DIPNet: Dynamic Identity Propagation Network for Video Object Segmentation

Ping Hu[1] pinghu@bu.edu
Jun Liu[2] jun_liu@sutd.edu.sg
Gang Wang[3] gangwang6@gmail.com
Vitaly Ablavsky[1] ablavsky@bu.edu
Kate Saenko[1] saenko@bu.edu
Stan Sclaroff[1] sclaroff@bu.edu

[1]Boston University [2]Singapore University of Technology and Design [3]Alibaba Group

Abstract

Many recent methods for semi-supervised Video Object Segmentation (VOS) have achieved good performance by exploiting the annotated first frame via one-shot fine-tuning or mask propagation. However, heavily relying on the first frame may weaken the robustness for VOS, since video objects can show large variations through time. In this work, we propose a Dynamic Identity Propagation Network (DIPNet) that adaptively propagates and accurately segments the video objects over time. To achieve this, DIPNet factors the VOS task at each time step into a dynamic propagation phase and a spatial segmentation phase. The former utilizes a novel identity representation to adaptively propagate objects' reference information over time, which enhances the robustness to videos' temporal variations. The segmentation phase uses the propagated information to tackle the object segmentation as an easier static image problem that can be optimized via light-weight fine-tuning on the first frame, thus reducing the computational cost. As a result, by optimizing these two components to complement each other, we can achieve a robust system for VOS. Evaluations on four benchmark datasets show that DIPNet provides state-of-the-art performance with time efficiency.

1. Introduction

Given annotations in the first frame, semi-supervised Video Object Segmentation (VOS) aims to identify and segment the target objects across the video [48, 9, 36, 33, 3, 4, 56, 54]. Recently, deep learning based methods [10, 43, 55, 30, 17, 8, 60, 57, 23, 45] have made remarkable progress for this task. However, many of these successful methods rely on one-shot fine-tuning or mask propagation, leading to a trade-off between efficiency and accuracy. For example, methods [2, 32, 25, 46] tackle the VOS task as a pure spatial segmentation problem, and apply computationally-heavy online fine-tuning to adapt models to memorize both the high-level semantic attributes and the low-level appearance of the target, leading to effectiveness but slow speed. On the other hand, approaches like [51, 6, 58] rely on the spatiotemporal connections to propagate masks from the first frame. Without online fine-tuning, this type of method is less adapted to testing videos, resulting in fast but less accurate performance. More crucially, due to the ground-truth annotations in the first frame, most of these methods choose to fix the first frame as a semantic reference rather than dynamically updating the reference frame over time. Yet, a fixed reference can weaken the robustness for segmentation, as video objects can show large variations over time. While a spatial prior has been utilized as auxiliary information in previous works [51, 58, 25, 34], it is still challenging to deal with internal variations of objects (see Fig. 1 for examples). As such, these methods may face difficulties in tackling online applications that require both fast speed and long-term accurate segmentation of video objects.

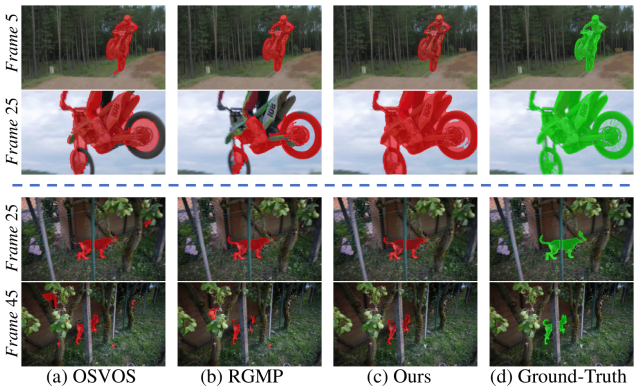

Figure 1. Two examples showing significant appearance variations of objects in different frames. (a) OSVOS [2] fine-tunes the model on the first frame. (b) RGMP [51] constantly propagates object's masks from the first frame. (c) Our DIPNet dynamically propagates the identity of objects and achieves robustness for video objects' temporal variations.

Unlike the previous methods that take the first frame as a

fixed semantic reference, in this paper we propose to adaptively update the reference information over time, so as to achieve robustness to objects' temporal variations. However, dynamically propagating object masks through time is non-trivial, since segmentation errors at each time step may be accumulated and amplified. To tackle this challenge, a propagation-based VOS system should meet two requirements: (i) the model should be robust to the quality of reference information; (ii) the model should produce accurate segmentation for each time step. To this end, we carefully design a Dynamic Identity Propagation Network that achieves these separately and complementarily with two components: the *Dynamic Identity Propagation* that effectively updates and propagates objects' information over time and the *Spatial Instance Segmentation* that accurately segments objects based on the propagated information.

Dynamic Identity Propagation. In order to adaptively update and propagate the reference information of objects over time, we propose a novel semantic identity representation that enables the robust extraction of instance information from previous frames. Recent works about *Class Activation Mapping* [61, 59, 40] have shown that in a deep CNN trained for classification, specific objects are encoded in specific channels of the highest-level feature maps (an example is shown in the supplementary material). Based on this, we end-to-end enhance the highest-level feature maps of ResNet [16] to encode instance-level information of the input into different channels for segmentation. As a result, we can represent an instance in the image with an Identity Attention vector, which is a channel-wise attention vector indicating the feature channels' relevance to the object. As the same objects in consecutive frames are encoded similarly, the object can be propagated by applying the Identity Attention vector estimated from previous frames to feature maps of the current frame (Fig. 3 gives an overview). The proposed representation also allows us to incorporate objects' information from multiple frames by averaging the Identity Attentions vectors. As our propagation model is based on deep CNNs' natural representation, it eases the difficulties for network training. Moreover, the Identity Attention operates at a high-level semantic space and ignores low-level details, thus can robustly update and propagate the reference information over time.

Spatial Instance Segmentation. After transferring the instance's information to the current frame via the Dynamic Identity Propagation module, the task is simplified to be a static image problem that segments objects with given high-level information as guidance. The segmentation network for this task can focus on segmenting objects' low-level details, and can be efficiently optimized via very slight fine-tuning on the first frame. As a result, we can achieve high segmentation accuracy while greatly reducing the computational cost comparing to previous one-shot fine-tuning methods [46, 2, 34].

Thus, in Dynamic Identity Propagation, we adaptively update objects' reference information along with their temporal variations; and in Spatial Instance Segmentation, the propagated identity information is utilized to facilitate accurate and efficient segmentation of objects for each frame. These two components complement each other to achieve robustness for temporal variations. Moreover, by tackling these two easier sub-tasks with respect to their specific targets, we reduce the difficulties for optimizing the entire system [39], and conveniently achieve a good state for both high accuracy and time efficiency. To sum up, our contributions are as follows: (1) We propose a novel semantic identity representation which allows us to robustly update and propagate objects' reference information in a dynamic way. (2) We develop a DIPNet that comprises a propagation step and a segmentation step, which can be separately optimized to complement each other. (3) We conduct evaluations on four benchmark datasets. Without extra training data, DIPNet achieves state-of-the-art accuracy at a fast speed.

2. Related Work

Traditional methods for semi-supervised VOS are usually built on tracking [48, 50, 12, 19], pixel trajectory [49, 9], object proposal [36, 44, 47, 24], and spatial-temporal lattices [33, 21]. Recently, many approaches employ deep CNNs to achieve good performance. Methods like [25, 34, 7, 18, 52, 22, 28] extract spatial-temporal structures from the motion. Models in [46, 2, 41, 34] are heavily fine-tuned on the first frame to achieve more effective models. One-shot finetuning is an effective way to adapt models to the testing scenarios, however for the spatiotemporal VOS task it always requires a large number of iterations to work effectively, leading to high computational cost. To achieve efficiency, some recent work proposes to propagate masks from the first frame based on the spatiotemporal connections between frames. Yang *et al.* [58] apply conditional batch normalization to modulate features of the current frame based on the first frame. Oh *et al.* [51] propagate the instance mask by concatenating the high-level feature maps and then decode them. By discarding the one-shot fine-tuning step, these methods greatly accelerate their speeds, while less adapted to the testing video, resulting in lower accuracy. Due to the well-annotated instance masks, both the one-shot learning based methods [46, 2, 41] and propagation based models like [51, 58] heavily relying on the semantic information in the first frame. However, video objects usually show variations and a fixed reference maybe not optimal. Although the online adaptation [46] can update models along with time, its speed is too slow. In contrast to previous methods [51, 58, 46, 2, 41, 34], our method relieves the challenge by a disentangled task-specific framework and a novel identity representation mechanism that allows for dy-

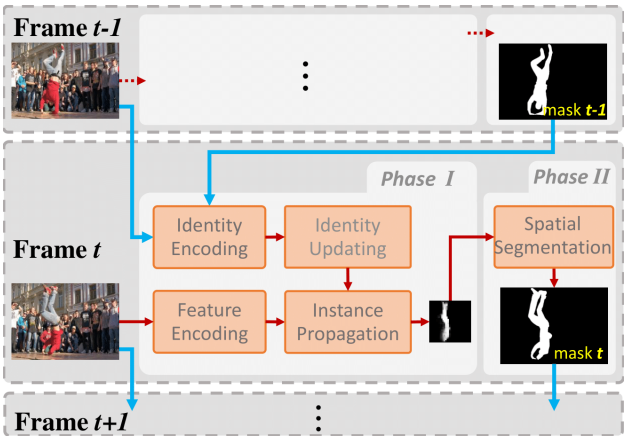

Figure 2. An overview of the proposed DIPNet. For each video object, we maintain a Dynamic Identity Attention vector that is initialized in the first frame and updated over time. At a time step t, there are two phases. In the Dynamic Identity Propagation (Phase I), we first employ an Identity Encoding Network to extract an Identity Attention vector for the object from the frame t-1, and use it to update the Dynamic Identity Attention; then, we apply the updated Dynamic Identity Attention vector to the feature maps of frame t to output a coarse mask that explicitly indicates the object's spatial and shape information. In the Spatial Instance Segmentation (Phase II), the target object is segmented based on the propagated coarse mask.

namically updating reference information along with time, so as to achieve robustness for video objects' temporal variations without losing time efficiency.

3. Dynamic Identity Propagation for VOS

In this section, we present each component of our method. An overview of is shown in Fig. 2.

3.1. Phase I: Dynamic Identity Propagation

Recent findings of *Class Activation Mapping* [61, 59, 40] show that for deep CNNs trained for image classification, specific objects in the input image are encoded by specific channels of the highest-level deep feature maps. Based on this, we further end-to-end enhance the highest-level feature maps of deep CNNs to encode instance-level information in different channels for object segmentation. As a result, an instance can be identified with an Identity Attention vector, which is actually a channel-wise attention vector showing each channel's relevance to the object. In the context of VOS, as consecutive frames are encoded similarly by the network, the information about an object can be propagated by multiplying the Identity Attention estimated from the previous frames to feature maps of the current frame. Moreover, since the proposed Identity Attention explicitly operates at high-level semantic space, our propagation model can be robust against low-level details in masks of previous segmentation results. Therefore we are able to dynamically and robustly update object information over time. Below we introduce the Dynamic Identity Propagation in detail, and an overview of this part is shown in Fig. 3.

Feature Encoding. We directly utilize ResNet [16] as feature extractor for video frames, as it encodes objects in different channels of the final-layer feature maps. Via end-to-end optimizing the propagation phase, we further enhance the feature maps to contain instance-level information in different channels. The last layer of the ResNet is removed, and the feature maps output by the block $Conv5\_X$ are utilized as output. The model takes a video frame of size 3*256*256 as input, and outputs a feature map of size 2048*8*8.

Identity Encoding Network. Given the feature maps of video frames encoded by the Feature Encoding Network, the Identity Encoding is designed to estimate the respective Identity Attention vectors for target objects. The structure of the Identity Encoding Network is shown in Fig. 3 (a). The network takes a video frame and the corresponding object mask as input, and outputs the 2048-dimension Identity Attention vector which shows the channels' relevance to the object. We build the network on ResNet50 [16]. After each of the first four consecutive convolutional blocks (i.e. $Conv1$, $Conv2\_X$, $Conv3\_X$, $Conv4\_X$), the target object's information is incorporated by applying its mask m as spatial attention on the feature maps f,

$$f' = f + m \odot f \tag{1}$$

where f' is the processed feature that will be the input for the next Conv-Block, $\odot$ represents the pixel-wise product between the binary mask and feature map. With such a scheme, we can robustly convert the object masks into high-level semantic representation. To further utilize overall information of the image and mask to help estimate the high-level semantic identity representation, at the end of the network, a fully-connected layer and a sigmoid layer are adopted to convert the feature maps from $Conv5\_X$ into a 2048-dimension Identity Attention vector that shows each channel's relevance to the current target.

Identity Updating Module. In order to dynamically update the reference information over time to achieve robustness for temporal variations in video, we maintain a Dynamic Identity Attention vector for each object. As shown in Fig. 3 (b), the Dynamic Identity Attention is updated by the results at the previous frames,

$$\alpha_t = \omega \cdot \beta_{t-1} + (1 - \omega) \cdot \alpha_{t-1} \tag{2}$$

where α_{t-1} and α_t are the Dynamic Identity Attention vectors before and after updating, β_{t-1} is the Identity Attention vector estimated from the results on $frame_{t-1}$, and ω is a parameter to control the influence of the earlier frames. The Dynamic Identity Attention vector is initialized with the first frame, and skips updating at frames with empty segmentation output. With such a design, the Dynamic Identity

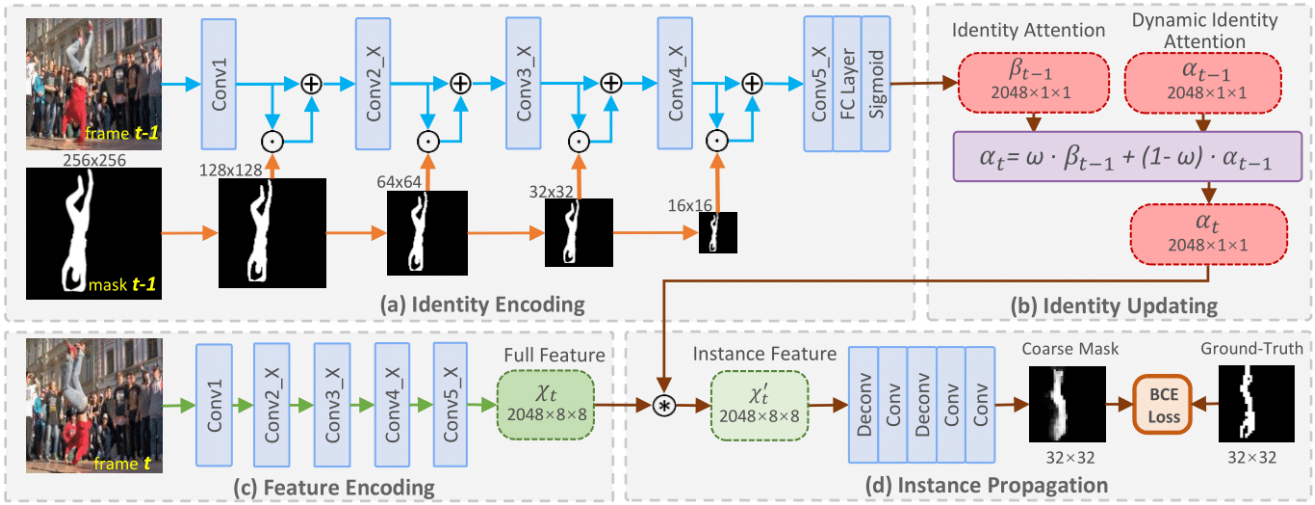

Figure 3. Overview of the propagation phase. Based on [61, 59, 40], we end-to-end enhance ResNet to encode different instances in different channels of the highest-level feature maps. Thereby an Identity Attention vector for an object is defined as a channel-wise attention vector that shows the feature channel's relevance to the object. (a) Identity Encoding Network estimates the object's Identity Attention vector. (b) Identity Updating Module adaptively maintains a Dynamic Identity Attention vector for each object. (c) Feature Encoding Network encodes video frames into feature maps. (d) Instance Propagation Module applies the Dynamic Identity Attention on the current frame's feature maps. ⊕ denotes pixel-wise addition. ⊙ means pixel-wise multiplication. ⊛ is channel-wise multiplication.

Attention is able to track the variations of objects over time, so that enhance the method's robustness.

Instance Propagation. Attention [53, 42] is a mechanism to re-weigh the nodes/channels with extra information in order to achieve better performance. In our case, on the one hand, with Feature Encoding Network, we get the feature maps that contains instance-level information in different channels; on the other hand, the Dynamic Identity Attention vector shows each channel's relevance to the target object. As a result, by applying the Dynamic Identity Attention vector as a channel-wise attention on the current frame's feature maps, the target object is propagated to the current frame. The process is shown in Fig. 3 (d)

At first, we multiply the Dynamic Identity Attention vector to the current frame's feature maps to extract features for the target objects. Then, we adopt two 3*3 Deconv layers, two 3*3 Conv layers and a 1*1 Conv layer to convert the extracted features to a 32*32 coarse mask for the target. All the layers except for the last one are followed by a BatchNorm Layer and a ReLU layer. When dealing with single-instance VOS, we add a sigmoid layer at the end. By outputting a propagated mask with resolution of 32*32, which is a relatively low resolution, we alleviate the network's burden of extracting low-level details. Thus allow the model to focus on propagating high-level information of the instance mask and ease the difficulties of training.

3.2. Phase II: Spatial Instance Segmentation

After the identity propagation phase, the object's information is transferred to the current frame as a coarse mask and the problem is simplified to a static image segmentation problem. At this phase, we aim to accurately segment the object based on the information of the propagated coarse mask. Comparing to previous one-shot finetuning methods [32, 25, 46] that drive the model to memorize both high-level semantic information and low-level appearance details of the target, the task for our model is much simpler: segmenting objects from the images based on the high-level guidance information.

Spatial Segmentation Network. In this part, we design our Spatial Instance Segmentation Network based on the Cascaded Refinement Network (CRN) proposed in [18] for static image segmentation. The CRN incorporates a coarse segmentation as spatial attention on the feature maps and segments the objects accurately. The original CRN accepts a 16*16 coarse mask, we modify the network structure so that the network takes a 512*512 image frame as well as the propagated 32*32 coarse mask as input, and outputs a refined 512*512 mask.

By formulating this step as a task of object segmentation guided by coarse masks, the segmentation network focuses on extracting low-level details and does not need to care too much about the high-level semantic information of the instance, thus alleviating the training burden. As a result, the model can be effectively adapted to testing videos via very slight finetuning on the first frame, and greatly saves the computational cost comparing to previous methods like [2, 46, 34].

3.3. Multi-Instance VOS

Multi-instance VOS is a more difficult task due to challenges like occlusion and similar appearances. To alleviate

these challenges, we propose a two-stage method to tackle multi-instance cases.

3.3.1 Foreground Region Segmentation

At first, we apply DIPNet to segment all the target instances as a single foreground object. Based on the foreground mask, we crop a tight region from the video frames, and re-size the region to the required input size. Then multi-instance segmentation is performed on these cropped patches. This step is beneficial for the accuracy, since cropping foreground patches helps to reduce the interference from background and make target instances more dominant.

3.3.2 Multiple Instance Segmentation

We differentiate instances on the coarse mask propagated by the Dynamic Instance Propagation phase. According to the disjoint constraint of instances, each pixel can only belong to one instance. Based on this, we formulate the multi-instance VOS as a task of pixel-wise multi-class classification, and adopt the softmax classifier that is widely used in semantic segmentation [11].

The model for multi-instance VOS is based on the same network in Fig. 3. To process multiple objects with a single network at the same time, we take advantage of the *Batch* dimension of the input tensor for deep networks. Given a frame from a video with n target objects, we stack the input images for the n objects on the *Batch* dimension to form input tensors with *Batchsize* = n. Then we apply the network on the input tensors to produce an output tensor with *Batchsize* = n. Lastly, we adopt the softmax function along the *Batch* dimension of the output to assign class probabilities to pixels as $p_i(\cdot) = \frac{e^{f_i(\cdot)}}{\sum_{j=0}^{n} e^{f_j(\cdot)}}$, where i is the object identity, p_i is probability map for the i-th object and n is the total number of objects, f_i is the network output, and $j = 0$ represents background. To compute f_0, we utilize the foreground as an extra object, and multiply the output for it with -1. After segmenting different instances, we apply the spatial refinement network on these coarse masks to generate full-resolution ones.

3.4. Network Training

By formulating the task of VOS as two sub-steps, we can easily optimize the system by training the two components separately. To effectively train our models, we adopt a three-stage training process [2] as follows.

Pre-Training on PASCAL VOC. We utilize the 11,355 images from PASCAL-VOC dataset [11, 14] to pre-train our model. To generate consecutive image pairs for training the temporal propagation model, we first randomly select an instance from a sample image as the target; then generate a second frame by shifting and rotating the image. For the spatial refinement network, we create training samples composed of an image, a ground-truth mask, and a coarse mask that is produced by randomly applying morphological operations (dilation and erosion kernels with sizes between 0 to 16 pixels) on the ground-truth mask and scaling it to 32*32. The temporal propagation network and refinement network are trained separately. We adopt the Binary Cross Entropy (BCE) loss and optimize both networks using SGD with batch-size 4 , learning rate 1e-4, and momentum 0.9 for 40 epochs.

Off-line training on DAVIS. To adapt models to the task of VOS, we fine-tune our models on the training split of DAVIS2016 (for single-instance task) and DAVIS2017 (for multi-instance task). To train the dynamic identity propagation module, we randomly select two frames as a training pair from every ten consecutive frames to form a training sample. For the spatial segmentation network, we randomly select frames from the training set to create a training sample as in the pre-training stage. We independently train the two networks both for 20 epochs using SGD with batch-size 4, learning rate 1e-4, and momentum as 0.9.

One-shot Finetuning for Testing. We also fine-tune our models on the first frame to adapt to the testing video. We don't find that fine-tuning the instance propagation network results in performance improvement. For efficiency at testing time, we only apply one-shot finetuning on the spatial refinement network. We make training samples using the first frame and apply morphological operations on the mask as in the previous two stages. To segment foreground regions, we train the spatial segmentation network for 200 iterations on the original scale with learning rate 1e-3 and momentum 0.9. For single-instance VOS, we further finetune the model 100 iterations on the cropped foreground patch; for multi-instance VOS, we further apply 200 iterations of finetuning for each instance.

4. Experiments

We evaluate our DIPNet on four benchmarks including DAVIS2016 [35] and Youtube-Object [20, 38] for single-instance VOS; SegtrakV2 [27], DAVIS2017 *validation* and *test-dev* [37] for multi-instance VOS. The region similarity $\mathcal{J}$ and the contour accuracy $\mathcal{F}$ [35] are utilized for evaluation. The region similarity is the mean intersection-over-union (mIoU) between the predicted segmentation map and the ground truth. The contour accuracy adopts the F-measure between the predicted contour and the ground truth. We built our method with PyTorch and all the experiments are performed with an Nvidia Titan Xp GPU.

4.1. Segmentation Performance

Single-object VOS. We compare the proposed DIPNet with two sets of recent state-of-the-art methods. The first set includes one-shot finetuning based methods OnAVOS [46],

Methods	DIPNet	DIPNet*	OnAVOS	OSVOS^S	CNIM	OSVOS	MSK	RGMP†	RGMP	PML	OSNM
$\mathcal{J}$ Mean ↑	0.858	0.836	**0.861**	0.856	0.834	0.798	0.797	0.824	0.815	0.757	0.74
Recall ↑	**0.973**	0.967	0.961	0.968	0.949	0.936	0.931	-	0.917	0.896	876
$\mathcal{F}$ Mean ↑	0.864	0.851	0.849	**0.875**	0.850	0.806	0.754	0.822	0.820	0.793	0.729
Recall ↑	0.956	**0.959**	0.897	0.956	0.921	0.926	0.871	-	0.908	0.934	0.870
One-shot	light	light	heavy	heavy	heavy	heavy	heavy	heavy	no	no	no
w/ Temp-Prop	✓	✓					✓	✓	✓	✓	✓
#Training Data	13.5k	13.5k	87k	85.1k	15.6k	2.1k	11k	26.6K	26.6K	85.1k	87.2k
Testing Speed (s/f)	1.09	0.70	15.57	(4.5)	(>60)	9.24	(12)	(1.87)	(0.13)	(0.28)	(0.14)

Table 1. Single-instance VOS Performance on DAVIS2016 with comparison to one-shot finetuning methods (OnAVOS [46], OSVOS^S [32], CNIM [1],OSVOS [2], MSK [34]) and temporal propagation methods (PML [5], OSNM [58], RGMP [51], RGMP† represents the results with one-shot finetuning). *DIPNet** indicates our method without foreground segmentation step. The best results are **boldfaced**, and the second are underlined. In the last row, the *s/f* is seconds per frame, and the numbers in parentheses are reported by the original papers.

DIPNet	OSVOS[2]	RCAL[13]	MSK[34]	OnAVOS[46]	OSNM[58]
.789	.783	.781	.777	.774	.690

Table 2. mIoU on Youtube-Object dataset.

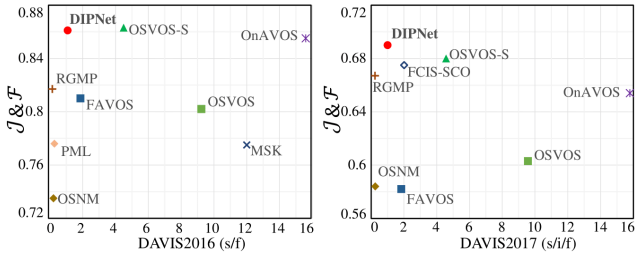

Figure 4. Accuracy versus runtime on *val* set of DAVIS2016 (left) and DAVIS2017 (right). The s/f and s/i/f represent seconds per frame and seconds per instance per frame, respectively.

OSVOS^S [32], CNIM [1], OSVOS [2], and MSK [34]. The other set comprises of temporal propagation based methods RGMP [51], PML [5], and OSNM [58]. The performance for DAVIS16 [35] is shown in Table 1 and the left part of Fig. 4. In general, one-shot finetuning methods achieve high accuracy but low efficiency, while temporal propagation based methods show high efficiency but low accuracy. Unlike these methods, our proposed DIPNet achieves both state-of-the-art accuracy and fast speed. Comparing to OnAVOS [46], our method achieves similar region similarity $\mathcal{J}$, but performs much better in contour accuracy $\mathcal{F}$. Furthermore, OnAVOS runs at a speed of 15.57 seconds per frame, as it applies one-shot fine-tuning, online adaptation, and post-processing with denseCRF [26] during testing, while our full method only takes about 1.09 seconds per frame, which is nearly 15x faster. The OSVOS^S incorporates the segments from existing instance semantic segmentation model [15] as extra information and achieves very similar accuracy but lower speed than ours. Comparing to propagation-based fast methods like RGMP, PML, and OSNM that optimize the task as propagation problem, our method is able to achieve a much higher accuracy without losing too much efficiency. Moreover, after one-shot fine-tuning, RGMP[58] only achieves 0.824 in mIoU at a speed 1.87s/f, which is still worse than ours in both speed and accuracy. This also shows that our framework can learn more effectively and efficiently from data. The performance on the Youtube-Object is presented in Table 2, and our method outperforms most of the recent methods. The state-of-the-art performance for single-instance VOS validates the effectiveness and efficiency of our DIPNet.

Multi-object VOS. The performance for multi-instance VOS on the *validation* split of DAVIS2017 [37] is shown in Table 3 and the right part of Fig. 4. We compare our method with recent models CNIM [1], FCIS-SCO [24], OSVOS^S [32], OnAVOS [46], OSVOS [2], FAVOS [6], and OSNM [58]. As shown in the table and figure, our DIPNet outperforms the one-shot-learning-based methods OnAVOS, and OSVOS by a large margin in both accuracy and efficiency. Although FCIS-SCO and OSVOS^S both adopt segments from existing instance segmentation methods [29, 15] to enhance the mask, our DIPNet still achieves better performance with higher efficiency. Comparing to the propagation-based methods FAVOS and OSNM, our method outperforms by a large gap of more than 0.1 in terms of both mean-$\mathcal{J}$ and mean-$\mathcal{F}$. The performance on the DAVIS2017 *test-dev* and SegtrackV2 is presented in Table 4 and Table 5 respectively. Our method outperforms most of the recent methods. CNIM outperforms our method on the DAVIS2017 dataset. However, our DIPNet only slightly finetunes on the first-frames, while CNIM utilizes optical flow and the entire video for inference. Moreover, CNIM takes more than one hour to finetune on synthesized training images for each testing video which makes it runs (>60 seconds per instance per frame) more than 60x slower than our DIPNet (1.06 seconds per instance per frame).

Running Time. During testing, DIPNet takes about 110 ms per frame for inference. To apply one-shot fine-tuning for the spatial segmentation model, DIPNet takes 40 seconds for 200 iterations at the original scale, and 20 seconds for 100 iterations at the cropped patch. As a result, our DIPNet takes 1.09 seconds per frame on DAVIS2016, Fig. 4 plots the $\mathcal{J}\&\mathcal{F}$ score and the running speed of each method. Our DIPNet shows a better trade-off than the other methods.

Methods		DIPNet	DIPNet*	CNIM	FCIS-SCO	OSVOS^S	OnAVOS	OSVOS	FAVOS	OSNM
$\mathcal{J}$	Mean ↑	0.653	0.587	**0.672**	0.665	0.647	0.616	0.566	0.546	0.525
	Recall ↑	0.766	0.662	0.745	**0.797**	0.742	0.674	0.638	0.611	0.609
$\mathcal{F}$	Mean ↑	0.716	0.651	**0.740**	0.688	0.713	0.691	0.639	0.618	0.571
	Recall ↑	**0.821**	0.739	0.816	**0.821**	0.807	0.754	0.738	0.723	0.661
One-shot		light	light	heavy	no	heavy	heavy	heavy	no	no
w/ Temp-Prop		✓	✓						✓	✓

Table 3. Multi-instance VOS performance for *validation* of DAVIS2017. *DIPNet** means to jointly segment multiple instances without foreground cropping step. The best results are **boldfaced**, and the second best are underlined.

DIPNet	OSVOS^S [32]	RGMP [51]	OnAVOS [46]	OSVOS [2]	OSMN [58]
.552	.529	.513	.499	.470	.377

Table 4. mIoU on DAVIS-17 *test-dev*.

DIPNet	MRNN [19]	RGMP [51]	MSK [34]	OFL [44]	OSVOS [2]
.738	.721	.711	.703	.675	.654

Table 5. mIoU on SegtrackV2 datasets.

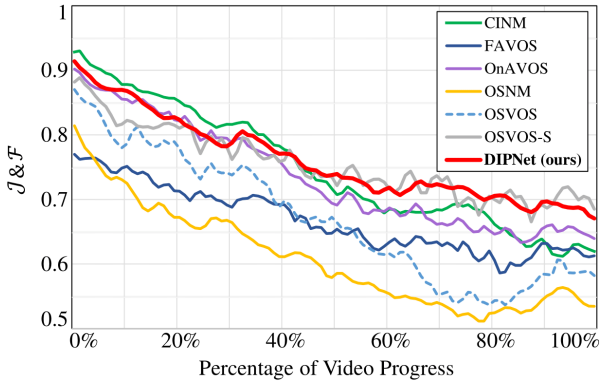

Figure 5. The average performance ($\mathcal{J}\&\mathcal{F}$) on DAVIS17 of different methods over time. "'0%' means the beginning of sequence. "'100%' represents the end of sequence. Performance on all the videos are normalized to the same length.

4.2. Method Analysis

4.2.1 Dynamic Identity Propagation

At first, we show in Fig. 5 how the performance of our DIPNet and other methods change over time. As we can see, unlike most of the other methods that decrease drastically over time, DIPNet achieves more stable accuracy. This shows that our method provides better robustness for video objects' temporal variations. OSVOS-S achieves a similar curve to ours, yet it is based on instance segmentation results of the MaskRCNN [15] and nearly 5x slower than ours. To investigate how the robustness for temporal variations is achieved by our dynamic identity propagation mechanism, we experiment with different ω in the Dynamic Identity Attention (Eq. 2). The performance for different ω is shown in Fig. 6. As we can see, a larger ω leads to a better performance, which shows that our dynamic reference is effective to achieve robustness for temporal variations of objects. Therefore, in practice we choose $\omega = 0.95$ for experiments. We also compare our model design with OSMN [58]. For a fair comparison, we also train

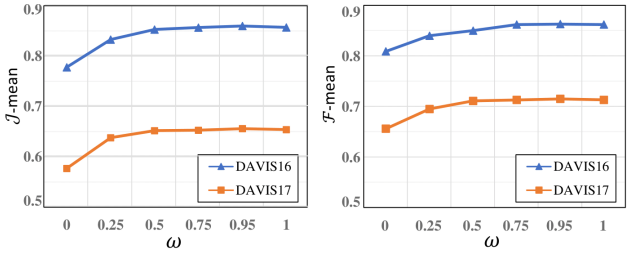

Figure 6. The performance for different ω in the Dynamic Identity Attention. $\omega = 0$ means only using the first frame as a reference, and $\omega = 1$ represents only relying on the previous frame

our model with DAVIS2017 as in OSMN [58]. Our unsupervised model achieves mIoU of 0.758 on DAVIS2016 which is higher than 0.740 of OSMN [58]. This shows that our Dynamic Identity Propagation is more effective.

4.2.2 Identity Attention Vector

In this section, we analyze the Identity Attention vector generated by the identity encoding network. We compute the 2048-d Identity Attention vectors for instances in all the frames. Then for the convenience of visualization, we apply t-SNE [31] to embed the Identity Attentions into a 2-d space. Based on the reduced vectors, we show in Fig. 7 an example for the distributions of those Identity Attentions vectors. As we can see, after training the distributions for different instances are better separated (left of Fig. 7 (c)) and the changes for the same instance between frames are smoother and more gradual (right of Fig. 7 (c)). To evaluate the effectiveness of the design of the identity encoding network, we also try directly concatenating the reference frame and mask as input for the Identity Encoding Network. Without one-shot finetuning on DAVIS2016, the concatenation structure achieves 0.713 in mIoU, which is worse than 0.734 of our proposed one. This shows that our network design is more robust to extract objects' semantic information from masks.

4.2.3 Ablation Study

We first show the effectiveness of the joint segmentation for multi-instance VOS. On the *validation* of DAVIS2017, without cropping the foreground, segmenting instances separately leads to a mIoU of 0.540. However, applying the

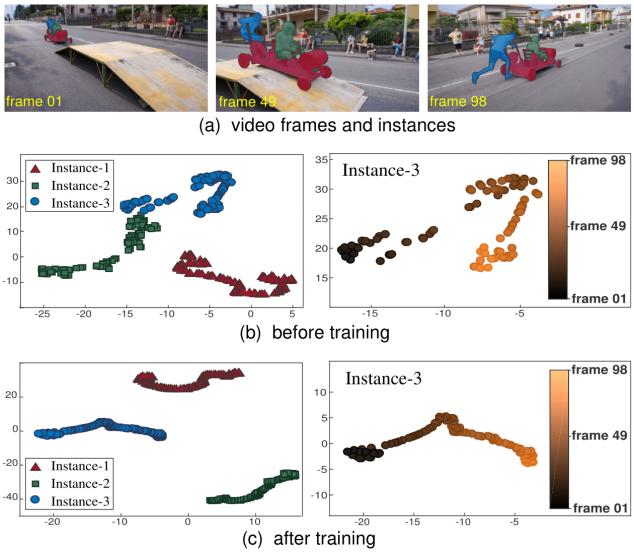

Figure 7. Distribution of the Identity Attention vectors for the three instances (sequentially labelled as red, green, and blue) in the sequence "soapbox" of DAVIS2017 using t-SNE [31]. (a) are examples of the video frames and the three instances. The left graphs of (b) and (c) are the distributions of the instances in all frames of the video. The right graphs of (b) and (c) show the details of the distributions for the instance *"Instance-3"* (labeled as blue in (a)) of the different frames, and the frame ID is indicated by the intensity of color.

IPN	SSN	DAVIS16	DAVIS17$_{fore}$	DAVIS17$_{multi}$
✓		0.643	0.671	0.395
	✓	0.825	0.819	0.486
✓	✓	0.836	0.832	0.587

Table 6. The performance (mIoU) for the Identity Propagation Network (IPN) and the Spatial Segmentation Network (SSN) in the proposed method. DAVIS16: single object segmentation. DAVIS17$_{fore}$: foreground segmentation. DAVIS17$_{multi}$: Jointly segmenting multiple instances.

joint segmentation increases the mIoU to 0.587. This is because segmenting instances separately may lead to overlaps between the instance masks which should be disjoint. The effectiveness of the proposed foreground cropping step can be shown by comparing the models without foreground cropping (DIPNet* in Table 1 and Table 3) with the models that adopt this step (DIPNet in Table 1 and Table 3). Directly applying our method leads to a mIoU of 0.836 on DAVIS16 and 0.587 on DAVIS17. While after applying the proposed foreground segmentation step, we get an improvement of 0.02 on DAVIS16 and 0.066 on DAVIS17. This shows that foreground cropping step is effective and necessary.

The effectiveness of different components is shown in Table 6. Without the Spatial Segmentation Network (SSN), the Identity Propagation Network (IPN) achieves low accuracy. This is because the IPN is designed to accept full-resolution masks, but when running by itself, it accepts previous coarse masks as input and the errors will be accumulated. The SSN can also run on its own by accepting the segmentation of previous frame as input. Due to the strong spatial-temporal continuity between consecutive frames, the SSN itself can achieve good performance. However, when combined with the IPN, the whole system can achieve better accuracy. Especially for the multi-instance task (DAVIS17$_{multi}$ in Table 6), combining the IPN and SSN leads to an improvement of 0.101 in mIoU. This shows the effectiveness of the proposed instance propagation mechanism, and also shows that the two components can complement each other effectively.

	Pre-Training	Off-Line	One-Shot
$\mathcal{J}$ Mean ↑	0.332	0.734	0.836
Recall ↑	0.341	0.840	0.967
$\mathcal{F}$ Mean ↑	0.340	0.742	0.851
Recall ↑	0.357	0.832	0.959

Table 7. Performance for different training stages on DAVIS2016.

As introduced in previous section, our method involves three stages of training, which are static image pre-training, DAVIS off-line training, and one-shot finetuning. We show the performance for models of these different stages in Table 7. After pre-training with images from the Pascal VOC dataset, the accuracy in mIoU is 0.332. Then fine-tuning on the DAVIS dataset helps to adapt model to the VOS task thus greatly improving the mIoU by 0.4. Finally, one-shot finetuning on the first frame further adapts the network to the testing videos, and leads to a further improvement of accuracy by 0.1 in mIoU. We also try jointly optimizing two components together as an end-to-end system at the one-shot finetuning step, but don't find accuracy improvement.

5. Conclusion

In this work, we present the accurate and fast Dynamic Identity Propagation Networks for semi-supervised video object segmentation. The task of VOS is explicitly formulated as a combination of two phases: a dynamic identity propagation step and a spatial segmentation step. In this way, the system can be more effectively optimized with limited data by separately optimizing models for their specific purpose. Furthermore, our method can be very efficiently adapted to each test video, and thus achieves state-of-the-art accuracy with high efficiency. Experiments on four benchmark datasets validate the effectiveness and efficiency of our method.

Acknowledgements. This work was supported in part by DARPA and NSF. The authors also acknowledge the support from Zhejiang Leading Innovation Research Program 2018R01017.

References

[1] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, 2018. 6

[2] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixe, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 1, 2, 4, 5, 6, 7

[3] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018. 1

[4] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. V. Gool. The 2019 davis challenge on vos: Unsupervised multi-object segmentation. *arXiv:1905.00737*, 2019. 1

[5] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, 2018. 6

[6] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 1, 6

[7] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, 2017. 2

[8] K. Duarte, Y. S. Rawat, and M. Shah. Capsulevos: Semi-supervised video object segmentation using capsule routing. In *ICCV*, 2019. 1

[9] S. Duffner and C. Garcia. Pixeltrack: a fast adaptive algorithm for tracking non-rigid objects. In *ICCV*, 2013. 1, 2

[10] S. Dutt Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, 2017. 1

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5

[12] M. Godec, P. M. Roth, and H. Bischof. Hough-based tracking of non-rigid objects. In *ICCV*, 2011. 2

[13] J. Han, L. Yang, D. Zhang, X. Chang, and X. Liang. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, 2018. 6

[14] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *ICCV*, 2017. 6, 7

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3

[17] P. Hu, G. Wang, X. Kong, J. Kuen, and Y. Tan. Motion-guided cascaded refinement network for video object segmentation. *IEEE Trans. on TPAMI*, 2019. 1

[18] P. Hu, G. Wang, X. Kong, J. Kuen, and Y.-P. Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, 2018. 2, 4

[19] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. MaskRNN: Instance Level Video Object Segmentation. In *NIPS*, 2017. 2, 7

[20] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *ECCV*, 2014. 5

[21] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, 2017. 2

[22] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident nnetwork. In *CVPR*, 2017. 2

[23] J. Johnander, M. Danelljan, E. Brissman, F. S. Khan, and M. Felsberg. A generative appearance model for end-to-end video object segmentation. In *CVPR*, 2019. 1

[24] Y. Jun Koh, Y.-Y. Lee, and C.-S. Kim. Sequential clique optimization for video object segmentation. In *ECCV*, 2018. 2, 6

[25] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. *arXiv preprint arXiv:1703.09554*, 2017. 1, 2, 4

[26] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 6

[27] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, 2013. 5

[28] X. Li and C. Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, 2018. 2

[29] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, 2017. 6

[30] J. Luiten, P. Voigtlaender, and B. Leibe. Premvos: Proposal-generation, refinement and merging for video object segmentation. In *ACCV*, 2018. 1

[31] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 7, 8

[32] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE Trans. on PAMI*, 2018. 1, 4, 6, 7

[33] N. Märki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 1, 2

[34] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A. Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 1, 2, 4, 6, 7

[35] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 5, 6

[36] F. Perazzi, O. Wang, M. Gross, and A. Sorkine-Hornung. Fully connected object proposals for video segmentation. In *ICCV*, 2015. 1, 2

[37] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 5, 6

[38] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 5

[39] S. Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017. 2

[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017. 2, 3, 4

[41] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, 2017. 2

[42] M. F. Stollenga, J. Masci, F. Gomez, and J. Schmidhuber. Deep networks with internal selective attention through feedback connections. In *NIPS*, 2014. 4

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

[44] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, 2016. 2, 7

[45] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *CVPR*, 2019. 1

[46] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 1, 2, 4, 5, 6, 7

[47] H. Wang, T. Raiko, L. Lensu, T. Wang, and J. Karhunen. Semi-supervised domain adaptation for weakly labeled semantic video object segmentation. In *ACCV*, 2016. 2

[48] S. Wang, H. Lu, F. Yang, and M.-H. Yang. Superpixel tracking. In *ICCV*, 2011. 1, 2

[49] W. Wang, J. Shen, J. Xie, and F. Porikli. Super-trajectory for video segmentation. In *ICCV*, 2017. 2

[50] L. Wen, D. Du, Z. Lei, S. Z. Li, and M.-H. Yang. Jots: Joint online tracking and segmentation. In *CVPR*, 2015. 2

[51] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, 2018. 1, 2, 6, 7

[52] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, 2018. 2

[53] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 4

[54] K. Xu, L. Wen, G. Li, L. Bo, and Q. Huang. Spatiotemporal cnn for video object segmentation. In *CVPR*, 2019. 1

[55] N. Xu, L. Yang, Y. Fan, J. Yang, D. Yue, Y. Liang, B. Price, S. Cohen, and T. Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *ECCV*, 2018. 1

[56] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018. 1

[57] S. Xu, D. Liu, L. Bao, W. Liu, and P. Zhou. Mhp-vos: Multiple hypotheses propagation for video object segmentation. In *CVPR*, 2019. 1

[58] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, 2018. 1, 2, 6, 7

[59] J. Zhang, Z. Lin, J. Brandt, X. Shen, S. Sclaroff, and S. A. Bargal. Top-down neural attention by excitation backprop. In *ECCV*, 2016. 2, 3, 4

[60] L. Zhang, Z. Lin, J. Zhang, H. Lu, and Y. He. Fast video object segmentation via dynamic targeting network. In *ICCV*, 2019. 1

[61] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 2, 3, 4