Video Object Segmentation Using Global and Instance Embedding Learning

Wenbin Ge[1]*, Xiankai Lu[2*†], Jianbing Shen[3,1]

[1] Beijing Institute of Technology [2] School of Software, Shandong University [3] Inception Institute of Artificial Intelligence

gewenbin292@bit.edu.cn, carrierlxk@gmail.com

Abstract

In this paper, we propose a feature embedding based video object segmentation (VOS) method which is simple, fast and effective. The current VOS task involves two main challenges: object instance differentiation and cross-frame instance alignment. Most state-of-the-art matching based VOS methods simplify this task into a binary segmentation task and tackle each instance independently. In contrast, we decompose the VOS task into two subtasks: global embedding learning that segments foreground objects of each frame in a pixel-to-pixel manner, and instance feature embedding learning that separates instances. The outputs of these two subtasks are fused to obtain the final instance masks quickly and accurately. Through using the relation among different instances per-frame as well as temporal relation across different frames, the proposed network learns to differentiate multiple instances and associate them properly in one feed-forward manner. Extensive experimental results on the challenging DAVIS [34] and Youtube-VOS [57] datasets show that our method achieves better performances than most counterparts in each case.

1. Introduction

Video object segmentation (VOS) aims at segmenting out the class-agnostic object(s) in the video. It has various applications in video editing, autonomous driving, robotics, human-computer interaction, *etc*. According to whether providing the annotation of the initial frame, VOS can be divided into two settings: *unsupervised* and *semi-supervised*. Unsupervised VOS needs to segment the primary objects automatically [11, 42, 64] while semi-supervised VOS needs to segment the specified objects from human interaction or predefined interesting target [33, 5, 52]. In this paper, we focus on the latter task, where the initial annotation of the object(s) are provided in the first frame.

Most current successful semi-supervised VOS approaches formulate VOS as a binary classification task and segment each instance separately [17, 55, 30] (Fig. 1(a)). This leads to two disadvantages: the computation efficiency is heavily affected by the instance number. Moreover, the spatial and temporal relation between different instances can not be sufficiently mined. To capture multiple instances per frame, some methods employ region proposal networks to generate instance masks [27, 58, 63]. Then, the re-identification network is used to find and associate multi-frame instances. However, the performance of these methods heavily relies on the region proposal model that lacks the temporal information during proposal generation. To guide the network to differentiate multiple instances, some recent methods append the instance mask as prior knowledge during the input stream [23]. Nevertheless, this leads to the unclear segmentation boundary as well as ambiguity

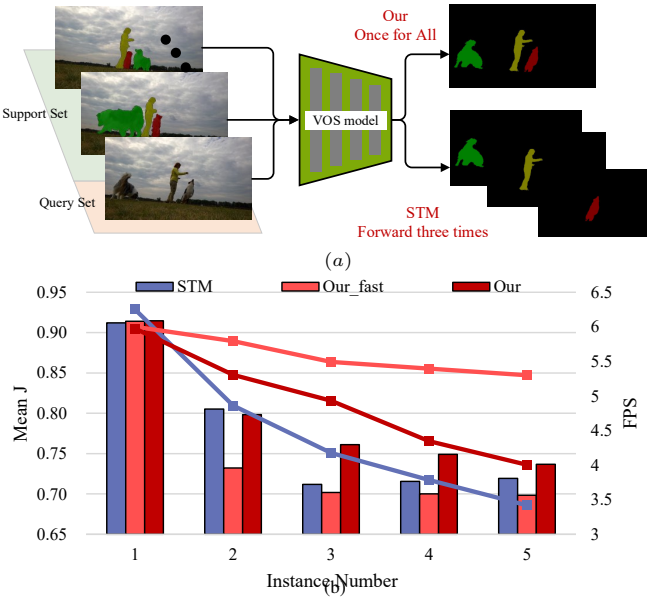

Figure 1: Illustration of our intuition. (a) For multiple instances case, STM [30] tackles each instance independently while our method handles them simultaneously. (b) Comparison of accuracy (in the histogram) and computation speed (by the curve) regarding different instance numbers. Our method takes the instance relation into consideration and achieves better and faster performance than STM.

*The first two authors contribute equally to this work.
†Corresponding author: *Xiankai Lu*.

accumulation due to unstable temporal mask propagation.

To alleviate these issues, we need to explore an effective framework that can comprehensively capture the intra-frame as well as inter-frames instance relationship. We break VOS into two parallel components: (1) an instance level representation that predicts a coarse mask for each local area, which can separate different instances automatically and simultaneously. (2) a global saliency map that segments out all foreground objects which can provide saliency details and realize pixel-wise alignment. Correspondingly, we design a two-branch network, the instance branch takes the instance-specific embedding as input and generates coarse instance score maps. The global branch takes the global object embedding as input and produces a global saliency map of the whole image to separates the foreground from the background. Finally, the coarse instance score maps are interacted with the global saliency map to generate refined instance masks. To maintain the temporal correspond of instances across frames, we make use of *memory-reading* [54, 38, 29, 31, 20] mechanism to incorporate temporal information for both instance embedding and global object embedding. During the segmentation, the proposed network handles all the instances in a single feed-forward pass.

We extensively evaluate our model on three widely-used video object segmentation datasets, namely DAVIS_{17} *val*, DAVIS_{17} *test-dev* and Youtube-VOS_{18}, showing its superior performance as well as higher speed over most existing methods. Our model is flexible, and we also evaluate a performance priority version of our approach (see Fig. 1(b)). Equipped with a learnable decoder, our method achieves state-of-the-art performances for comprehensive evaluation on these VOS datasets.

We summarize the main contributions as follows:

- We propose a new insightful solution for VOS based on embedding learning from instance segmentation. We construct a two-stream network in which one stream focuses on instance embedding learning while the other stream focuses on global feature embedding learning.
- We integrate both the temporal as well as spatial relation between instances into the embedding learning which helps to obtain more accurate mask boundary.
- We propose a temporal attention mechanism to effectively exploit support set information to facilitate the query frame segmentation. This module can adaptively assign different weights to each element in the support set during the network inference.
- We extensively evaluate the proposed method on recent large-scale benchmark datasets. The proposed method performs favorably against state-of-the-arts.

2. Related Works

2.1. Video Object Segmentation

The task of VOS is commonly formulated as a temporal label propagation problem. Traditional methods usually build spatio-temporal graph over the pixels [40], regions [1, 37, 50] and superpixels [49, 41] to infer the labels for subsequent frames. Current top-leading deep learning based VOS models are mainly built upon two critical techniques, *i.e.*, online finetuning and cross-frame feature matching.

Video Object Segmentation with Online Finetuning. Given the ground-truth segmentation mask of the initial frame, semi-supervised VOS methods finetune the network for each video with samples generated in the first frame. Therefore, semi-supervised VOS can be referred to one-shot VOS (O-VOS). OSVOS [5] firstly applies a pre-trained convolutional network for semantic segmentation, and finetunes in the first frame to segment out foreground and background. OnAVOS [44] and OSVOS-S [28] further extend OSVOS by online learning with unlabeled data in the subsequent frame, and by adding instance-level semantic information. Another approach is to learn to propagate the segmentation mask from the previous frame to the next frame after finetuning in the first frame, such as *optical flow* in MaskTrack [32], LucidTracker [19], PreMVOS [27] and CRN [15] or *deep recurrent network* module in MaskRNN [16] and DyeNet [21]. Bao *et al*. [2] integrate Markov Random Field into VOS for iterative refining segmentation results. While these approaches have achieved satisfying performance, they have one critical drawback in common: the network learning in the first frame as well as subsequent frame updating is quite time-consuming.

Video Object Segmentation with Feature Matching. To avoid invoking expensive fine-tuning procedure in the first frame of semi-supervised VOS, some recent works resort to some techniques in few-shot learning. One representative work is OSMN [60] in which a first-frame related visual modulator is learned for adapting the segmentation rapidly. More recently, with the emergence of large-scale VOS datasets [34, 57], *matching*-based networks [17, 55, 42, 24, 48, 47, 26, 43, 23, 30, 63, 18, 36, 22] usually train a prototypical Siamese network to find the most matching pixel (or feature) between the first frame (or a segmented frame) and the query frame, and then perform label prediction accordingly. Moreover, more advanced internal meta learners [35, 3, 25] have been exploited for updating VOS network quickly.

2.2. Feature Embedding Learning

Compared to the traditional VOS methods, most deep learning based methods have achieved better performance without complex graph-based model design as well as post-processing. The reason mainly attributes to the powerful

feature embedding learning. Considering most current VOS datasets [35, 3, 6] are instance-level annotated, it is meaningful to learn more instance-specific feature embedding. However, most methods mentioned above formulate feature embedding learning as a binary segmentation task by handling each instance independently. This binary supervision ignores mining the global context between each instance as well as the wealthy annotation information which is provided by instance-level annotations.

Some works have paid attention to learning more instance-related information by pixel-wise metric learning[8], location-sensitive embedding[10] or attention mechanism[23]. The differences from previous methods are significant: **1)** Our method takes multiple instances relation into consideration to facilitate the instance embedding learning. **2)** Our method provides an elegant architecture to simultaneously segment multiple objects.

Recently, one-stage instance segmentation[62, 4, 51, 56, 7, 52, 39] in static images becomes popular via omitting the proposal generation step in two-stage methods (*e.g.*, Mask-RCNN [13]). Without instance-level proposals in the two-stage methods, to distinguish instances, most of the one-stage methods try to learn an *instance-specific embedding* from the global *pixel embedding*. By taking advantage of instance embedding learning, for instance, anchor-free based proposal embedding in EmbedMask[62], instance mask coefficients in YOLACT [4], position-sensitive instance features in Blendmask [7], polygon surrounded contour in PolarMask [56], point based local shape in CenterMask [52] and instance-aware mask head in CondInst[39], these methods have achieved comparable even better performance than two-stage ones. The success of the one-stage instance segmentation inspires us to design a VOS method that can generate all the instances simultaneously.

3. Proposed Algorithm

3.1. Revisiting Label Propagation in VOS

Before elaborating on our method (§3.2), we first give a brief introduction to the common label propagation methods in VOS. One representative methodology[8, 65, 58, 20] computes the similarity between the reference frames I_r and the query frame (*i.e.*, current frame) I_t, then estimates the label based on the learned similarity matrix directly:

$$\mathcal{P}^t = F_s\left(\phi(I_t), \phi(I_r)\right)\mathcal{P}^{t-1}, \tag{1}$$

where F_s denotes the similarity function with feature extraction ϕ. $\mathcal{P}^{t-1}$ and $\mathcal{P}^t$ represent the labels and prediction in the reference and current frames, respectively.

Another popular label propagation strategy is to integrate the label propagation into the feature learning at the input level or feature level[17, 55, 30, 18]:

$$\mathcal{P}^t = F_g(I_t, I_r, \mathcal{P}^{t-1}), \tag{2}$$

where F_g denotes the end-to-end segmentation network. These methods tend to handle each object individually and perform better than direct label propagation methods at the cost of slow inference speed.

3.2. Feature Embedding Learning in VOS

In contrast to much prior work on decomposing instance VOS into multiple binary segmentation subtasks or propagate the mask directly, our approach focuses on fully exploiting the spatio-temporal relation in a video sequence. This enables us to construct an elegant model that is both strong in performance and fast in inference. Specifically, we advocate a new solution that learns instance-specific embedding for instance video object segmentation. Inspired by instance segmentation [7, 52, 39] and conditional convolution [59], our heuristic approach makes the training process be consistent with the inference stage. In the following, we will describe each of the components in more detail.

Global feature embedding learning. Fig. 2 depicts our network structure. Our framework handles each video in a sequential manner. Prior frames that can be regarded as support set in few-shot learning [3] are fed together with the annotated or segmented *objects mask* to the support encoder for obtaining the support feature embeddings $\{\boldsymbol{m}_i\}_T \in \mathbb{R}^{T\times W\times H\times C}$, where T is the size of support set, W, H and C represent the width, height and channel, respectively. In this way, we can obtain the *objects*-level embedding. Current frame (*i.e.*, query image) is fed into the query encoder to output the query embedding $\boldsymbol{q} \in \mathbb{R}^{W\times H\times C}$.

To learn instance-specific feature embedding while considering temporal information from past frames, following [54, 38, 29, 31], we employ *Query-to-Memory* mechanism to implement the temporal feature combination. Specifically, the embedded *key* and *value* are computed from the support and query embeddings directly:

$$\begin{aligned}
\mathbf{k}^S &= f_k(\{\boldsymbol{m}_i\}_T) = \{\mathbf{k}_i^S\}_T \in \mathbb{R}^{T\times W\times H\times C/8} \\
\mathbf{v}^S &= f_v(\{\boldsymbol{m}_i\}_T) = \{\mathbf{v}_i^S\}_T \in \mathbb{R}^{T\times W\times H\times C/2} \\
\mathbf{k}^Q &= f_k(\boldsymbol{q}) = \{\mathbf{k}^Q\} \in \mathbb{R}^{W\times H\times C/8} \\
\mathbf{v}^Q &= f_v(\boldsymbol{q}) = \{\mathbf{v}^Q\} \in \mathbb{R}^{W\times H\times C/2}.
\end{aligned} \tag{3}$$

In the memory reading [54, 38, 29, 31], the global correlation $\boldsymbol{A}$ between current frame and support set can be formulated as:

$$\boldsymbol{A} = \text{softmax}(\underbrace{\mathbf{k}^S}_{TWH\times C/8} \cdot \underbrace{\mathbf{k}^{Q^\top}}_{C/8\times WH}) \in \mathbb{R}^{(TWH)\times(WH)}. \tag{4}$$

Once obtained $\boldsymbol{A}$, the global feature embedding about the query image is computed as:

$$\boldsymbol{Emb}_{global} = [\underbrace{\mathbf{v}^Q}_{W\times H\times C/2}, \underbrace{\mathbf{v}^{S^\top}\boldsymbol{A}}_{W\times H\times C/2}] \in \mathbb{R}^{W\times H\times C}, \tag{5}$$

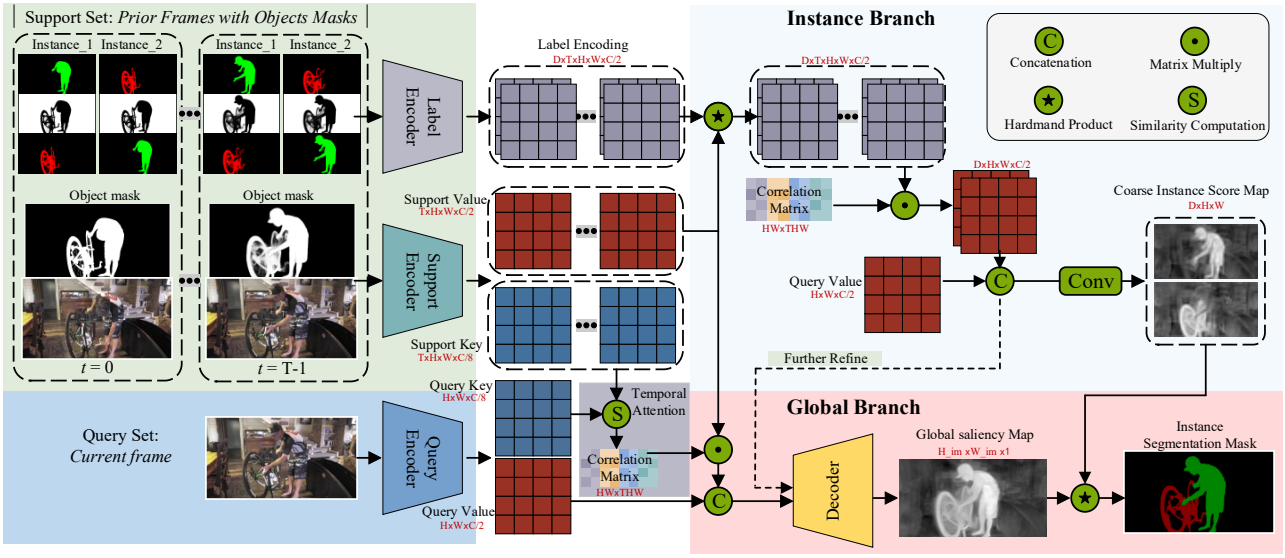

Figure 2: Illustration of our instance embedding based VOS method. Previous frames are fed together with the pre-defined or self-segmented object masks to the support encoder to obtain support embedding. Instance labels are encoded into one-hot tensors and fused with support embeddings to obtain the instance-level embeddings. Current frame is fed into the query encoder to output the query embedding. The query embedding interacts with support embedding under two levels: the first level is to mine instance-specific information via instance branch while the second level is to obtain global object context via global branch. After that, the global decoder predicts a binary global saliency map while the instance branch forms the coarse instance maps directly by differential cropping.

where the first term in concatenation $[\cdot]$ reflects the content of the current frame while the second term reflects the transferred global objects-level feature from support frames.

The global feature embedding aims to represent the salience of each pixel in the whole image, *i.e.*, whether the pixel belonging to an object area or not[52]. Based on global feature embedding, the prediction is a class-agnostic binary map $\boldsymbol{F}_{global}$ (*i.e.*, objects map) that provides a strong Foreground-Background objects prior for current frame in the video[61]:

$$\boldsymbol{F}_{global} = f_{decoder}(\boldsymbol{Emb}_{global}) \in \mathbb{R}^{1\times W_{im}\times H_{im}}, \quad (6)$$

where $f_{decoder}$ is a learnable decoder that comes from [30], W_{im}, H_{im} means the width and height of the output.

Instance-specific feature embedding learning. In addition to transferring global objects information from the support frames using the learned objects embedding, we employ *Query-to-Memory* mechanism to learn instance-specific feature embedding. To incorporate instance-level information, we merge the one-hot label tensor in which each channel represents one instance[55] into the instance-level embedding learning.

Let $\{\mathcal{P}^i\}_T$ denote the one-hot label tensors [55] in the support frames, we introduce a label encoder module. It takes the label tensors as input and predicts an instance level target mask representation with high dimensional space:

$$\widetilde{\mathcal{P}}^i = f_\theta(\mathcal{P}^i) \in \mathbb{R}^{D\times W\times H\times C/2}, \quad (7)$$

where $f_\theta(\cdot)$ denotes the label encoder. For each instance label tensor $\mathcal{P}^i \in \mathbb{R}^{W_{im}\times H_{im}\times 3}$, it has three channels (See Fig. 2): The first channel contains a certain instance, the second channel is about the rest instances, and last channel only contains the background. In this way, our label encoder can suppress the distraction from other instances and maintain the instance-specific information[17]. We set the maximum instances number $D = 6$.

After that, $\widetilde{\mathcal{P}}^i$ is multiplied to the value of support embeddings $\mathbf{v}^S$. This operation allows each support value only contain the instance-specific information:

$$\boldsymbol{V}_{instance} = \underbrace{\mathbf{v}^S}_{T\times W\times H\times C/2} \star \underbrace{\{\widetilde{\mathcal{P}}^i\}_T}_{D\times T\times W\times H\times C/2} \in \mathbb{R}^{D\times T\times W\times H\times C/2}, \quad (8)$$

where '$\star$' denotes dimension-wise Hadamard product. After that, we conduct instance-level memory reading through the correlation matrix $\mathbf{A}$ (*i.e.*, Eq.4) and obtain the instance-level pixel embedding as:

$$\boldsymbol{Emb}_{instance} = [\underbrace{\mathbf{v}^Q}_{D\times W\times H\times C/2}, \underbrace{\boldsymbol{V}^\top_{instance}\boldsymbol{A}}_{D\times W\times H\times C/2}] \in \mathbb{R}^{D\times W\times H\times C}. \quad (9)$$

Based on the instance embedding, we can generate the coarse instance score map directly with a small fully convolution network:

$$\boldsymbol{F}_{coarse} = f_{coarse}(\boldsymbol{Emb}_{instance}) \in \mathbb{R}^{D\times W\times H}, \quad (10)$$

where f_{coarse} denotes the small fully convolution network. Each channel of the $\boldsymbol{F}_{coarse}$ is a heatmap for the corresponding instance.

Instance mask generation. To construct the final instance mask, we take advantage of both $\boldsymbol{F}_{global}$ (*i.e.*, Eq. 6) and

$\boldsymbol{F}_{coarse}$ (*i.e.*, Eq. 10) collaboratively. The instance coarse map indicates the coarse area for each instance and the cropped global saliency map realizes precise segmentation. Specifically, we calculate the Hadamard product of these two items:

$$\widetilde{\boldsymbol{F}}_{final} = \underbrace{\mathcal{U}(\boldsymbol{F}_{coarse})}_{D\times W_{im}\times H_{im}} \star \underbrace{\boldsymbol{F}_{global}}_{1\times W_{im}\times H_{im}} \in \mathbb{R}^{D\times W_{im}\times H_{im}}, \quad (11)$$

where $\mathcal{U}(\cdot)$ is upsampling operation that resizes the coarse instance maps into the current spatial resolution. Finally, all the instance masks can be assembled using soft max aggregation [55] across the D-dimension.

Overall, our method generates all the instance masks simultaneously during both the training phase and the inference phase. To further improve the final segmentation result, we replace the upsampling function $\mathcal{U}(\cdot)$ in Eq. 11 with learnable decoder [30] to refine the coarse instance maps (see dashed line in Fig. 2):

$$\boldsymbol{F}_{final} = f_{decoder}(\boldsymbol{Emb}_{instance}) \in \mathbb{R}^{D\times W_{im}\times H_{im}}. \quad (12)$$

3.3. Network Architecture

We implement the whole method via an end-to-end network. We use ResNet50 [14] as the backbone network for the query and support encoders, except for the input channels. The initial weights come from pre-trained Mask R-CNN [13]. The query encoder takes an RGB query frame as input, while, for the support encoder, input is an RGB support frame concatenated with the class-agnostic binary mask from global feature embedding. For instance embedding module, we implement the *label encoder* (Eq. 7) as a fully convolutional network that consists of two convolution layers followed by batch normalization layers. The rest newly added layers in Eqs. 3 and 10 are implemented by 1×1 convolution layers.

Temporal Attention Mechanism. In the context of VOS, each instance may exhibit appearance variation as well as underlying occlusion. For video with fast appearance change, the latest segmented frames help to locate the targets in the current frame[55], meanwhile, the first annotated frame yet is able to re-identify targets when they re-appear after occlusion[21]. Therefore, it is meaningful to guide the network to assign different weights to the memory *key* under different scenarios. In particular, we present a temporal attention module which consists of convolution layers and pooling layers. For each *key* in the support set, the proposed temporal attention module takes this *key*, the first frame *key* (*i.e.*, as the first frame and its annotation always provide the most reliable information) and the query *key* together as input, and then regresses a normalized weight with a small network. Because there are T frames in the support set, the length of total weights is $T\times1$. Next, this adaptive weight

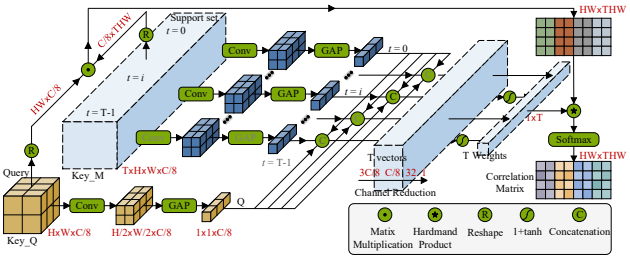

Figure 3: Diagram of the proposed temporal attention mechanism. For each element (*i.e.*, *key*) in the support set, the proposed temporal attention module first computes a global summarization of the first *key*, the selected one and the query *key*. This is implemented by 1×1 convolution layers and global average pooling followed by a concatenation operation. With the combined feature, three 1×1 convolution layers are used for regressing a weight vector.

is applied to the similarity matrix to emphasize the importance of certain support images. The detailed architecture of temporal attention can be seen in Fig. 3.

3.4. Implementation Details

Network Training. Following the training protocol in [3, 43, 31], we sample video clips with length $T+1$from a video to simulate the inference procedure. The first T frames in the clips build the support set while the last frame is used for the query set. We feed them with corresponding annotations into the network and compute the loss functions as:

$$\begin{aligned}\mathcal{L}_{all} &= \mathcal{L}_{mask} + \mathcal{L}_{instance} \\ &= \text{BCE}(\boldsymbol{F}_{global}, \{\mathcal{P}^i\}_T) + \lambda\text{CE}(\boldsymbol{F}_{final}, \{\mathcal{P}^i\}_T),\end{aligned} \quad (13)$$

where $\lambda>0$ is a pre-defined tradeoff parameter. T frames annotations in total are involved in the "recurrence training" procedure [55, 12]. BCE and CE are binary cross entropy and cross entropy loss, respectively. Objects annotations in BCE can be obtained by simply combining the instance-level labels. In this way, we can sufficiently use both objects-level and instance-level annotations for network supervision. We utilize videos from Youtube-VOS$_{18}$ [57] and DAVIS$_{17}$ [34] datasets to train the network. The video clip length$T+1$is set to 4. We employ a random 740×384 crop for each frame with random flipping, rotation and scaling for data augmentation. Our model is implemented on *PyTorch* and trained on four NVIDIA Tesla V100 GPUs with 32 GB memory per card. The batch size is set to 16. We optimize the loss function with Adam optimizer using "poly" learning schedule, with the base learning rate of $1e$-5 and power of 1.0. λ is set to 1.0.

Network Inference. After training, we apply our learned model for the unseen video sequences directly. Similar to the previous works [55, 30, 18], we process each video in a temporally sequential manner. For the first T frames, we take all of them as the support images with corresponding

predictions. Starting from $T+1$ frame, we perform a temporal sampling [45] from the processed frames to construct the support set. As the first frame and its annotation always provide the most reliable information, we absorb the first frame into the support set every time. Our full model runs at a speed of 6.6 FPS. We also evaluate a faster version of our approach that achieves a speed of 9.1 FPS, with only a slight degradation in segmentation accuracy. All our implementations, trained models, and segmentation results will be released to provide the full details of our approach.

4. Experiments

4.1. Experimental Setup

To demonstrate the effectiveness of our method, we apply it to two famous one-shot VOS datasets including DAVIS$_{17}$ [34] and Youtube-VOS$_{18}$ [57].

DAVIS$_{17}$ contains 150 video sequences, totaling 10,459 annotated frames. The whole dataset is divided into three subsets: *training* set (60 videos), *val* set (30 videos) and *test-dev* set (30 videos). Following the standard evaluation protocol [34], we adopt two evaluation metrics: region similarity $\mathcal{J}$ which denotes the mean IoU between the prediction and ground truth, and boundary accuracy $\mathcal{F}$ which is average similarity measure between the boundary of the prediction and the ground truth.

Youtube-VOS$_{18}$ is the latest large-scale dataset for video object segmentation. Similar to DAVIS$_{17}$, this dataset contains three subsets: *training* set (3,471 videos), *val* set (507 videos) and *test-dev* set (541 videos). The *training* set covers 65 categories, the *val* set contains additional 26 unseen categories while the *test-dev* set contains 29 unseen categories. These unseen categories are used for measuring the generalization ability of the evaluation methods. $\mathcal{J}$ and $\mathcal{F}$ are separately computed for the seen and unseen sets.

4.2. Ablation Study

In this section, we analyze the effect of the individual components of our method to the final performance. Table 1 shows the diagnostic experiment results on the DAVIS$_{17}$ *val* set [34]. The experimental results are evaluated by mean $\mathcal{J}$ and mean $\mathcal{F}$. All the variants are retrained independently with their specific network architectures.

Importance of Instance Embedding. We first study the effect of instance embedding learning. As seen, removing instance embedding leads to huge performance degradation (mean $\mathcal{J}$: 80.2→76.4, mean $\mathcal{F}$: 85.3→81.8). This performance gap attributes to that the proposed instance embedding learning can mine instance relationship to facilitate object segmentation.

Effectiveness of Global Embedding. Next, we assess the impact of global embedding according to the segmentation results. From the third row of Table 1, it clearly shows that global embedding learning also helps to improve the segmentation performance. This suggests that leveraging the global context information in objects granularity provides a prior knowledge for the instance-level discrimination.

Aspects	Module	Davis$_{17}$ mean $\mathcal{J}$	$\Delta\mathcal{J}$	mean $\mathcal{F}$	$\Delta\mathcal{F}$
Reference	**Full model** (5 Support Images)	80.2	-	85.3	-
Embedding Learning	*w/o.* Instance Embedding	76.4	-3.8	81.8	-3.5
	w/o. Global Embedding	79.0	-1.2	83.8	-1.5
Multi-Variants Analysis	*w/o.* Temporal Attention	78.9	-1.3	84.2	-1.1
	w/o. Refine Decoder	78.7	-1.5	83.7	-1.6
Support Set Size (T)	First	69.8	-10.4	76.5	-8.8
	Previous	69.4	-10.8	74.6	-10.7
	2	79.9	-0.3	85.0	-0.3
	3	79.9	-0.3	84.9	-0.4
	4	79.8	-0.4	85.0	-0.3
	5	80.2	0.0	85.3	0.0
	6	80.0	-0.2	85.2	-0.1

Table 1: **Ablation study on DAVIS$_{17}$ [34] `val` set** (§4.2). The mean $\mathcal{J}$ and mean $\mathcal{F}$ are adapted.

Temporal Attention Mechanism. It is also of interest to verify the effectiveness of the proposed temporal attention mechanism. We can see that removing this module also leads to the performance degradation (-1.0 and -0.8 in terms of mean $\mathcal{J}$ and $\mathcal{F}$). This observation suggests that assigning various weights for the support images adaptively leads to a balance between target appearance variation and object re-identification when compared to fixed weight.

Support Set Size. Finally, we investigate the influence of support set size on the final performance in-depth. With more reference images in the support set (1→5), better performance can be obtained. However, more support images (5→6) has a slight influence on the final performance. The reason is that most videos in these datasets are short-term sequences, and five support images are enough for capturing global information. In addition, performances that correspond to the first frame and previous frame are inferior to all other cases. This further proves the importance of the proposed temporal attention.

4.3. Quantitative Results

We compare the proposed method with several strong baselines on the three aforementioned challenging datasets.

DAVIS$_{17}$ [34] *val* set and *test-dev* set: The performance of our network on DAVIS$_{17}$ is shown in Table 2 and Table 3 in comparison with more than eighteen VOS methods. Overall, our model outperforms all the contemporary methods and achieves better performance in terms of mean $\mathcal{J}\&\mathcal{F}$ (82.7), mean $\mathcal{J}$ (80.2) and mean $\mathcal{F}$ (85.3) on *val* set. Notably, our method obtains a much higher score for both region similarity and contour accuracy compared to several representative *divide-and-rule* scheme methods that handle each instance independently, such as RGMP [55], RANet [53], STM [30] and GC [22]. Meanwhile, compared to *region proposal* based methods: PReMVOS [27], CFBI [61] and DMMNet [63], our method outperforms these methods by

Table 2: **Evaluation of O-VOS on DAVIS$_{17}$** `val` set (§4.3), with region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$ and average of $\mathcal{J}\&\mathcal{F}$. Speed is also reported.

Method		OSMN[60]	SIMMASK[46]	FAVOS[9]	RVOS[42]	OSVOS[5]	AGAME[18]	OnAVOS[44]	RGMP[55]	OSVOS-S[28]	RANet[53]	FEELVOS[43]
$\mathcal{J}\&\mathcal{F}$	Mean ↑	54.8	65.4	58.2	60.6	60.3	71.0	65.4	66.7	68.0	65.7	71.5
$\mathcal{J}$	Mean ↑	52.5	54.3	54.6	57.5	56.6	68.5	61.6	64.8	64.7	63.2	69.1
	Recall↑	60.9	62.8	61.1	65.2	63.8	78.4	67.4	74.1	74.2	73.7	79.1
	Decay↓	21.5	19.3	14.1	24.9	26.1	14.0	27.9	18.9	15.1	18.6	17.5
$\mathcal{F}$	Mean ↑	57.1	58.5	61.8	63.6	63.9	73.6	69.1	68.6	71.3	68.2	74.0
	Recall↑	66.1	67.5	72.3	73.2	73.8	83.4	75.4	77.7	80.7	78.8	83.8
	Decay↓	24.3	21.0	18.0	28.2	27.0	15.8	26.6	19.6	18.5	19.7	20.1
Times (s)		0.13	0.028	1.8	1.8	7.0	0.07	13	0.13	4.5	0.13	0.5

Method		CINM[2]	PReMVO[27]	DMMNet[63]	STM[30]	AGSS[23]	TVOS[65]	FRTM[35]	GC[22]	CFBI [61]	Ours_fast	**Ours**
$\mathcal{J}\&\mathcal{F}$	Mean ↑	70.6	77.8	70.7	81.8	67.4	72.3	76.9	71.4	81.9	81.2	**82.7**
$\mathcal{J}$	Mean ↑	67.2	73.9	68.1	79.2	69.9	69.9	74.0	69.3	79.1	78.7	**80.2**
	Recall↑	74.5	83.1	77.3	88.7	-	83.7	85.9	-	-	89.0	**90.0**
	Decay↓	24.6	16.2	16.8	8.0	-	13.9	8.8	-	-	**5.8**	6.3
$\mathcal{F}$	Mean ↑	74.0	81.8	73.3	84.3	64.9	74.7	79.8	73.5	84.6	83.7	**85.3**
	Recall↑	81.6	88.9	82.7	91.8	-	86.6	92.1	-	-	93.0	**93.9**
	Decay↓	26.2	19.5	23.5	10.5	-	18.4	13.1	-	-	**8.0**	8.6
Times (s)		38	70	2.7	0.18	0.10	0.03	0.11	-	0.45	0.11	0.15

Table 3: **Evaluation of O-VOS on DAVIS$_{17}$** `test-dev` set (§4.3), with region similarity $\mathcal{J}$, boundary accuracy $\mathcal{F}$ and average of $\mathcal{J}\&\mathcal{F}$. Speed is also reported.

Method		OSMN[60]	FAVOS[9]	OSVOS[5]	OnAVOS[44]	$OSVOS^S$[28]	RGMP[55]	FEELVOS[43]	Lucid[19]
$\mathcal{J}\&\mathcal{F}$	Mean ↑	41.3	43.6	50.9	52.8	57.5	52.9	57.8	66.7
$\mathcal{J}$	Mean ↑	37.7	42.9	47.0	49.9	52.9	51.3	55.1	63.4
$\mathcal{F}$	Mean ↑	44.9	44.2	54.8	55.7	62.1	54.4	60.4	69.9
Times (s)		0.13	1.8	7.0	13	1.8	0.13	0.5	>10000

Method		CINM[2]	DyeNet[21]	PReMVOs[27]	STM[30]	AGSS[23]	TVOS[65]	Ours_fast	**Ours**
$\mathcal{J}\&\mathcal{F}$	Mean ↑	67.5	68.2	71.6	72.3	57.2	63.1	73.4	**75.2**
$\mathcal{J}$	Mean ↑	64.5	65.8	67.5	69.3	54.8	58.8	70.0	**72.0**
$\mathcal{F}$	Mean ↑	70.5	70.5	75.7	75.2	59.7	67.4	76.8	**78.3**
Times (s)		38	0.5	70	0.18	0.11	0.03	0.11	0.15

a large margin (82.7 *vs* 77.8 and 82.7 *vs* 70.7). Finally, our approach outperforms all embedding based methods: FEELVOS [43], TVOS [65] and AGSS [65]. We attribute our performance improvement to the consideration of the instance relation during network learning. Furthermore, we report the segmentation speed by averaging the inference times for all instances. We can see that our fast version yields better performance than recent higher framerates methods (*e.g.*, FRTM [35] and TVOS [65]) meanwhile maintains the favorable speed.

For completeness, we also evaluate our approach on DAVIS$_{17}$ *test-dev* set. This subset is much more challenging than *val* set due to that the heavy and long-term occlusions among instances belonging to the same category are more frequent. The performance of all the compared methods degrade on this dataset, however, our method still outperforms all compared methods. We attribute the performance advances to the help of the proposed temporal attention mechanism, which plays the role of re-identification function. Ours outperforms these methods with a $\mathcal{J}$ & $\mathcal{F}$ score of 75.2. In addition, our fast version is faster than all previous approaches, maintaining a $\mathcal{J}$ & $\mathcal{F}$ score of 73.4.

Youtube-VOS$_{18}$ [57]: Moreover, we report the segmentation results on Youtube-VOS$_{18}$ in Table 4. Our approach obtains a final score of 80.6, significantly outperforming recent state-of-the-arts models: STM [30], TVOS [65], FRTM [35] and EGMP [25]. Considering Youtube-VOS$_{18}$ contains more multiple-instances sequences, the performance promotion over STM on this dataset is more significant than that on DAVIS$_{17}$ (79.4→80.6). Notably that there are 507 sequences with more than 12,576 frames in total, such average promotion over STM is remarkable. Compared to memory-based methods: S2S [57] and FRTM [35], our model achieves much higher performance (*i.e.*, 80.6 *vs* 64.4, 80.6 *vs* 71.3), which verifies the effectiveness of our two-stream embedding learning mechanism. Our method again outperforms other embedding learning based VOS competitors (*e.g.*, FEELVOS [43] and AGSS [65]) across all the metrics. Moreover, our method performs favorably on both *seen* and *unseen* categories (mean $\mathcal{J}$: 80.7 and 75.0).

4.4. Qualitative Results

Fig. 4 shows some visual results at different time steps (uniformly sampled percentage w.r.t. the whole video length) with the main counterpart: STM [30]. Benefit from taking the instance relation into account during the

Table 4: **Evaluation of O-VOS on Youtube-VOS$_{18}$** `val` set (§4.3), with region similarity $\mathcal{J}$ and boundary accuracy $\mathcal{F}$. "Overall": averaged over the four metrics.

	Method	MSK [32]	OSMN [60]	RGMP [55]	OnAVOS [44]	RVOS [42]	OSVOS [5]	S2S [57]	AGAME [18]	PreMVOS [27]	DMMNet [63]	STM [30]	AGSS [23]	TVOS [65]	FRTM [35]	EGMP [25]	Ours_fast	**Ours**
	Overall	53.1	51.2	53.8	55.2	56.8	58.8	64.4	66.1	66.9	58.0	79.4	71.3	67.8	71.3	80.2	79.0	**80.6**
seen	Mean$\mathcal{J}\uparrow$	59.9	60.0	59.5	60.1	63.6	59.8	71.0	67.8	71.4	60.3	79.7	71.3	67.1	72.2	80.7	79.4	**80.7**
	Mean$\mathcal{F}\uparrow$	59.5	60.1	-	62.7	67.2	60.5	70.0	-	75.9	63.5	84.2	75.2	69.4	76.1	**85.1**	83.9	85.0
unseen	Mean$\mathcal{J}\uparrow$	45.0	40.6	45.2	46.6	45.5	54.2	55.5	60.8	56.5	50.6	72.8	65.5	63.0	64.5	74.0	72.6	**75.0**
	Mean$\mathcal{F}\uparrow$	47.9	44.0	-	51.4	51.0	60.7	61.2	-	63.7	57.4	80.9	73.1	71.6	72.7	80.9	80.1	**81.9**

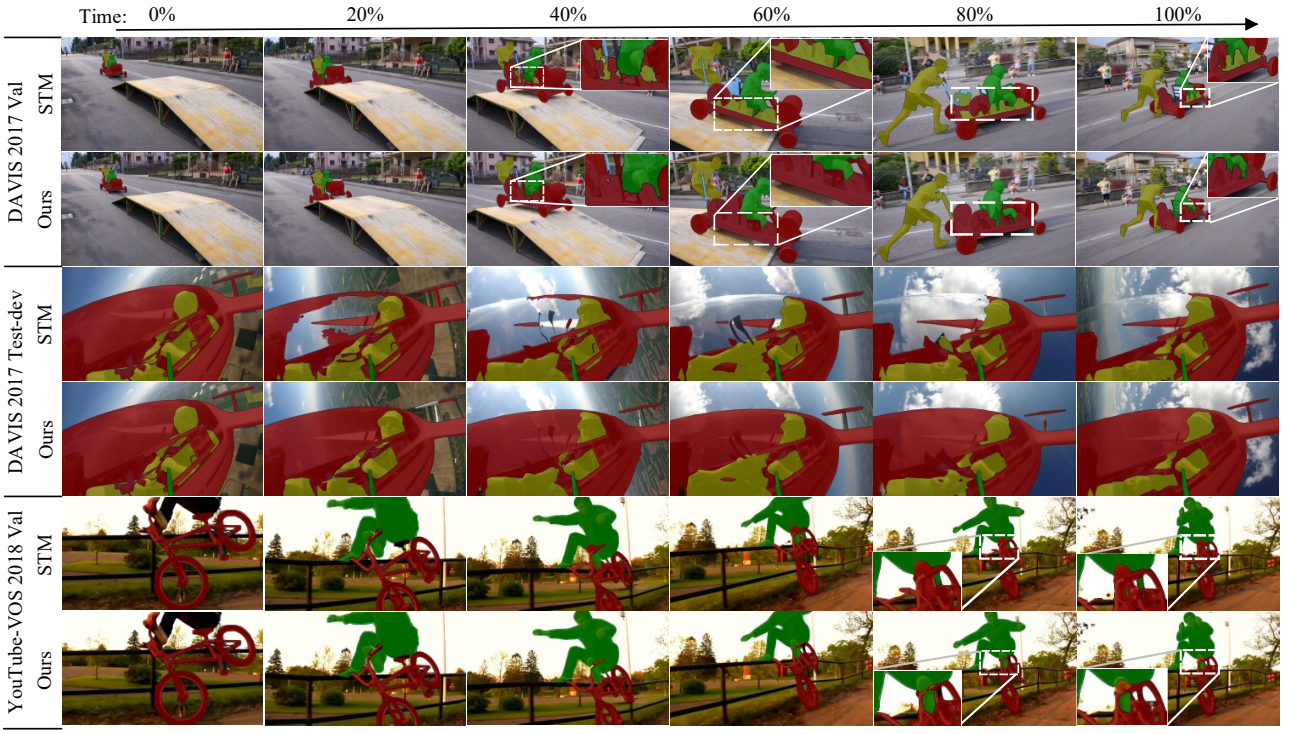

Figure 4: **Qualitative comparison between STM [30] and ours** on DAVIS$_{17}$ *val*, DAVIS$_{17}$ *test-dev* and Youtube-VOS$_{18}$ (§4.4). For the case in Youtube-VOS$_{18}$, the annotations of *bike* and *human* are provided in different frames separately.

instance embedding learning, our method yields more precise segmentation compared to STM. In the first sequence (*i.e.*, *soapbox*), while STM can segment out the primary instances, they predict a confusion label for the cart body. In the second video (*i.e.*, *aerobatics*), it can be observed that STM quickly fails to segment out helicopter part. By comparison, equipped with temporal attention mechanism, our method can capture the temporal coherence and succeed in tracking the whole body. On the third video (*i.e.*, *9c81364fc6*), we can see that STM can not capture the accurate boundary between the objects while our method can discriminate the detailed boundary well by considering the instance relation. On these challenging videos, STM can not tackle the scenario well, by using the relation to facilitate embedding learning, our method can handle different targets successfully. Overall, both quantitative and qualitative results verify the effectiveness of the proposed embedding based VOS approach.

5. Conclusion

In this paper, we have proposed a semi-supervised VOS method from the embedding learning view. We started with the observation that most current VOS methods omit the underlying relationship between different instances during network learning. Based on this insight, we proposed to learn a global feature embedding for capturing the saliency information and local instance-specific embedding to distinguish each instance in an end-to-end way. Correspondingly, both objects-level and instance-level supervisions from instance annotation are sufficiently exploited to guide network learning. Besides, to balance the demand trade-off between target appearance variation and re-detection for occlusion, we introduced a temporal attention mechanism for support images. Experiment showed that each component of our method is highly effective and achieved satisfying results on DAVIS$_{17}$ and Youtube-VOS$_{18}$.

References

[1] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *CVPR*, 2014. 2

[2] 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 *CVPR*, 2018. 2, 7

[3] Goutam Bhat, Felix Jremo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *ECCV*, 2020. 2, 3, 5

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019. 3

[5] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 2, 7, 8

[6] Sergi Caelles, Jordi Pont-Tuset, Federico Perazzi, Alberto Montes, Kevis-Kokitsi Maninis, and Luc Van Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019. 3

[7] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. Blendmask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 2020. 3

[8] Yuhua Chen, Jordi Pont-Tuset, Alberto Montes, and Luc Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 3

[9] 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 *CVPR*, 2018. 7

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

[11] Alon Faktor and Michal Irani. Video segmentation by non-local consensus voting. In *BMVC*, 2014. 1

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. 5

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *ICCV*, 2017. 3, 5

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

[15] Ping Hu, Gang Wang, Xiangfei Kong, Jason Kuen, and Yap-Peng Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, 2018. 2

[16] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *NeurIPS*, 2017. 2

[17] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, 2018. 1, 2, 3, 4

[18] Joakim Johnander, Martin Danelljan, Emil Brissman, Fahad Shahbaz Khan, and Michael Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019. 2, 3, 5, 7, 8

[19] Anna Khoreva, Rodrigo Benenson, Eddy Ilg, Thomas Brox, and Bernt Schiele. Lucid data dreaming for video object segmentation. *International Journal of Computer Vision*, 127(9):1175–1197, 2019. 2, 7

[20] Zihang Lai, Erika Lu, and Weidi Xie. MAST: A memory-augmented self-supervised tracker. In *CVPR*, 2020. 2, 3

[21] Xiaoxiao Li and Chen Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018. 2, 5, 7

[22] Yu Li, Zhuoran Shen, and Ying Shan. Fast video object segmentation using the global context module. In *ECCV*, 2020. 2, 6, 7

[23] Huaijia Lin, Xiaojuan Qi, and Jiaya Jia. Agss-vos: Attention guided single-shot video object segmentation. In *ICCV*, 2019. 1, 2, 3, 7, 8

[24] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *CVPR*, 2019. 2

[25] 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. 2, 7, 8

[26] Xiankai Lu, Wenguan Wang, Jianbing Shen, David Crandall, and Jiebo Luo. Zero-shot video object segmentation with co-attention siamese networks. *IEEE TPAMI*, 2020. 2

[27] Jonathon Luiten, Paul Voigtlaender, and Bastian Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018. 1, 2, 6, 7, 8

[28] K-K Maninis, Sergi Caelles, Yuhua Chen, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. Video object segmentation without temporal information. *IEEE TPAMI*, 41(6):1515–1530, 2018. 2, 7

[29] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, 2016. 2, 3

[30] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. In *ICCV*, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[31] Seoung Wug Oh, Joon-Young Lee, Ning Xu, and Seon Joo Kim. Video object segmentation using space-time memory networks. *ICCV*, 2019. 2, 3, 5

[32] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2, 8

[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 *CVPR*, 2016. 1

[34] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 1, 2, 5, 6

[35] Andreas Robinson, Felix Jaremo Lawin, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Learning fast and robust target models for video object segmentation. In *CVPR*, 2020. 2, 3, 7, 8

[36] Hongje Seong, Junhyuk Hyun, and Euntai Kim. Kernelized memory network for video object segmentation. In *ECCV*, 2020. 2

[37] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing

Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *ECCV*, 2018. 2

[38] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NIPS*, 2015. 2, 3

[39] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. *ECCV*, 2020. 3

[40] David Tsai, Matthew Flagg, Atsushi Nakazawa, and James M Rehg. Motion coherent tracking using multi-label mrf optimization. *International Journal of Computer Vision*, 100(2):190–202, 2012. 2

[41] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, 2016. 2

[42] 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 *CVPR*, 2019. 1, 2, 7, 8

[43] 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 *CVPR*, 2019. 2, 5, 7

[44] Paul Voigtlaender and Bastian Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 2, 7, 8

[45] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 6

[46] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019. 7

[47] Wenguan Wang, Xiankai Lu, Jianbing Shen, David J Crandall, and Ling Shao. Zero-shot video object segmentation via attentive graph neural networks. In *ICCV*, 2019. 2

[48] Wenguan Wang, Jianbing Shen, Xiankai Lu, Steven CH Hoi, and Haibin Ling. Paying attention to video object pattern understanding. *IEEE TPAMI*, 2020. 2

[49] Wenguan Wang, Jianbing Shen, Fatih Porikli, and Ruigang Yang. Semi-supervised video object segmentation with super-trajectories. *IEEE TPAMI*, 41(4):985–998, 2018. 2

[50] Wenguan Wang, Jianbing Shen, Ruigang Yang, and Fatih Porikli. Saliency-aware video object segmentation. *IEEE TPAMI*, 40(1):20–33, 2017. 2

[51] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. In *ECCV*, 2020. 3

[52] Yuqing Wang, Zhaoliang Xu, Hao Shen, Baoshan Cheng, and Lirong Yang. Centermask: single shot instance segmentation with point representation. In *CVPR*, 2020. 1, 3, 4

[53] Ziqin Wang, Jun Xu, Li Liu, Fan Zhu, and Ling Shao. Ranet: Ranking attention network for fast video object segmentation. In *ICCV*, 2019. 6, 7

[54] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *ICLR*, 2015. 2, 3

[55] Seoung Wug Oh, Joon-Young Lee, Kalyan Sunkavalli, and Seon Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 1, 2, 3, 4, 5, 6, 7, 8

[56] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020. 3

[57] 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 *ECCV*, 2018. 1, 2, 5, 6, 7, 8

[58] Shuangjie Xu, Daizong Liu, Linchao Bao, Wei Liu, and Pan Zhou. Mhp-vos: Multiple hypotheses propagation for video object segmentation. In *CVPR*, 2019. 1, 3

[59] Brandon Yang, Gabriel Bender, Quoc V Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *NIPS*, 2019. 3

[60] Linjie Yang, Yanran Wang, Xuehan Xiong, Jianchao Yang, and Aggelos K Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 2, 7, 8

[61] Zongxin Yang, Yunchao Wei, and Yi Yang. Collaborative video object segmentation by foreground-background integration. In *ECCV*, 2020. 4, 6, 7

[62] Hui Ying, Zhaojin Huang, Shu Liu, Tianjia Shao, and Kun Zhou. Embedmask: Embedding coupling for one-stage instance segmentation. *arXiv*, 2019. 3

[63] Xiaohui Zeng, Renjie Liao, Li Gu, Yuwen Xiong, Sanja Fidler, and Raquel Urtasun. Dmm-net: Differentiable mask-matching network for video object segmentation. In *ICCV*, 2019. 1, 2, 6, 7, 8

[64] Lu Zhang, Jianming Zhang, Zhe Lin, Radomr Mch, Huchuan Lu, and You He. Unsupervised video object segmentation with joint hotspot tracking. In *ECCV*, 2020. 1

[65] Yizhuo Zhang, Zhirong Wu, Houwen Peng, and Stephen Lin. A transductive approach for video object segmentation. In *CVPR*, 2020. 3, 7, 8