object Relation Module(ORM) to fuse the object-level information of the first frame as a prior into the inference of entire video stream to maintain target consistency.

In Object Relation Module, we start from the first value v^F and the query value v^Q. The module structure is illustrated in Figure 5. According to the ground truth mask, for each object we select the foreground feature in the first value v^F into a value set $F\{f_i\}$, where i denotes the location that belongs to certain object mask. Inspired by [14], we design a cross relation mechanism to merge object-level feature into the query value. For both $F\{f_i\}$ and $v^Q(j)$, we conduct non-local operation and output respective non-local relation feature $F_Q\{f_i\}$ and $v_F^Q(j)$ as follows:

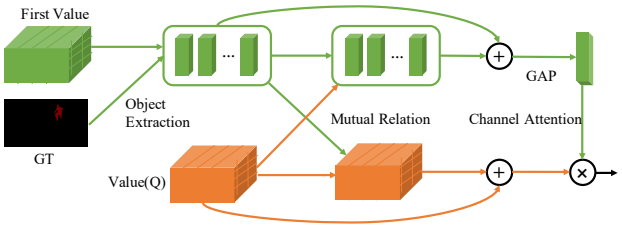

Figure 5. Process of Object Relation Module.

$$F_Q\{(f_i)\} = \frac{1}{d}\sum_j f(F\{f_i\}, v^Q(j)) * g(v^Q(j)) \tag{7}$$

$$v_F^Q(j) = \frac{1}{d}\sum_i f(v^Q(j), F\{f_i\}) * g(F\{f_i\}) \tag{8}$$

where $d = H*W$ is the normalization factor and g is a 1×1 convolutional layer. f denotes dot product between two vectors. Then the original feature is enhanced by the non-local relation feature via element-wise sum. Furthermore, we conduct global average pooling on the enhanced first value feature followed by two fully-connected layers and Sigmoid function as in the design of SENet[15], serving as the channel-wise attention. Thus, the query value can adaptively re-weighting the importance coefficient over channels through the instruction from object-level feature. The process is summarized as follows, where GAP indicates global average pooling:

$$v^Q(j) = v^Q(j) + v_F^Q(j) \tag{9}$$

$$F\{f_i\} = F\{f_i\} + F_Q\{(f_i)\} \tag{10}$$

$$y^{ORM}(j) = v^Q(j) * GAP(F\{f_i\}) \tag{11}$$

Object Relation Module encodes object-sensitive information flow into the feature extraction. The output is merged with Position Guidance Module and concatenated with the memory value from Global Retrieval Module as the final feature. We employ the decoder described in [45, 29] to gradually upsample the feature map combined with residual skip connections to estimate the object mask. We apply soft aggregation[45, 29] to merge the multi-object predictions.

3.5. Training Strategy

Pre-training on static images. As widely used in recent VOS task[45, 29, 36], we simulate fake video dataset

with static images to pre-train the network for better parameter initialization. We leverage image segmentation datasets[8, 37, 25] for pre-training. A synthetic clip contains three frames. Specifically, one image is sampled from real dataset and generates other two fake images by applying random affine transforms.

Main-training on real videos without temporal limit. In this step, we leverage video object segmentation datasets to train the model. Different from the original main training setting in [29], we do not limit video sampling intervals. Three frames are randomly selected from a video sequence and we randomly shuffle the order of them. Only objects that appear in all three frames are selected as foreground objects. This strategy encourages the network to strength retrieval capability since the target object will appear in all possible regions.

Fine-tuning on real videos as sequence. At inference of video object segmentation, the mask results is computed frame by frame sequentially. Therefore, in this training stage we further fine-tune the model to reduce the gap between training and testing. We sample three frames with time-order and the skip number is randomly selected from 1 to 5. The predicted soft mask result is used to compute memory embeddings. This training mechanism construct training samples with sequence information, which benefits the training of PGM.

Training Details. We initialize the network with ImageNet pretrained parameters. During pre-training, we conduct translation, rotation, zooming and bluring to transform images and randomly crop 384×384 patches. We minimize the cross-entropy loss using Adam optimizer with learning rate of 5e-4. During main-training and fine-tuning, we randomly crop a 640×384 patch around the maximum bounding box of all objects in three frames. Adam optimizer with learning rate of 1e-5 is used in main-training and SGD optimizer with learning rate of 3e-4 for fine-tuning. We use 8 Tesla V100 GPUs. Pre-training takes 25 hours(10 epoch). Training without temporal limit takes 12 hours(200 epoch). Training as sequence takes 3 hours(50 epoch). We do not apply post-processing or online training.

4. Experiments

We evaluate our model on DAVIS[33, 34] and YouTube-VOS[46], two popular VOS benchmarks with multiple objects. For YouTube-VOS, we train our model on the YouTube-VOS training set and report the result on YouTube-VOS 2018 validation set. For the evaluation on DAVIS, we train our model on DAVIS 2017 training set with 60 videos. Both DAVIS 2016 and 2017 are evaluated using an identical model trained on DAVIS 2017 for a fair comparison with the previous works. We also report the result trained with both DAVIS 2017 and YouTube-VOS(3471 videos) following recent works.

		Seen		Unseen	
	Overall	$\mathcal{J}$	$\mathcal{F}$	$\mathcal{J}$	$\mathcal{F}$
OSMN[48]	51.2	60.0	60.1	40.6	44.0
MSK[32]	53.1	59.9	59.5	45.0	47.9
RGMP[45]	53.8	59.5	-	45.2	-
OnAVOS[40]	55.2	60.1	62.7	46.6	51.4
RVOS[38]	56.8	63.6	67.2	45.5	51.0
OSVOS[3]	58.8	59.8	60.5	54.2	60.7
S2S[47]	64.4	71.0	70.0	55.5	61.2
A-GAME[20]	66.1	67.8	-	60.8	-
PreMVOS[27]	66.9	71.4	75.9	56.5	63.7
BoLTVOS[41]	71.1	71.6	-	64.3	-
DMM[51]	58.0	60.3	63.5	50.6	57.4
CapsuleVOS[51]	62.3	67.3	53.7	68.1	59.9
GC[22]	73.2	72.6	75.6	68.9	75.7
AFB_URR[23]	79.6	78.8	83.1	74.1	82.6
GraphMem[26]	80.2	80.7	85.1	74.0	80.9
CFBI[49]	81.4	81.1	85.8	75.3	83.4
LWTL[11]	81.5	80.4	84.9	**76.4**	**84.4**
KMN[36]	81.4	81.4	85.6	75.3	83.3
STM[29]	79.4	79.7	84.2	72.8	80.9
LCM	**82.0**	**82.2**	**86.7**	75.7	83.4

Table 1. The quantitative evaluation on Youtube-VOS 2018 validation dataset.

The evaluation metric is the average of $\mathcal{J}$ score and $\mathcal{F}$ score. $\mathcal{J}$ score calculates the average IoU between the prediction and the ground truth mask. $\mathcal{F}$ score calculates an average boundary similarity between the boundary of the prediction and the ground truth mask.

4.1. Compare with the State-of-the-art Methods

Youtube-VOS[46] is the largest dataset for video segmentation which consists of 4453 high-resolution videos. In detail, the dataset contains 3471 videos in the training set (65 categories), 474 videos in the validation set (additional 26 unseen categories). We train our model on Youtube-VOS training set and evaluate it on Youtube-VOS-18 validation set.

As shown in Table 1, our approach LCM obtains a final score of 82.0%, significantly outperforming our baseline STM(79.4%) of 2.6%. It demonstrates the effectiveness of our proposed modules on typical memory-based methods. Compared with other recent works, LCM also achieves state-of-the-art performance. CFBI[49] is built on a strong pipeline with COCO[25] pre-trained DeepLabV3+[5] and a well-designed segmentation head. KMN applies a Hide-and-Seek training strategy which improves the diversity and accuracy of training data and is a general data-augmentation for any other memory-based VOS methods including LCM. Without these enhancements, our performance is still higher. This result demonstrates the ro-

	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Overall
Validation Set			
OSVOS[3]	56.6	63.9	60.3
PReMVOS[27]	73.9	81.7	77.8
OSVOS^s[28]	64.7	71.3	68.0
OSMN[48]	52.5	57.1	54.8
VideoMatch[17]	56.5	68.2	62.4
RGMP[45]	64.8	68.6	66.7
A-Game[20]	67.2	72.7	70.0
FAVOS[7]	54.6	61.8	58.2
FEELVOS[39](+YV)	69.1	74.0	71.5
DMM[51]	68.1	73.3	70.7
RANet[43]	63.2	-	65.7
GC[22]	69.3	73.5	71.4
AFB_URR[23]	73.0	76.1	74.6
LWTL[11](+YV)	79.1	84.1	81.6
CFBI[49](+YV)	79.1	84.6	81.9
GraphMem[26](+YV)	80.2	85.2	82.8
KMN[36](+YV)	80.0	85.6	82.8
STM[29]	69.2	74.0	71.6
LCM	73.1	77.2	75.2
STM[29](+YV)	79.2	84.3	81.8
LCM(+YV)	**80.5**	**86.5**	**83.5**
Test-dev Set			
PReMVOS[27]	67.5	75.7	71.6
RGMP[45]	51.3	54.4	52.9
FEELVOS[39](+YV)	55.2	60.5	57.8
RANet[43]	53.4	-	55.3
CFBI[49](+YV)	71.1	78.5	74.8
KMN[36](+YV)	74.1	80.3	77.2
STM[29](+YV)	69.3	75.2	72.2
LCM(+YV)	**74.4**	**81.8**	**78.1**

Table 2. The quantitative evaluation on DAVIS-2017 validation and test-dev dataset. (+YV) indicates training with both DAVIS and Youtube-VOS.

bustness and generalization of our approach on a complex dataset.

DAVIS 2017[34] is a multi-object extension of DAVIS 2016 and it is more challenging than DAVIS 2016 since the model needs to consider the difference between various objects. The validation set of DAVIS 2017 consists of 59 objects in 30 videos. In this section we evaluate our model on both DAVIS 2017 validation and test-dev benchmark.

The results are compared to state-of-the-art approaches in Table 2. Our method shows state-of-the-art results. When applying both DAVIS and Youtube-VOS datasets for training, LCM achieves 83.5%, surpassing our baseline STM of 1.7%. And LCM also shows higher performance than other existing methods including online-learning methods and offline-learning methods. Follwing recent work, we also report the result with only DAVIS for training. And

	Time(s)	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Overall
OSVOS[3]	9	79.8	80.6	80.2
MaskRNN[16]	-	80.7	80.9	80.8
LSE[9]	-	82.9	80.3	81.6
CINN[1]	30	83.4	85.0	84.2
PReMVOS[27]	32.8	84.9	88.6	86.8
OnAVOS[40]	13	86.1	84.9	85.5
RANet[43]	4	86.6	87.6	87.1
FEELVOS[39]	0.45	81.1	82.2	81.7
RGMP[45]	0.13	81.5	82.0	81.8
A-Game[20]	0.07	82.0	82.2	82.1
FAVOS[7]	1.8	82.4	79.5	81.0
DMVOS[44]	0.035	87.8	87.5	88.0
RANet[43]	0.03	85.5	85.4	85.5
GC[22]	0.04	87.6	85.7	86.6
CFBI[49]	0.18	88.3	90.5	89.4
KMN[36]	0.12	89.5	**91.5**	90.5
STM[29]	0.112	88.7	89.9	89.3
LCM	0.118	**89.9**	91.4	**90.7**

Table 3. The quantitative evaluation on DAVIS-2016 validation dataset. The running time of STM is our reimplement result.

LCM outperforms the baseline STM of 3.6%. In addition, we report the result on the DAVIS testing split and also shows best results of 78.1%, surpassing STM by a significant margin($+5.9$). By employing similar approaches in LCM together with other tricks such as better backbone, strong segmentation head, multi-scale testing and model ensemble, we achieve 84.1% on the DAVIS challenge split and rank the 1st in the DAVIS 2020 challenge semi-supervised VOS task.

DAVIS 2016[33] consists of 20 videos annotated with high-quality masks each for a single target object. As shown in Table 3, LCM also achieves state-of-the-art performance. Compared to other methods, LCM is slightly higher than KMN of 0.2%. Since DAVIS 2016 is relatively a simple dataset and its performance highly relies on the precision of segmentation detail. A possible reason is that the Hide-and-Seek can provide more precise boundaries as described in KMN. Compared to the baseline STM, LCM shows better accuracy (89.3vs.90.7).

We also report the running time on DAVIS2016. We use 1 Tesla P100 GPU for inference. The increased running time brought by PGM and ORM is no more than 6% compared with the baseline STM. We also compare it with other existing methods and our LCM maintains a comparable fast inference speed with higher performance.

4.2. Qualitative Results.

We show the qualitative results compared with memory-based method STM in Figure 6. We use the author's officially released pre-computed results. The result shows that

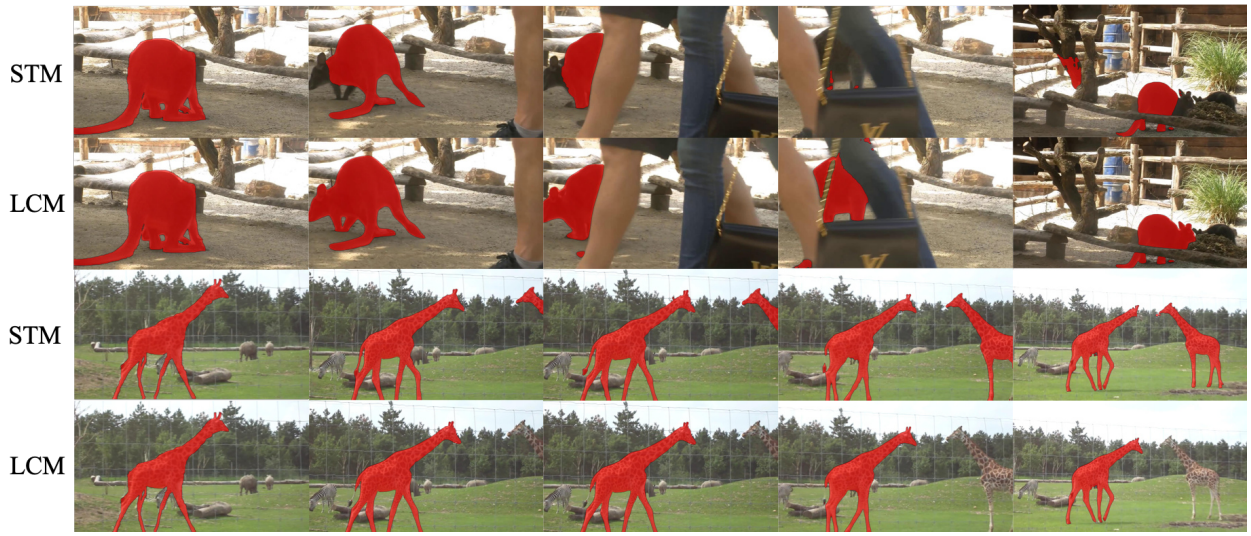

Figure 6. Qualitative results of our proposed LCM. Our model is more robust under challenging situation such as occlusion, appearance change and similar objects.

LCM can reduce typical errors in memory-based method and is more robust under challenging situation such as occlusion, appearance change and similar objects.

4.3. Ablation Study

We conduct an ablation study on DAVIS 2017 validation set to demonstrate the effectiveness of our approach.

Network Sub-module. We experimentally analyze the effectiveness of our proposed three sub-modules. In this experiment, we do not apply pre-training step for saving time and directly use DAVIS and Youtube-VOS to train our model. The result is shown in Table 4. When applying all three proposed modules, LCM achieves 79.2% on DAVIS 2017 validation set without pre-training. The performance drops to 77.8% and 78.4% respectively When we disable Position Guidance Module or Object Relation Module. Without both modules, the result degrades to 76.9%, which demonstrates the importance of these two modules. Furthermore, when disabling Global Retrieval Module, the performance heavily drops from 79.2% to 67.5%. The reason is that Global Retrieval Module is the fundamental module of LCM otherwise a large amount of information is absent without memory pool.

Training Strategy. We experimentally analyze the impact of our training strategy. The result is shown in Table 5. When only conducting pre-training and training without temporal limit, the performance achieves 82.9%, which is already a state-of-the-art performance. When only conducting pre-training and training as sequence, the result degrades to 80.7%.The reason is that small sampling interval makes the model incapable to learn appearance change and fast motion. Consequently, our framework has the best performance when combining all three training stages.

GRM	PGM	ORM	$\mathcal{J}$ Mean	$\mathcal{F}$ Mean	Overall
✓	✓	✓	77.1	81.4	79.2
✓		✓	75.5	80.1	77.8
✓	✓		76.0	80.8	78.4
✓			74.6	79.2	76.9
	✓	✓	65.2	69.8	67.5

Table 4. Ablation study of the network sub-module on DAVIS 2017 validation without pre-training.

Training Strategy	$\mathcal{J}$	$\mathcal{F}$	Avg
Combining three training stages	80.5	86.5	83.5
w/o training as sequence	79.9	85.9	82.9
w/o training without temporal limit	77.9	83.5	80.7

Table 5. Ablation study of the training strategy on DAVIS 2017 validation.

5. Conclusion

This paper investigates the problem of memory-based video object segmentation(VOS) and proposes Learning position and target Consistency of Memory-based video object segmentation(LCM). We follow memory mechanism and introduce Global Retrieval Module(GRM) to conduct pixel-level matching. Moreover, we design Position Guidance Module(PGM) for learning position consistency. And we integrate object-level information with Object Relation Module(ORM). Our approach achieves state-of-the-art performance on VOS benchmark.

References

[1] Linchao Bao, Baoyuan Wu, and Wei Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5977–5986, 2018. 7

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 2

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017. 6, 7

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 4

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 6

[6] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1189–1198, 2018. 2

[7] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7415–7424, 2018. 7

[8] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014. 6

[9] Hai Ci, Chunyu Wang, and Yizhou Wang. Video object segmentation by learning location-sensitive embeddings. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 501–516, 2018. 7

[10] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020. 2

[11] Bhat Goutam, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *16th European Conference on Computer Vision*, 2020. 6, 7

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[14] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. In *Advances in Neural Information Processing Systems*, pages 2725–2734, 2019. 2, 3, 5

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 5

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *Advances in neural information processing systems*, pages 325–334, 2017. 7

[17] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G Schwing. Videomatch: Matching based video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 54–70, 2018. 2, 7

[18] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, and Chi-Keung Tang. Fast video object segmentation with temporal aggregation network and dynamic template matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8879–8889, 2020. 2

[19] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 603–612, 2019. 2

[20] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8953–8962, 2019. 6, 7

[21] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 90–105, 2018. 2

[22] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. *arXiv preprint arXiv:2001.11243*, 2020. 6, 7

[23] Yongqing Liang, Xin Li, Navid Jafari, and Jim Chen. Video object segmentation with adaptive feature bank and uncertain-region refinement. *Advances in Neural Information Processing Systems*, 33, 2020. 6, 7

[24] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3949–3957, 2019. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[26] Xiankai Lu, Wenguan Wang, Danelljan Martin, Tianfei Zhou, Jianbing Shen, and Van Gool Luc. Video object segmentation with episodic graph memory networks. In *ECCV*, 2020. 6, 7

[27] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for

video object segmentation. In *Asian Conference on Computer Vision*, pages 565–580. Springer, 2018. 2, 6, 7

[28] Kevis-Kokitsi Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *arXiv preprint arXiv:1709.06031*, 2017. 7

[29] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9226–9235, 2019. 1, 2, 3, 5, 6, 7

[30] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018. 4

[31] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2663–2672, 2017. 2

[32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2663–2672, 2017. 6

[33] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016. 1, 6, 7

[34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 6, 7

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2

[36] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. *arXiv preprint arXiv:2007.08270*, 2020. 2, 5, 6, 7

[37] Jianping Shi, Qiong Yan, Li Xu, and Jiaya Jia. Hierarchical image saliency detection on extended cssd. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):717–729, 2015. 6

[38] Carles Ventura, Miriam Bellver, Andreu Girbau, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Rvos: End-to-end recurrent network for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5277–5286, 2019. 6

[39] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9481–9490, 2019. 2, 7

[40] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. *arXiv preprint arXiv:1706.09364*, 2017. 6, 7

[41] Paul Voigtlaender, Jonathon Luiten, and Bastian Leibe. Boltvos: Box-level tracking for video object segmentation. *arXiv preprint arXiv:1904.04552*, 2019. 6

[42] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[43] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 3978–3987, 2019. 7

[44] Peisong Wen, Ruolin Yang, Qianqian Xu, Chen Qian, Qingming Huang, Runmin Cong, and Jianlou Si. Dmvos: Discriminative matching for real-time video object segmentation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2048–2056, 2020. 7

[45] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7376–7385, 2018. 2, 5, 6, 7

[46] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018. 1, 6

[47] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 585–601, 2018. 6

[48] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6499–6507, 2018. 2, 6, 7

[49] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. *arXiv preprint arXiv:2003.08333*, 2020. 2, 6, 7

[50] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019. 2

[51] Xiaohui Zeng, Renjie Liao, Li Gu, Yuwen Xiong, Sanja Fidler, and Raquel Urtasun. Dmm-net: Differentiable mask-matching network for video object segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3929–3938, 2019. 6, 7

[52] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 267–283, 2018. 2

[53] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6688–6697, 2019. 2